EMERGING TECHNOLOGIES AND SYSTEMS FOR BIOLOGICALLY PLAUSIBLE IMPLEMENTATIONS OF NEURAL FUNCTIONS

EDITED BY: Erika Covi, Elisa Donati, Stefano Brivio and Hadi Heidari
PUBLISHED IN: Frontiers in Neuroscience



frontiers Research Topics



frontiers

Frontiers eBook Copyright Statement

The copyright in the text of individual articles in this eBook is the property of their respective authors or their respective institutions or funders. The copyright in graphics and images within each article may be subject to copyright of other parties. In both cases this is subject to a license granted to Frontiers.

The compilation of articles constituting this eBook is the property of Frontiers.

Each article within this eBook, and the eBook itself, are published under the most recent version of the Creative Commons CC-BY licence. The version current at the date of publication of this eBook is CC-BY 4.0. If the CC-BY licence is updated, the licence granted by Frontiers is automatically updated to the new version.

When exercising any right under the CC-BY licence, Frontiers must be attributed as the original publisher of the article or eBook, as applicable.

Authors have the responsibility of ensuring that any graphics or other materials which are the property of others may be included in the CC-BY licence, but this should be checked before relying on the CC-BY licence to reproduce those materials. Any copyright notices relating to those materials must be complied with.

Copyright and source acknowledgement notices may not be removed and must be displayed in any copy, derivative work or partial copy which includes the elements in question.

All copyright, and all rights therein, are protected by national and international copyright laws. The above represents a summary only. For further information please read Frontiers' Conditions for Website Use and Copyright Statement, and the applicable CC-BY licence.

ISSN 1664-8714
ISBN 978-2-88976-000-8
DOI 10.3389/978-2-88976-000-8

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

EMERGING TECHNOLOGIES AND SYSTEMS FOR BIOLOGICALLY PLAUSIBLE IMPLEMENTATIONS OF NEURAL FUNCTIONS

Topic Editors:
Erika Covi, NaMLab gGmbH, Germany
Elisa Donati, University of Zurich, Switzerland
Stefano Brivio, Institute for Microelectronics and Microsystems, Italian National Research Council, Italy
Hadi Heidari, University of Glasgow, United Kingdom

Citation: Covi, E., Donati, E., Brivio, S., Heidari, H., eds. (2022). Emerging Technologies and Systems for Biologically Plausible Implementations of Neural Functions. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88976-000-8

Table of Contents

05 ***Editorial: Emerging Technologies and Systems for Biologically Plausible Implementations of Neural Functions***
Erika Covi, Elisa Donati, Stefano Brivio and Hadi Heidari

07 ***Efficient Implementation of Cerebellar Purkinje Cell With the CORDIC Algorithm on LaCSNN***
Xinyu Hao, Shuangming Yang, Jiang Wang, Bin Deng, Xile Wei and Guosheng Yi

23 ***Prediction in Autism by Deep Learning Short-Time Spontaneous Hemodynamic Fluctuations***
Lingyu Xu, Xiulin Geng, Xiaoyu He, Jun Li and Jie Yu

35 ***Digital Biologically Plausible Implementation of Binarized Neural Networks With Differential Hafnium Oxide Resistive Memory Arrays***
Tifenn Hirtzlin, Marc Bocquet, Bogdan Penkovsky, Jacques-Olivier Klein, Etienne Nowak, Elisa Vianello, Jean-Michel Portal and Damien Querlioz

49 ***Memristor-Based Edge Detection for Spike Encoded Pixels***
Daniel J. Mannion, Adnan Mehonic, Wing H. Ng and Anthony J. Kenyon

60 ***LAO-NCS: Laser Assisted Spin Torque Nano Oscillator-Based Neuromorphic Computing System***
Hooman Farkhani, Tim Böhnert, Mohammad Tarequzzaman, José Diogo Costa, Alex Jenkins, Ricardo Ferreira, Jens Kargaard Madsen and Farshad Moradi

73 ***On-Chip TaO_x-Based Non-volatile Resistive Memory for* in vitro *Neurointerfaces***
Maksim Zhuk, Sergei Zarubin, Igor Karateev, Yury Matveyev, Evgeny Gornev, Gennady Krasnikov, Dmitiry Negrov and Andrei Zenkevich

81 ***Spatial Properties of STDP in a Self-Learning Spiking Neural Network Enable Controlling a Mobile Robot***
Sergey A. Lobov, Alexey N. Mikhaylov, Maxim Shamshin, Valeri A. Makarov and Victor B. Kazantsev

91 ***Topological Properties of Neuromorphic Nanowire Networks***
Alon Loeffler, Ruomin Zhu, Joel Hochstetter, Mike Li, Kaiwei Fu, Adrian Diaz-Alvarez, Tomonobu Nakayama, James M. Shine and Zdenka Kuncic

102 ***Equilibrium Propagation for Memristor-Based Recurrent Neural Networks***
Gianluca Zoppo, Francesco Marrone and Fernando Corinto

110 ***Neurohybrid Memristive CMOS-Integrated Systems for Biosensors and Neuroprosthetics***
Alexey Mikhaylov, Alexey Pimashkin, Yana Pigareva, Svetlana Gerasimova, Evgeny Gryaznov, Sergey Shchanikov, Anton Zuev, Max Talanov, Igor Lavrov, Vyacheslav Demin, Victor Erokhin, Sergey Lobov, Irina Mukhina, Victor Kazantsev, Huaqiang Wu and Bernardo Spagnolo

124 ***Bio-Inspired Techniques in a Fully Digital Approach for Lifelong Learning***
Stefano Bianchi, Irene Muñoz-Martin and Daniele Ielmini

138 ***Double MgO-Based Perpendicular Magnetic Tunnel Junction for Artificial Neuron***
Dong Won Kim, Woo Seok Yi, Jin Young Choi, Kei Ashiba, Jong Ung Baek, Han Sol Jun, Jae Joon Kim and Jea Gun Park

147 ***On-Chip Training Spiking Neural Networks Using Approximated Backpropagation With Analog Synaptic Devices***
Dongseok Kwon, Suhwan Lim, Jong-Ho Bae, Sung-Tae Lee, Hyeongsu Kim, Young-Tak Seo, Seongbin Oh, Jangsaeng Kim, Kyuho Yeom, Byung-Gook Park and Jong-Ho Lee

161 ***Blood Pressure Modulation With Low-Intensity Focused Ultrasound Stimulation to the Vagus Nerve: A Pilot Animal Study***
Ning Ji, Wan-Hua Lin, Fei Chen, Lisheng Xu, Jianping Huang and Guanglin Li

171 ***Toward New Modalities in VEP-Based BCI Applications Using Dynamical Stimuli: Introducing Quasi-Periodic and Chaotic VEP-Based BCI***
Zahra Shirzhiyan, Ahmadreza Keihani, Morteza Farahi, Elham Shamsi, Mina GolMohammadi, Amin Mahnam, Mohsen Reza Haidari and Amir Homayoun Jafari

188 ***Non-linear Memristive Synaptic Dynamics for Efficient Unsupervised Learning in Spiking Neural Networks***
Stefano Brivio, Denys R. B. Ly, Elisa Vianello and Sabina Spiga

204 ***Adaptive Extreme Edge Computing for Wearable Devices***
Erika Covi, Elisa Donati, Xiangpeng Liang, David Kappel, Hadi Heidari, Melika Payvand and Wei Wang

231 ***Markov Chain Abstractions of Electrochemical Reaction-Diffusion in Synaptic Transmission for Neuromorphic Computing***
Margot Wagner, Thomas M. Bartol, Terrence J. Sejnowski and Gert Cauwenberghs

frontiers in Neuroscience

EDITORIAL
published: 31 March 2022
doi: 10.3389/fnins.2022.863680

Editorial: Emerging Technologies and Systems for Biologically Plausible Implementations of Neural Functions

Erika Covi[1*†], *Elisa Donati*[2*†], *Stefano Brivio*[3*†] *and Hadi Heidari*[4*†]

[1] NaMLab gGmbH, Dresden, Germany, [2] Institute of Neuroinformatics, University of Zurich, Eidgenössische Technische Hochschule Zürich (ETHZ), Zurich, Switzerland, [3] CNR—IMM, Unit of Agrate Brianza, Agrate Brianza, Italy, [4] Microelectronics Lab, James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom

Keywords: memristive devices, neuromorphic circuit, sensors, learning, editorial

Editorial on the Research Topic

Emerging Technologies and Systems for Biologically Plausible Implementations of Neural Functions

The human brain is a complex and fascinating biological machine. With 20 W only, the activity of 100 billion neurons and 3 orders of magnitude more (10^{15}) synapses in a volume as small as a shoebox allow us to learn, process, sense, and perceive a vast amount of information from the external environment in real-time. The human brain features a distributed system based on slow and unreliable components. Yet, it is able to learn from experience and compute unstructured data reliably with extreme energy efficiency. Therefore, our brain's real-time and low-power cognitive processes have always been the ultimate ambition in terms of building artificial systems for Edge Information-Extraction and Computing, User-Specific Applications such as Healthcare, Autonomous vehicles, Robotics, and the Internet of Things. Studies on the brain lead to models describing its operating and computational principles, which in turn can be used as a guideline to build new devices, circuits, and systems emulating brain functionality.

Neuromorphic Very Large-Scale Integration (VLSI) circuits model neural networks using a synthetic biology approach whereby they attempt to understand the properties of brain-inspired neural networks by building biologically plausible artifacts that reproduce the physics of the biological systems they model. Neuromorphic circuits can exhibit very slow, biologically plausible, time constants, facilitating the artificial system and/or real-world interaction. Despite the slow time constants, the neuromorphic neural processing chips have fast response times, thanks to a distributed memory, which improves the latency typical of conventional von Neumann architectures. For these reasons, neuromorphic systems can be developed to carry out sensory data analysis and information extraction and solve problems in noisy and uncertain settings and constraint satisfactory problems. In addition, these systems are able to learn from experience, leading to significant progress in the perceptive abilities of e.g., robots, security, and healthcare systems.

Recently, emerging technologies, encompassing memristive and spintronic devices, have been investigated to further improve the memory performance and to complement Complementary Metal Oxide Semiconductor (CMOS) technology, in power-limited neuromorphic systems on edge. Thanks to their excellent performance in terms of high scalability, low latency, low-power operation, and their ability to reversibly change their conductance upon applying proper electrical stimuli, these devices are being researched to emulate artificial synaptic or neural behaviors. Furthermore, their intrinsic physical properties are well suited to implement spike-based

OPEN ACCESS

Edited and reviewed by:
Timothée Masquelier, Centre National de la Recherche Scientifique (CNRS), France

***Correspondence:**
Erika Covi
erika.covi@namlab.com
Elisa Donati
elisa@ini.uzh.ch
Stefano Brivio
stefano.brivio@mdm.imm.cnr.it
Hadi Heidari
Hadi.Heidari@glasgow.ac.uk

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *27 January 2022*
Accepted: *03 March 2022*
Published: *31 March 2022*

Citation:
Covi E, Donati E, Brivio S and Heidari H (2022) Editorial: Emerging Technologies and Systems for Biologically Plausible Implementations of Neural Functions. Front. Neurosci. 16:863680. doi: 10.3389/fnins.2022.863680

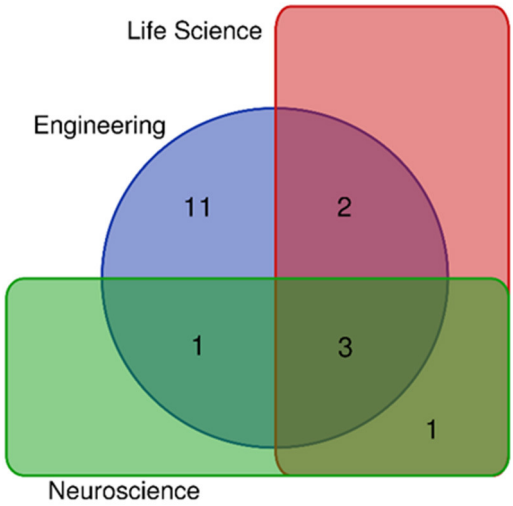


FIGURE 1 | Distribution of the contributions across disciplines.

time, rate-sensitive operations locally, and support edge-of-chaos dynamics, as well as fundamental computing primitives belonging to biological neurons and synapses.

This Research Topic provides an overview of the avant-garde artificial biologically plausible sensing, computing, and perception paradigms and technologies enabling biologically plausible neuromorphic systems. Contributions cover the following areas:

1. Sensors for biological signals and external environmental stimuli.
2. Emerging devices, circuits, and systems enabling neuromorphic paradigms.
3. Emerging technologies and device models to emulate synaptic plasticity and learning.
4. Biologically plausible models that are implementable in neuromorphic sensing, computing, and perception systems.

The 18 articles in this collection span and merge several disciplines, from the field of engineering to life science and neuroscience, with a significant portion of cross-disciplinary works, as illustrated in **Figure 1**. Furthermore, this collection provides a good representation of worldwide research, collecting contributions from four continents and 14 countries.

Among the authors, only 14% are female and the percentage of female first/last authors is 16% each. The data evidence that there is still a long path ahead to achieve gender balance.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Covi, Donati, Brivio and Heidari. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 15 October 2019
doi: 10.3389/fnins.2019.01078

Efficient Implementation of Cerebellar Purkinje Cell With the CORDIC Algorithm on LaCSNN

*Xinyu Hao, Shuangming Yang, Jiang Wang, Bin Deng, Xile Wei and Guosheng Yi**

School of Electrical and Information Engineering, Tianjin University, Tianjin, China

Purkinje cell is an important neuron for the cerebellar information processing. In this work, we present an efficient implementation of a cerebellar Purkinje model using the Coordinate Rotation Digital Computer (CORDIC) algorithm and implement it on a Large-Scale Conductance-Based Spiking Neural Networks (LaCSNN) system with cost-efficient multiplier-less methods, which are more suitable for large-scale neural networks. The CORDIC-based Purkinje model has been compared with the original model in terms of the voltage activities, dynamic mechanisms, precision, and hardware resource utilization. The results show that the CORDIC-based Purkinje model can reproduce the same biological activities and dynamical mechanisms as the original model with slight deviation. In the aspect of the hardware implementation, it can use only logic resources, so it provides an efficient way for maximizing the FPGA resource utilization, thereby expanding the scale of neural networks that can be implemented on FPGAs.

Keywords: Purkinje, multiplier-less, coordinate rotation digital computer (CORDIC), field-programmable gate array (FPGA), digital implementation

OPEN ACCESS

Edited by:
Stefano Brivio,
Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Saeed Safari,
University of Tehran, Iran
Chao Wang,
University of Science and Technology of China, China

****Correspondence:***
Guosheng Yi
guoshengyi@tju.edu.cn

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *30 May 2019*
Accepted: *24 September 2019*
Published: *15 October 2019*

Citation:
Hao X, Yang S, Wang J, Deng B, Wei X and Yi G (2019) Efficient Implementation of Cerebellar Purkinje Cell With the CORDIC Algorithm on LaCSNN. Front. Neurosci. 13:1078. doi: 10.3389/fnins.2019.01078

INTRODUCTION

The cerebellum is a very important part of the human brain and associated with many important functions with a large number of incoming and outgoing connections between the brain, brainstem, and spinal cord. These functions are not only relevant to motor control including error correction (Doya, 2000; Llinas, 2009), tracking movements (Paulin, 1993; Miall et al., 2000), and coordinated movements (Thach et al., 1992; Heck et al., 2007) but also relevant to many non-motor functions such as linguistic prediction, word generation, emotional control, and so on (Leiner et al., 1993; Schmahmann and Caplan, 2006; Pleger and Timmann, 2018). Purkinje cells (PCs) make up the middle layer of the cerebellum, Purkinje layer, which is responsible for receiving information from the cerebellar granule cell (GC) synapses through parallel fibers (PF) and climbing fibers (CF) in brainstem. In addition to being all the constituent cells of the cerebellar Purkinje layer, PCs also directly connect to deep cerebellar nuclei cells, which are the main output cells of cerebellum. So, it is obvious that PCs play the most important role in the information processing of the cerebellum. Besides, PCs are responsible for cerebellar motor learning (Gilbert and Thach, 1977) with the information stored in the synapses with granule cells. The information is presented as the variation of synaptic strength according to the error signals carried by CFs through spike timing-dependent plasticity (STDP), which consists of long-term potential (LTP) and long-term depression (LTD) (Ito and Kano, 1982; Han et al., 2000; Medina et al., 2000). This learning mechanism can be obviously observed in classical eyeblink conditioning experiments (Bao et al., 2002) and cerebellar

vestibulo-ocular reflex (VOR) (Blazquez et al., 2003; Masuda and Amari, 2008), which are mainly caused by the function of PCs.

There are two calculation modes for simulation spiking neurons or spiking neural networks, serial computing mode, and the parallel computing mode (Yang S. M. et al., 2019). The serial computing mode is mainly based on some computer simulation software that is incompatible with the parallel computing features of real neural systems. In order to achieve these in a more biological way, more and more neuroscientists prefer to implement neurons and neural networks with parallel computing mode. Analog very Large-Scale Integration (VLSI), Graphics Processing Unit (GPU), and Field Programmable Gate Array (FPGA) are the three most used platforms with parallel computing capacity. Analog VLSI is an efficient analog-based method for hardware implementation of spiking neurons and neural networks because it can realize the non-linear function directly (Han, 2005; Hsieh and Tang, 2012). However, it cannot be flexibly changed once formed, so it is more suitable for well-defined circuits. In addition, its high cost and long development cycle also limit the application range. GPU provides a digital implementation method for spiking neurons and neural networks with its powerful parallel calculation ability and many researches have been carried on GPUs (Igarashi et al., 2011; Yamazaki and Igarashi, 2013). However, the kernel-launch method used on GPU and the limited bandwidths are obstacles for dealing with a lot of data. Compared to the two methods above, FPGA has many advantages for realizing the neural circuits. On one hand, the flexible reconfigurability and parallel computing architecture can perfectly meet the requirements for exploring characteristics of not only spiking neurons but also the large-scale spiking neural networks; on the other hand, its low area and power consumption also make it popular in neurosciences (Yang et al., 2017, 2018a). In this work, the neuron is implemented on the Large-Scale Conductance-Based Spiking Neural Networks (LaCSNN) system first proposed by Yang S. et al. (2019). The system consists of six Altera EP3SL340 FPGAs and is designed to simulate large-scale spiking neural networks with digital neuromorphic architecture. Its powerful storage capacity, high calculation speed, and sufficient resources make it an effective tool for neuroscience researches.

Although the advantages of FPGA are very prominent, the disadvantages are also distinct. Most of the resources on FPGA are logic resources; the lack of memory and multiplier resources often limits the scale when implementing neural networks. As a kind of digital systems, it is difficult to implement the non-linear functions directly. To solve these problems, many methods have been proposed. One of the most frequently used methods is to store the function values in a storage area with continuous address space in advance, which is called look up table (LUT) realization. When used, the function value can be obtained by addressing. This method is very easy but costs much memory resources. Besides, the use of LUTs increases the duration of reconstruction when changing model parameters. Another method, Taylor series approximation, is to replace the non-linear function in the neighborhood with an n-order polynomial approximation for a certain error. This method can make a trade-off between LUT resources and multiplier resources with different approximation order, but it still needs these resources (Lee and Burgess, 2003). The piece-wise linear (PWL) approximation (Julian et al., 1999) is a more efficient method to solve these problems but there are two main cons: one is there will be unavoidable error due to the use of several linear segments; the other is that it needs to recalculate when the non-linear function changes. So, in this work, we propose a non-multiplier and non-LUT method with the CORDIC algorithm for implementing the cerebellar Purkinje model on FPGA.

One of the main reasons for implementing single neurons with optimization algorithms on FPGA is to lay a foundation for realizing large-scale spiking neural networks. Many researches have been carried out in recent years. Yang et al. (2018b) propose a series of techniques for implementing a conductance-based neuron model that is beneficial for building large-scale neural networks. Soleimani et al. (2012) implement a classic Izhikevich model using PWL method to prove that the method can simplify the hardware implementation with showing similar dynamic behaviors. Ambroise et al. (2013) also implement an Izhikevich model on FPGA, but it is mainly to propose an architecture to reproduce a neural network with only one computation core (one neuron) based on one multiplier. Bonabi et al. (2012) implement a Hodgkin–Huxley (H–H) single neuron with the CORDIC algorithm and some LUTs that show high precision with more compact used logic.

There are also many researches about implementing the CORDIC algorithm on FPGA. Valls et al. (2002) evaluate some methods for the CORDIC algorithm and realize a variable precision method using conventional arithmetic on FPGA. Liu et al. (2014) implement a modified CORDIC

TABLE 1 | Conductance parameters of cerebellar Purkinje cell.

Current type	g_i	E_i	x_∞ (α_x)	τ_x (β_x)
Potassium	10	−95	$1/\{1+\exp[-(V+29.5)/10\,]\}$	$0.25+4.375\exp[(V+10)/10\,],\ V\le -10$ $0.25+4.375\exp[-(V+10)/10\,],\ V>-10$
Sodium	125	50	$1/\{1+\exp[(V+59.4)/10\ .7]\}$ $m_\infty=1/\{1+\exp[-(V+34.5)/10\,]\}$	$0.15+1.15/\{1+\exp[(V+33.5)/15\,]\}$
Calcium	1	125	$1.6/\{1+\exp[-0.072(V-5)]\}$	$0.02(V+8.9)/\{\exp[(V+8.9)/5]-1\}$
M	0.75	−95	$0.02/\{1+\exp[-(V+20)/5]\}$	$0.01\exp[-(V+43)/18]$
Leak	2	−70	–	–

algorithm that reduces the utilization of ROM resources and power consumption. Garcia et al. (2006) realize a pipelined CORDIC architecture with solution for overflow and quadrant correction and successfully generating sine and cosine waves. Muñoz et al. (2010) propose a floating-point CORDIC FPGA implementation for calculating transcendental functions. The FPGA implementation of the CORDIC algorithm can give full play to the advantages of FPGA and utilize hardware resources to realize an optimization scheme combining hardware and algorithm. The pipelined computational structure of FPGA can also enhance the real-time performance of the CORDIC algorithm, minimizing the computational delay due to the iterative operations. Therefore, the CORDIC algorithm can be widely applied to real-time high-quality signal processing with high-performance requirements.

The remaining parts of this work are arranged as follows. In section Neuron Model, the original model and modified CORDIC model of cerebellar PC are presented. The CORDIC algorithm used is also introduced in this section. Section Hardware Implementation Based on LaCSNN describes the details of hardware implementation. The results of software simulation and hardware simulation are shown in section Results. We also compare and analyze the result between the original model and the CORDIC model with various evaluation indicators for both the two simulations. The behaviors of a network with this neuron are also presented. section Discussion illustrates the discussion and conclusion for this work.

NEURON MODEL

Original Purkinje Model

During the exploration of PCs, many mathematic models have been built for different research interests (De Schutter and Bower, 1994a,b; Khaliq et al., 2003). Many models are either too detailed to form a large-scale neural network or too simple to have many basic biological characteristics. For the starting point of our implementation, which is to propose a method for simplifying a single neuron model with relatively high biological plausibility and make contributions to build large-scale networks on FPGA, we choose an H–H (Hodgkin and Huxley, 1952)-based model proposed by Miyasho et al. (2001) and Middleton et al. (2008), which consists of 32 ionic channels and simplified by Kramer et al. (2008) to 5. The membrane potential is shown as follows:

$$\begin{aligned} C\frac{dV}{dt} &= -g_k n^4 (V - E_k) - g_{Na} m_\infty (V)^3 h (V - E_{Na}) \\ &\quad -g_{Ca} c^2 (V - E_{Ca}) - g_M M(V - E_M) \\ &\quad -g_L (V - E_L) - I \end{aligned} \tag{1}$$

where V represents the membrane potential, C represents the membrane capacitance, and g_i and E_i ($i \in \{k, Na, Ca, M, L\}$) are the maximum ionic conductance and reversal potentials for different ion channels, respectively. There are five ionic currents and an external stimulus current I in this model: a potassium current $I_k = g_k n^4 (V - E_k)$, a sodium current $I_{Na} = g_{Na} m_\infty (V)^3 h (V - E_{Na})$, a calcium current $I_{Ca} = g_{Ca} c^2 (V - E_{Ca})$, an M-current $I_M = g_M M (V - E_M)$, and a leak current $I_L = g_L (V - E_L)$. n, m, h, c, and M are gating variables for different ionic currents and the dynamics are described as follows:

$$x_\infty = \frac{\alpha_x}{\alpha_x + \beta_x}, \ \tau_x = \frac{1}{\alpha_x + \beta_x} \tag{2}$$

$$\frac{dx}{dt} = \frac{x_\infty - x}{\tau_x} \tag{3}$$

x_∞ is the state variable, τ_x is the time constant for $x \epsilon \{n, m, h, c, M\}$, α_x, and β_x are relevant functions, and all of these are functions of membrane potential V. The detailed parameter values and the description for ionic currents dynamics are provided in **Table 1**.

CORDIC-Based Purkinje Mode

In order to make the implementation more suitable for building large-scale neural network and improve the calculation speed, we modify the original Purkinje model to save memory and multiplier resources with the CORDIC algorithm and introduce as follows.

The CORDIC algorithm is originally developed in Volder (1959) as an algorithm for calculating trigonometric and hyperbolic functions and first used in navigation systems. Then, a unified CORDIC algorithm is proposed in Walther (1971). By introducing a coordinate system parameter m, the circular rotation, hyperbolic rotation, and linear rotation are unified into the same CORDIC iterative equations, which provide a premise for the multifunction of the same hardware implementation. The essence of the CORDIC algorithm is to approximate a certain rotation angle by using a set of constant angle bases. It is possible

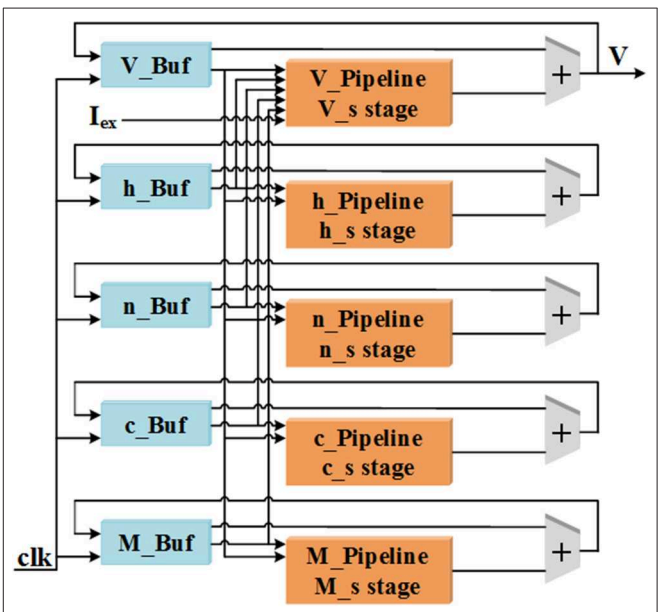


FIGURE 1 | The diagram of pipeline structure for the five variables used in cerebellar Purkinje model.

to accurately calculate many non-linear functions by using vector repeated rotation. Its iterative equation is as follows:

$$\begin{cases} X_{i+1} = X_i - mq_i Y_i \cdot 2^{-i} \\ Y_{i+1} = X_i + q_i X_i \cdot 2^{-i} \\ Z_{i+1} = Z_i - q_i \cdot \theta_i \end{cases} \quad (4)$$

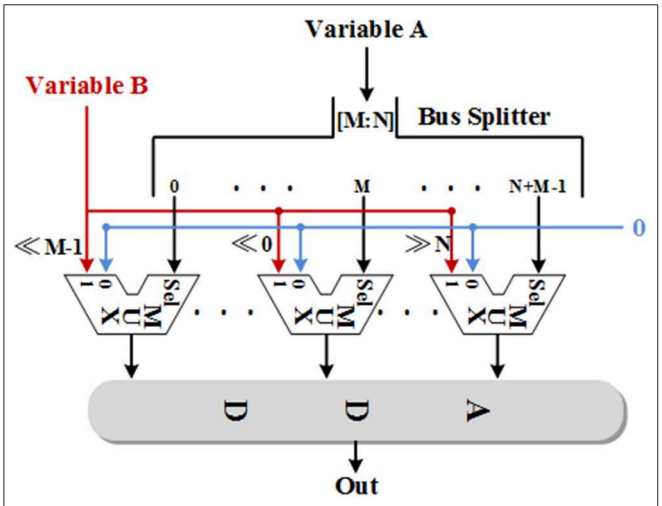


FIGURE 2 | The schematic diagram of the functional shift multiplier (FSM).

where X_i and Y_i are the value before rotation, X_{i+1} and Y_{i+1} are the value after rotation, q_i is the direction of rotation, θ_i and the relationship between m value and rotation mode are both described in Equation (5).

$$\theta_i = \begin{cases} \tanh^{-1}\left(2^{-i}\right), & m = -1, \text{ hyperbolic rotation} \\ 2^{-i}, & m = 0, \text{ linear rotation} \\ \arctan(2^{-i}), & m = 1, \text{ circular rotation} \end{cases} \quad (5)$$

The exponential operations and divisions used in this paper are calculated in the hyperbolic rotation and linear rotation modes, respectively. The division can be easily gotten with $\theta_i = 2^{n-i}\left(\theta_i = 2^{i-n}\right)$, where n determines the calculation range. As for the exponential operations, since through hyperbolic rotation we can only obtain the values of $cosh\theta$ and $\sinh\theta$, e^{θ} needs to be calculated with the basic relationship between hyperbolic functions $\sinh\theta + cosh\theta = e^{\theta}$. According to Equation (5), we can know that the convergence domain is limited by $\tanh^{-1}\left(2^{-i}\right)$. In detail, the maximum value it can be calculated is determined by the sum of all the angles, which is approximately equal to 1.1182. It is obvious that it cannot meet the calculation requirements of this model. So, before calculating, the input variable needs to be preprocessed to expand the convergence domain. Suppose the input variable is θ, it can be divided into integer part A and fractional part b after being divided by ln2 just as Equation (6)

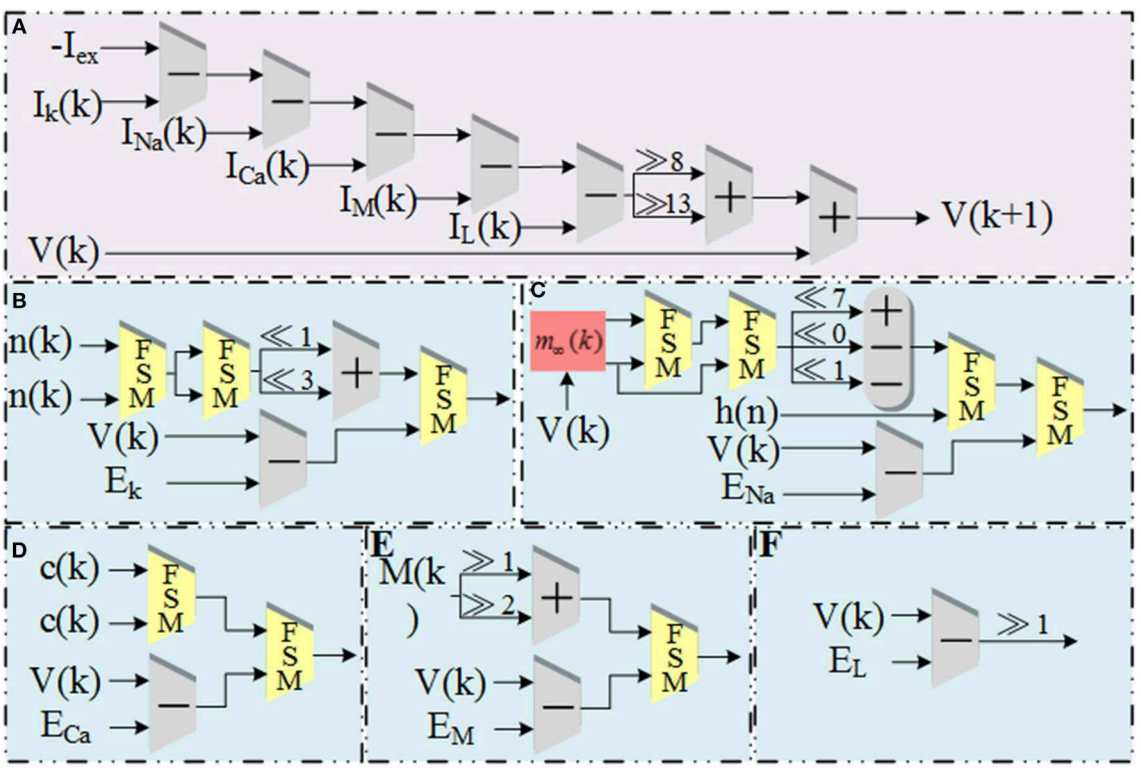


FIGURE 3 | The schematic diagram of data flow for V and currents in the modified model. **(A)** The pipeline of "*V*." **(B)** The pipeline of "I_k." **(C)** The pipeline of "I_{Na}." **(D)** The pipeline of "I_{Ca}." **(E)** The pipeline of "I_M." **(F)** The pipeline of "I_L.".

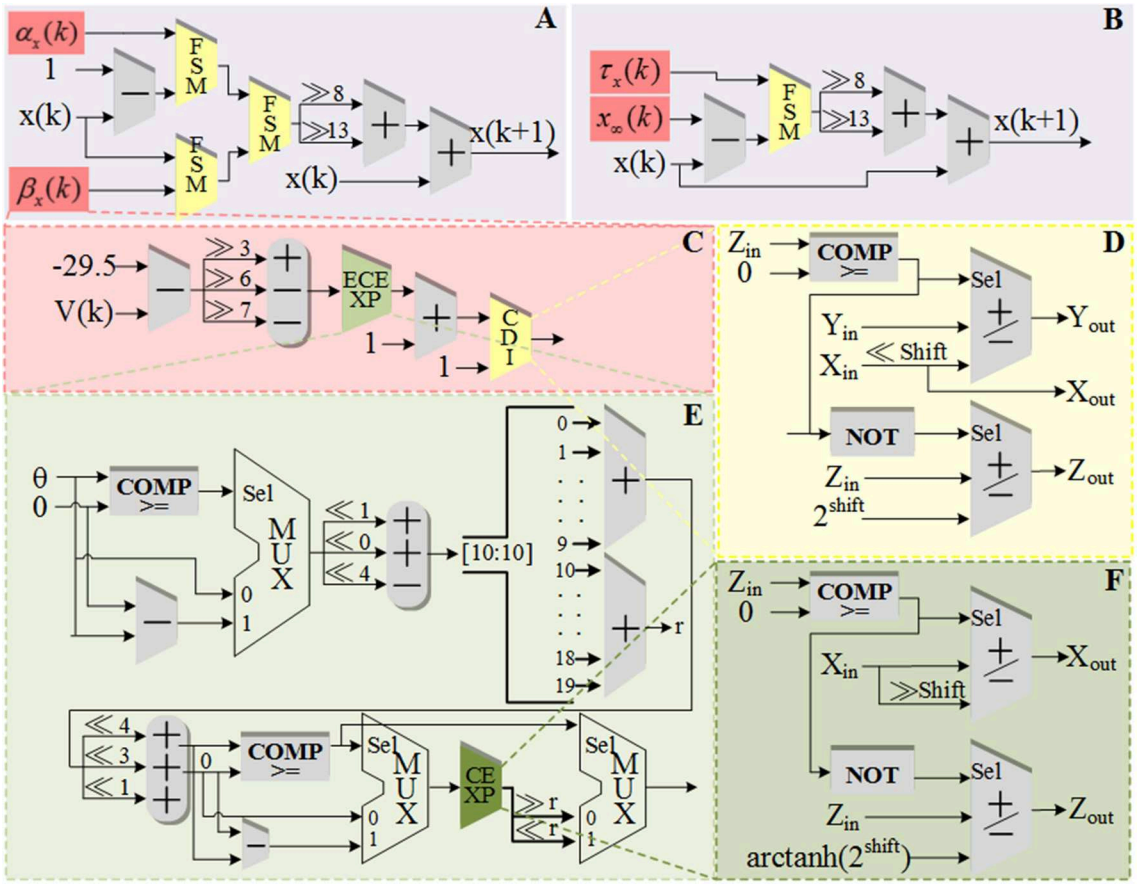


FIGURE 4 | The schematic diagram of data flow for gating variables, a state variable and the detailed structures of one iteration in each CORDIC operations. **(A)** The pipeline of "*c*" and "*M*." **(B)** The pipeline of "*h*" and "*n*." **(C)** The pipeline of "h_∞." **(D)** One iteration of CORDIC division (CDI). **(E)** One iteration of expand CORDIC exponential (ECEXP). **(F)** One iteration of and CORDIC exponential (CEXP) in order.

and the exponential operation will be $e^{\theta} = 2^{A} \cdot e^{bln2}$.

$$\frac{\theta}{ln2} = A + b \tag{6}$$

There are two reasons for choosing $ln2$, one is the exponential operation of integer part can be transmitted to power of 2 directly, which can be easily implemented by shifting; the other one is that $b \cdot ln2$ is smaller than 1.1182, which is just within the convergence domain. With this method, we can perform exponential operations in any range. After careful consideration, in the case of ensuring high precision and minimizing resource consumption, the iterations in this work are chosen as: 10 for exponential operations and 12 for divisions with $n = 2$ for θ_i.

HARDWARE IMPLEMENTATION BASED ON LaCSNN

To the best of our knowledge, there are no works on FPGA implementation for cerebellar PC model based on H–H form. The detailed implementation method is described in the following.

In order to be implemented on a digital system, the differential equations of the Purkinje model should be solved with the Euler method. The Euler method is suitable for hardware implementation with its easy operation and adequate precision. The discretization results with a mathematical finite-difference method are shown as follows, Equation (7) is for membrane potential and Equation (8) shows the results of other variables:

$$\begin{aligned} \mathrm{V}[\mathrm{k}+1] &= \mathrm{V}[\mathrm{k}] \\ &+ \left\{ \begin{array}{l} g_{\mathrm{k}} n^4 \left(\mathrm{V}[\mathrm{k}] - \mathrm{E_k}\right) - g_{\mathrm{Na}} \mathrm{m}_\infty{}^3 \mathrm{h} \left(\mathrm{V}[\mathrm{k}] - \mathrm{E_{Na}}\right) \\ -g_{\mathrm{Ca}} \mathrm{c}^2 \left(\mathrm{V}[\mathrm{k}] - \mathrm{E_{Ca}}\right) \\ -g_{\mathrm{M}} \mathrm{M} \left(\mathrm{V}[\mathrm{k}] - \mathrm{E_M}\right) - g_{\mathrm{L}} \left(\mathrm{V}[\mathrm{k}] - \mathrm{E_L}\right) - \mathrm{I} \end{array} \right\}^* \left(\frac{\Delta \mathrm{t}}{\mathrm{C}}\right) \end{aligned} \tag{7}$$

$$\left\{ \begin{array}{ll} \mathrm{h}[\mathrm{k}+1] & = \mathrm{h}[\mathrm{k}] + \frac{\mathrm{h}_\infty - \mathrm{h}[\mathrm{k}]}{\tau_{\mathrm{h}}}{}^* \Delta \mathrm{t} \\ \mathrm{n}[\mathrm{k}+1] & = \mathrm{n}[\mathrm{k}] + \frac{\mathrm{n}_\infty - \mathrm{n}[\mathrm{k}]}{\tau_{\mathrm{n}}}{}^* \Delta \mathrm{t} \\ \mathrm{c}[\mathrm{k}+1] & = \mathrm{c}[\mathrm{k}] + \left(\alpha_{\mathrm{c}}^* \left(1 - \mathrm{c}[\mathrm{k}]\right) - \beta_{\mathrm{c}}^* [\mathrm{k}]\right)^* \Delta \mathrm{t} \\ \mathrm{M}[\mathrm{k}+1] & = \mathrm{M}[\mathrm{k}] + \left(\alpha_{\mathrm{M}}^* \left(1 - \mathrm{M}[\mathrm{k}]\right) - \beta_{\mathrm{M}}^* \mathrm{M}[\mathrm{k}]\right)^* \Delta \mathrm{t} \end{array} \right. \tag{8}$$

where k is the iterations and Δt is the time step for the Euler method. Generally speaking, the precision is inversely

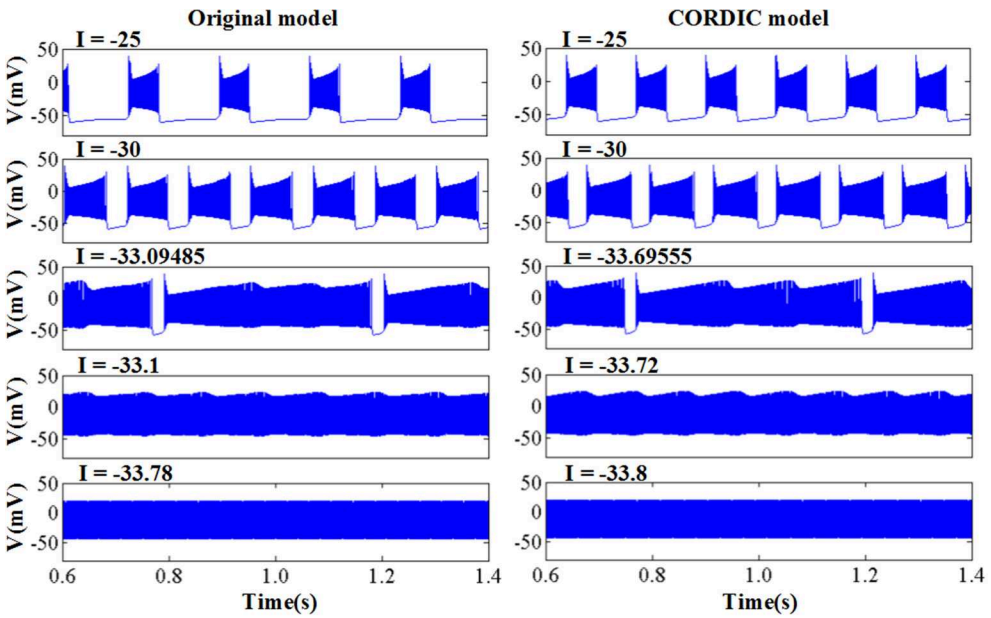


FIGURE 5 | The voltage activities of original model and the modified CORDIC model.

proportional to the value of the time step, and in this work, the time step is set to be 0.004 ms, which is the same in Traub et al. (2003).

Since floating-point operations take up a lot of resources and require a long calculation time, the FPGA implementation usually uses fixed-point calculations under the premise of meeting the computing needs. The bit width of fixed-point calculation is another important factor affecting precision or even implementation result besides the time step. The selection of bit width can be divided into an integer part and a fractional part, which can be estimated according to the software simulation results. For example, the range of V in this work is -60 to 40 mV, so the integer part should be 7 at least for $2^7/2 > |-60|$. If we need the precision to be 0.001, the bit width of the fractional part that directly determines it should be 10. In the calculation process of this work, the range of most variables is from -100 to 100, so the bit width of the integer part for most logical operation modules is 8. It should be pointed out that one of the variables reached 8,000 in the process of calculating h_∞, so the bit width of the integer part for related logical operation modules is 14. In order to guarantee the precision of spiking and dynamics, the bit width of the fractional part is chosen to be 15.

In this work, all of the variables including V, n, h, c, and M in Equations (7) and (8) are designed to be realized with pipeline structures. The overall pipeline schematic is shown in **Figure 1**. There are two parts for one pipeline structure, the "Pipeline" includes all the calculations in Equations (7) and (8) and the "Buf" consists of a certain number of buffers to store the calculation results for each variable. This implementation method can improve the throughout and calculation efficiency of the LaSCNN system.

It is well-known that one of the factors limiting the size of the network implemented on the FPGA is the limited multiplier and memory resources. Due to most of the resources on FPGA is logic resources, all the multiplication, division, and exponential operations are replaced by adders and shifters in this work. The division and exponential operations are implemented with the CORDIC method as described in section Neuron Model and the multiplications are implemented through two methods, which are more efficient than the CORDIC algorithm. On one hand, we can only use shifters and adders for the multiplications with a constant multiplicand. The main idea of this method is decomposing the constant into a summation of several $(-1)^k 2^n$ with different values of k and n and shifting the multiplier according to the values. k is 0 or 1, which determines the sign bit and the absolute value of n determines the number of bits that the multiplier need to be shifted. The direction of shifting is decided by the sign bit of n. If n is negative, the multiplier needs to be shifted to the left; if n is positive, the multiplier needs to be shifted to the right. On the other hand, the rest of multiplications are realized with functional shift multipliers (FSMs) with the structure shown in **Figure 2**. As we can see, one of the variables is split into single bits through a bus splitter and output to the multiplexers as enable signals. That is, if a bit of this variable is 0, the output of the corresponding multiplexer is 0; if a bit of this variable is 1, the output of the corresponding multiplexer will be the value of the other variable after shifting. The number of shifters is related to the bit position of the previous variable. Finally, add all the values from multiplexers and then the multiplication result can be obtained. It is worth noting that there is always one slow variable in a multiplication and splitting this variable is a better choice when using FSMs. Due to each additional bit of an FSM consumes a shifter and a

multiplexer, all the FSMs used in this work are designed to fit the inputs in order to minimize the use of logical resources. The overall bit width is between 13 and 18.

Besides, when implementing the exponential operations with the CORDIC algorithm, we find that the iterative structure of X and Y is very similar, so we merge the two iterations and then the iterative Equation (4) becomes like Equation (9), which can save about one-third of the logical resources without changing the results. Under the premise of ensuring accuracy, the iterations of CORDIC is set to 20 for division and 10 for exponential operations in this work.

$$\begin{cases} X_{i+1} = X_i + q_i X_i \cdot 2^{-i} \\ Z_{i+1} = Z_i - q_i \cdot \tanh^{-1}\left(2^{-i}\right) \end{cases} \quad (9)$$

The schematic diagram of dataflow for V is shown in **Figure 3**. **Figure 3A** shows the data flow of V with ionic currents and external current in the modified model. The detailed structures for the ionic currents are shown in **Figures 3B–F**. The schematic diagrams for the other four variables and the CORDIC algorithm are shown in **Figure 4**. **Figure 4A** is the structure for c and M and **Figure 4B** is for h and n. Due to space limitations, we only give a typical example for CORDIC-based non-linear function that includes all the compartments used in other functions in the **Figure 4C**. **Figures 4D–F** show the detailed structure for non-linear operations realized with the CORDIC algorithm. For the sake of simplicity, the figures only give the structure for one iteration of each operation and it will need several same structures with different values of "shift" for realizing the calculation. There is no LUTs and multipliers in all the designs so we can get a non-multiplier and non-LUT implementation through this method, which has potential for large-scale cerebellum realization on LaCSNN.

RESULTS

Comparison of Software Simulation Results

The original and CORDIC-based cerebellar Purkinje model are both simulated with MATLAB v2014a. The time step for software simulation is 0.004 ms. The membrane potential waveforms of two models are shown in **Figure 5**. As shown in the figure and taking the original model as an example, the burst activity with increasing amplitude can be seen when $I = -25$ (Bursting I). With the decrease of I, the interburst intervals decrease and the bursting becomes more durable (Bursting II). When $I = -33.09485$ (I'), a value in the critical region, the voltage activity presents bursts interspersed with amplitude modulation, which is the new type of activity founded in Kramer et al. (2008). The continuous decrease of I will make bursting disappear gradually, from the only spiking amplitude modulation ($I = -33.1$) to the complete fast spiking ($I < -33.2$). The CORDIC model can successfully reproduce the same voltage activities as performed in the original model but with different values of I. That is caused by the differences of the non-linear function realized with two different methods. There are inevitable errors of the CORDIC algorithm due to the iterative operations, but it will not affect overall results and can meet our requirements. To show it clearly, the detailed spiking waveforms for fast spiking of the two models are shown in **Figure 6**. We can see that there exists a certain but small difference in spiking interval and the disparity of amplitude is also limited.

TABLE 2 | The value of RMSE and mAE of non-linear function realized with CORDIC.

CORDIC functions	RMSE	mAE
n_∞	0.0013	0.0098
τ_n	0.0010	0.0068
h_∞	8.80×10^{-4}	0.0073
τ_h	0.0012	0.0102
m_∞	0.0013	0.0082
α_c	0.0012	0.0146
β_c	0.0015	0.0299
α_M	1.42×10^{-5}	1.28×10^{-4}
β_M	5.22×10^{-5}	4.29×10^{-4}

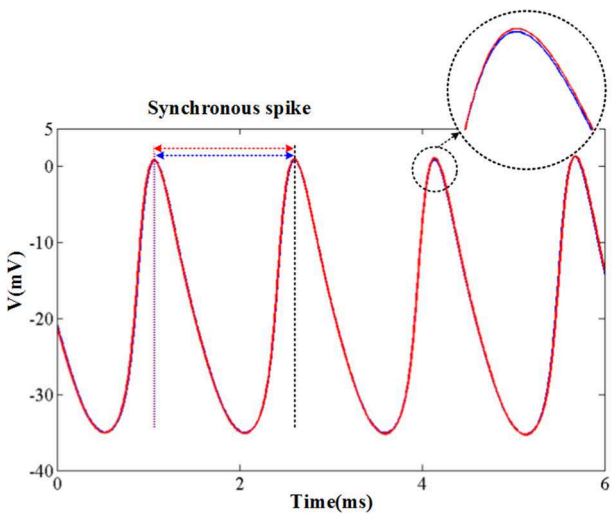


FIGURE 6 | The spikes under fast spiking mode of the two models. The blue lines and red lines are for original model and the CORDIC model, respectively. The solid lines represent the spike waveform. The dash lines show the spike moment and the black one shows the synchronous spike moment of the two models. Horizontal arrows represent spiking intervals.

TABLE 3 | The value of ERRt and Corr of five different types of spikes.

Spike type	ERRt	Corr
Bursting I	0.0011	0.9980
Bursting II	0.0006	0.9890
Bursting with amplitude modulation	0.0037	0.9835
Amplitude modulation	0.0041	0.9922
Fast spiking	0.0005	0.9817

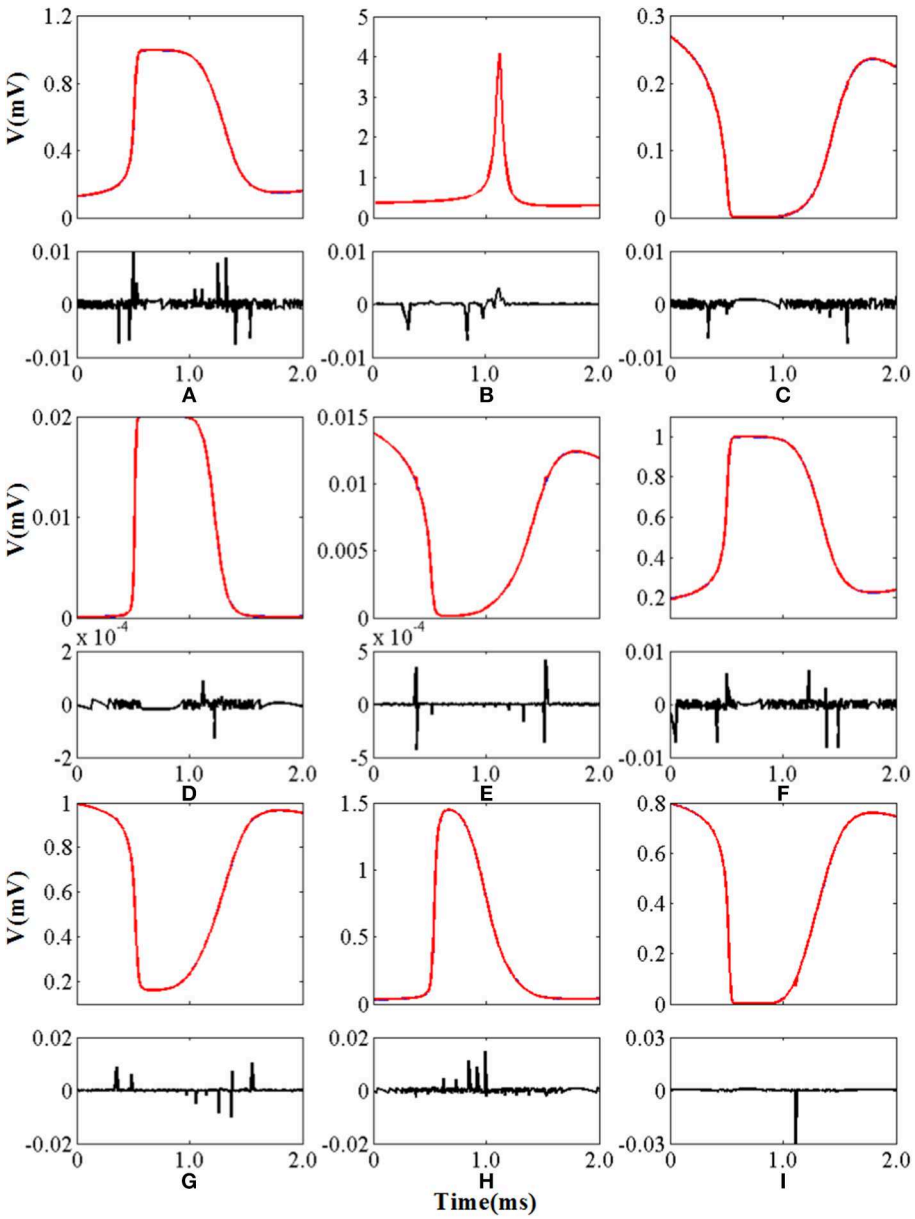


FIGURE 7 | The curve and absolute error of nine nonlinear functions. The top panel of each figure is the function curve and the bottom panel of each figure is the absolute error. Blue lines represent the original nonlinear functions and red lines represent the CORDIC functions. **(A–I)** Represents n_∞, τ_n, h_∞, τ_h, α_c, β_c, α_M, β_M, and m_∞ in order.

Error Analysis

In order to evaluate the CORDIC model more accurately, we use different methods to quantify the error between the two models to get a more comprehensive understanding of the CORDIC model. The detailed description of the method is as follows.

1) Maximum absolute error (mAE):

The absolute error (AE) is defined as the difference between the absolute values of the voltage of the two models. The maximum absolute error is defined as the difference between the voltage maximum absolute values of the two models. The two indexes can be calculated with the following equation:

$$\begin{cases} \mathrm{AE\,(i)} &= |\mathrm{F_{ori}\,(i)} - \mathrm{F_{CORDIC}\,(i)}| \qquad\quad , i = 1, 2, \ldots, N \\ \mathrm{mAE} &= \max\left(|F_{\mathrm{ori}}\,(\mathrm{i}) - F_{\mathrm{CORDIC}}\,(\mathrm{i})|\right), i = 1, 2, \ldots, N \end{cases} \tag{10}$$

where $F_{\mathrm{ori}}(i)$ represents the relevant value of the original model and the $F_{\mathrm{CORDIC}}(i)$ represents the relevant value of the CORDIC model. The symbol $|\cdot|$ is used to get the absolute value and the

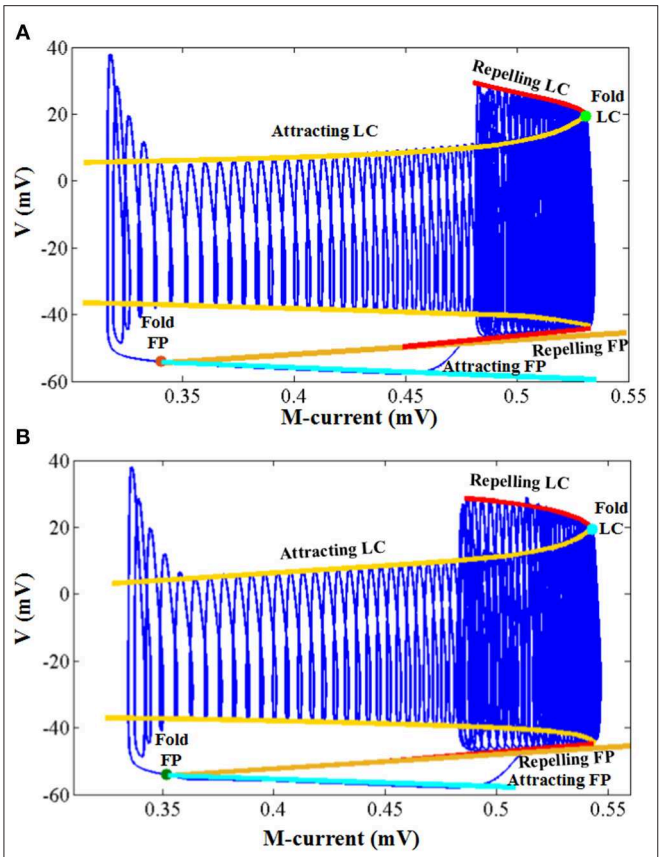


FIGURE 8 | Comparison of the dynamics between the original model **(A)** and the CORDIC model **(B)** with $I = I'$. The "FP" and "LC" represent fixed points and limit cycles, respectively.

max(·) is used to get the maximum value. In order to get the error of the CORDIC algorithm in detail, we calculated the mAE of all nine non-linear functions realized by the CORDIC algorithm under a complete spiking waveform for a more convincing effect. The values are summarized in **Table 2**.

2) Root mean square error (RMSE):

The root mean square error is a typical measurement index for two value differences and very sensitive to very large or very small errors. We also calculate the RMSEs of all nine non-linear functions with the equation below and summarized in **Table 2**.

$$\mathrm{RSME} = \sqrt{\frac{1}{N}\sum_{i-1}^{N}\left(\mathrm{F_{ori}(i)} - \mathrm{F_{CORDIC}(i)}\right)^2} \tag{11}$$

3) Error of spikes' timing (ERRt):

The error of spikes' timing reflects the difference in spiking interval between the two models. It can not only directly reflect the difference in spiking periodicity but also indirectly reflect the difference in the shape of the spiking waveform. To calculate the spiking interval, we should find a synchronous spike at first just as **Figure 5**. Then, measure the time interval between the synchronous spike and the previous or next spike. The error can be calculated as follows:

$$\begin{cases} \mathrm{ERRt} &= \left|\frac{\Delta \mathrm{T_{CORDIC}} - \Delta \mathrm{T_{ori}}}{\Delta \mathrm{T_{ori}}}\right| \\ \Delta \mathrm{T} &= \mathrm{t_{syn}} - \mathrm{t_{pre}} \text{ or } \Delta \mathrm{T} = \mathrm{t_{nex}} - \mathrm{t_{syn}} \end{cases} \tag{12}$$

where ΔT_{CORDIC} represents the spiking time interval of the CORDIC model and the ΔT_{ori} represents the spiking time interval of the original model.

4) Correlation coefficient (Corr):

The correlation coefficient is an amount of linear correlation between the two groups of data. For the spiking waveforms of the two kinds of neurons, the larger the correlation coefficient is, the more similar the two waveforms are, and the maximum value of Corr is 1. As shown in Equation (13), the Corr is generally defined as the ratio of covariance to variance product of two sets of data. The covariance and the variance of the two sets of data can be obtained by Equation (14).

$$\mathrm{Corr} = \frac{\mathrm{cov}(\mathrm{V_{ori}}, \mathrm{V_{CORDIC}})}{\sigma(\mathrm{V_{ori}})\sigma(\mathrm{V_{CORDIC}})} \tag{13}$$

$$\begin{cases} \mathrm{cov}\left(\mathrm{V_{ori}}, \mathrm{V_{CORDIC}}\right) \\ = \sum_{i=1}^{n}\left(\mathrm{V_{ori}}\,(i) - \overline{\mathrm{V_{ori}}}\right)\left(\mathrm{V_{CORDIC}}\,(\mathrm{i}) - \overline{\mathrm{V_{CORDIC}}}\right) \\ \sigma(\mathrm{V}) = \sqrt{\sum_{\mathrm{i}-1}^{\mathrm{n}}\left(\mathrm{V(i)} - \overline{\mathrm{V}}\right)^2} \end{cases} \tag{14}$$

We calculate the ERRt at 20 different times and take the average as the final value and the Corr is calculated with the membrane voltage values 6 ms (about 3–4 complete spiking) after the start of spiking synchronization. Both the two indexes are measured with five spiking modes and the values are summarized in **Table 3**. It can be seen from the table that the ERRt of the five spiking modes are <0.005, which indicates that the difference between the spiking intervals is small. In addition, the Corr of the five discharge modes is also around 0.99, which indicates that the spiking waveform is very similar.

The curves of the nine non-linear functions of the original model and CORDIC model are shown in **Figure 7** with the shape of AE below. As can be seen from **Figure 7** and the two tables, the model implemented using the CORDIC algorithm has very small errors calculated by various methods and can meet the needs for building spiking neurons.

Dynamic Analysis

In order to evaluate the difference between the two models more comprehensively, we learn and compare the dynamical mechanisms in different discharge modes. Because small errors can cause large differences in dynamical diagrams, it is a good way to measure model consistency. For more convincing, we have implemented the dynamic mechanisms in Kramer et al. (2008): the bifurcation diagrams of voltage V and the slow variable M.

The simulation results of the entire system and its associated bifurcation diagram for bursting with amplitude modulation are

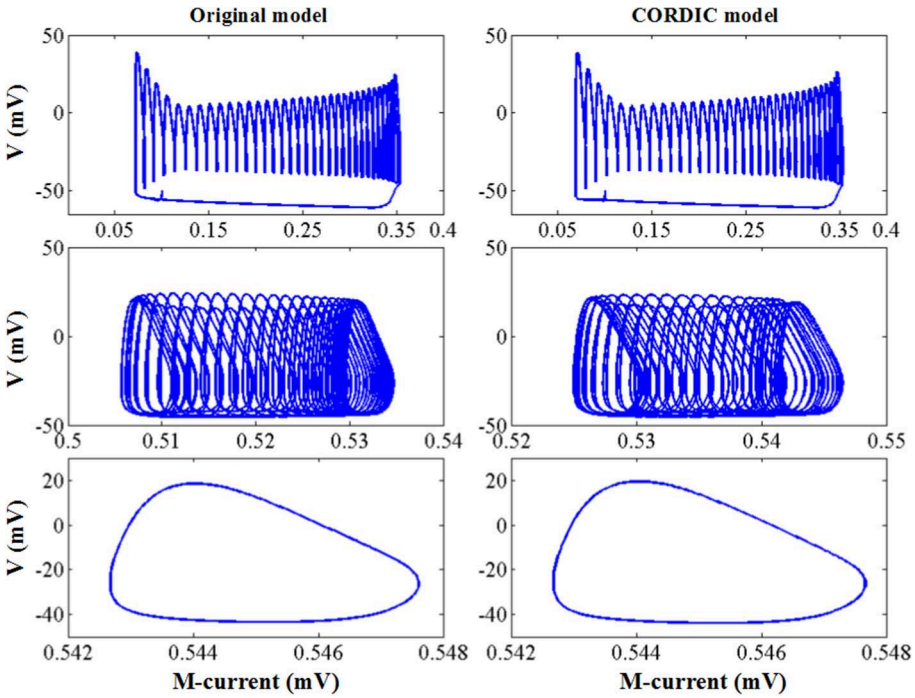


FIGURE 9 | The dynamic of the original model **(left)** and CORDIC model **(right)** for bursting, amplitude modulation, and fast spiking.

shown in **Figure 8A**, which is obtained from the original model, and **Figure 8B**, which is from the CORDIC-based model. For a clearer description, the portion of the M-current at (0.483, 0.546) mV is referred to as the fast subsystem. When the M-current is reduced to less than the voltage at fold of fixed points in the fast subsystem, the rapid discharge begins and the attracting and repelling fixed points are also merged at this point. After that, the voltage increases rapidly and the system enters the fast subsystem along the attraction curve of the limit cycle. During this period, the M-current gradually increases until it reaches a fold of limit cycles in the fast subsystem. Finally, the M-current decreases, and the dynamics of the system temporarily follows the repelling branch of limit cycles until the return fixed points (light gray) or limit cycles. It can be seen from the figure that the small errors of CORDIC make the shapes of the two figures slightly different, but the CORDIC models can still reproduce the results in the original paper very well.

Figure 9 shows the other three bifurcation diagrams for bursting, amplitude modulation, and fast spiking. M-current and calcium current play major roles in the switching of the spiking mode. When the hyper-polarization due to the M-current works ($I < I'$), the bursting occurs due to the victory of hyperpolarization, then the cell enters the stationary phase of bursting and spiking stops. When the calcium current works, its depolarizing effect prevents the hyperpolarization. Then, the stationary phase no longer appears, with the amplitude modulation spiking instead. As I continues to decrease, there only exists fast spiking. There are still small differences in these figures, but they are also able to reproduce the dynamical mechanisms that the cell follows.

Network Behavior

In this section, we present a network of two coupled PCs to verify the proposed method. The two PCs (V_{pre} and V_{post}) are all in the form of Equation (1) each with an extra added synaptic current I_{syn}. The pre-PC is set to an excitatory cell and the post-PC is set to an inhibitory cell. The pre one receives excitatory current I_{syn_post}through $GABA_A$ receptors and the post one receives inhibitory current I_{syn_pre} through AMPA receptors. The detailed synaptic current is shown as follows:

$$\begin{cases} \frac{dz}{dt} &= \frac{1+tanh(\frac{V}{10})}{2}\frac{1-z}{\tau_1} - \frac{z}{\tau_2} \\ I_{syn_pre} &= Wz_{pre}(V_{post} - V_{inh}) \\ I_{syn_post} &= Wz_{post}(V_{pre} - V_{ex}) \end{cases} \tag{15}$$

where W is neuron connection weight, z is the synaptic activation variable, and τ_1 and τ_2 are time delay constants, which for $GABA_A$ receptors are 0.5, 10 and for AMPA receptors are 0.2, 2, respectively. The other parameter values are: $W = 0.5$, $V_{ex} = 0$, $V_{inh} = -50$, I for pre and post cell is -25 and -34, respectively.

The simulation results are shown in **Figure 10**. When the two neurons are uncoupled, they both present the spike mode according to the value of I, bursting mode for the pre cell with $I = -25$, and fast spiking mode for the post cell with $I = -34$. When the two neurons are coupled, the bursting period of the pre cell becomes longer due to the excitatory synaptic current and the spiking mode of the post cell turns into bursting due to the

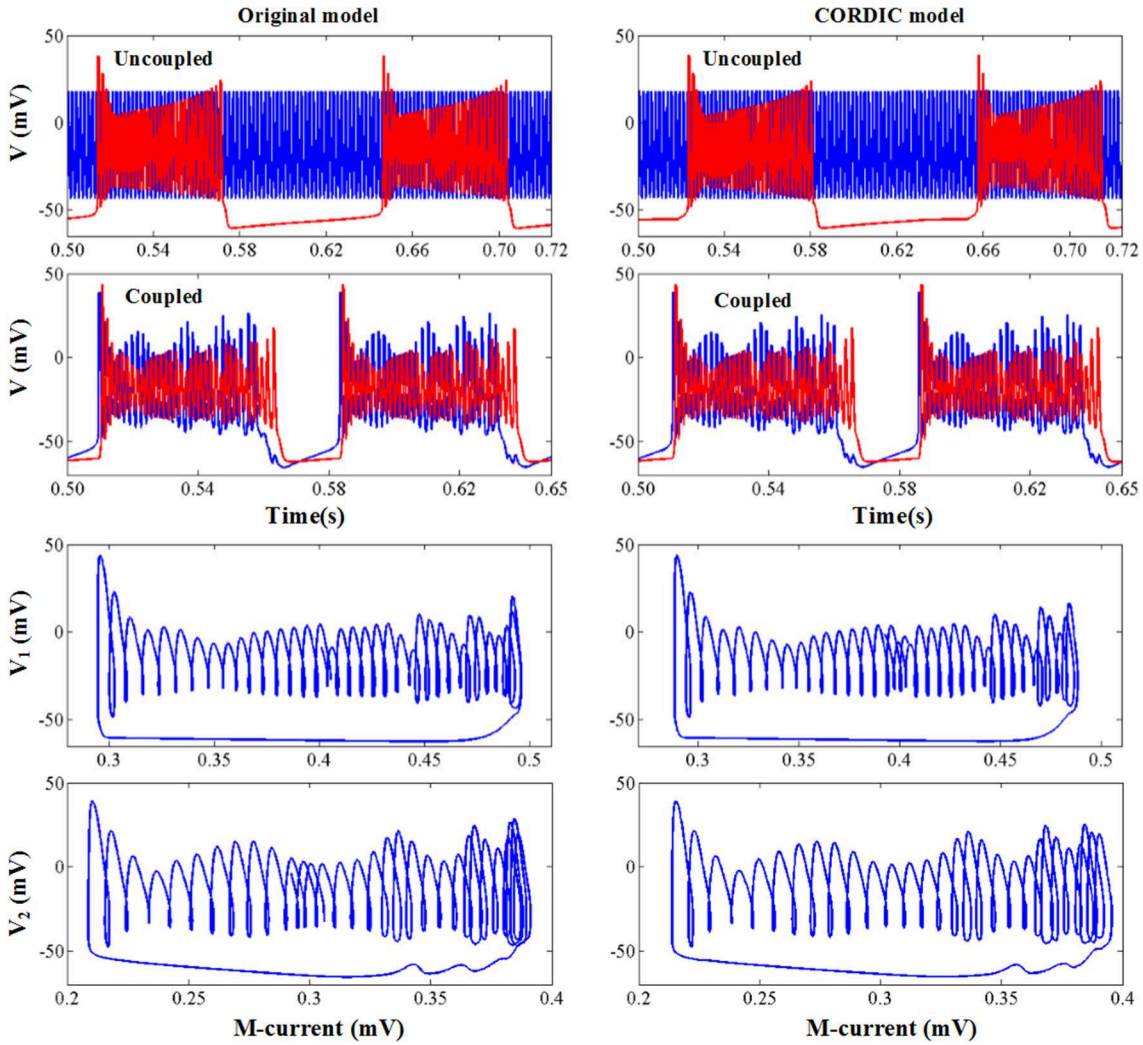


FIGURE 10 | The network behavior of the original model **(left)** and the CORDIC model **(right)**.

inhibitory synaptic current. It is worth mentioning that the peak value of each spike changes with time due to the interaction of the two neurons, and the dynamic behavior shows corresponding changes. We can see from **Figure 10** that no matter the spiking behaviors or the dynamic behaviors, the original model and the CORDIC model show a high degree of consistency, which indicates that the proposed method is also applicable for the neural network.

Hardware Implementation Result

The modified CORDIC cerebellar Purkinje model is built with the DSP Builder aided design toolbox in Simulink and then transformed to VHLD hardware language that can be compiled in Quartus II and downloaded to the LaCSNN system through USB-Blaster with Joint Test Action Group (JTAG) mode. In order to facilitate observation, the digital outputs from FPGA are transmitted to analog signals through a 16-bit dual-channel DA converter. The converter is also connected to an oscilloscope where the voltage activity of the model can be observed directly. The LaCSNN system and the voltage activity on the oscilloscope screen are both shown in **Figure 11**. The x-label and the y-label represent the time and voltage, respectively.

The comparison of the software simulation results and the FPGA implementation results for voltage activity is shown in **Figure 12**. To clearly present the difference between the two results, we give a partial spiking waveform. The overall shape of the voltage is the same, but the period and amplitude are different. The main reason is the usage of the approximation method and the fixed-point calculation on the hardware. The bifurcation diagrams for bursting, amplitude modulation, fast spiking, and bursting with amplitude modulation of the two simulation methods are shown in **Figure 13**; for the same reason, the basic shape of these diagrams is the same but the voltage values have deviations.

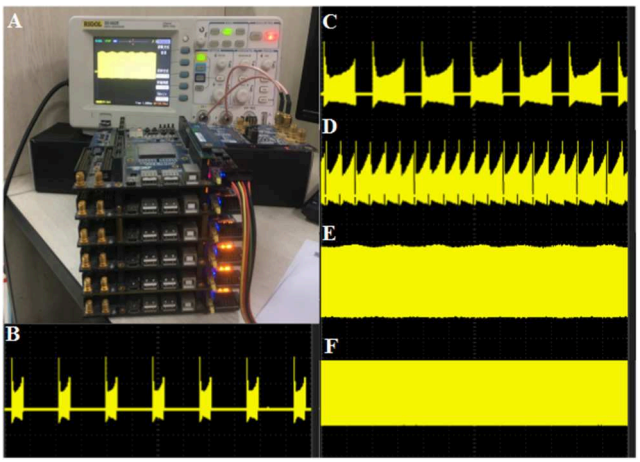


FIGURE 11 | The hardware implementation results of the CORDIC cerebellar Purkinje cell on the LaCSNN system. The x-label is time with 100 ms/cell and the y-label is voltage with 50 mV/cell. **(A)** The LaCSNN system and the oscilloscope. **(B)** The membrane potential of "Bursting I" mode. **(C)** The membrane potential of "Bursting II" mode. **(D)** The membrane potential of "Bursting with amplitude modulation" mode. **(E)** The membrane potential of "Amplitude modulation" mode. **(F)** The membrane potential of "Fast spiking" mode.

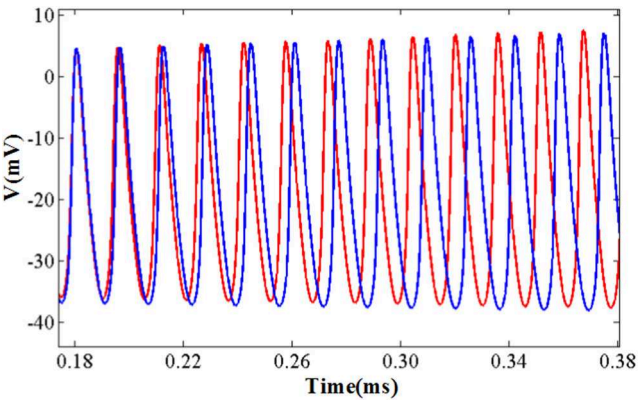


FIGURE 12 | The detailed spiking shape of software simulation and hardware implementation under the fast spiking mode. The blue line represents the software simulation result and the red line represents the hardware implementation result.

The resource utilization, working frequency, and power dissipation of the original and CORDIC model are summarized in **Table 4**. Due to the unroll iteration structure and a mass of multiplications, the logical elements used by the CORDIC model is more than the original model. However, the memory bits used by LUTs and the DSP block 18-bit elements used by multipliers can be reduced to zero. The power dissipation is a little more also due to the unroll iteration structure. For a clearer explanation, the same contents of the key algorithm of this method are summarized in **Table 5**. Comparing the three key algorithms, we can conclude that the FSM is more efficient than the CORDIC with less logic resources and high working frequency, which is why we do not use the CORDIC algorithm to realize multiplications. With the number of iteration increases (20 for division and 10 for exponent), the working frequency decreases due to the iterative structure, which affects the working frequency of the entire model. More importantly, it is obvious that there's no need for memory and multiplier resources for realizing the non-linear operations with high frequency and low power dissipation. It proves that, through this method, we can efficiently convert memory resources and multiplier resources into logical resources, which is of great significance to maximize the use of FPGA on-chip resources and improve the scale of neural network implementation.

DISCUSSION

There is a bottleneck for realizing a large-scale neural network with high biological precision neurons such as the model in this paper based on the H–H neuron model. These models have many conductance-based ionic currents that usually contain many non-linear functions and greatly increase the computational complexity. To solve this problem, many previous studies are working on FPGA resource optimization for spiking neurons with different methods (Ahmadi and Zwolinski, 2010; Bonabi et al., 2014; Hayati et al., 2016; Akbarzadeh-Sherbaf et al., 2018). Ahmadi and Zwolinski (2010) propose a method with PWL approximation for implementing the Izhikevich model. The non-linear operations in the model are only multiplications for there are no detailed ionic currents. The model complexity is relatively simple so the reference meaning for building high biological precision neurons is limited. Bonabi et al. (2014) implement an H–H-based model and a two-mini-column network with the CORDIC algorithm but it is only used for calculating exponent operations, but there are still some things to do to implement a large-scale neural network, because the multiplication and division operations account for a large proportion of the model and they still need multipliers and memory resources. Besides, there is no simplification for the iterative structure as we have done. Akbarzadeh-Sherbaf et al. (2018) use a general PWL approach to implement a randomly connected network with H–H models. If we just focus on one H–H model, the PWL approach can successfully realize the non-linear functions and improve the working frequency, but the precision is lower than the CORDIC algorithm for a sharp curve will certainly appear at the junction of the two linear sections. Besides, the approximate range of each linear part is only applicable to that set by the designer, so the linearization must be redesigned each time the model changes, and any unexpected values may get unexpected behaviors. As for the GPU platform, there may not be many researches on implementing a single neuron on it, but many researches have been carried on for the comparison between GPUs and FPGAs about implementing spiking neural networks (Cheung et al., 2012, 2016; Luo et al., 2016). The results show that GPUs can speed up the simulations with multi-core processors and parallel computing capacity, but compared to FPGA, two obvious cons still exist. One is the small on-chip

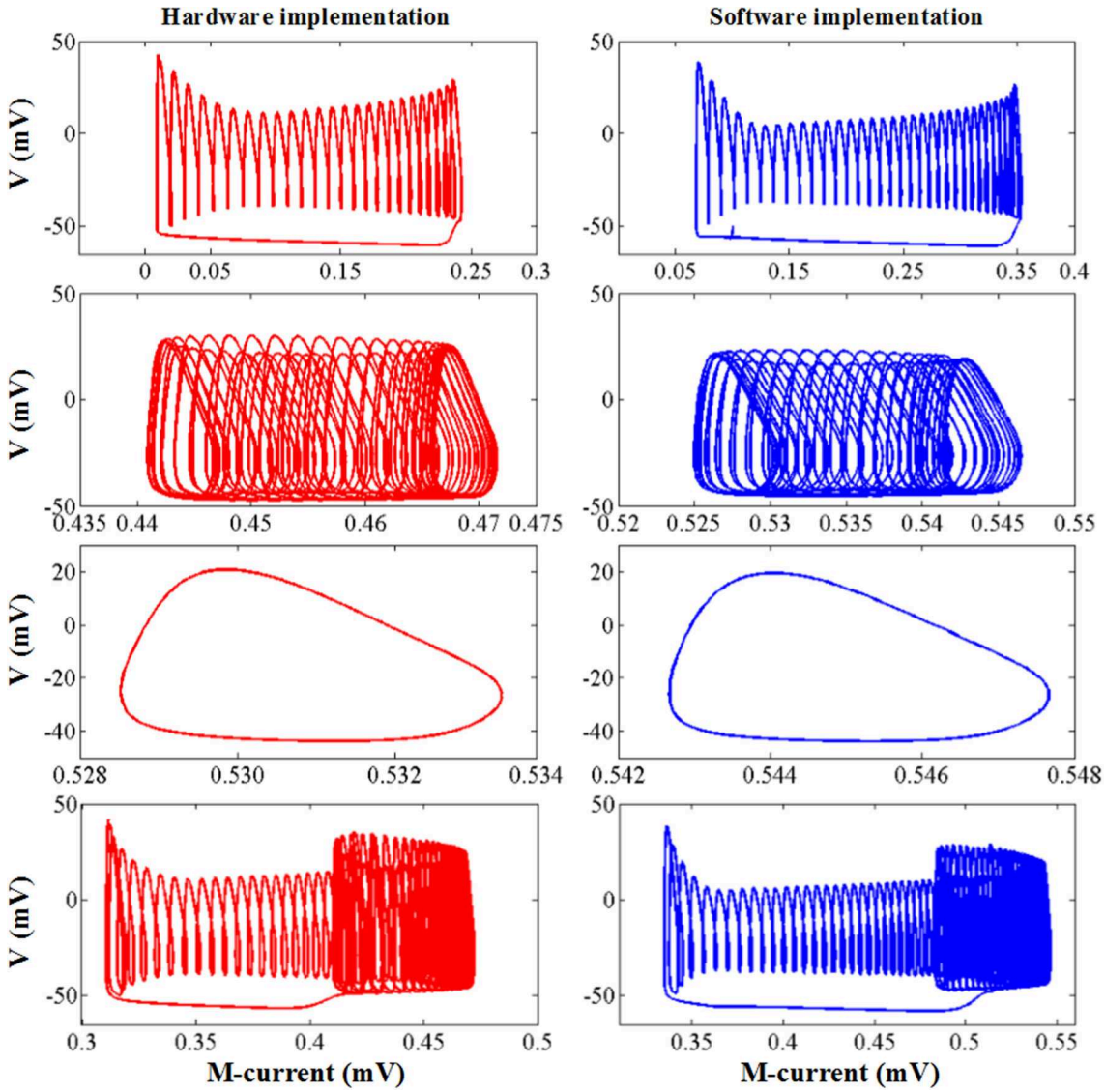


FIGURE 13 | The dynamic behavior of software simulation and hardware implementation under bursting, amplitude modulation, fast spiking, and bursting with amplitude modulation mode. The blue line represents the software simulation results and the red line represents the hardware implementation results.

memory and bandwidth, which limits the scale, the other is the high-power consumption of the desktop system. Besides, the calculation speed of GPUs is also lower than FPGAs in these works.

In order to save multiplier resources on FPGA, many multiplier-less methods have been proposed with different application ranges. Both Jokar and Soleimani (2017) and Hayati et al. (2016) propose a multiplier-less structure with the PWL approach that needs to linearize each function that contains multiplication of variables. The multiplier-less implementation in Agostini et al. (2005) and Gomar and Ahmadi (2014) are simple for there are all constant number multiplications in their models, which can be easily replaced by adders and shifters. Thomas and Luk (2013) replace the multipliers with LUTs and block RAMs, which use more LUT resources to save multiplier resources. Our work presents an FSM, which is common to all multiplication operations and easy to use. With this method, users do not need to redesign the whole approximation using the PWL approach, and all of the multiplications can be realized just by adjusting the supported bit width, even simpler than the method implementing the constant number multiplications. The working frequency of the FSM is 195.92 MHz as shown in **Table 5**, so the lower working frequency of the cell model compared to the model mentioned above is only due to the unavoidable iterative structure of the CORDIC algorithm and the complexity of this model.

This paper presents a multiplier-less and LUT-less CORDIC method to realize the conductance-based cerebellar Purkinje

TABLE 4 | The resource utilization of hardware implementation for the two kinds of models on Altera Stratix III EP3SL340H1152C2.

FPGA resources	Total available	Original model	Modified model
Total logical elements	270,400	1,821	43,543
Dedicated logical registers	270,400	520	468
Total pins	744	29	29
Total memory bits	16,662,528	307,200	0
DSP block 18-bit elements	576	236	0
Total PLLs	4	1	1
Max frequency	–	28.15 MHz	53.44 MHz
Total power dissipation	–	275.40 mW	445.91 mW

TABLE 5 | The resource utilization of hardware implementation for CORDIC algorithm and FSM on Altera Stratix III EP3SL340H1152C2.

FPGA resources	Total available	CORDIC-Exponent	CORDIC-Division	14-bit FSM
Total logical elements	270,400	644	1,222	303
Dedicated logical registers	270,400	565	537	0
Total pins	744	20	20	20
Total memory bits	16,662,528	0	0	0
DSP block 18-bit elements	576	0	0	0
Total PLLs	4	0	0	0
Max frequency	–	120.45 MHz	73.44 MHz	195.92 MHz
Total power dissipation	–	159.23 mW	164.81 mW	158.62 mW

model on FPGA. This can be used for the trade-off among logic resources, memory resources, and multiplier resources, which can be adopted to make full use of the FPGA resources to build a large-scale neural network. All of the calculation modules in our work, the FSM, CDI, and ECEXP, can be directly used for any other models without any extra operation. Besides, the modified pipelined parallel CORDIC algorithm can significantly reduce the resource consumption and the complexity of the hardware implementation architecture.

CONCLUSION

In this work, we present an efficient implementation of a modified cerebellar PC using the CORDIC algorithm with recently found new dynamic performance. Through the analysis of various errors of the two single-neuron models and the comparison of waveforms and network behaviors from different aspects, it can be concluded that the original model and the CORDIC-based model are consistent in biological activities and dynamic mechanisms. After that, we use the non-multiplier and non-LUT methods and implement the CORDIC model on the LaCSNN system. The implementation results are observed on the oscilloscope through the DA conversion module, which are also consistent with the results of the software simulation. By comparing the resource utilization of the original model and the CORDIC model in FPGA implementation, we can conclude that the method used in this paper can transform the use of multiplier resources and memory resources into logical resources, so as to maximize the utilization of FPGA on-chip resources and expand the network scale that can be achieved. This work provides an effective method for realizing large-scale spiking neural networks of cerebellum or many other spiking neural networks on FPGAs.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the manuscript/supplementary files.

AUTHOR CONTRIBUTIONS

XH, SY, XW, JW, BD, and GY made significant contributions to the conception and design of manuscripts and the drafting of manuscripts, as well as critical changes to important intellectual content. All authors have approved publications in their current form. All authors agree to be responsible for all aspects of the work to ensure proper investigation and resolution of issues related to the accuracy or completeness of any part of the work.

FUNDING

This study was supported by the National Natural Science Foundation of China (Grant Nos. 61771330 and 61871287).

REFERENCES

Agostini, L. V., Porto, R. C., Bampi, S., and Silva, I. S. (2005). "A FPGA based design of a multiplierless and fully pipelined JPEG compressor," in *DSD 2005: 8th Euromicro Conference on Digital System Design, Proceedings* (Porto), 210–213. doi: 10.1109/DSD.2005.6

Ahmadi, A., and Zwolinski, M. (2010). "A modified Izhikevich model for circuit implementation of spiking neural networks," in *IEEE Latin American*

Symposium on Circuits and Systems (LASCAS) (Foz do Iguacu), 192–195. doi: 10.1109/LASCAS.2010.7410243

Akbarzadeh-Sherbaf, K., Abdoli, B., Safari, S., and Vahabie, A. H. (2018). A scalable FPGA architecture for randomly connected networks o Hodgkin-Huxley neurons. *Front. Neurosci.* 12:698. doi: 10.3389/fnins.2018.00698

Ambroise, M., Levi, T., Bornat, Y., and Saighi, S. (2013). "Biorealistic spiking neural network on FPGA," in *2013 47th Annual Conference on Information Sciences and Systems (CISS)* (Baltimore, MD).

Bao, S. W., Chen, L., Kim, J. J., and Thompson, R. F. (2002). Cerebellar cortical inhibition and classical eyeblink conditioning. *Proc. Natl. Acad. Sci. U.S.A.* 99, 1592–1597. doi: 10.1073/pnas.032655399

Blazquez, P. M., Hirata, Y., Heiney, S. A., Green, A. M., and Highstein, S. M. (2003). Cerebellar signatures of Vestibulo-Ocular reflex motor learning. *J. Neurosci.* 23, 9742–9751. doi: 10.1523/JNEUROSCI.23-30-09742.2003

Bonabi, S. Y., Asgharian, H., Bakhtiari, R., Safari, S., and Ahmadabadi, M. N. (2012). "FPGA implementation of Hodgkin-Huxley neuron model," in *IJCCI* (Barcelona), 522–528.

Bonabi, S. Y., Asgharian, H., Safari, S., and Ahmadabadi, M. N. (2014). FPGA implementation of a biological neural network based on the Hodgkin-Huxley neuron model. *Front. Neurosci.* 8:379. doi: 10.3389/fnins.2014.00379

Cheung, K., Schultz, S. R., and Luk, W. (2012). "A large-scale spiking neural network accelerator for FPGA systems," in *International Conference on Artificial Neural Networks* (Berlin; Heidelberg: Springer), 113–120. doi: 10.1007/978-3-642-33269-2_15

Cheung, K., Schultz, S. R., and Luk, W. (2016). NeuroFlow: a general purpose spiking neural network simulation platform using customizable processors. *Front. Neurosci.* 9:516. doi: 10.3389/fnins.2015.00516

De Schutter, E., and Bower, J. M. (1994a). An active membrane model of the cerebellar Purkinje cell II. Simulation of synaptic responses. *J. Neurophysiol.* 71, 401–419. doi: 10.1152/jn.1994.71.1.401

De Schutter, E., and Bower, J. M. (1994b). An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice. *J. Neurophysiol.* 71, 375–400. doi: 10.1152/jn.1994.71.1.375

Doya, K. (2000). Complementary roles of basal ganglia and cerebellum in learning and motor control. *Curr. Opin. Neurobiol.* 10, 732–739. doi: 10.1016/S0959-4388(00)00153-7

Garcia, E. O., Cumplido, R., and Arias, M. (2006). "Pipelined CORDIC design on FPGA for a digital sine and cosine waves generator," in *2006 3rd International Conference on Electrical and Electronics Engineering* (Veracruz), 1–4. doi: 10.1109/ICEEE.2006.251917

Gilbert, P. F., and Thach, W. T. (1977). Purkinje cell activity during motor learning. *Brain Res.* 128, 309–328. doi: 10.1016/0006-8993(77)90997-0

Gomar, S., and Ahmadi, A. (2014). Digital multiplierless implementation of biological adaptive-exponential neuron model. *IEEE Trans. Circuits Syst. I* 61, 1206–1219. doi: 10.1109/TCSI.2013.2286030

Han, I. S. (2005). "Biologically plausible VLSI neural network implementation with asynchronous neuron and spike-based synapse," in *IEEE IJCNN* (Montreal, QC), 3244–3248.

Han, V. Z., Grant, K., and Bell, C. C. (2000). Reversible associative depression and nonassociative potentiation at a parallel fiber synapse. *Neuron* 27, 611–622. doi: 10.1016/S0896-6273(00)00070-2

Hayati, M., Nouri, M., Abbott, D., and Haghiri, S. (2016). Digital multiplierless realization of two-coupled biological hindmarsh-rose neuron model. *IEEE Trans. Circuits Syst. II* 63, 463–467. doi: 10.1109/TCSII.2015.2505258

Heck, D. H., Thach, W. T., and Keating, J. G. (2007). On-beam synchrony in the cerebellum as the mechanism for the timing and coordination of movement. *Proc. Natl. Acad. Sci. U.S.A.* 104, 7658–7663. doi: 10.1073/pnas.0609966104

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Hsieh, H. Y., and Tang, K. T. (2012). VLSI implementation of a bio-inspired olfactory spiking neural network. *IEEE Trans. Neural Netw. Learn. Syst.* 23, 1065–1073. doi: 10.1109/TNNLS.2012.2195329

Igarashi, J., Shouno, O., Fukai, T., and Tsujino, H. (2011). Real-time simulation of a spiking neural network model of the basal ganglia circuitry using general purpose computing on graphics processing units. *Neural Netw.* 24, 950–960. doi: 10.1016/j.neunet.2011.06.008

Ito, M., and Kano, M. (1982). Long-lasting depression of parallel fiber-Purkinje cell transmission induced by conjunctive stimulation of parallel fibers and climbing fibers in the cerebellar cortex. *Neurosci. Lett.* 33, 253–258. doi: 10.1016/0304-3940(82)90380-9

Jokar, E., and Soleimani, H. (2017). Digital multiplierless realization of a calcium-based plasticity model. *IEEE Trans. Circuits Syst. II* 64, 832–836. doi: 10.1109/TCSII.2016.2621823

Julian, P., Desages, A., and Agamennoni, O. (1999). High-level canonical piecewise linear representation using a simplicial partition. *IEEE Trans. Circuits Syst. I* 46, 463–480. doi: 10.1109/81.754847

Khaliq, Z. M., Gouwens, N. W., and Raman, I. M. (2003). The contribution of resurgent sodium current to high-frequency firing in Purkinje neurons: an experimental and modeling study. *J. Neurosci.* 23, 4899–4912. doi: 10.1523/JNEUROSCI.23-12-04899.2003

Kramer, M. A., Traub, R. D., and Kopell, N. J. (2008). New dynamics in cerebellar Purkinje cells: torus canards. *Phys. Rev. Lett.* 101:068103. doi: 10.1103/PhysRevLett.101.068103

Lee, B., and Burgess, N. (2003). "Some results on Taylor-series function approximation on FPGA," in *Asilomar Conference on Signals, Systems and Computers,* 2198–2202. doi: 10.1109/ACSSC.2003.1292370

Leiner, H. C., Leiner, A. L., and Dow, R. S. (1993). Cognitive and language functions of the human cerebellum. *Trends Neurosci.* 16, 444–447. doi: 10.1016/0166-2236(93)90072-T

Liu, Y. D., Fan, L. H., and Ma, T. Y. (2014). A modified CORDIC FPGA implementation for wave generation. *Circuits Syst. Signal Process.* 33, 321–329. doi: 10.1007/s00034-013-9638-8

Llinas, R. R. (2009). Inferior olive oscillation as the temporal basis for motricity and oscillatory reset as the basis for motor error correction. *Neuroscience* 162, 797–804. doi: 10.1016/j.neuroscience.2009.04.045

Luo, J. W., Coapes, G., Mak, T., Yamazaki, T., Tin, C., and Degenaar, P. (2016). Real-time simulation of passage-of-time encoding in cerebellum using a scalable FPGA-based system. *IEEE Trans. Biomed. Circuits Syst.* 10, 742–753. doi: 10.1109/TBCAS.2015.2460232

Masuda, N., and Amari, S. I. (2008). A computational study of synaptic mechanisms of partial memory transfer in cerebellar vestibulo-ocular-reflex learning. *J. Comput. Neurosci.* 24, 137–156. doi: 10.1007/s10827-007-0045-7

Medina, J. F., Nores, W. L., Ohyama, T., and Mauk, M. D. (2000). Mechanisms of cerebellar learning suggested by eyelid conditioning. *Curr. Opin. Neurobiol.* 10, 717–724. doi: 10.1016/S0959-4388(00)00154-9

Miall, R. C., Imamizu, H., and Miyauchi, S. (2000). Activation of the cerebellum in co-ordinated eye and hand tracking movements: an fMRI study. *Exp. Brain Res.* 135, 22–33. doi: 10.1007/s002210000491

Middleton, S. J., Racca, C., Cunningham, M. O., Traub, R. D., Monyer, H., Knopfel, T., et al. (2008). High-frequency network oscillations in cerebellar cortex. *Neuron* 58, 763–774. doi: 10.1016/j.neuron.2008.03.030

Miyasho, T., Takagi, H., Suzuki, H., Watanabe, S., Inoue, M., Kudo, Y., et al. (2001). Low-threshold potassium channels and a low-threshold calcium channel regulate Ca2+ spike firing in the dendrites of cerebellar Purkinje neurons: a modeling study. *Brain Res.* 891, 106–115. doi: 10.1016/S0006-8993(00)03206-6

Muñoz, D. M., Sanchez, D. F., Llanos, C. H., and Ayala-Rincón, M. (2010). "FPGA based floating-point library for CORDIC algorithms," in *2010 VI Southern Programmable Logic Conference (SPL)* (Ipojuca: IEEE), 55–60. doi: 10.1109/SPL.2010.5483002

Paulin, M. G. (1993). The role of the cerebellum in motor control and perception. *Brain Behav. Evol.* 41, 39–50. doi: 10.1159/000113822

Pleger, B., and Timmann, D. (2018). The role of the human cerebellum in linguistic prediction, word generation and verbal working memory: evidence from brain imaging, non-invasive cerebellar stimulation and lesion studies. *Neuropsychologia* 115, 204–210. doi: 10.1016/j.neuropsychologia.2018.03.012

Schmahmann, J. D., and Caplan, D. (2006). Cognition, emotion and the cerebellum. *Brain* 129, 290–292. doi: 10.1093/brain/awh729

Soleimani, H., Ahmadi, A., and Bavandpour, M. (2012). Biologically inspired spiking neurons: piecewise linear models and digital implementation. *IEEE Trans. Circuits Syst. I* 59, 2991–3004. doi: 10.1109/TCSI.2012.2206463

Thach, W. T., Goodkin, H., and Keating, J. (1992). The cerebellum and the adaptive coordination of movement. *Annu. Rev. Neurosci.* 15, 403–442. doi: 10.1146/annurev.ne.15.030192.002155

Thomas, D. B., and Luk, W. (2013). Multiplierless algorithm for multivariate Gaussian random number generation in FPGAs. *IEEE Trans. VLSI Syst.* 21, 2193–2205. doi: 10.1109/TVLSI.2012.2228017

Traub, R. D., Buhl, E. H., Gloveli, T., and Whittington, M. A. (2003). Fast rhythmic bursting can be induced in layer 2/3 cortical neurons by enhancing persistent Na+ conductance or by blocking BK channels. *J. Neurophysiol.* 89, 909–921. doi: 10.1152/jn.00573.2002

Valls, J., Kuhlmann, M., and Parhi, K. K. (2002). Evaluation of CORDIC algorithms for FPGA design. *J. VLSI Signal Process. Syst.* 32, 207–222. doi: 10.1023/A:1020205217934

Volder, J. E. (1959). The CORDIC trigonometric computing technique. *Electron. Comput. IRE Trans*. EC-8, 330–334. doi: 10.1109/TEC.1959.5222693

Walther, J. S. (1971). "A unified algorithm for elementary functions," in *May, Spring Joint Computer Conference* (Atlantic City, NJ).

Yamazaki, T., and Igarashi, J. (2013). Realtime cerebellum: a large-scale spiking network model of the cerebellum that runs in realtime using a graphics processing unit. *Neural Netw.* 47, 103–111. doi: 10.1016/j.neunet.2013.01.019

Yang, S., Wang, J., Deng, B., Liu, C., Li, H., Fietkiewicz, C., et al. (2019). Real-time neuromorphic system for large-scale conductance-based spiking neural networks. *IEEE Trans. Cybern.* 49, 2490–2503. doi: 10.1109/TCYB.2018.2823730

Yang, S. M., Deng, B., Li, H. Y., Liu, C., Wang, J., Yu, H. T., et al. (2018a). FPGA implementation of hippocampal spiking network and its real-time simulation on dynamical neuromodulation of oscillations. *Neurocomputing* 282, 262–276. doi: 10.1016/j.neucom.2017.12.031

Yang, S. M., Hao, X. Y., Deng B., Wei, X. L., Li, H. Y., and Wang, J. (2019). A survey of brain-inspired artificial intelligence and its engineering. *Life Res.* 1, 23–29. doi: 10.12032/life2018-0711-005

Yang, S. M., Wei, X. L., Deng, B., Liu, C., Li, H. Y., and Wang, J. (2018b). Efficient digital implementation of a conductance-based globus pallidus neuron and the dynamics analysis. *Phys. A* 494, 484–502. doi: 10.1016/j.physa.2017.11.155

Yang, S. M., Wei, X. L., Wang, J., Deng, B., Liu, C., Yu, H. T., et al. (2017). Efficient hardware implementation of the subthalamic nucleus-external globus pallidus oscillation system and its dynamics investigation. *Neural Netw.* 94, 220–238. doi: 10.1016/j.neunet.2017.07.012

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Hao, Yang, Wang, Deng, Wei and Yi. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 08 November 2019
doi: 10.3389/fnins.2019.01120

Prediction in Autism by Deep Learning Short-Time Spontaneous Hemodynamic Fluctuations

Lingyu Xu[1,2], Xiulin Geng[2], Xiaoyu He[2], Jun Li[3,4] and Jie Yu[2]**

[1] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China, [2] School of Computer Engineering and Science, Shanghai University, Shanghai, China, [3] Guangdong Provincial Key Laboratory of Optical Information Materials and Technology, South China Academy of Advanced Optoelectronics, South China Normal University, Guangzhou, China, [4] Key Lab for Behavioral Economic Science & Technology, South China Normal University, Guangzhou, China

This study aims to explore the possibility of using a multilayer artificial neural network for the classification between children with autism spectrum disorder (ASD) and typically developing (TD) children based on short-time spontaneous hemodynamic fluctuations. Spontaneous hemodynamic fluctuations were collected by a functional near-infrared spectroscopy setup from bilateral inferior frontal gyrus and temporal cortex in 25 children with ASD and 22 TD children. To perform feature extraction and classification, a multilayer neural network called CGRNN was used which combined a convolution neural network (CNN) and a gate recurrent unit (GRU), since CGRNN has a strong ability in finding characteristic features and acquiring intrinsic relationship in time series. For the training and predicting, short-time (7 s) time-series raw functional near-infrared spectroscopy (fNIRS) signals were used as the input of the network. To avoid the over-fitting problem and effectively extract useful differentiation features from a sample with a very limited size (e.g., 25 ASDs and 22 TDs), a sliding window approach was utilized in which the initially recorded long-time (e.g., 480 s) time-series was divided into many partially overlapped short-time (7 s) sequences. By using this combined deep-learning network, a high accurate classification between ASD and TD could be achieved even with a single optical channel, e.g., 92.2% accuracy, 85.0% sensitivity, and 99.4% specificity. This result implies that the multilayer neural network CGRNN can identify characteristic features associated with ASD even in a short-time spontaneous hemodynamic fluctuation from a single optical channel, and second, the CGRNN can provide highly accurate prediction in ASD.

Keywords: ASD, fNIRS, neural network, time series, CGRNN model

OPEN ACCESS

Edited by:
Stefano Brivio, Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Keum-Shik Hong, Pusan National University, South Korea
Guoqi Li, Tsinghua University, China

****Correspondence:***
Jun Li
jun.li@coer-scnu.org
Jie Yu
jieyu@shu.edu.cn

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *19 June 2019*
Accepted: *03 October 2019*
Published: *08 November 2019*

Citation:
Xu L, Geng X, He X, Li J and Yu J (2019) Prediction in Autism by Deep Learning Short-Time Spontaneous Hemodynamic Fluctuations. Front. Neurosci. 13:1120. doi: 10.3389/fnins.2019.01120

INTRODUCTION

Autism spectrum disorder (ASD) refers to a group of neurodevelopmental disorders, including autism and Asperger's syndrome (AS). The current diagnostic criteria for ASD focus on two core symptoms: social communication impairment, restricted interests, and repetitive behaviors (Sharma et al., 2018). Due to the complexity and diversity of ASD, it often takes a long time from detection of the behavioral signs to the definitive diagnosis, which inevitably leads to the lagging of necessary treatment or intervention. In recent years, the ASD prevalence is increasing rapidly

(e.g., 1 in 59, with the prevalence of 4:1 male to females), therefore the study in ASD has drawn significant attention to the public (Christensen et al., 2016). To overcome the drawback that the diagnosis of ASD relies on behavioral observation solely, a variety of studies have been performed, including those brain imaging studies to find characteristics associated with this disorder. On the other hand, with the advance of machine learning, in particular, deep-learning artificial neural network, it may become possible for neurologists to use these machine-learning algorithms to analyze the brain image data collected from ASD and perform image-based early diagnosis of ASD. In addition to this, machine-learning may also play a promising role in ASD intervention, for instance, using personalized intelligent robots to interact with ASD individuals to improve their behaviors (Amaral et al., 2017; Rudovic et al., 2018).

A large variety of brain image studies have demonstrated functional and structural abnormalities in brains of ASD. For example, magnetic resonance imaging (MRI) studies have uncovered that individuals with ASD present an aberrant age-related brain growth trajectory in the frontal area (Elizabeth and Eric, 2005; Lainhart, 2010; Courchesne et al., 2011), which strongly suggests that functional brain measurement at young ages is crucial for revealing ongoing abnormalities in ASD. Libero et al. (2015) utilized multimodal brain imaging modalities [structural MRI, diffusion tensor imaging (DTI), and hydrogen proton magnetic resonance spectrum (1H-MRS)] to investigate neural structure in the same group of individuals (19 adults with ASD and 18 adults with TD) and used the decision tree with fractional anisotropy (FA), radial diffusivity (RD), and cortical thickness as features to perform classification between ASD and TD. This combination method overcomes the discrepancy problem arising from using each imaging method separately (Libero et al., 2015). Some functional brain studies have shown atypical brain activation in response to various cognitive tasks or decreased resting-state functional connectivity (RSFC). These characteristics could also be used for differentiating between individuals with ASD and TD individuals (Kaiser and Pelphrey, 2012; Murdaugh et al., 2012; Deshpande et al., 2013). For example, Iidaka calculated the correlation matrix of resting-state functional magnetic resonance imaging (RS-fMRI) time series and then sent the matrix as input to a probabilistic neural network (PNN) for the classification, which demonstrated that the inherent connection matrix generated by RS-fMRI data might serve as biomarkers for predicting ASD (Iidaka, 2015).

Functional near-infrared spectroscopy (fNIRS) as an optical brain imaging modality utilizing near-infrared light to probe human brain functional activity, is advancing rapidly in techniques and applications. Hong et al. (2018) investigated a brain-computer interface framework for hybrid fNIRS and electroencephalography (EEG) for locked-in syndrome (LIS) patients, and found that the prefrontal cortex is identified as a suitable brain region for imaging. They also studied hybrid fNIRS and EEG for early detection of hemodynamic responses (Hong and Khan, 2017; Khan et al., 2018). Furthermore, they developed a new vector phase diagram to differentiate the initial dip phase and the delayed hemodynamic response (HR) phase of oxy-hemoglobin changes (ΔHbO) (Zafar and Hong, 2018). Very recently fNIRS was also adopted in the investigation of atypical brain activity associated with ASD (Adelina and Bravo, 2011; Jung et al., 2016; Li and Yu, 2018). For instance, Mitsuru et al. measured brain hemodynamic fluctuations of bilateral Brodmann area 10 (BA10) in 3- to 7-year-old ASD and TD children under conscious conditions. They found that slow hemodynamic fluctuations showed abnormal functional connections in ASD (Mitsuru et al., 2013).

Thus far, most of the classifications between ASD patients and normal controls depend on prior characteristic features extracted empirically from brain images. However, due to the complexity and limited knowledge about the pathogenic mechanism of ASD, the hidden factors associated with ASD, which can be used for accurate differentiation between ASD and normal controls, are not easy to be observed and identified merely through reading the brain images. Since the deep-learning artificial neural network is a data-driven method, has the ability to find characteristics hidden in the complete data set. We hypothesize that deep-leaning model might be used for the prediction of ASD through brain images, in particular, our fNIRS data collected from children with ASD, though deep learning based approaches have not been well studied (Ilias et al., 2016; Dvornek et al., 2017; Chiarelli et al., 2018).

On the other hand, a critical challenge for acquiring brain images of most of brain imaging modalities such as MRI/functional magnetic resonance imaging (fMRI), magnetoencephalography (MEG), single photon emission computed tomography (SPECT) and positron emission tomography (PET), et al. is that the subject has to be strictly still during image acquisition that could last 5–10 min or longer. It is not an easy task for conscious (not sedated) children, in particular children with ASD. Therefore if the characteristics of ASD can extract from brain images collected in a short time, it is of great practical significance for brain imaging study in ASD. Even though EEG and fNIRS are not as sensitive to motion as those imaging techniques mentioned above, the artifact caused by head movement still can deteriorate the time-series signals, resulting in an inaccurate result. Thus we aim at two goals in this study: (1) exploring the possibility of using a deep-learning neural network to extract features associated with ASD from fNIRS signals; and (2) using a short-time (e.g., 7 s) fNIRS time series to perform accurate classification between ASD and TD children.

To test our hypothesis and realize the goals, we collected approximate 8-minute spontaneous hemodynamic fluctuations from the bilateral inferior frontal gyrus and temporal lobe by an fNIRS setup in 25 children with ASD and 22 TD children. To analyze the fNIRS data [i.e., time-series of oxygenated hemoglobin (HbO_2), deoxygenated hemoglobin (Hb), and total hemoglobin (HbT = HbO_2 + Hb)], we designed a multilayer neural network consisting of CNN and GRU as a combined unit (called CGRNN) for learning and predicting ASD. The CGRNN is powerful in recognizing characteristic features, identifying the relationship among data in the sequence, and has low computation cost. To test the possibility of using short-time spontaneous hemodynamic fluctuations for the differentiation between ASD and TD,

we segregated the long-time (i.e., 8 min) data sequence into many overlapped short-time (i.e., 7 s) sub-sequences, and then sent them as the input to the CGRNN for the training and classification. The result demonstrated that even using the short-time hemodynamic fluctuation from a single optical channel, we could achieve a rather high accurate classification with 92.2% accuracy, 85.0% sensitivity, and 99.4% specificity. Receiver operating characteristic curve (ROC) Curves also showed that the performance of CGRNN for the classification between ASD and TD is better than GRU, CNN, and Long Short-Term Memory (LSTM) model, implying that CGRNN is a suitable deep-learning neural network for predicting ASD by using spontaneous hemodynamic fluctuations recorded by fNIRS.

MATERIALS AND METHODS

fNIRS Data Collection

In this study, we used a continuous wave fNIRS system (FOIRE-3000, Shimadzu Corporation, Tokyo, Japan) to record spontaneous hemodynamic fluctuations. fNIRS uses near-infrared light to probe brain activity in terms of HbO_2 and Hb. As an optical imaging modality, fNIRS is relatively low cost, portability, safety, low noise (compared to fMRI), and easiness to use. Unless EEG and MEG, its data are not much susceptible to electrical noise. At the same time, it can measure the blood flow changes in the local capillary network caused by neuron firings (Naseer and Hong, 2015). FOIRE-3000 has 16 light sources and 16 detectors. Each light source emits three different wavelengths (780, 805, and 830 nm) near-infrared light in an alternating way. The back reflected light which has passed through the cortex is received by neighboring light detectors. Each source-detector (SD) pair forms a detection channel with a fixed SD distance of 3.0 cm. The fNIRS is used to measures the change in light intensity of the three wavelengths, which is converted to the concentration change in hemoglobin (e.g., HbO_2, Hb, and HbT) by the modified Beer-Lambert law. Neural activity can induce a change in hemoglobin concentration in the local region of the cortex through the neurovascular coupling, which is the basic principle of fNIRS (and fMRI).

Twenty-five children with ASD and twenty-two TD children with an average age of 9.3 ($\pm$1.4) and 9.5 ($\pm$1.6) respectively were recruited in this study. They were all right-handed. Among them, the ASD group consisted of eighteen boys and seven girls. The TD group included eighteen boys and four girls. Experienced clinicians diagnosed all ASD patients in hospitals. Before fNIRS data collection, each subject was informed about the experimental protocol and written informed consent was obtained from his/her parents. During the data collection, the subject sat in a dark, quiet room with their eyes closed and tried to stay still. The spontaneous (or resting-state) hemodynamic fluctuations were recorded from the bilateral inferior frontal and temporal regions on each subject. The experimental protocol is following the ethical standards of the Academic Ethics Committee of South China Normal University (Zhu et al., 2014). It meets the Helsinki Declaration (Inc, 2009).

Figure 1A represents the location of fNIRS measurement channels. Yellow circles indicate light sources and green circles represent light detectors. The number (1–44) in the white square is the number for the channel (each channel consists of a pair of a light source and light detector). **Figure 1B** displays the location of each channel on the brain cortices. The probing area included the bilateral inferior frontal gyrus (1–10 for the left and 23–32 for the right) and bilateral temporal lobe (11–22 for the left and 33–44 for the right). In locating channel positions, the international 10–10 system for EEG was referenced. For each subject approximate 8-minute spontaneous hemodynamic fluctuations were recorded with $\sim$70 millisecond time resolution, corresponding to a sampling rate of 14.29 Hz.

Data Analysis

Data Processing Flow

Figure 2 gives an overview of our data processing flow. The flow divides into two parts. The first part is to increase the sample data for the problem of the small amount of original hemoglobin data. We use the fNIRS time series data as input, and traverse the ASD and TD raw hemoglobin data in the form of sliding window. Each data set can be transformed into a series of continuous and partially overlapping sub-sequence. Each sub-sequence is the data within a labeled sliding window. Hence, the expansion of the small sample data set is performed. The second part, we propose a multilayer neural network, CGRNN model, which combines CNN and GRU, and demonstrate its utility on the accurate classification between ASD and TD with a short-time fNIRS time series. The function in each part of the model is listed below:

(1) The first part of the model uses three-layer CNN to complete the local mode recognition of fNIRS time series. CNN can extract local sub-sequence from the input sequence. Its primary process is to perform the equivalent input transformation on each sub-sequence. So the pattern learned from a specific position of the series can be recognized at any other place later. Thus we can complete the identification of the regional pattern of the sequence and strengthen the generalization ability of feature recognition.
(2) The second part of the model adds the max-pooling layer to prevent over-fitting. Max-pooling layer compresses the data in the form of down-sampling to reduce the parameter information and avoids over-fitting. Moreover, it extracts the maximal value of the feature sequence and further excavates the intrinsic characteristics of data.
(3) The third part of the model utilizes GRU to enhance time series association. We make the feature sequences extract by max-pooling as an input of the GRU. The GRU layer is presented in **Figure 2**. The reset gate (R_t) and the update gate (Z_t) in the GRU are used to capture short-term and long-term dependencies in the sequence. Therefore, GRU can remember the features in the order of time and infer results from features, which serve to strengthen the correlation of series.

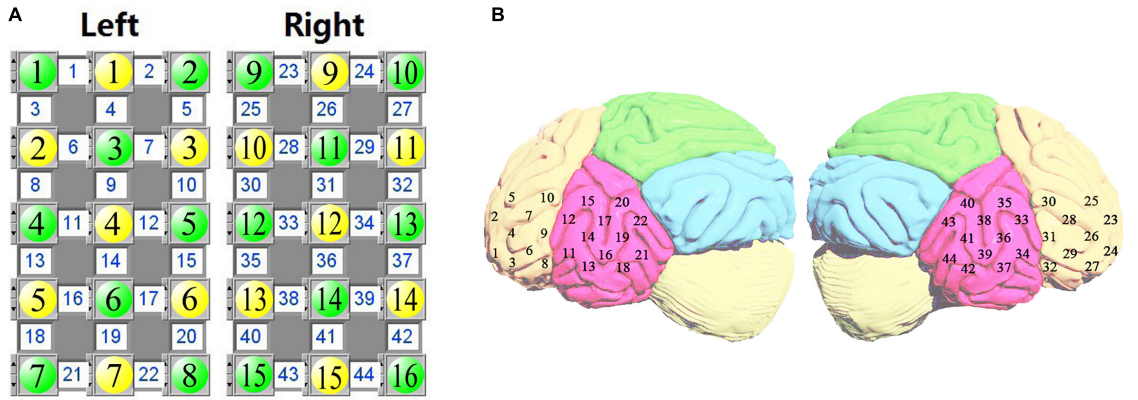


FIGURE 1 | The source-detector configuration where the yellow circles indicate the sources, the green circles indicate the detectors, and the white square between a source and a detector is a channel **(A)**. Location of fNIRS measurement channels over the inferior frontal and temporal cortex **(B)**.

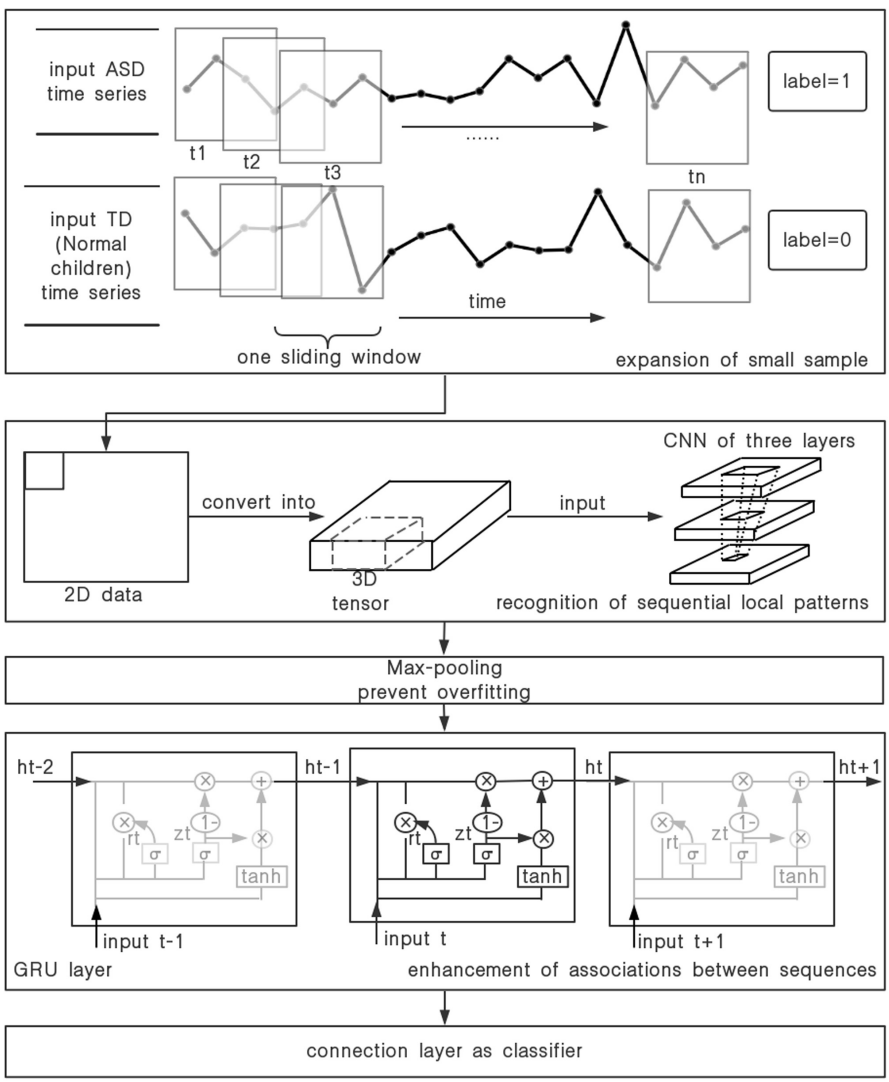


FIGURE 2 | The process of the time-series data.

(4) The fourth part is the construction of classifier. The distributed feature learned by the full connection layer maps to the sample tag space.

Expansion of Small Sample

Because of the less number of fNIRS time series, over-fitting is easy to occur in the training of CGRNN model. Thus we expand the data set in the form of setting the sliding window and excavate the distinctive features of the small sample data set further, which maximize the predictive ability of the CGRNN. More specifically, the time-series of one attribute define as "m" in one channel hemoglobin data. One uses a sliding window with step "s" and width "w" (s < < w) to divide the time series, and obtains N = ł(m−w + s)/sł (m > > w) sub-sequences of length w where the symbol "ł" represents the rounding toward minus infinity. Finally, m divides into N overlapping sub-sequences. And the set of sub-sequence is T. There are T = {T_1, T_2,.. T_N}. If the original information is ASD, the sub-sequence collection label is 1. If the original data is TD, the sub-sequence set label is 0. The process of sliding window is shown in **Figure 3**.

CGRNN Model

The CGRNN model is proposed by a combination of CNN and GRU. Among them, CNN is responsible for the identification of the local mode of the original hemoglobin time series, which uses its translation invariant property to extract subtle but distinct features from fNIRS signals, distinguishing the heterogeneity characteristics of ASD from TD in various feature combinations. We use CNN to improve the CGRNN prediction ability. Specifically, first of all, to learn the spatial hierarchies of hemoglobin, the original data is converted into the three-dimensional tensor (samples, time, feature). Then the three-dimensional tensors are put into CNN part for training. CNN uses the convolution kernel to perform the same input transformation for the input data. Local sub-sequences can be extracted from the entire sequence. Patterns learned from one location in the series can be identified at any other location afterward.

Our network uses three convolution layers to introduce the special hierarchical structure of the space filter by making the continuous convolution layer window grow larger. Further, Rectified Linear Unit (RELU) has the function of making some neurons lose activity and reducing the complexity of network structure. Hence, each layer of CNN aggregates a RELU. Finally, the max-pooling layer is utilized to reduce the occurrence of over-fitting further and extract the maximum value of distinctive features by down-sampling. In other word, the data are compressed to reduce the parameter information and excavate the useful information further.

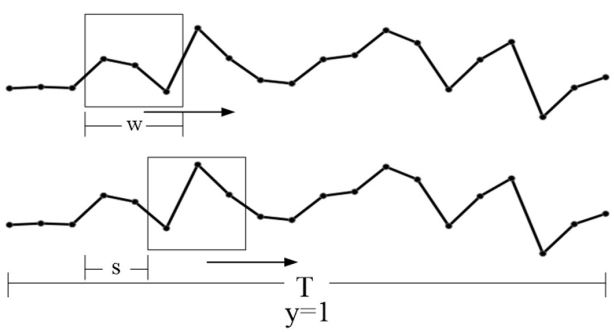


FIGURE 3 | The set of the sliding window (w) and step size (s).

The original time series is prone to gradient disappearance in the course of CNN training, which results in the invalid training. So we use GRU behind the max-pooling layer to solve this problem. GRU network model is a sequence structure that prevents the gradual disappearance of early information by carrying information across multiple time steps. It mainly contains two gate functions (reset gate and update gate). The reset gate and the update gate are utilized to capture the short-term and long-term dependence in the sequence, respectively. GRU remembers the features in the order of time and infers results from features, which serves to strengthen the correlation of series. The specific calculation formula is as follows:

$$Z_t = \sigma\left(W_z \bullet \left[h_{t-1}, x_t\right]\right) \tag{1}$$

$$r_t = \sigma\left(W_r \bullet \left[h_{t-1}, x_t\right]\right) \tag{2}$$

Where h_{t-1} is the hidden state at time t–1, $x_t$is input. W stands for weight. Z_t represents the update gate, which determines how much previous information is retained. r_t stands for the reset gate. It provides a mechanism to discard past implicit states that have nothing to do with the future, i.e., reset gate determines how much past information has been forgotten. The activation function of both gates is the sigmoid function with a range of {0, 1}.

$$\tilde{h}_t = \tanh\left(W \bullet \left[r_t * h_{t-1}, x_t\right]\right) \tag{3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{4}$$

$\tilde{h}_t$ denotes candidate hidden state. It uses the reset gate to control the inflow of the last hidden state that contains the past information. If the reset gate is approximately 0, the previous implicit state will discard.

h_t is the hidden state at time t, which uses update gate Z_t to update the last hidden state h_{t-1} and candidate hidden state $\tilde{h}_t$. Update gate controls the important degree of the implicit state of the past at the current moment. If the update gate is approximately 1, the former implicit state will be saved and passed to the present moment.

This design can deal with the gradient attenuation of CNN and capture the widely spaced dependencies in time-series better. Finally, CGRNN model builds a classifier in the form of adding a full connection layer on the end of GRU. The distributed feature of learned by the full-connection layer maps to the sample tag space. The CGRNN flow is illustrated in **Figure 4**.

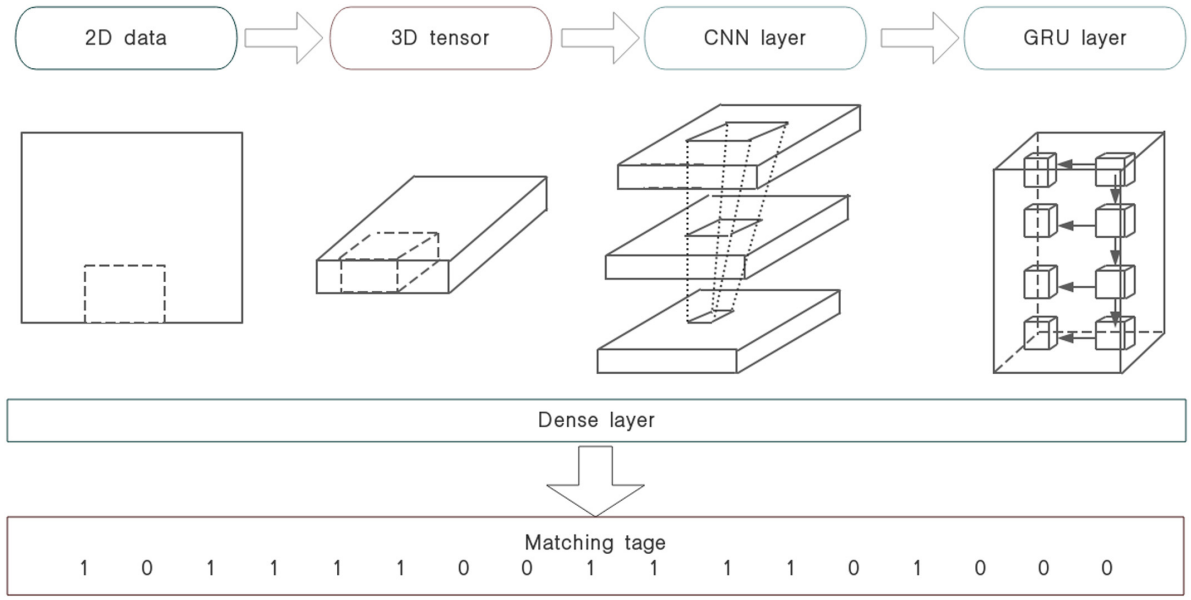


FIGURE 4 | CGRNN flow.

RESULTS

Analysis of Different Channels

To prevent training samples too few to fitting, we adjusted the experimental parameters and divided the data by the sliding window of 100 and step of 50 finally, i.e., considered 7-second data as a sub-sequence. Each sub-sequence corresponded to a specific label. Then we converted the processed data into the three-dimensional tensor (3552, 100, and 1) for the input. We used the data of 28 people to train model. Most of people were divided into 136 sub-sequences. But for a few people the number of sub-sequences was less than 136 because of the recording time was shorter than 8 min (e.g., some children could not tolerate 8-minute measurement, so we had to stop fNIRS recording early). So the final sample size was 3552. Our CGRNN network used three convolution layers. The three convolution layers, respectively had 32, 64, and 128 filters, their kernel_sizes were all five. In each convolution layer, the filter transforms the matrix of a child node in the current layer into the matrix of a unit node in the next layer. The node matrix processed by the filter is determined by the filter size, namely kernel_size. Among of them, the filter slides at regular intervals on the neural network matrix of the current layer and does dot product. In other words, the element of the filter at each position are multiplied by the corresponding element of the input sample and we add up the overall result. We assume that the input sample is [a, b, c, d], and one of the filters is [2, 3], the interval is 1. Then we can get [$a \times 2 + b \times 3$, $b \times 2 + c \times 3$, $c \times 2 + d \times 3$] through the calculation of dot product. The result is called feature map, and the number of feature map is the same as the number of filters. For example, the first convolution layer uses 32 filters, 32 filter maps are obtained after convolution calculation and serve as the input of the second convolution layer.

The input of GRU includes the input sample at current time and the hidden state of the previous sample. The hidden state of the previous time and the current time is multiplied by the weight matrix. Then the added data are sent to the update gate, that is, multiply by the sigmoid function. Therefore, the update gate determines how much previous information and current information is retained. The operation of reset gate is similar to update gate. However, the weight matrix of the reset gate is different from that of the update date because the reset gate determines how much past information has been forgotten it. So GRU not only capture the short-term dependence, but also capture the long-term dependence in the sequence. Furthermore, the CGRNN model were trained using the RELU active function, the binary_cross-entropy loss function, and the Adam optimizer with the default parameter values. The dropout rate during training was fixed to 0.5. Learning rate was 0.01. Models were initialized using default settings. The output were 128 filters, each filter was an 88 dimensional vector. Then, the hemoglobin data were divided into three parts: training set, validation set, and test set. Their proportion is 3:1:1. We trained the CGRNN in the training set, evaluated the generalization ability of the model in the verification set, and saved the optimal model with the smallest loss function. Finally, the model was tested on the test set.

To evaluate the performance of CGRNN classifier, this study applies sensitivity, specificity, and accuracy to the test results of CGRNN classifier (Cheng et al., 2019). Among them, ASD calls positive class, TD calls negative class. Sensitivity means the proportion of actual positives that are correctly classified in all positives. Specificity is the proportion of actual negatives that are correctly identified in all negatives. Accuracy defines the percentage of correct diagnoses among all diagnoses. More specifically, we consider testing every 7-second data of one attribute with a single-channel. There are 44 channels

for each person's data in the test set. Each channel consists of three attributes (HbO_2, Hb, and HbT). Each attribute divides into sub-sequences in the form of the sliding window. Every sub-sequence of 7 s tests its sensitivity, specificity, and accuracy. Finally, the average of sub-sequences denotes the final sensitivity, specificity, and accuracy. The test results are presented in **Figures 5**, **6**.

The hemodynamic activity recorded from the bilateral inferior frontal and temporal lobes vary significantly, which may imply that not every location (or channel) is closely associated with ASD, or sensitive enough for the discrimination, so it is not necessary to achieve good accuracy for the classification in every channel. If the characteristics of ASD can extract from brain images collected in few channels (which might be closely associated with ASD), it is of great practical significance for brain imaging study in ASD.

Figure 5 shows that CGRNN classifier performs well on HbO_2, Hb, and HbT. Although the classification results of different channels under the same attribute are quite different, some channel classification effects are significant. Therefore, CGRNN classifier perform an accurate distinction between ASD and TD children. Moreover, though HbT is the sum of HbO_2 and Hb, it may provide richer discriminative information than HbO_2 and Hb. For instance, the accuracy of HbO_2, Hb, HbT in channel three are 76.8, 64.4, and 88.7%, respectively.

Accuracy represents the overall diagnostic accuracy of ASD and TD. Hence, **Figure 6** sorts the accuracy from large to small and shows the top ten channels with their corresponding attributes. The classification effect is evaluated by accuracy, as shown in **Figure 6**. The first is the Hb of channel 10: 92.2% accuracy, 85.0% sensitivity, and 99.4% specificity. The second is the HbO_2 of channel 43: 90.8% accuracy, 87.9% sensitivity, and 93.8% specificity. The third is the HbT of channel 25: 90.0% accuracy, 81.6% sensitivity, and 98.5% specificity. Most of the functional imaging modalities such as fMRI, MEG, and SPECT, et al., is that the imaging data usually requires recording and analyzing for several minutes. However, the CGRNN model proposed in this paper performs better classification effect by using only 7-second data and has practical application value.

Classification Effect of CGRNN

Moreover, to visualize the classification performance of the CGRNN model further, we randomly select four pairs of ASD and TD children to display the results of the test data. To expand the number of test samples and make the predictive result more accurate, we splice together the data from the same column. Specifically, for each people, we splice 6,859 rows of one column data into 13,718 rows of one column data. Then the time series for most of people is divided into 273 sub-sequences by using a sliding window with a width of 100, and a step of 50. The predictive result for each sub-sequence is displayed in **Figure 7A**. The average of the accuracy is shown in **Figure 7B**.

The classification effect of ASD and TD children is illustrated in **Figures 7A,B**. **Figure 7A** is the real distribution of test data. The *X*-axis represents the randomly selected four pairs

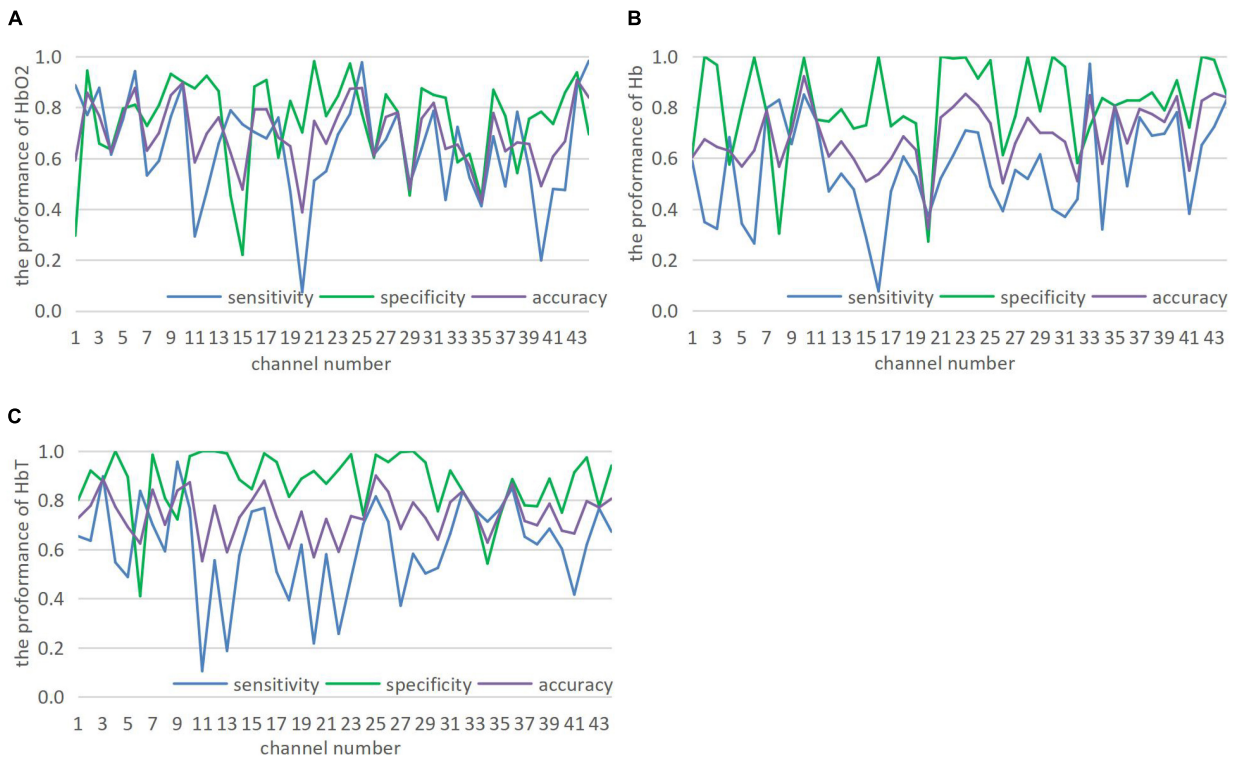
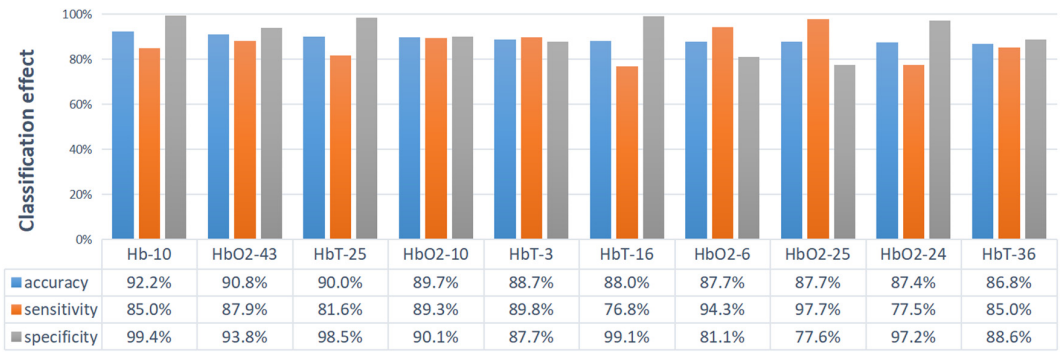


FIGURE 5 | (A–C) Show the prediction of 44 channels in HbO_2, Hb, and HbT three attributes.



	Hb-10	HbO2-43	HbT-25	HbO2-10	HbT-3	HbT-16	HbO2-6	HbO2-25	HbO2-24	HbT-36
accuracy	92.2%	90.8%	90.0%	89.7%	88.7%	88.0%	87.7%	87.7%	87.4%	86.8%
sensitivity	85.0%	87.9%	81.6%	89.3%	89.8%	76.8%	94.3%	97.7%	77.5%	85.0%
specificity	99.4%	93.8%	98.5%	90.1%	87.7%	99.1%	81.1%	77.6%	97.2%	88.6%

FIGURE 6 | The display diagram of the top 10 channels with corresponding properties that have the best classification effect. Ten sets of data are sorted from large to small by accuracy.

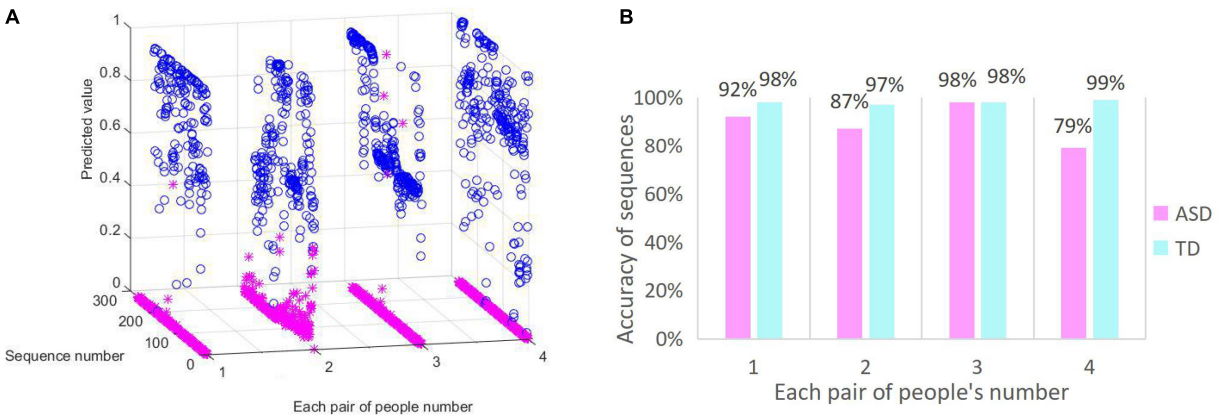


FIGURE 7 | CGRNN classification effect. **(A)** The predictive distribution of test data. **(B)** The accuracy of sequences diagnosis.

of people in the test set. Each pair consists of an ASD child and a TD child. ASD children express by blue, and TD children express by pink. *Y*-axis represents a collection of sub-sequences. *Z*-axis represents the predicted value of CGRNN model. Predictive values range from 0 to 1. The predictive value of TD children is less than 0.5 for accurate forecasting, and the predicted value of ASD is more than 0.5 for precise prediction. Predicted values of ASD data are represented by "o." The Others are represented by "*." **Figure 7B** shows the diagnostic accuracy of sub-sequences. The *X*-axis represents the four pairs of people randomly selected in the test set. The *Y*-axis is the percentage of correctly diagnosed sub-sequences in the whole sequences.

We can see that the predictive values of ASD are basically between 0.5 and 1.0 from **Figure 7A**. Most of the predicted values of TD concentrate in the vicinity of zero. So there are four clear lines on the *Y*-axis, and these are stacked by "*." In other words, most of the TD test results are correct. As shown in **Figure 7B**, the average predictive value of ASD children greater than 0.5 is 89% and the average predictive value of TD children less than 0.5 is 98%. Because each sub-sequence is a 7-second fNIRS time series, we can assume that the recognition accuracy of 7 s of ASD data is 89%, and the recognition accuracy of 7 s of TD data is 98%. Therefore, the CGRNN model can effectively distinguish between ASD and TD children in a short time (7 s).

Since our model is the first ASD classification using a single channel of fNIRS time series data that employs neural network model, there are no canonical comparison partners. We thus compared our model with some widely used traditional classification algorithms such as Logistic Regression (LR), k-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM) classification methods. LR is a generalized linear regression analysis model, it is often used in the binary classification of disease diagnosis. KNN is a well-known machine learning classification algorithm, it determines the category of the sample to be divided according to the category of the nearest sample or samples. RF is a classifier that uses multiple trees to train and predict samples, its classification performance is much better than LR and KNN algorithm. With limited sample size, SVM has stronger ability of generalization in comparison with other existing machine learning algorithms. Firstly, we divided data set in the form of the sliding window ($w = 100$, $s = 50$), the input to these models were two-dimension data (the number of samples, window_size). Then we used GridSearchCV to tune hyper parameter and

made feature engineering. By contrast, the CGRNN model outperformed the other four traditional models as shown in **Table 1**.

DISCUSSION

ROC Comparison Under Different Neural Network Models

For further determining the efficiency of our model, we respectively, applied CNN, LSTM, and GRU to the classification between the children with ASD and TD children based on short-time spontaneous hemodynamic fluctuations. CNN is a feedforward neural network, it not only plays an important role in computer vision tasks, but also has an impact on the time-series analysis. CNN can extract features from local time-series data by using convolution, modularize represented information, utilize data more efficiently. LSTM can overcome the limitation of vanishing gradient in time-series analysis of RNN, thus can capture long-term dependence in time sequential learning. Compared with LSTM, GRU has one less gating unit, which leads to fewer parameters and easy convergence, its computation cost is lower. In these neural network models, the same input data sets as in the CGRNN model were used. The best parameters for each model was selected by validation. We tested the accuracy of each model in 44 channels, and displayed the results in **Figure 8**. The red dot shown the four neural network models (CNN, LSTM, and GRU) also have good performance in the classification for channel 10.

Then we further used ROC to aggregate characteristics of "True Positive Rate" (TPR) and "False Positive Rate" (FPR) and evaluate CGRNN model classification by comparing with ROC curves of different models. TPR represents the proportion of true positives that are correctly classified in all true positives. FPR denotes the percentage of false positives that are correctly classified in all false positives. Empirical ROC curve takes TPR and FPR as ordinate and abscissa, respectively. The TPR and FPR points show different diagnostic locations. These are connected to compose ROC curves. Without considering the effect of misdiagnosis and missed diagnosis, we make the diagnosis point closest to the top left corner (0, 1) as the cut-off point (Fawcett, 2005).

Figure 9 showed the comparison of ROC curves of different models. Since hemoglobin data of channel 10 had the best-classified effect, we selected channel 10 to respectively, verify the ASD classified effect of CGRNN, GRU, CNN, LSTM four different neural network models. Each test selected three thresholds, i.e., max value, min value, and mid-value. Each threshold corresponded to a point (FPR, TPR). All coordinate points were connected to draw the ROC curve these points were used to identify different algorithm performance visually.

TABLE 1 | Accuracy of different classification models.

Model	LR	KNN	RF	SVM	CGRNN
Accuracy	61.5%	65.0%	80.2%	81.2%	92.2%

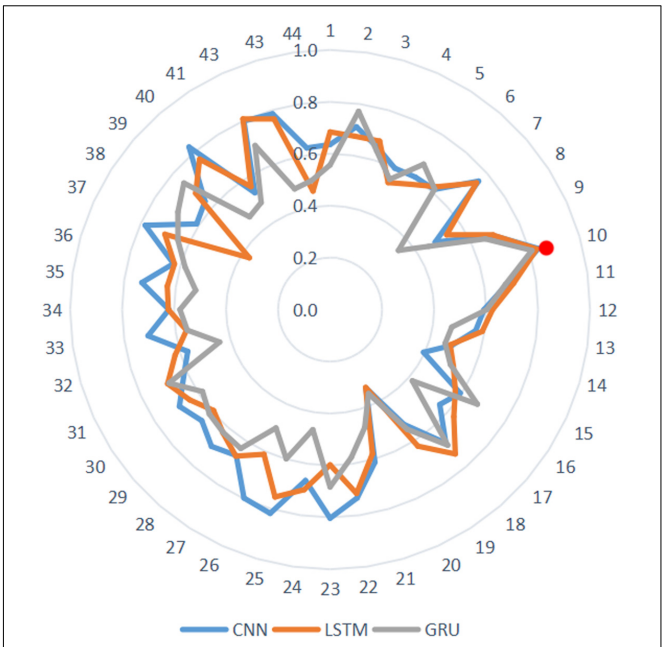


FIGURE 8 | The performance of 44 channels.

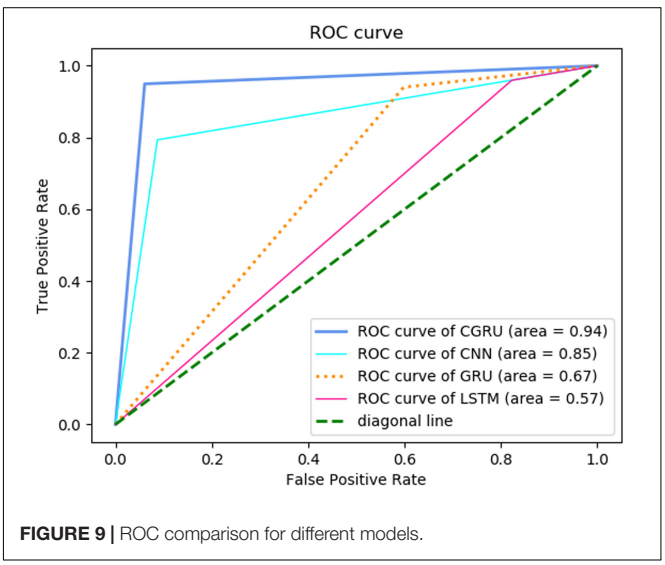


FIGURE 9 | ROC comparison for different models.

CGRNN model was closest to the upper left corner in the ROC curve, so it had the best classification effect. The second was the CNN model, which was helpful to extract hemoglobin features. The third was the GRU model, which cannot play a useful role in classification. Finally, the LSTM model had the worst classification effect. Besides, we compared the area (AUC) under the ROC curve of each model. AUC is a comprehensive measure of all possible classification effect. It regards as the probability that the model randomly arranges the positive sample above negative sample. Generally speaking, the larger the AUC value is, the better the classification effect is (Fawcett, 2005). The comparison shown that the AUC of GRU algorithm was the largest. Therefore, the diagnosis of CGRNN algorithm is the most valuable.

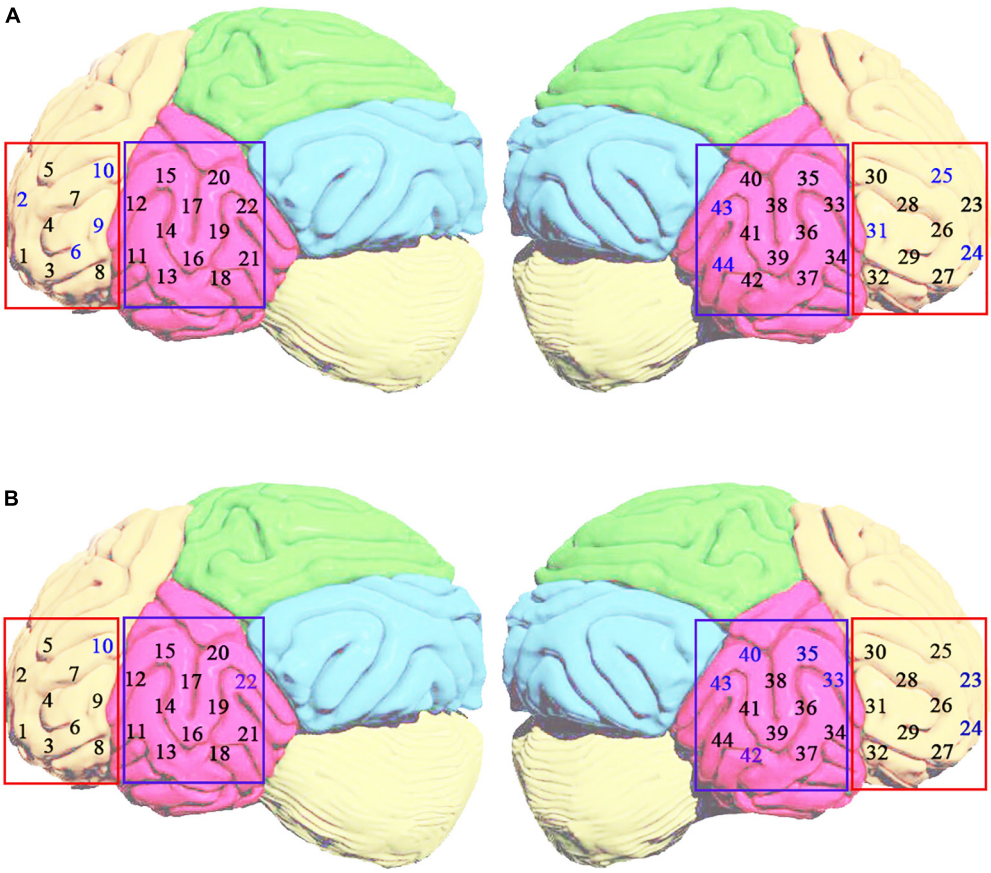


FIGURE 10 | The distribution of best classification performance (i.e., accuracy >80.0%) channels (blue number) based on HbO_2 **(A)** and Hb **(B)** attributes in the left and right brain regions. The yellow area indicates the frontal lobe. The rose-red area represents the temporal lobe.

Furthermore, most of the previous studies utilized multiple feature variables to perform effective differentiation between ASD and TD. No one applied seven seconds' data of single-channel to achieve a better classification effect. The CGRNN uses the 7-second test data of the Hb attribute of channel 10 to have 92.2% accuracy, 85.0% sensitivity, and 99.4% specificity. Therefore, the CGRNN model cannot only diagnose patients with autism efficiently and accurately but also avoid the misdiagnosis of healthy people.

Comparison of the Classification Ability of Brain Regions

Figure 10 displays the location of selected channels with good differentiating ability (accuracy >80.0%) in the HbO_2 and Hb attribute. For HbO_2 (**Figure 10A**), there are seven channels locating in the frontal lobe (4 on the left, 3 on the right), and two channels situated in the temporal lobe (both on the right), indicating that the HbO_2 data of the frontal is more discriminative than the temporal area. Among them, the most discriminative channel 10 locates in the left frontal region. Overall, on the HbO_2, the data from the left and right hemisphere has little difference in classification ability (four channels on the left and five channels on the right). For the Hb data (**Figure 10B**), there are seven channels on the right and two channels on the left, indicating that the Hb data from the right brain is more separable.

CONCLUSION

Our study aims to explore the feasibility of using a multilayer artificial neural network for the classification between children with ASD and TD children based on short-time spontaneous hemodynamic fluctuations.

The contribution of this study has three aspects. First of all, a multilayer neural network called CGRNN was used, which combined three-layered CNN and one-layered GRU. Since CGRNN has a strong ability in finding characteristics associated with ASD and acquiring intrinsic relationship in fNIRS time-series, it can accurately predict ASD by using a short fNIRS time series, which is of great significance for brain imaging research on ASD.

Secondly, different from using small sample data of fNIRS, we expanded the data in the form of the sliding window and combined the CGRNN model to excavate the intrinsic characteristics of the data and improved its predictive ability. The result showed our model performed better than the other

four traditional algorithms such as LR, KNN, RF, and SVM. Furthermore, we used ROC curve to compare our model with CNN, LSTM and GRU neural network model to demonstrating the reliability of our model.

Finally, we demonstrated that though HbT is the sum of HbO_2 and Hb, it may provide richer discriminative information than HbO_2 and Hb. On HbO_2 attribute, the hemodynamic signal from the frontal lobe rather than the temporal lobe leads to a better classification. On Hb attribute, hemodynamic signal from the right hemisphere contains more discriminative information between ASD and TD than the left hemisphere.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this manuscript will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

LX designed the outline of this study and analyzed the data. XG and XH contributed to writing and modifying manuscript. JY participated in the logical construction and design of study and supervised the whole process. JL analyzed the feasibility of the article from a perspective of fNIRS brain imaging and revisited the manuscript critically. All authors contributed to manuscript revision, read and approved the submitted version.

FUNDING

The National Program on Key Research Project under Grant No. 2016YFC1401900, the National Natural Science Foundation of China (NSFC) under Grant No. 81771876, and Guangdong Provincial Key Laboratory of Optical Information Materials and Technology (Grant No. 2017B030301007).

ACKNOWLEDGMENTS

The authors express heartfelt thanks to all the participants for the trial and the tutors who supported us. The authors benefit from tutors' professional guidance, patient encouragement, and selfless help. Thank you for all of them.

REFERENCES

Adelina, P., and Bravo, M. D. C. (2011). Near-infrared spectroscopy: a methodology-focused review. *Semin. Fetal Neonatal Med.* 16, 42–49. doi: 10.1016/j.siny.2010.05.003

Amaral, C. P., Simões, M. A., Mouga, S., Andrade, J., and Castelo-Branco, M. (2017). A novel brain computer interface for classification of social joint attention in autism and comparison of 3 experimental setups: a feasibility study. *J. Neurosci. Methods* 290, 105–115. doi: 10.1016/j.jneumeth.2017.07.029

Cheng, H., Yu, J., Xu, L., and Li, J. (2019). Power spectrum of spontaneous cerebral homodynamic oscillation shows a distinct pattern in autism spectrum disorder. *Biomed. Opt. Express* 10, 1383–1392. doi: 10.1364/BOE.10.001383

Chiarelli, A. M., Croce, P., Merla, A., and Zappasodi, F. (2018). Deep learning for hybrid EEG-fNIRS brain-computer interface: application to motor imagery classification. *J. Neural. Eng.* 15:036028. doi: 10.1088/1741-2552/aaaf82

Christensen, D. L., Bilder, D. A., Zahorodny, W., Pettygrove, S., Durkin, M. S., Fitzgerald, R. T., et al. (2016). Prevalence and characteristics of autism spectrum disorder among 4-year-old children in the autism and developmental disabilities monitoring network. *J. Dev. Behav. Pediatr.* 37, 1–8. doi: 10.1097/dbp.0000000000000235

Courchesne, E., Campbell, K., and Solso, S. (2011). Brain growth across the life span in autism: age-specific changes in anatomical pathology. *Brain. Res.* 1380, 138–145. doi: 10.1016/j.brainres.2010.09.101

Deshpande, G., Libero, L. E., Sreenivasan, K. R., Deshpande, H. D., and Kana, R. K. (2013). Identification of neural connectivity signatures of autism using machine learning. *Front. Hum. Neurosci.* 7:670. doi: 10.3389/fnhum.2013.00670

Dvornek, N. C., Ventola, P., Pelphrey, K. A., and Duncan, J. S. (2017). Identifying autism from resting-state fMRI using long short-term memory networks. *Mach. Learn. Med. Imaging* 10541, 362–370. doi: 10.1007/978-3-319-67389-9_42

Elizabeth, R., and Eric, C. (2005). When is the brain enlarged in autism? a meta-analysis of all brain size reports. *Biol. Psychiatry*. 58, 1–9. doi: 10.1016/j.biopsych.2005.03.026

Fawcett, T. (2005). An introduction to ROC analysis. *Pattern Recognit. Lett.* 27, 861–874. doi: 10.1016/j.patrec.2005.10.010

Hong, K.-S., and Khan, M. J. (2017). Hybrid brain–computer interface techniques for improved classification accuracy and increased number of commands: a review. *Front. Neurorobot.* 11:35. doi: 10.3389/fnbot.2017.00035

Hong, K.-S., Khan, M. J., and Hong, M. J. (2018). Feature extraction and classification methods for hybrid fNIRS-EEG brain-computer interfaces. *Front. Hum. Neurosci.* 12:246. doi: 10.3389/fnhum.2018.00246

Iidaka, T. (2015). Resting state functional magnetic resonance imaging and neural network classified autism and control. *Cortex* 63, 55–67. doi: 10.1016/j.cortex.2014.08.011

Ilias, S., Tahir, N. M., Jailani, R., and Che, Z. C. H. (2016). "Classification of autism children gait patterns using neural network and support vector machine," in *Proceedings of the IEEE Symposium Computer Applications & Industrial Electronics*, (Penang: IEEE), 52–56.

Inc, W. M. A. (2009). Declaration of helsinki ethical principles for medical research involving human subjects. *J. Indian Med. Assoc.* 14, 233–238. doi: 10.1515/9783110208856.233

Jung, C. E., Strother, L., Feil-Seifer, D. J., and Hutsler, J. J. (2016). Atypical asymmetry for processing human and robot faces in autism revealed by fNIRS. *PLoS One* 11:e0158804. doi: 10.1371/journal.pone.0158804

Kaiser, M. D., and Pelphrey, K. A. (2012). Disrupted action perception in autism: behavioral evidence, neuroendophenotypes, and diagnostic utility. *Dev. Cogn. Neurosci.* 2, 25–35. doi: 10.1016/j.dcn.2011.05.005

Khan, M. J., Ghafoor, U., and Hong, K.-S. (2018). Early detection of hemodynamic responses using EEG: a hybrid EEG-fNIRS study. *Front. Hum. Neurosci.* 12:479. doi: 10.3389/fnhum.2018.00479

Lainhart, J. E. (2010). Advances in autism neuroimaging research for the clinician and geneticist. *Am. J. Med. Genet. C Semin. Med. Genet.* 142C, 33–39. doi: 10.1002/ajmg.c.30080

Li, Y., and Yu, D. (2018). Variations of the functional brain network efficiency in a young clinical sample within the autism spectrum: a fNIRS investigation. *Front. Physiol.* 9:67. doi: 10.3389/fphys.2018.00067

Libero, L. E., Deramus, T. P., Lahti, A. C., Deshpande, G., and Kana, R. K. (2015). Multimodal neuroimaging based classification of autism spectrum disorder

using anatomical, neurochemical, and white matter correlates. *Cortex* 66, 46–59. doi: 10.1016/j.cortex.2015.02.008

Mitsuru, K., Yuko, Y., Kiyomi, S., Sanae, U., Hirotoshi, H., Toshio, M., et al. (2013). Anterior prefrontal hemodynamic connectivity in conscious 3- to 7-year-old children with typical development and autism spectrum disorder. *PLoS One* 8:e56087. doi: 10.1371/journal.pone.0056087

Murdaugh, D. L., Shinkareva, S. V., Deshpande, H. R., Wang, J., Pennick, M. R., and Kana, R. K. (2012). Differential deactivation during mentalizing and classification of autism based on default mode network connectivity. *PLoS One* 7:e50064. doi: 10.1371/journal.pone.0050064

Naseer, N., and Hong, K.-S. (2015). fNIRS-based brain-computer interfaces: a review. *Front. Hum. Neurosci.* 9:3. doi: 10.3389/fnhum.2015.00003

Rudovic, O., Lee, J., Dai, M., Schuller, B., and Picard, R. (2018). Personalized machine learning for robot perception of affect and engagement in autism therapy. *Science* 3:eaao6760. doi: 10.1126/scirobotics.aao6760

Sharma, S. R., Gonda, X., and Tarazi, F. I. (2018). Autism spectrum disorder classification, diagnosis and therapy. *Pharmacology* 190, 91–104. doi: 10.1016/j.pharmthera.2018.05.007

Zafar, A., and Hong, K.-S. (2018). Neuronal activation detection using vector phase analysis with dual threshold circles: a functional near-infrared spectroscopy study. *Int. J. Neural Syst.* 28:1850031. doi: 10.1142/S0129065718500314

Zhu, H., Fan, Y., Guo, H., Huang, D., and He, S. (2014). Reduced interhemispheric functional connectivity of children with autism spectrum disorder: evidence from functional near infrared spectroscopy studies. *Biomed. Opt. Express* 5, 1262–1274. doi: 10.1364/BOE.5.001262

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Xu, Geng, He, Li and Yu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 09 January 2020
doi: 10.3389/fnins.2019.01383

Digital Biologically Plausible Implementation of Binarized Neural Networks With Differential Hafnium Oxide Resistive Memory Arrays

Tifenn Hirtzlin[1†], Marc Bocquet[2†], Bogdan Penkovsky[1], Jacques-Olivier Klein[1], Etienne Nowak[3], Elisa Vianello[3], Jean-Michel Portal[2] and Damien Querlioz[1]*

[1] C2N, Univ Paris-Sud, Université Paris-Saclay, CNRS, Palaiseau, France, [2] Aix Marseille Univ, Université de Toulon, CNRS, IM2NP, Marseille, France, [3] CEA, LETI, Grenoble, France

OPEN ACCESS

Edited by:
Erika Covi,
Politecnico di Milano, Italy

Reviewed by:
Doo Seok Jeong,
Hanyang University, South Korea
Irene Munoz,
Politecnico di Milano, Italy

****Correspondence:***
Damien Querlioz
damien.querlioz@c2n.upsaclay.fr

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *06 August 2019*
Accepted: *06 December 2019*
Published: *09 January 2020*

Citation:
Hirtzlin T, Bocquet M, Penkovsky B, Klein J-O, Nowak E, Vianello E, Portal J-M and Querlioz D (2020) Digital Biologically Plausible Implementation of Binarized Neural Networks With Differential Hafnium Oxide Resistive Memory Arrays. Front. Neurosci. 13:1383. doi: 10.3389/fnins.2019.01383

The brain performs intelligent tasks with extremely low energy consumption. This work takes its inspiration from two strategies used by the brain to achieve this energy efficiency: the absence of separation between computing and memory functions and reliance on low-precision computation. The emergence of resistive memory technologies indeed provides an opportunity to tightly co-integrate logic and memory in hardware. In parallel, the recently proposed concept of a Binarized Neural Network, where multiplications are replaced by exclusive NOR (XNOR) logic gates, offers a way to implement artificial intelligence using very low precision computation. In this work, we therefore propose a strategy for implementing low-energy Binarized Neural Networks that employs brain-inspired concepts while retaining the energy benefits of digital electronics. We design, fabricate, and test a memory array, including periphery and sensing circuits, that is optimized for this in-memory computing scheme. Our circuit employs hafnium oxide resistive memory integrated in the back end of line of a 130-nm CMOS process, in a two-transistor, two-resistor cell, which allows the exclusive NOR operations of the neural network to be performed directly within the sense amplifiers. We show, based on extensive electrical measurements, that our design allows a reduction in the number of bit errors on the synaptic weights without the use of formal error-correcting codes. We design a whole system using this memory array. We show on standard machine learning tasks (MNIST, CIFAR-10, ImageNet, and an ECG task) that the system has inherent resilience to bit errors. We evidence that its energy consumption is attractive compared to more standard approaches and that it can use memory devices in regimes where they exhibit particularly low programming energy and high endurance. We conclude the work by discussing how it associates biologically plausible ideas with more traditional digital electronics concepts.

Keywords: binarized neural networks, resistive memory, memristor, in-memory computing, biologically plausible digital electronics, ASICs

1. INTRODUCTION

Through the progress of deep learning, artificial intelligence has made tremendous achievements in recent years. Its energy consumption in graphics or central processing units (GPUs and CPUs) remains, however, a considerable challenge, limiting its use at the edge and raising the question of the sustainability of large-scale artificial intelligence-based services. Brains, by contrast, manage intelligent tasks with highly reduced energy usage. One key difference between GPUs and CPUs on the one hand and brains on the other hand is how they deal with memory. In GPUs and CPUs, memory and arithmetic units are separated, both physically and conceptually. In artificial intelligence algorithms, which require large amounts of memory access, considerably more energy is spent moving data between logic and memory than in doing actual arithmetic (Pedram et al., 2017). In brains, by contrast, neurons—which implement most of the arithmetic—and synapses—which are believed to store long-term memory—are entirely colocated. A major lead for reducing the energy consumption of artificial intelligence is therefore to imitate this strategy and design non-von Neumann systems where logic and memory are merged (Indiveri and Liu, 2015; Querlioz et al., 2015; Editorial, 2018; Yu, 2018). There is new interest in this idea today with the advent of novel nanotechnology-based non-volatile memories, which are compact and fast and can be embedded at the core of the Complementary Metal Oxide Semiconductor (CMOS) process (Prezioso et al., 2015; Saïghi et al., 2015; Wang et al., 2015; Covi et al., 2016; Serb et al., 2016; Ambrogio et al., 2018; Yu, 2018). Another key difference between processors and the brain is the basic nature of computations. GPUs and CPUs typically perform all neural network computations with precise floating-point arithmetic. In brains, most of the computation is done in a low-precision analog fashion within the neurons (Klemm and Bornholdt, 2005; Faisal et al., 2008), resulting in asynchronous spikes as an output, which is therefore binary. A second idea for cutting the energy consumption of artificial intelligence is therefore to design systems that operate with much lower-precision computation.

In recent years, considerable research has been conducted to implement neural networks using analog resistive memory as synapses—the device conductance implementing the synaptic weights. To a large extent, neural network computation can be done using analog electronics: weight/neuron multiplication is performed based on Ohm's law, and addition is natively implemented with Kirchoff's current law (Prezioso et al., 2015; Serb et al., 2016; Ambrogio et al., 2018; Li et al., 2018; Wang et al., 2018). This type of implementation is, to a certain extent, very biologically plausible, as it reproduces the two strategies mentioned above. The challenge of this implementation, however, is that it requires relatively heavy analog or mixed-signal CMOS circuitry such as operational amplifiers or Analog-to-Digital Converters, resulting in significant area and energy overhead.

In parallel, a novel class of neural networks has recently been proposed—Binarized Neural Networks (or the closely related XNOR-NETs) (Courbariaux et al., 2016; Rastegari et al., 2016). In these neural networks, once trained, synapses as well as neurons assume only binary values, meaning $+1$ or -1. These neural networks therefore have limited memory requirements and also rely on highly simplified arithmetic. In particular, multiplications are replaced by one-bit exclusive NOR (XNOR) operations. Nevertheless, Binarized Neural Networks can achieve near state-of-the-art performance on vision tasks (Courbariaux et al., 2016; Rastegari et al., 2016; Lin et al., 2017) and are therefore extremely attractive for realizing inference hardware. The low precision of Binarized Neural Networks and in particular the binary nature of neurons—which is reminiscent of biological neurons spikes—also endows them with biological plausibility: they can indeed be seen as a simplification of spiking neural networks.

Great effort has been devoted to developing hardware implementations of Binarized Neural Networks. Using nanodevices, one natural intuition would be to adopt the strategy proposed for conventional neural networks and perform arithmetic in an analog fashion using Kirchoff's law (Yu et al., 2016; Yu, 2018). However, Binarized Neural Networks are very digital in nature and are multiplication-less. These networks can therefore provide an opportunity to benefit, at the same time, from both bioinspired ideas and the achievements of Moore's law and digital electronics. In this work, we propose a fully digital implementation of binarized neural networks incorporating CMOS and nanodevices, and implementing the biological concepts of tight memory and logic integration, and low-precision computing. As memory nanodevices, we use hafnium oxide-based resistive random access memory (OxRAM), a compact and fast non-volatile memory cell that is fully compatible with the CMOS process (Grossi et al., 2016).

However, one significant challenge to implementing a digital system with memory nanodevices is their inherent variability (Ielmini and Wong, 2018; Ly et al., 2018), which causes bit errors. Traditional memory applications employ multiple error-correcting codes (ECCs) to solve this issue. ECC decoding circuits have large areas and high energy consumption (Gregori et al., 2003) and add extra time to data access due to syndrome computation and comparison. Moreover, the arithmetic operations of error-syndrome computation are actually more complicated than those of a Binarized Neural Network. This solution is difficult to implement in a context where memory and logic are tightly integrated, especially when part of the computation is performed during sensing. This is one of the main reasons that the state of the art of RRAM for in-memory computing does not correct errors and is not compatible with technologies with errors (Chen et al., 2017, 2018). In this paper, we introduce our solution. We design, fabricate, and test a differential oxide-based resistive memory array, including all peripheral and sensing circuitry. This array, based on a two-transistor/two-resistor (2T2R) bit cell, inherently reduces bit errors without the use of ECC, and we show that it is particularly well-adapted for in-memory computing. We then design and simulate a fully binarized neural network based on this memory array. We show that the XNOR operations can be integrated directly within the sense operation of the memory array and that the resulting system can be highly energy efficient. Based on neural networks on multiple datasets (MNIST, CIFAR-10,

ImageNet, and ECG data analysis), we evaluate the number of bit errors in the memory that can be tolerated by the system. Based on this information, we show that the memory nanodevices can be used in an unconventional programming regime, where they feature low programming energy (less than five picoJoules per bit) and outstanding endurance (billions of cycles).

Partial and preliminary results of this work have been presented at a conference (Bocquet et al., 2018). This paper adds additional measurements of OxRAMs with shorter programming pulses, an analysis of the impact of bit errors on more datasets (ImageNet and ECG data analysis), and a detailed comparison and benckmarking of our approach with processors, non-binarized ASIC neural networks, and analog RRAM-based neural networks.

2. MATERIALS AND METHODS

2.1. Differential Memory Array for In-memory Computing

In this work, we fabricated a memory array for in-memory computing with its associated peripheral and sensing circuits. The memory cell relies on hafnium oxide (HfO_2) oxide-based resistive Random Access Memory (OxRAM). The stack of the device is composed of a HfO_2 layer and a titanium layer. Both layers have a thickness of ten nanometers, and they grow between two titanium nitride (TiN) electrodes. Our devices are embedded within the back-end-of-line of a commercial 130-nm CMOS logic process (**Figure 1A**), allowing tight integration of logic and non-volatile memory (Grossi et al., 2016). The devices are integrated on top of the fourth (copper) metallic layer.

We chose hafnium oxide OxRAMs because they are known to provide non-volatile memories compatible with the modern CMOS process and only involve foundry-friendly materials and process steps.

After a one-time forming process, such devices can switch between low-resistance and high-resistance states (LRS and HRS) by applying positive or negative electrical pulses, respectively. Our work could be reproduced with other types of memories. NOR flash cells, which are readily available in commercial processes, could be used, and their potential for neuromorphic inference has been proven Merrikh-Bayat et al. (2017). However, they suffer from high programming voltages (higher than ten volts) requiring charge pumps, have limited endurance, and are not scalable to the most advanced technology nodes (Dong et al., 2017). Emerging memories such as phase change memory or spin torque magnetoresitive memory could also be used adopting the strategies presented in this paper. These technologies do not require a forming process, and they can bring enhanced reliability with regards to OxRAMs but come with an increased process cost (Chen, 2016).

Conventionally, OxRAMs are organized in a "One Transistor-One Resistor" structure (1T1R), where each nanodevice is associated with one access transistor (Chen, 2016). The LRS and HRS are used to mean the zero and one logic values or

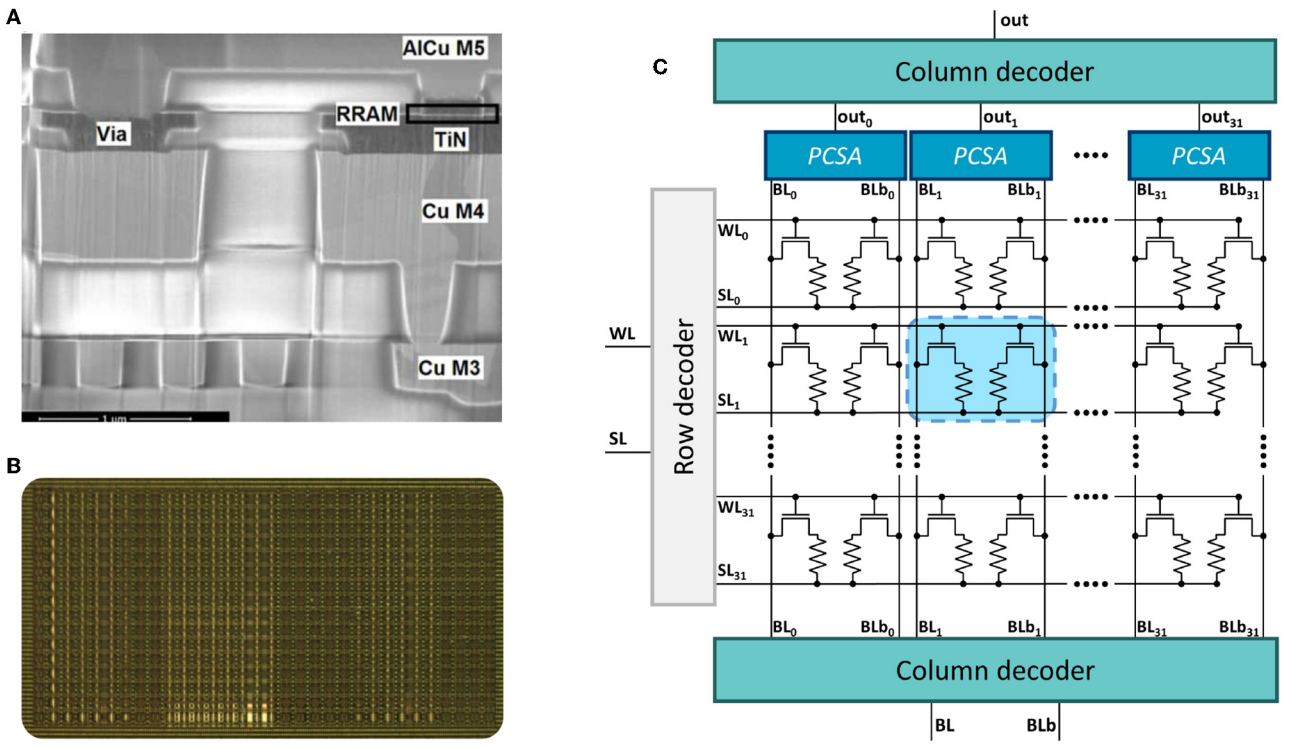


FIGURE 1 | (A) Scanning Electron Microscopy image of the back-end-of-line of the CMOS process integrating an OxRAM device. **(B)** Photograph and **(C)** simplified schematic of the one kilobit in-memory computing-targeted memory array characterized in this work.

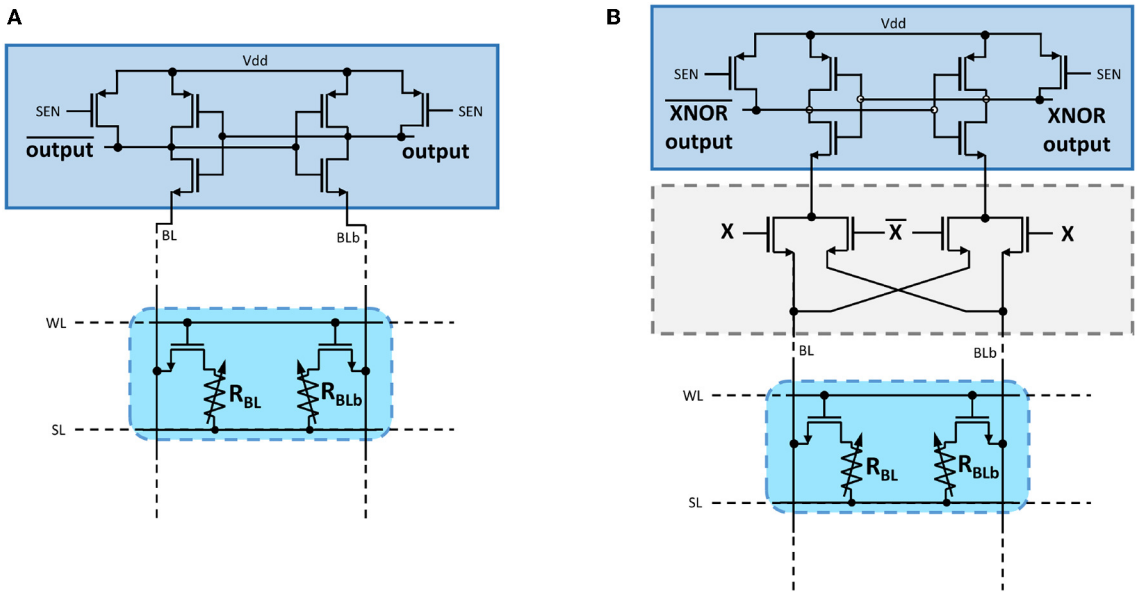


FIGURE 2 | (A) Schematic of the precharge sense amplifier used in this work to read 2T2R memory cells. **(B)** Schematic of the precharge sense amplifier augmented with an XNOR logic operation.

the inverse. The read operation is then achieved by comparing the electrical resistance of the nanodevice to a reference value intermediate between the typical values of resistances in HRS and LRS. Unfortunately, due to device variability, OxRAMs are prone to bit errors: the HRS value can become lower than the reference resistance, and the LRS value can be higher than the reference resistance. The device variability includes both device-to-device mismatch and the fact that, within the same device, the precise value of HRS and LRS resistance changes at each programming cycle (Grossi et al., 2018).

To limit the number of bit errors, in this work, we fabricated a memory array with a "Two Transistors-Two Resistors" structure (2T2R), where each bit of information is stored in a pair of 1T1R structures. A photograph of the die is presented in **Figure 1B** and its simplified schematic in **Figure 1C**. Information is stored in a differential fashion: the pair LRS/HRS means logic value zero, while the pair HRS/LRS means logic value one. In this situation, readout is performed by comparing the resistance values of the two devices. We therefore expect bit errors to be less frequent, as a bit error only occurs if a device programmed in LRS is more resistive than its complementary device programmed in HRS. This concept of 2T2R memory arrays has already been proposed, but its benefits in terms of bit error rate have never been demonstrated until this work (Hsieh et al., 2017; Shih et al., 2017).

The programming of devices in our array is made sequentially, i.e., on a device-by-device basis. The first time that the memory array is used, all devices are "formed." To form the device of row i and column j, the bit line BL_j, connected to the bottom electrode of the memory device, is set to ground, and the word line WL_i is set to a voltage chosen to limit the current to a "compliance value" of 200 μA. A voltage ramp is applied to the sense line SL_i connected to the top electrode of the memory device, increasing from 0 to 3.3 V at a ramp rate of 1,000 V/s. This forming operation is performed only once over the lifetime of the device. To program a device into its LRS (SET operation), the bit line BL_j is set to ground, while the sense line SL_i is set to $2V$. The word line WL_i is again set to a voltage chosen to limit the current to a compliance value, ranging from 20 to 200 μA depending on the chosen programming condition. To program a device into its HRS (RESET operation), a voltage of opposite sign needs to be applied to the device, and the current compliance is not needed. The sense line SL_i is therefore set to the ground, while the word line WL_i is set to a value of 3.3 V, and the bit line BL_j to a "RESET voltage" ranging from 1.5 to 2.5 V depending on the chosen programming condition. For both SET and RESET operations, the programming duration can range from 0.1 to 100 μs. During programming operations, all bit, select, and word lines corresponding to non-selected devices are grounded, with the exception of the bit line of the complementary device of the selected device: this one is programmed to the same voltage as the one applied to the sense line to avoid any disturbing effect on the complementary device.

In our fabricated circuit, the readout operation is performed with precharge sense amplifiers (PCSA) (Zhao et al., 2009, 2014) (**Figure 2A**). These circuits are highly energy-efficient due to their operation in two phases, precharge and discharge, avoiding any direct path between the supply voltage and ground. First, the sense signal (SEN) is set to ground and SL to the supply voltage, which precharges the two selected complementary nanodevices as well as the comparing latch at the same voltage. In the second phase, the sense signal is set to the supply voltage, and the voltages on the complementary devices are discharged to ground through SL. The branch with the lowest resistance discharges

faster and causes its associated inverter output to discharge to ground, which latches the complementary inverter output to the supply voltage. The two output voltages therefore represent the comparison of the two complementary resistance values. In our test chip, the read time is approximately 10 *ns* and results from the high capacitive load associated with our probe testing setup. Without this high capacitive load, the switching time would be determined by the time to resolve the initial metastability of the circuit. This switching time can be as fast as 100 *ps* in a scaled technology (Zhao et al., 2014).

We fabricated a differential memory array with 2,048 devices, therefore implementing a kilobit memory array. Each column of complementary nanodevices features a precharge sense amplifier, and rows and columns are accessed through integrated CMOS digital decoders. The pads of the dies are not protected from electrostatic discharge, and the dies were tested with commercial 22-pad probe cards. In all the experiments, voltages are set using a home-made printed circuit board, and pulse voltages are generated using Keysight B1530A pulse generators. In the design, the precharge sense amplifiers can optionally be deactivated and bypassed, which allows the nanodevice resistance to be measured directly through external precision source monitor units (Keysight B1517a).

2.2. Design of In-memory Binarized Neural Network Based on the Differential Memory Building Block

This work aims at implementing Binarized Neural Networks in hardware. In these neural networks, the synaptic weights, as well as the neuronal states, can take only two values, $+1$ and -1, while these parameters assume real values in conventional neural networks. The equation for neuronal value A_j in a conventional neural network is:

$$A_j = f\left(\sum_i W_{ji}X_i + b_j\right), \tag{1}$$

where X_i are the neuron inputs, W_{ji} the synaptic weights values, b_j a bias term, and f the neural activation function, which introduces non-linearity into the network. Typical examples of activation functions are the sigmoid function, the softmax function, and the hyperbolic tangent function. In Binarized Neural Networks, the activation function is much simpler, as it is substituted by the sign function, as shown in Equation (2):

$$A_j = \text{sign}\left(\underset{i}{\text{POPCOUNT}}\left(XNOR\left(W_{ji}, X_i\right)\right) - T_j\right). \tag{2}$$

In this equation, T_j is the so-called threshold of the neuron, and it is learned during training. POPCOUNT is the function that counts the number of ones in a series of bits, and sign is the sign function.

The training process of binarized neural networks differs from that of conventional neural networks. During training, the weights assume real weights in addition to the binary weights, which are equal to the sign of the real weights. Training employs the classical error backpropagation equations with several adaptations. The binarized weights are used in the equations of both the forward and the backward passes, but the real weights change as a result of the learning rule (Courbariaux et al., 2016). Additionally, as the activation function of binarized neural networks is the sign function and is not differentiable, we consider the sign function as the first approximation of the hardtanh function,

$$\text{Hardtanh(x)} = \text{Clip(x}, -1, 1), \tag{3}$$

and we use the derivative of this function as a replacement for the derivative of the sign function in the backward pass. This replacement is a key element for training BNN successfully. The clip interval in Equation (3) is not learned and is chosen to be between -1 and 1 for all neurons. Using a larger interval would indeed increase the vanishing gradient effect, while using a smaller interval would lead to derivatives higher than one, which can cause exploding gradient effects.

Finally, the Adam optimizer is used to stabilize learning (Kingma and Ba, 2014). A technique known as batch-normalization is employed at each layer of the neural network (Ioffe and Szegedy, 2015). Batch-normalization shifts and scales the neuronal activations over a batch during the training process. This method is used optionally in conventional neural networks to accelerate and stabilize learning. Using this technique becomes essential when training binarized neural networks to reach high accuracies, as it ensures that neuronal activations utilize both $+1$ and -1 values. At inference time, batch-normalization is no longer necessary, and the threshold learned by this technique can be used directly as the neuronal threshold in Equation (2).

With this learning technique, binarized neural networks function surprisingly well. They can achieve near state-of-the-art performance on image recognition tasks such as CIFAR-10 and ImageNet (Lin et al., 2017). After learning, the real weights serve no more purpose and can be discarded. This makes binarized neural networks exceptional candidates for hardware implementation of neural network inference. Not only are their memory requirements minimal (one bit per neuron and synapse), but their arithmetic is also vastly simplified. Multiplication operations of Equation (1) are expensive in terms of area and energy consumption, and they are replaced by one-bit exclusive NOR (XNOR) operations in Equation (2). Additionally, the real sums in Equation (1) are replaced by POPCOUNT operations, which are equivalent to integer sums with a low bit width.

It is possible to implement ASIC Binarized Neural Networks with solely CMOS (Ando et al., 2017; Bankman et al., 2018). However, a more optimal implementation would rely on emerging non-volatile memories and associate logic and memory as closely as possible. This approach can provide non-volatile neural networks and eliminate the von Neumann bottleneck entirely: the nanodevices can implement the synaptic weights, while the arithmetic can be done in CMOS. Most of the literature proposing the use of emerging memories as synapses relies on an ingenious technique to perform the multiplications and additions of Equation (1) that relies on analog electronics: the multiplications are done based on Ohm's law and the addition

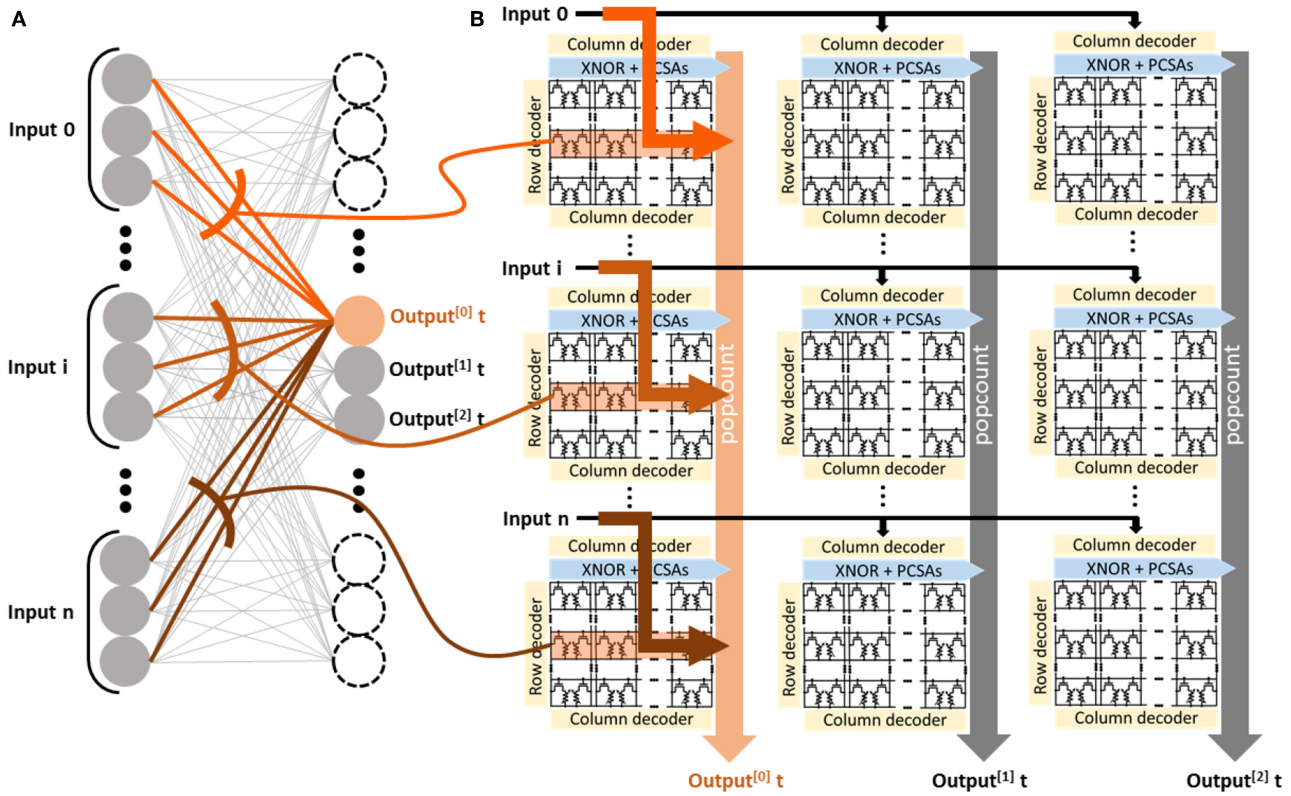


FIGURE 3 | (A) Schematization of the implemented Binarized Neural Network highlighting connections to one specific neuron. **(B)** Schematization of the full architecture to implement the Binarized Neural Network in the "parallel to sequential" configuration. The system assembles a memory block surrounded by logic circuits and moves minimal data between the blocks. The architecture is presented with three rows and three columns (i.e., $N = M = 3$) of kilobit memory blocks (i.e., $n = 32$).

based on Kirchoff's current law (Yu et al., 2016; Ambrogio et al., 2018). This analog approach can be transposed directly to binarized neural networks (Tang et al., 2017; Sun et al., 2018a,b; Yu, 2018). However, binarized neural networks are inherently digital objects that rely, as previously remarked, on simple logic operation: XNOR operations and low bit-width sums. Therefore, here, we investigate their implementation with purely digital circuitry. This concept also recently appeared in Natsui et al. (2018) and Giacomin et al. (2019) and in our preliminary version of this work (Bocquet et al., 2018). Our work is the first one to present measurements on a physical memory array that include the effect of bit errors.

A first realization is that the XNOR operations can be realized directly within the sense amplifiers. For this, we follow the pioneering work of Zhao et al. (2014), which shows that a precharge sense amplifier can be enriched with any logic operation. In our case, we can add four additional transistors in the discharge branches of a precharge sense amplifier (**Figure 2B**). These transistors can prevent the discharge and allow the implementation of the XNOR operation between input voltage X and the value stored in the complementary OxRAM devices in a single operation.

Based on the basic memory array with PCSAs enriched with XNOR, we designed the whole system implementing a Binarized Neural Network. The overall architecture is presented in **Figure 3**. It is inspired by the purely CMOS architecture proposed in Ando et al. (2017), adapted to the constraints of OxRAM. The design consists of the repetition of basic cells organized in a matrix of N by M cells. These basic cells incorporate a $n \times n$ OxRAM memory block with XNOR-enriched PCSAs and POPCOUNT logic. The whole system, which aims at computing the activation of neurons (Equation 2), features a degree of reconfigurability to adapt to different neural network topologies: it can be used either in a "parallel to sequential" or in a "sequential to parallel" configuration.

The parallel to sequential configuration (presented in **Figure 3**) can deal with layers with up to $n \times N$ input neurons and up to $n \times M$ output neurons. In this situation, at each clock cycle, the system computes the activations of M output neurons in parallel. At each clock cycle, each basic cell reads an entire row of its OxRAM memory array while performing the XNOR operation with input neuron values. The results are used to compute the POPCOUNT operation over a subset of the indices i in Equation (2), using fully digital five-bit counters embedded within the cell. Additional logic, called "popcount tree" and only activated in this configuration, computes the full POPCOUNT value operation over a column by successively adding the five-bit-wide partial POPCOUNT values. The activation value of the

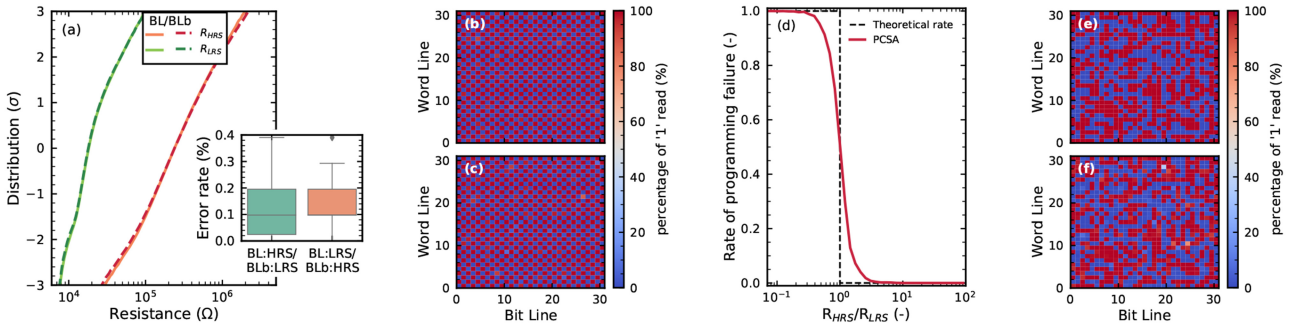


FIGURE 4 | (a) Distributions of the LRS and HRS of the OxRAM devices in an array programmed with a checkerboard pattern. RESET voltage of 2.5V, SET current of 55 μA, and programming time of 1 μs. **(b,c)** Proportion of 1 values read by the on-chip precharge sense amplifier, over 100 whole-array programming cycles of a memory array, for the two complementary checkerboards configuration. **(d)** Rate of programming failure indicated of the precharge sense amplifier circuits as a function of the ratio between HRS and LRS resistance (measured by a sense measure unit) in the same configuration as **(a–c)**. **(e–f)** Proportion of 1 values read by the on-chip precharge sense amplifier, over 100 whole-array programming cycles of a memory array, for the last layer of a binarized neural network trained on MNIST (details in body text).

neuron is obtained by subtracting the complete POPCOUNT value at the bottom of the column from a threshold value stored in a separate memory array; the signed bit of the result gives the activation value. At the next clock cycle, the next rows in the OxRAM memory arrays are selected, and the activations of the next M neurons are computed.

The sequential to parallel configuration (not presented), by contrast, can be chosen to deal with a neural network layer with up to n^2 inputs neurons and up to NM output neurons. In this configuration, each basic cell of the system computes the activation of one neuron A_j. The input neurons X_i are presented sequentially by subsets of n inputs. At each clock cycle, the digital circuitry therefore computes only a part of Equation (2). The partial POPCOUNT is looped to the same cell to compute the whole POPCOUNT sequentially. After the presentations of all inputs, the threshold is subtracted, the binary activation is extracted, and Equation (2) has been entirely computed.

This whole system was designed using synthesizable SystemVerilog. The memory blocks are described in behavioral SystemVerilog. We synthesized the system using the 130-nm design kit used for fabrication, as well as using the design kit of an advanced commercial 28-nm process for scaling projection.

All simulations reported in the results sections were performed using Cadence Incisive simulators. The estimates for system-level energy consumption were obtained using the Cadence Encounter tool. We used Value Change Dump (VCD) files extracted from simulations of practical tasks so that the obtained energy values would reflect the operation of the system realistically.

3. RESULTS

3.1. Differential Memory Allows Memory Operation at Reduced Bit Error Rate

This section first presents the results of electrical characterization of the differential OxRAM arrays. We program the array with checkerboard-type data, alternating zero and one bits, using programming times of one microsecond. For programming devices in HRS (RESET operation), the access transistor is fully opened, and a reset voltage of 2.5 V is used. For programming devices in LRS (SET operation), the gate voltage of the access transistor is chosen to ensure a compliance current of 55μA. **Figure 4a** shows the statistical distribution of the LRS and HRS of the cells, based on 100 programming cycles of the full array. This graph uses a standard representation in the memory field, where the y axis is expressed as the number of standard deviations of the distribution (Ly et al., 2018). The figure superimposes distributions of left (BL) and right (BLb) columns of the array, and no significant difference is seen between BL and BLb devices. The LRS and HRS distributions are clearly separate but overlap at a value of three standard deviations, which makes bit errors possible. If a 1T1R structure were used, a bit error rate of 0.012 (1.2%) would be seen with this distribution. By contrast, at the output of the precharge sense amplifiers, a bit error rate of 0.002 (0.2%) is seen, providing a first indication of the benefits of the 2T2R approach. **Figures 4b,c** show the mean error (using the 2T2R configuration) on the whole array for the two types of checkerboards. We see that all devices can be programmed in HRS and LRS. A few devices have an increased bit error rate. This graph highlights the existence of both cycle-to-cycle and device-to-device variability and the absence of "dead" cells.

We now validate in detail the functionality of the precharge sense amplifiers. The precise resistance of devices is first measured by deactivating the precharge sense amplifiers and using the external source monitor units. Then, the precharge sense amplifiers are reactivated, and a sense operation is performed. **Figure 4d** plots the mean measurement of the sense amplifiers as a function of the ratio between the two resistances that are being compared, superimposed on the ideal behavior of a sense amplifier. The sense amplifiers show excellent functionality but can make mistakes if the two resistances differ by less than a factor of five. Finally, **Figures 4e,f** show the results of repeating the experiments of **Figures 4b,c** in a more realistic situation and on a different die. We trained a memory array 100

times with weights corresponding to the last layer of a binarized neural network trained on the MNIST task of handwritten digit recognition. As in the checkerboard case, no dead cell is seen, and a similar degree of cycle-to-cycle and device-to-device variation is exhibited.

The programming rates are strongly dependent on the programming conditions. **Figure 5** shows the mean number of incorrect bits on a whole array for various combinations of programming time (from 0.1 to 100 μs), RESET voltage (between 1.5 and 2.5 Volts), and SET compliance current (between 28 and 200 μA). We observe that the bit error rate depends extensively on these three programming parameters, the SET compliance current having the most significant impact.

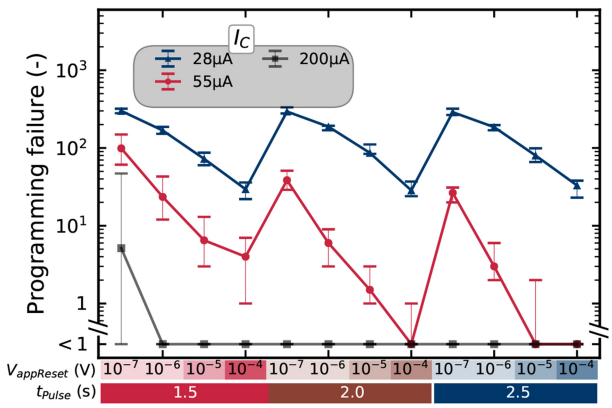


FIGURE 5 | Number of errors for different programming conditions, as measured by precharge sense amplifier, for a 2T2R configuration on a kilobit memory array. The "< 1" label means that no errors were detected. The error bars present the minimum and maximum number of detected errors over five repetitions of the experiments.

In **Figure 6**, we look more precisely at the effects of cycle-to-cycle device variability and device aging. A device and its complementary device were programmed through 700 million cycles. **Figures 6A,B** show the distribution of the LRS and HRS of the device under test and its complementary device after different number of cycles, ranging from the first one to the last one. We can observe that as the devices are cycled, the LRS and HRS distributions become less separated and start to overlap at a lower number of standard deviations. This translates directly to the mean resistance of the devices in HRS and LRS (**Figures 6C,D**), which become closer as the device ages. More importantly, the aging process impacts the device bit error rate (**Figure 6E**): the bit error rate of the device and its complementary device increase by several orders of magnitudes over the lifetime of the device. The same effect is seen on the bit error rate resulting from the precharge sense amplifier (2T2R), but it remains at a much lower level: while the 1T1R bit error rate goes above 10^{-3} after a few million cycles, the 2T2R remains below this value over the 700 million cycles. This result highlights that the concept of cyclability depends on the acceptable bit error rate and that the cyclability at constant bit error rate can be considerably extended by using the 2T2R structure. It should also be highlighted that the cyclability depends tremendously on the programming conditions. **Figures 7A,B** shows endurance measurements with a reset voltage of 1.5 V (all other programming conditions are identical to **Figures 6A–E**). We can see that the device experiences no degradation through more than ten billion cycles. Over that time, the 2T2R bit error rate remains below 10^{-4}.

We now aim at quantifying and benchmarking the benefits of the 2T2R structure more precisely. We performed extensive characterization of bit error rates on the memory array in various

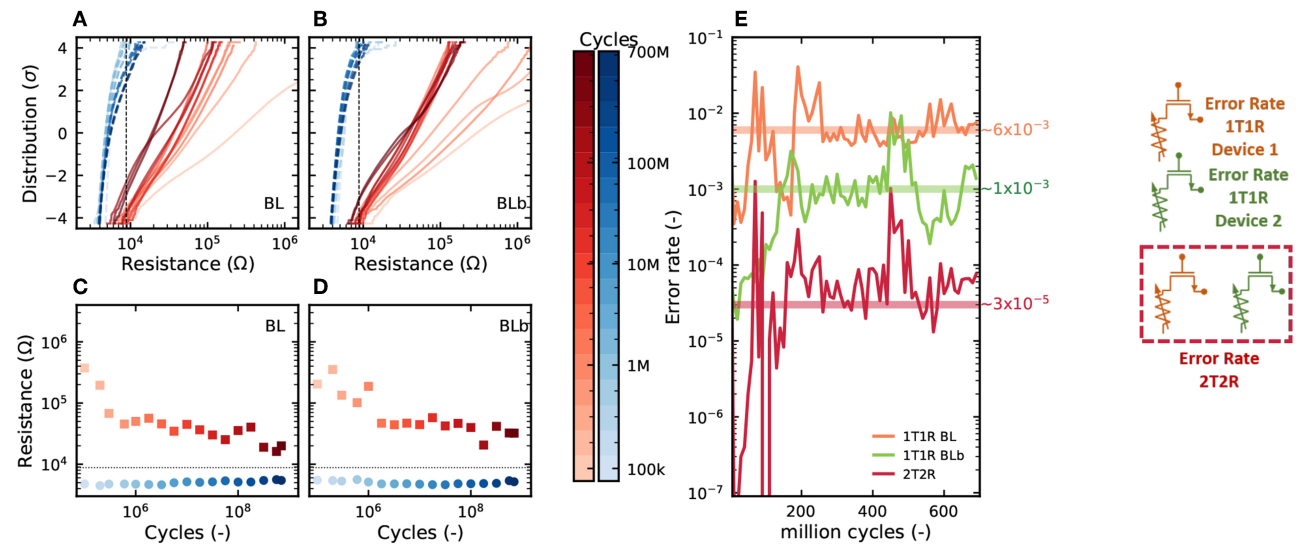


FIGURE 6 | (A,B) Distribution of resistance values, **(C,D)** mean resistance value, and **(E)** mean bit error rate over 10 million cycles, as measured by precharge sense amplifier, in the 2T2R configuration, as a function of the number of cycles for which a device has been programmed. RESET voltage of 2.5 V, SET current of 200 μA, and programming time of 1 μs.

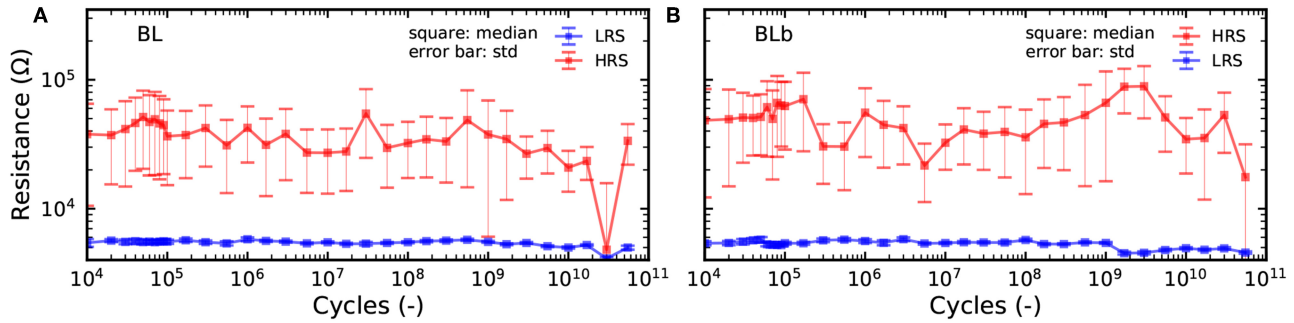


FIGURE 7 | (A,B) Mean resistance value of the BL and BLb device over 10 thousand cycles for measurements of a device pair over 5×10^{10} cycles. RESET voltage of 1.5V, SET current of 200 μA, and programming time of 1 μs.

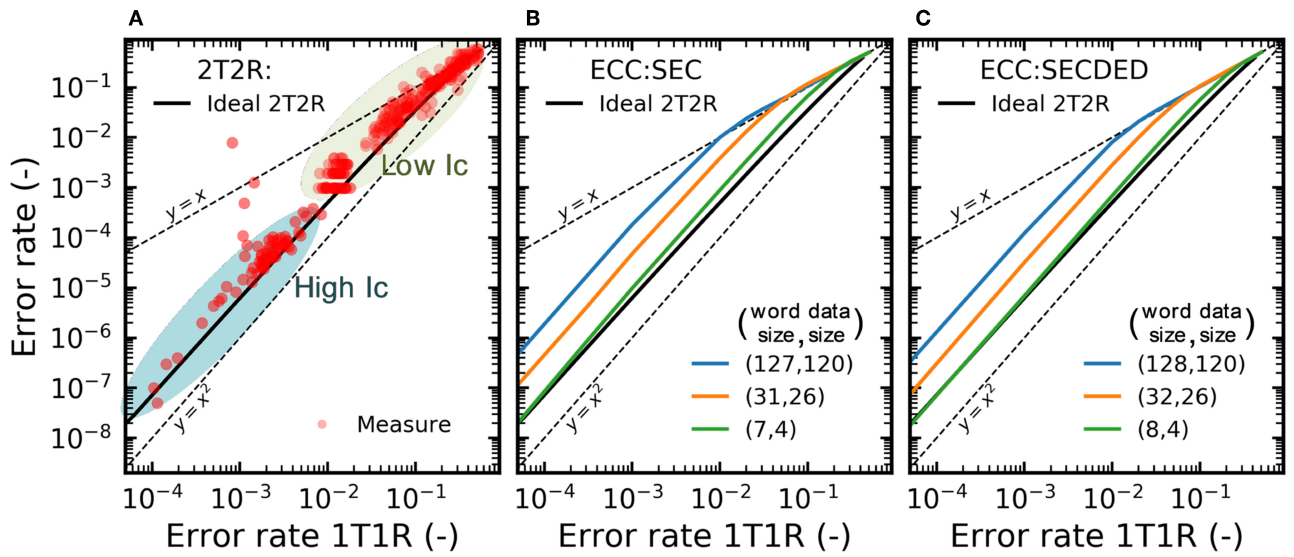


FIGURE 8 | (A) Experimental bit error rate of the 2T2R array, as measured by precharge sense amplifier, as a function of the bit error rate obtained with individual (1T1R) RRAM devices under the same programming conditions. The detailed methodology for obtaining this graph is presented in the body text. Bit error rate obtained with **(B)** Single Error Correcting (SEC) and **(C)** Single Error Correcting Double Error Detection (SECDED) ECC as a function of the error rate of the individual devices.

regimes. **Figure 8A** presents different experiments where the 2T2R bit error rate is plotted as a function of the bit error rate that would be obtained by using a single device programmed in the same conditions. The different points are obtained by varying the compliance current Ic during SET operations, and the graph associates two types of experiments:

- The points marked as "Low Ic" are obtained using whole-array measurement, where devices are programmed with a low SET compliance current to ensure a high error rate. Each device in the memory array is programmed once (following the checkerboard configuration), and all synaptic weights are read using the on-chip precharge sense amplifiers. The plotted bit error rate is the proportion of weights for which the read weight differs from the weight value targeted by the programming operation.
- The points marked as "High Ic" are obtained by measurements on a single device pair. A single 2T2R structure in the array is programmed ten million times by alternating programming to $+1$ and -1 values. The value programmed in the 2T2R structure is read using an on-chip precharge sense amplifier after each programming operation. The plotted bit error rate is the proportion of read operations for which the read weight differs from the targeted value.

We can see that the 2T2R bit error rate is always lower than the 1T1R one. The difference is larger for a lower bit error rate and reaches four orders of magnitude for a 2T2R bit error rate of 10^{-8}. The black line represents calculations where the precharge sense amplifier is supposed to be ideal (i.e., to follow the idealized dotted characteristics of **Figure 4C**).

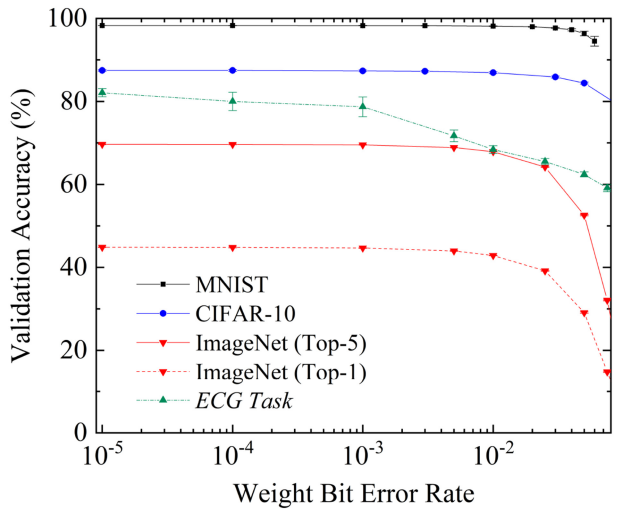


FIGURE 9 | Recognition rate on the validation datasets of the fully connected neural network for MNIST, the convolutional neural network for CIFAR10, and AlexNet for ImageNet (Top 5 and Top 1) and in the ECG analysis task, as a function of the bit error rate over the weights during inference. Each experiment was repeated five times; the mean recognition rate is presented. Error bars represent one standard deviation.

To interpret the results of the 2T2R approach in a broader perspective, we benchmark them with standard error-correcting codes. **Figures 8B,C** show the benefits of two codes, using the same plotting format as **Figure 8A**: a Single Error Correction (SEC) and a Single Error Correction Double Error Detection (SECDED) code, presented with different degrees of redundancy. These simple codes, formally known as Hamming and extended Hamming codes, are widely used in the memory field. Interestingly, we see that the benefit of these codes are very similar to the benefit of our 2T2R approach with an ideal sense amplifier, at equivalent memory redundancy (e.g., SECDED(8,4)), although our approach uses no decoding circuit and performs the equivalent of error correction directly within the sense amplifier. By contrast, ECCs can also reduce bit errors, to a lesser extent, using less redundancy, but the required decoding circuits utilize hundreds to thousands of logic gates (Gregori et al., 2003). In a context where logic and memory are tightly integrated, these decoding circuits would need to be repeated many times, and as their logic is much more complicated than that of binarized neural networks, they would be the dominant source of computation and energy consumption. ECC circuits are also incompatible with the idea of integrating XNOR operations within the sense amplifiers and cause important read latency.

3.2. Do All Errors Need to Be Corrected?

Based on the results of the electrical measurements, and before discussing the whole system, it is important to determine the OxRAM bit error rate levels that can be tolerated for applications. To answer this question, we performed simulations of binarized neural networks on four different tasks:

- MNIST handwritten digit classification (LeCun et al., 1998), the canonical task of machine learning. We use a fully connected neural network with two 1024-neuron hidden layers.
- The CIFAR-10 image recognition task (Krizhevsky and Hinton, 2009), which consists of recognizing 32 × 32 color images spread between ten categories of vehicles and animals. We use a deep convolutional network with six convolutional layers using kernels of 3 × 3 and a stride of one, followed by three fully connected layers.
- The ImageNet recognition task, which consists of recognizing 224 × 224 color images out of 1000 classes. This task is considerably more difficult than MNIST and CIFAR-10. We use the historic AlexNet deep convolutional neural network (Krizhevsky et al., 2012).
- A medical task involving the analysis of electrocardiography (ECG) signals: automatic detection of electrode misplacement. This binary classification challenge takes as input the ECG signals of twelve electrodes. The experimental trial data are sampled at 250 Hz and have a duration of three seconds each. To solve this task, we employ a convolutional neural network composed of five convolutional layers and two fully connected layers. The convolutional kernel (sliding window) sizes decrease from 13 to 5 in each subsequent layer. Each convolutional layer produces 64 filters detecting different features of the signal.

Fully binarized neural networks were trained on these tasks on NVIDIA Tesla GPUs using Python and the PyTorch deep-learning framework. Once the neural networks were trained, we ran them on the dataset validation sets, artificially introducing errors into the neural network weights (meaning some +1 weights were replaced by −1 weights, and reciprocally). Using this technique, we could emulate the impact of OxRAM bit errors. **Figure 9** shows the resulting validation accuracy as a function of the introduced bit error rate for the four tasks considered. In the case of ImageNet, both the Top-1 (proportion of validation images where the right label is the top choice of the neural network) and the Top-5 (proportion of validation images where the right label is within the top five choices of the neural network) are included.

On the three-vision tasks (MNIST, CIFAR-10, and ImageNet), we see that extremely high levels of bit errors can be tolerated: up to a bit error rate of 10^{-4}, the network performs as well as with no errors. Minimal performance reduction starts to be seen with bit error rates of 10^{-3} (the Top-5 accuracy on ImageNet is degraded from 69.7% to 69.5%). At bit error rates of 0.01, the performance reduction becomes significant. The reduction is more substantial for harder tasks: MNIST accuracy is only degraded from 98.3% to 98.1%, CIFAR-10 accuracy is degraded from 87.5% to 86.9%, while ImageNet Top-5 accuracy is degraded from 69.7% to 67.9%.

The ECG task also shows extremely high bit error tolerance, but bit errors have an effect more rapidly than in the vision tasks. At a bit error rate of 10^{-3}, the validation accuracy is reduced from 82.1% to 78.7%, and at a bit error rate of 0.01, to 68.4%. This difference between vision and ECG tasks probably originates in the fact that ECG signals carry a lot less redundant information

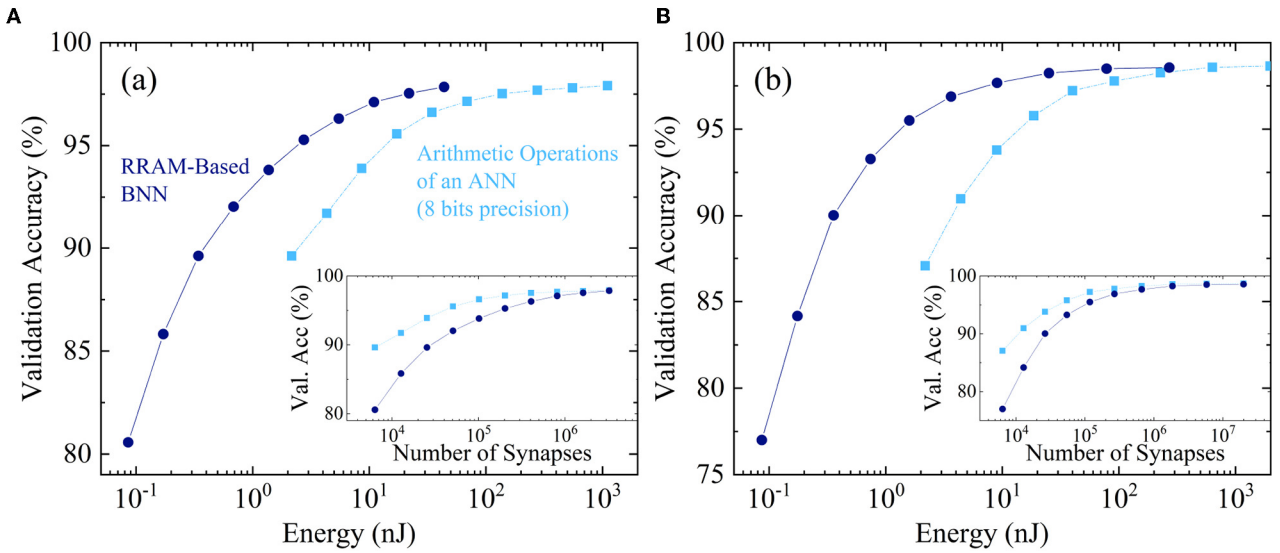


FIGURE 10 | Dark blue circles: MNIST validation accuracy as a function of the inference energy of our Binarized Neural Network hardware design. Light blue squares: same, as a function of the energy used for arithmetic operation in a real-valued neural network employing eight-bit fixed-point arithmetic. The different points are obtained by varying the number of hidden neurons in **(A)** a one hidden layer neural network and **(B)** a two hidden layers neural network. Insets: number of synapses in each situation.

than images. Nevertheless, we see that even for ECG tasks, high bit error rates can be accepted with regards to the standards of conventional digital electronics.

4. DISCUSSION

4.1. Projection at the System Level

4.1.1. Impact of In-Memory Computation

We now use all the paper results to discuss the potential of our approach. Based on our ASIC design, using the energy-evaluation technique described at the end of the Methods section, we find that our system would consume 25 *nJ* to recognize one handwritten digit, using a fully connected neural network with two hidden layers of 1,024 neurons. This is considerably less than processor-based options. For example, Lane et al. (2016) analyses the energy consumption of inference on CPUs and GPUs: operating a fully connected neural network with two hidden layers of 1,000 neurons requires 7 to 100 millijoules on a low-power CPU (from NVIDIA Tegra K1 or Qualcomm Snapdragon 800 systems on the chip) and 1.3 millijoules on a low-power GPU (NVIDIA Tegra K1).

These results are not surprising due to the considerable overhead for accessing memory in modern computers. For example, Pedram et al. (2017) shows that accessing data in a static RAM cache consumes around fifty times more energy than the integer addition of this data. If the data is stored in the external dynamic RAM, the ratio is increased to more than 3,000. Binarized Neural Networks require minimal arithmetic: no multiplication and only integer addition with a low bit width. When operating a Binarized Neural Network on a CPU or GPU, almost the entirety of the energy is used to move data, and the inherent topology of the neural network is not exploited to reduce data movement. Switching to in-memory or near-memory computing approaches therefore has the potential to reduce energy consumption drastically for such tasks. This is especially true as, in inference hardware, synaptic weights are static and can be programmed to memory only once if the circuit does not need to change function.

4.1.2. Impact of Binarization

We now look specifically at the benefits of relying on Binarized Neural Networks rather than real-valued digital ones. Binarized Neural Networks feature considerably simpler architecture than conventional neural networks but also require an increased number of neurons and synapses to achieve equivalent accuracy. It is therefore essential to compare the binarized and real-value approaches.

Most digital ASIC implementations of neural networks have an inference function with eight-bit fixed-point arithmetic, the most famous example being the tensor processing units developed by Google (Jouppi et al., 2017). At this precision, no degradation is usually seen for inference with regards to 32- and 64-bis floating-point arithmetic.

To investigate the benefits of Binarized Neural Networks, **Figure 10** looks at the energy consumption for inference over a single MNIST digit. We consider two architectures: a neural network with a single hidden layer (**Figure 10A**) and another with two hidden layers (**Figure 10B**), and we vary the number of hidden neurons. **Figures 10A,B** plot on the *x*-axis the estimated energy consumption of a Binarized Neural Network using our architecture based on the flow presented in the Methods section. It also plots the energy required for the arithmetic operations

(sum and product) of an eight-bit fixed-point regular neural network, neglecting the overhead that is considered for the Binarized Neural Network. For both types of networks, the y-axis shows the resulting accuracy in the MNIST task. We see that at equivalent precision, the Binarized Neural Network always consumes less energy than the arithmetic operations of the real-valued one. It is remarkable that the energy benefit depends significantly on the targeted accuracy and should therefore be investigated on a case-by-case basis. The highest energy benefits, a little less than a factor ten, are seen at lower targeted precision.

Binarized Neural Networks have other benefits with regards to real-valued digital networks: if the weights are stored in RRAM, the programming energy is reduced due to the lower memory requirements of Binarized Neural Networks. The area of the overall circuit is also expected to be reduced due to the absence of multipliers, which are high-area circuits.

4.1.3. Comparison With Analog Approaches

As mentioned in the introduction, a widely studied approach for implementing neural networks with RRAM is to rely on an analog electronics strategy, where Ohm's law is exploited for implementing multiplications and Kirchoff's current law for implementing additions (Prezioso et al., 2015; Serb et al., 2016; Shafiee et al., 2016; Ambrogio et al., 2018; Li et al., 2018; Wang et al., 2018). The digital approach presented in this paper cannot be straightforwardly compared to the analog approach: the detailed performance of the analog approach depends tremendously on its implementation details, device specifics, and the size of the neural network. Nevertheless, several points can be raised.

First, the programming of the devices is much simpler in our approach than in the analog one: one only needs to program a device and its complementary device in LRS and HRS, which can be achieved by two programming pulses. It is not necessary to verify the programming operation, as the neural network has inherent bit-error tolerance. Programming RRAM for analog operation is a more challenging task and usually requires a sequence of multiple pulses (Prezioso et al., 2015), which leads to higher programming energy and device aging.

For the neural network operation, the analog approach and ours function differently. Our approach reads synaptic values using the sense amplifier, which is a highly energy-efficient and fast circuit that can operate at hundreds of picoseconds in advanced CMOS nodes (Zhao et al., 2014). This sense amplifier inherently produces the multiplication operation, and then the addition needs to be performed using a low bit-width digital integer addition circuit. The ensemble of a read operation and the corresponding addition typically consumes fourteen femtojoules in our estimates in advanced node. In the analog approach, the read operation is performed by applying a voltage pulse and inherently produces the multiplication through Ohm's law but also the addition though Kirchoff law. This approach is attractive, but, on the other hand, requires the use of CMOS analog overhead circuitry such as an operational amplifier, which can bring high energy and area overhead. Which approach is the most energy-efficient between ours and the analog one will probably depend tremendously on memory size, application, and targeted accuracy.

Another advantage of the digital approach is that it is much simpler to design, test, and verify as it relies on all standard VLSI design tools. On the other hand, an advantage of the analog approach is that it may, for a small memory size, function without access transistors, resulting in higher memory densities (Prezioso et al., 2015).

TABLE 1 | RRAM Properties with different programming conditions.

Programming condition	Strong (Figure 6)	Optimized endurance (Figure 7)	Optimized programming energy
SET compliance current	$200\,\mu A$	$200\,\mu A$	$200\,\mu A$
RESET voltage	$2.5\,V$	$1.5\,V$	$2\,V$
Programming time	$1\,\mu s$	$1\,\mu s$	$100\,ns$
2T2R bit error rate (before aging)	$< 10^{-7}$	$< 10^{-4}$	$< 10^{-5}$
Programming energy (SET/RESET)	$200 \sim 400\,pJ$	$150 \sim 400\,pJ$	$20 \sim 30\,pJ$
Cyclability (number of cycles at BER $< 10^{-3}$)	$> 10^8$	$> 10^{10}$	$> 10^8$

4.1.4. Impact in Terms of Programming Energy and Device Aging

A last comment is that the bit error tolerance of binarized neural networks can have considerable benefits at the system level. **Table 1** summarizes the measured properties of RRAM cells under different programming conditions, chosen from those presented in **Figure 5**. We consider only the conditions with bit error rates below 10^{-3} (i.e., corresponding to a "$<$ 1" data point in **Figure 5**), as this constraint makes them appropriate for use for all tasks considered in section 3.2. The "strong" programming conditions are the ones presented in **Figure 6**. They feature a low bit error rate before aging but high programming energy. The other two columns correspond to two optimized choices. The conditions optimized for programming energy are the conditions of **Figure 5** with bit error rates below 10^{-3} and the lowest programming energy. They use a lower RESET voltage ($2.0\,V$) than the strong conditions and a shorter programming time ($100\,ns$). The cyclability of the device—defined as the number of programming cycles a cell can perform while retaining a bit error rate below 10^{-3}—remains comparable to the strong programming conditions. The conditions optimized for endurance are, by contrast, the conditions of **Figure 5** with a bit error rate below 10^{-3} and the highest cyclability: more than 10^{10} cycles, as already evidenced in **Figure 7**. These conditions use a low RESET voltage $1.5\,V$ but require a programming time of $1\,\mu s$.

4.2. Conclusion

This work proposes an architecture for implementing binarized neural networks with RRAMs and incorporates several biologically plausible ideas:

- Fully co-locating logic memory,
- Relying only on-low precision computation (through the Binarized Neural Network concept),
- Avoiding multiplication altogether, and
- Accepting some errors without formal error correction.

At the same time, our approach relies on conventional microelectronics ideas that are non-biological in nature:

- Relying on fixed-point arithmetic to compute sums, whereas brains use analog computation,
- Using sense amplifier circuits, which are not brain-inspired, and
- Using a differential structure to reduce errors, a traditional electrical engineering strategy.

Based on these ideas, we designed, fabricated, and extensively tested a memory structure with its peripheral circuitry and designed and simulated a full digital system based on this memory structure. Our results show that this structure allows neural networks to be implemented without the use of Error-Correcting Codes, which are usually used with emerging memories. Our approach also features very attractive properties in terms of energy consumption and can allow that use of RRAM devices in a "weak" programming regime, where they have low programming energy and outstanding endurance. These results highlight that although in-memory computing cannot efficiently rely on Error-Correcting Codes, it can still function without stringent requirements on device variability if a differential memory architecture is chosen.

When working on bioinspiration, drawing the line between bio-plausibility and embracing the differences between the nanodevices of the brain and electronic devices is always challenging. In this project, we highlight that digital electronics can be enriched by biologically plausible ideas. When working with nanodevices, it can be beneficial to incorporate device physics questions into the design, and not necessarily to target the level of determinism that we have been accustomed to by CMOS.

This work opens multiple prospects. On the device front, it could be possible to develop more integrated 2T2R structures to increase the density of the memories. The concept of this work could also be adapted to other kinds of emerging memories, such as phase-change memories and spin torque magnetoresistive memories. At the system level, we are now in a position to fabricate larger systems and to investigate the extension of our concept to more varied forms of neural network architecture such as convolutional and recurrent networks. In the case of convolutional networks, a dilemma appears between taking the in-memory computing approach to the fullest degree, by replicating physically convolutional kernels or implementing some sequential computation to minimize resources, as works have started to evaluate already. These considerations open the way for truly low-energy artificial intelligence for both servers and embedded systems.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request from the corresponding author.

AUTHOR CONTRIBUTIONS

EV and EN were in charge of fabrication and of the initial RRAM characterization. J-MP performed the CMOS design of the memory array. MB performed the electrical characterization. TH designed the BNN systems. TH, BP, and DQ performed the BNN simulations. DQ wrote the initial version of the paper. J-OK, EV, J-MP, and DQ planned and supervised the project. All authors participated in data analysis and the writing of the paper.

FUNDING

This work was supported by European Research Council Starting Grant NANOINFER (reference: 715872) and Agence Nationale de la Recherche grant NEURONIC (ANR-18-CE24-0009).

REFERENCES

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R. M., Boybat, I., Nolfo, C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558:60. doi: 10.1038/s41586-018-0180-5

Ando, K., Ueyoshi, K., Orimo, K., Yonekawa, H., Sato, S., Nakahara, H., et al. (2017). "Brein memory: a 13-layer 4.2 k neuron/0.8 M synapse binary/ternary reconfigurable in-memory deep neural network accelerator in 65 nm CMOS," in *Proceedings of the VLSI Symposium on Circuits* (IEEE), C24–C25.

Bankman, D., Yang, L., Moons, B., Verhelst, M., and Murmann, B. (2018). An always-on 3.8muJ/86 % cifar-10 mixed-signal binary cnn processor with all memory on chip in 28-nm CMOS. *IEEE J. Solid State Circ.* 54, 158–172. doi: 10.1109/JSSC.2018.2869150

Bocquet, M., Hirztlin, T., Klein, J.-O., Nowak, E., Vianello, E., Portal, J.-M., et al. (2018). "In-memory and error-immune differential rram implementation of binarized deep neural networks," in *IEDM Technical Digest* (San Francisco, CA: IEEE), 20.6.1.

Chen, A. (2016). A review of emerging non-volatile memory (NVM) technologies and applications. *Solid State Electron.* 125, 25–38. doi: 10.1016/j.sse.2016.07.006

Chen, W. H., Li, K. X., Lin, W. Y., Hsu, K. H., Li, P. Y., Yang, C. H., et al. (2018). "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *Proceedings of the ISSCC* (San Francisco, CA), 494–496.

Chen, W. H., Lin, W. J., Lai, L. Y., Li, S., Hsu, C. H., Lin, H. T., et al. (2017). "A 16Mb dual-mode ReRAM macro with sub-14ns computing-in-memory and memory functions enabled by self-write termination scheme," in *IEDM Technical Digest* (San Francisco, CA), 28.2.1–28.2.4. doi: 10.1109/IEDM.2017.8268468

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks: training deep neural networks with weights and activations constrained to $+1$ or -1. *arXiv:1602.02830*.

Covi, E., Brivio, S., Serb, A., Prodromakis, T., Fanciulli, M., and Spiga, S. (2016). Analog memristive synapse in spiking networks implementing unsupervised learning. *Front. Neurosci.* 10:482. doi: 10.3389/fnins.2016.00482

Dong, Q., Kim, Y., Lee, I., Choi, M., Li, Z., Wang, J., et al. (2017). "11.2 a 1mb embedded nor flash memory with 39μw program power for mm-scale high-temperature sensor nodes," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)* (San Francisco, CA: IEEE), 198–199.

Editorial (2018). Big data needs a hardware revolution. *Nature* 554:145. doi: 10.1038/d41586-018-01683-1

Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. *Nat. Rev. Neurosci.* 9:292. doi: 10.1038/nrn2258

Giacomin, E., Greenberg-Toledo, T., Kvatinsky, S., and Gaillardon, P.-E. (2019). A robust digital rram-based convolutional block for low-power image processing and learning applications. *IEEE Trans. Circ. Syst. I Regul. Papers* 66, 643–654. doi: 10.1109/TCSI.2018.2872455

Gregori, S., Cabrini, A., Khouri, O., and Torelli, G. (2003). On-chip error correcting techniques for new-generation flash memories. *Proc. IEEE* 91, 602–616. doi: 10.1109/JPROC.2003.811709

Grossi, A., Nowak, E., Zambelli, C., Pellissier, C., Bernasconi, S., Cibrario, G., et al. (2016). "Fundamental variability limits of filament-based RRAM," in *IEDM Technical Digest* (San Francisco, CA: IEEE), 4–7. doi: 10.1109/IEDM.2016.7838348

Grossi, A., Vianello, E., Zambelli, C., Royer, P., Noel, J.-P., Giraud, B., et al. (2018). "Experimental investigation of 4-kb rram arrays programming conditions suitable for TCAM," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26, 2599–2607. doi: 10.1109/TVLSI.2018.2805470

Hsieh, W.-T., Chih, Y.-D., Chang, J., Lin, C.-J., and King, Y.-C. (2017). "Differential contact RRAM pair for advanced CMOS logic NVM applications," in *Proceedings of the SSDM* (Sendai), 171. doi: 10.7567/SSDM.2017.D-1-02

Ielmini, D., and Wong, H.-S. P. (2018). In-memory computing with resistive switching devices. *Nat. Electr.* 1:333. doi: 10.1038/s41928-018-0092-2

Indiveri, G., and Liu, S.-C. (2015). Memory and information processing in neuromorphic systems. *Proc. IEEE* 103, 1379–1397. doi: 10.1109/JPROC.2015.2444094

Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., et al. (2017). "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the ISCA* (Toronto, ON: IEEE), 1–12.

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv:1412.6980*.

Klemm, K., and Bornholdt, S. (2005). Topology of biological networks and reliability of information processing. *Proc. Natl. Acad. Sci. U.S.A.* 102, 18414–18419. doi: 10.1073/pnas.0509132102

Krizhevsky, A., and Hinton, G. (2009). *Learning Multiple Layers of Features From Tiny Images*. Technical report, Citeseer.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 1097–1105.

Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L., et al. (2016). "Deepx: a software accelerator for low-power deep learning inference on mobile devices," in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks* (Vienna: IEEE Press), 23.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., et al. (2018). Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks. *Nat. Commun.* 9:2385. doi: 10.1038/s41467-018-04484-2

Lin, X., Zhao, C., and Pan, W. (2017). "Towards accurate binary convolutional neural network," in *Advances in Neural Information Processing Systems* (Long Beach, CA), 345–353.

Ly, D. R. B., Grossi, A., Fenouillet-Beranger, C., Nowak, E., Querlioz, D., and Vianello, E. (2018). Role of synaptic variability in resistive memory-based spiking neural networks with unsupervised learning. *J. Phys. D Appl. Phys.* 51:444002. doi: 10.1088/1361-6463/aad954

Merrikh-Bayat, F., Guo, X., Klachko, M., Prezioso, M., Likharev, K. K., and Strukov, D. B. (2017). High-performance mixed-signal neurocomputing with nanoscale floating-gate memory cell arrays. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 4782–4790. doi: 10.1109/TNNLS.2017.2778940

Natsui, M., Chiba, T., and Hanyu, T. (2018). Design of MTJ-based nonvolatile logic gates for quantized neural networks. *Microelectr. J.* 82, 13–21. doi: 10.1016/j.mejo.2018.10.005

Pedram, A., Richardson, S., Horowitz, M., Galal, S., and Kvatinsky, S. (2017). Dark memory and accelerator-rich system optimization in the dark silicon era. *IEEE Design Test* 34, 39–50. doi: 10.1109/MDAT.2016.2573586

Prezioso, M., Merrikh-Bayat, F., Hoskins, B., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521:61. doi: 10.1038/nature14441

Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). "XNOR-net: imagenet classification using binary convolutional neural networks," in *Proceedings of the ECCV* (Amsterdam: Springer), 525–542.

Saïghi, S., Mayr, C. G., Serrano-Gotarredona, T., Schmidt, H., Lecerf, G., Tomas, J., et al. (2015). Plasticity in memristive devices for spiking neural networks. *Front. Neurosci.* 9:51. doi: 10.3389/fnins.2015.00051

Serb, A., Bill, J., Khiat, A., Berdan, R., Legenstein, R., and Prodromakis, T. (2016). Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses. *Nat. Commun.* 7:12611. doi: 10.1038/ncomms12611

Shafiee, A., Nag, A., Muralimanohar, N., Balasubramonian, R., Strachan, J. P., Hu, M., et al. (2016). "ISAAC: a convolutional neural network accelerator with *in-situ* analog arithmetic in crossbars," in *ACM SIGARCH Computer Architecture News*, Vol. 44 (New York, NY), 14–26.

Shih, Y.-H., Hsu, M.-Y., Lin, C. J., and King, Y.-C. (2017). "Twin-bit via RRAM in 16nm FinFET logic technologies," in *Proceedings of the SSDM* (Sendai), 137.

Sun, X., Peng, X., Chen, P.-Y., Liu, R., Seo, J.-S., and Yu, S. (2018a). "Fully parallel rram synaptic array for implementing binary neural network with $(+1, -1)$ weights and $(+1, 0)$ neurons," in *Proceedings of the ASP-DAC* (Jeju: IEEE Press), 574–579. doi: 10.1109/ASPDAC.2018.8297384

Sun, X., Yin, S., Peng, X., Liu, R., Seo, J.-S., and Yu, S. (2018b). XNOR-RRAM: a scalable and parallel resistive synaptic architecture for binary neural networks. *Algorithms* 2:3. doi: 10.23919/DATE.2018.8342235

Tang, T., Xia, L., Li, B., Wang, Y., and Yang, H. (2017). "Binary convolutional neural network on RRAM," in *Proceedings of the ASP-DAC* (Tokyo: IEEE), 782–787.

Wang, Z., Ambrogio, S., Balatti, S., and Ielmini, D. (2015). A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning in neuromorphic systems. *Front. Neurosci.* 8:438. doi: 10.3389/fnins.2014.00438

Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., et al. (2018). Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electr.* 1:137. doi: 10.1038/s41928-018-0023-2

Yu, S. (2018). Neuro-inspired computing with emerging nonvolatile memorys. *Proc. IEEE* 106, 260–285. doi: 10.1109/JPROC.2018.2790840

Yu, S., Li, Z., Chen, P.-Y., Wu, H., Gao, B., Wang, D., et al. (2016). "Binary neural network with 16 mb rram macro chip for classification and online training," in *IEDM Technical Digest* (San Francisco, CA: IEEE), 16–12.

Zhao, W., Chappert, C., Javerliac, V., and Noziere, J.-P. (2009). High speed, high stability and low power sensing amplifier for MTJ/CMOS hybrid logic circuits. *IEEE Trans. Magn.* 45, 3784–3787. doi: 10.1109/TMAG.2009.2024325

Zhao, W., Moreau, M., Deng, E., Zhang, Y., Portal, J.-M., Klein, J.-O., et al. (2014). Synchronous non-volatile logic gate design based on resistive switching memories. *IEEE Trans. Circ. Syst. I Regul. Pap.* 61, 443–454. doi: 10.1109/TCSI.2013.2278332

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Hirtzlin, Bocquet, Penkovsky, Klein, Nowak, Vianello, Portal and Querlioz. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 17 January 2020
doi: 10.3389/fnins.2019.01386

Memristor-Based Edge Detection for Spike Encoded Pixels

Daniel J. Mannion, Adnan Mehonic, Wing H. Ng and Anthony J. Kenyon*

Department of Electronic and Electrical Engineering, University College London, London, United Kingdom

Memristors have many uses in machine learning and neuromorphic hardware. From memory elements in dot product engines to replicating both synapse and neuron wall behaviors, the memristor has proved a versatile component. Here we demonstrate an analog mode of operation observed in our silicon oxide memristors and apply this to the problem of edge detection. We demonstrate how a potential divider exploiting this analog behavior can prove a scalable solution to edge detection. We confirm its behavior experimentally and simulate its performance on a standard testbench. We show good performance comparable to existing memristor based work with a benchmark score of 0.465 on the BSDS500 dataset, while simultaneously maintaining a lower component count.

Keywords: memristor, edge detection, computer vision, spiking neural networks, neuromorphic computing

OPEN ACCESS

Edited by:
Stefano Brivio,
Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Wei Wang,
Politecnico di Milano, Italy
Antonio La Porta,
IBM Research–Zurich, Switzerland

****Correspondence:***
Daniel J. Mannion
daniel.mannion.13@ucl.ac.uk

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *28 June 2019*
Accepted: *09 December 2019*
Published: *17 January 2020*

Citation:
Mannion DJ, Mehonic A, Ng WH and Kenyon AJ (2020) Memristor-Based Edge Detection for Spike Encoded Pixels. Front. Neurosci. 13:1386. doi: 10.3389/fnins.2019.01386

1. INTRODUCTION

Interest in the application of memristors was originally driven by their potential as non-volatile memory elements. Subsequent work has demonstrated their capability to compute, to emulate biological synapses, and even to perform some of the functions of the biological neuron. Resistance switches, a sub-class of memristors, can be seen as devices that switch between two or more discrete resistance states. However, they exhibit very rich resistance dynamics under a variety of electrical stimuli. Here we demonstrate that it is not necessary to fully switch such devices to obtain useful functionality. They can be operated in an analog regime to perform elementary computing tasks: in our example, edge detection, and potentially far more. This suggests the possibility of reconfigurable networks of memristors in which different sections of an array can simultaneously store data, perform Boolean logic, generate spikes, integrate multiple inputs, and perform a variety of machine intelligence-related tasks. The memristor thus becomes a simple two terminal building block for a reconfigurable set of computing architectures.

When we consider memristors as computational elements it is largely as accelerators of mathematical operations such as the dot product operator. These accelerations then lead to the speed up of conventional algorithms further down the line. In this work we take the premise a step further by showing that a unique combination of volatile device behaviors with a potential divider arrangement accelerates not just a single operation but the entire computational problem of edge detection in a single stage.

Before detailing previous memristor based studies, we should acknowledge that the field of low power vision is a well researched field with promising alternatives such as event cameras (Gallego et al., 2019). Event cameras encode images in a different manner than a typical charge-coupled device. Rather than reporting the absolute value of a pixel, an event camera signals when the change in a pixel's value exceeds a threshold. Encoding images in this manner when combined with absolute pixel values, allows for more efficient algorithms one example being the edge detection and tracking of Kueng et al. (2016). The technique can be considered a novel sensor technology requiring further

processing by a central processing unit (CPU) or graphics processing unit (GPU) to handle the unique data produced by the camera.

In contrast, memristor based techniques work with absolute pixel values and are intended as hardware accelerators with the aim of reducing computational overhead. Alone they will struggle to achieve the same performance as event cameras, which exploit the sophisticated processing possible on a CPU/GPU. For example, in the tracking work of Kueng et al. (2016) edge detection is achieved through a combination of both a Harris corner detector and a Canny edge detector. Therefore, memristor techniques should not be considered the end solution but instead as accelerators to be used in conjunction with other systems.

The work presented in this paper follows the latter approach in that it works with absolute pixel values and is intended as a hardware accelerator. However, it differentiates itself from existing techniques in both function and form. Firstly, it does not simply accelerate a single operation, such as a crossbar does for the dot product operation, but instead accelerates the entire process of edge detection. The output requires no additional processing except for the reading of spikes. Secondly, we replace the commonly used crossbar structure, which is the standard in memristive image processing, and instead use a potential divider built from our volatile devices. The combination of volatile devices placed in a potential divider arrangement is unique and has not been used before in the application of image processing. It differs entirely from studies using non-volatile devices in a crossbar (Yakopcic and Taha, 2017; Khokhar and Khalid, 2018; Li et al., 2018), differing in both device behavior and circuit layout. Where potential dividers have been used before in image processing, they have been non-volatile and required the frequent reprogramming of weights. For example, the study that most closely resembles our own approach is the use of memristive threshold logic to detect moving objects (Maan et al., 2015). Although their memristors are also in a potential divider arrangement their use is more complicated. They are non-volatile devices requiring a training phase between frames, in which their conductances are reprogrammed depending on the previous frames values. In contrast, our approach requires no programming nor training phase, instead operating on the fly. This makes for a simpler circuit design.

We will begin by outlining the origin and basis of the existing memristor based techniques and then detail how our approach differs.

In conventional computing there exist a number of algorithms to carry out edge detection, one example being the Sobel algorithm (Duda, 1973). In this the gradient across neighboring pixels is calculated from the scalar dot product of the pixel in question and its surrounding pixels with a predetermined 3×3 matrix, referred to as a kernel. Two different kernels are used, one determining the horizontal gradient, G_x and the other the vertical gradient, G_y.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The results of these two dot products are combined to find the overall gradient using Equation (1).

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{1}$$

As a result, Sobel relies on a number of dot product operations being carried out across the image. However, because dot products involve the passing of data back and forth between processors and memory they lead to bottlenecks and inefficiencies (Fatahalian et al., 2004). Therefore, one approach to improving the efficiency of edge detection is to accelerate the dot product operator.

Circuits designed to accelerate dot product operations are named dot product engines (DPEs). An effective DPE can be implemented using the memristor (Chua, 1971; Strukov et al., 2008). The memristor is a two terminal device similar to a resistor with the exception that its conductance is not fixed. Instead it can be adjusted with an applied voltage. For example, voltages in one polarity may make the device more conductive while voltages in the opposite polarity will make it less conductive. Therefore, a memristor's conductance depends on the potentials applied to it in the past and so can be considered a form of memory.

This memory property is exploited when implementing memristive DPEs. Memristors are arranged into a crossbar array with the value of one matrix element encoded in the conductance of the memristor and the value of the second encoded in the applied voltage (Alibart et al., 2013). The multiplication of matrix elements is carried out by applying the voltage to the memristor producing an output current as defined by Ohm's law. As is required in the dot product operator, the output currents for each element are summed in accordance with Kirchoff's current law. Crossbar arrays consisting of memristors have previously been shown to be effective DPEs (Hu et al., 2016).

The Sobel algorithm was implemented by Can Li et. al. on a memristive DPE with good performance (Li et al., 2018). They made use of non-volatile, analog memristors in a crossbar structure. A variation on directly implementing Sobel on a DPE is to instead teach the same algorithm to a neural network. Memristive crossbar arrays are still used as DPEs. However, the multiplication is of inputs and trained network weights, not of the matrices directly derived from Sobel. Yakopcic et al. constructed a multilayer perceptron network and trained it to replicate the Sobel algorithm, again with good performance (Yakopcic and Taha, 2017). Their system used non-volatile memristors with 128 discrete conductance states. More bespoke implementations depart further from the conventional crossbar or neural network architectures and replace the dot product entirely. Fuzzy XOR gates implemented with memristors can determine pixel gradients (Merrikh-Bayat et al., 2014) and swarm computations, based on the behavior of ants, have been replicated with grids of memristors (Pajouhi and Roy, 2018).

Although these approaches aim to improve efficiency through changes to circuit design, none consider the encoding of their signals, instead choosing to use only continuous real encoding. An alternative is to use spike-encoded signals. Information is represented in either spike timings, spike shape, or both. A spike-based circuit is typically analog and computes at the arrival

of a spike. It is argued that its inactivity between spikes can result in a more power efficient network (Joubert et al., 2012). This has led to the development of neural networks that exploit such spiking signals, called spiking neural networks (SNNs), with a variety of CMOS implementations (Indiveri et al., 2006). Spiking neural networks operate using unsupervised learning rules, one example being the spike-timing-dependent plasticity rule (STDP). Applications such as character recognition via STDP have been demonstrated in both Von-Neumann computer systems (Diehl and Cook, 2015) and memristive systems (Covi et al., 2016). However, computation is not solely restricted to the use of STDP learning rules. Memristors exhibiting generic analog behavior have been used in the learning of spatiotemporal patterns and sound localization (Wang et al., 2018) as well as in the sorting of spike patterns (Werner et al., 2016).

In our application, pixel values are encoded into the frequency of spike trains. Therefore, the detection of edges is equivalent to determining the difference in frequency between neighboring pixels. We present a simple potential divider circuit consisting of two amorphous silicon oxide memristors whose switching characteristics we have detailed previously (Mehonic et al., 2017; Munde et al., 2017; Kenyon et al., 2019). The potential divider design is able to indicate the difference in frequency of its inputs with the amplitude of spikes at its output. By inserting our memristive potential divider between neighboring pixels we can detect differences in pixel values and in turn identify edges.

We will begin by describing the devices used in this work and how their behavior differs from typical memristors. We then go on to detail our circuit and experimental data confirming its behavior. Finally, we describe our model of the circuit and present simulated results for a collection of images.

2. DEVICES

Our devices are of a metal-insulator-metal (MIM) structure with a sputtered silicon oxide insulator layer. They consist of a gold top electrode with a wetting layer of titanium on the oxide and a bottom electrode of molybdenum. The device size is $200 \times 200\ \mu m$. **Figure 1A** details the dimensions of each layer. More details regarding fabrication can be found in Mehonic et al. (2017).

These devices were originally developed as binary memory cells, able to switch between two distinct low and high resistance states (Mehonic et al., 2017) but have also demonstrated a number of neuromorphic uses such as in replicating synapse functionality (Zarudnyi et al., 2018), neuronal spiking and integration (Mehonic and Kenyon, 2016) as well as more conventional machine learning techniques such as interference (Mehonic et al., 2019). A typical application is as elements within random access memories, referred to as resistive random access memory (RRAM). RRAM devices switch between their two distinct resistance states in response to a sufficiently large voltage being applied to the device. Although we have previously shown our devices behave in this manner we do not use this conventional RRAM behavior in this work. Instead, we use an analog operating mode which is obtained through a change in the initial stressing of the device.

2.1. Analog Operation and Current Transients

It is well known the behavior of a memristor is defined by its device history. One of the key stages in this history is the initial stressing of the device, in which the device transitions from a pristine to an operational state. This initial stressing is typically carried out using a voltage sweep and is referred to as electroforming. After electroforming, the device exhibits binary switching behavior as we have shown in Mehonic et al. (2017). However, by modifying this initial stressing we find the device can be forced into an alternative operating mode, one that does not exhibit discrete jumps in resistance but instead analog and volatile changes. A characteristic feature of this operating mode is the observed transient in current in response to constant potentials, shown in **Figure 1B**. Therefore, we have devices able to exhibit either digital or analog behaviors depending on how the device is initially stressed.

In order to induce the analog operating mode the device is not electroformed with a voltage sweep but instead has a constant current driven through the device at the top electrode. The magnitude of the stressing current can range from -10 to -100μA and should be maintained until the change in device conductance slows and levels out. An example of this forming process is included as an inset to **Figure 1B**. It should be stressed this is not an operating condition but an initial step in order to induce the analog regime and so could be considered a kind of electroforming. However, it should not be confused with the electroforming typically associated with memristors. This process is a smooth transition, very different from the discrete jumps of electroforming. After removing the current bias and allowing the device to relax, it is now in the analog regime and will exhibit the characteristic current transients.

Transients similar to those in **Figure 1B** have been documented before in other MIM structures. Studies of barium strontium titanate capacitors (Zafar et al., 1998; Saha and Krupanidhi, 2001) revealed transients of the same form and timescale. Although it may seem reasonable to believe transients are the result of capacitive charging this theory fails in two ways. Firstly, the change in conductance is in the opposite sense to what would be expected. If we were simply charging the electrode we would expect a decrease in current, instead we observe an initial increase. Secondly, the latter half of the transient, which is a decrease in current, occurs over a duration of 50s. Considering our device is driven by a low impedance voltage source, we would expect significantly shorter timescales. The capacitance of the device can be approximated to be 35 to 40 pF; assuming a relative dielectric constant between 3.5 to 4, an area of $4 \times 10^{-4}(\text{cm}^2)$ and a thickness of 35 nm. If we then assume the combined source and lead resistances were to be at the extreme end, say 100 Ω, the time constant of the system would range from 3.5 to 4 ns, many orders of magnitude smaller than what is observed. Instead, the cause of this behavior is thought to be drifting oxygen vacancies in turn modulating electronic conduction (Meyer et al., 2005; Zhong et al., 2010).

When driven with a negative bias at the top electrode, current transients consist of two parts; an initial increase in conductance followed by a later decrease. The increase in conductance is

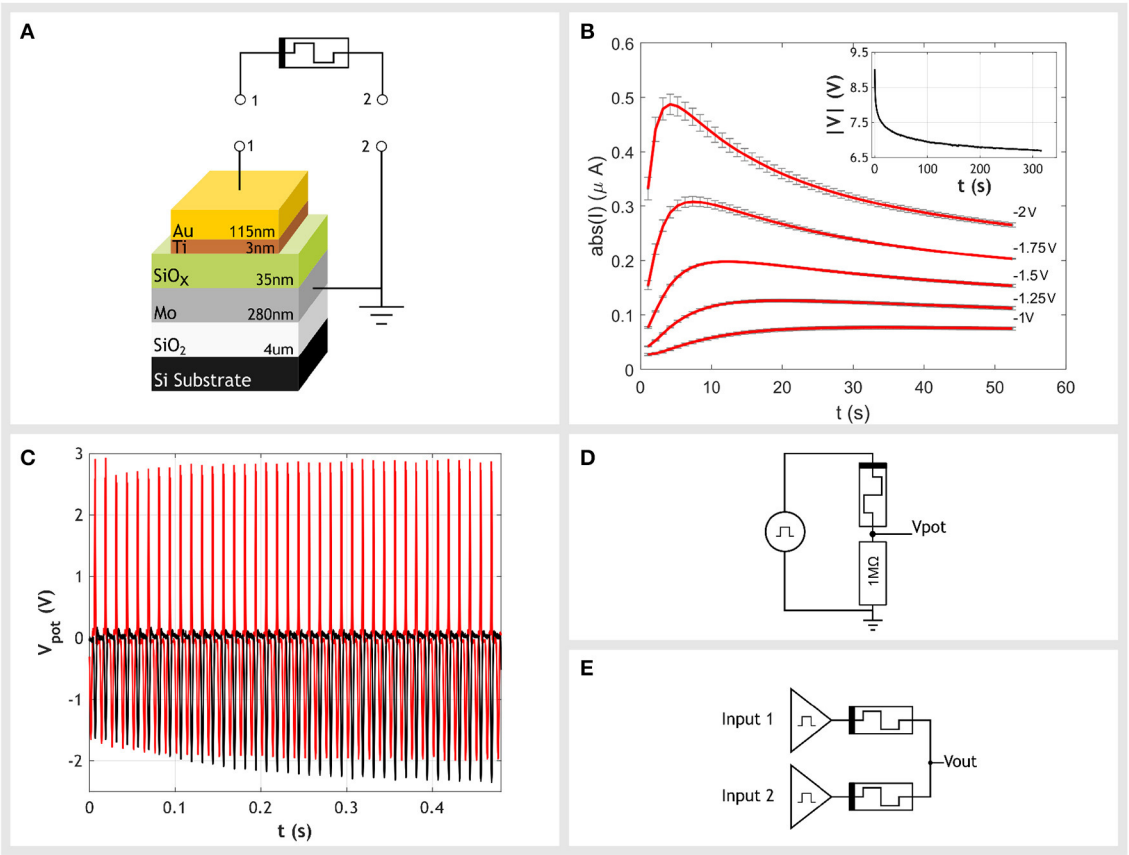


FIGURE 1 | (A) Our devices use an active layer of 35 nm sputtered amorphous silicon oxide. The bottom contact is a 280 nm layer of molybdenum and the top contact is a 115 nm layer of gold with a 3 nm wetting layer of titanium. **(B)** Examples of the current transients which occur when constant negative voltages are applied to the top electrode with respect to the bottom electrode. Transients consist of two parts. There is an initial increase in conductance and a subsequent decrease. In this work we operate only within the first region, the increase in conductance. Inset is a plot of the absolute voltage across the device during the initial stressing stage. A constant current of -10μA is driven through the device. The applied voltage decreases over time, indicating the reduction in device resistance that occurs as a result. **(C)** Negative and positive voltages have an opposite effect on the device's conductance. When a train of negative voltage spikes were applied to the device in a potential divider setup with a fixed $1M\Omega$ resistor, the voltage of spikes at the output increases over time (black trace), corresponding to an increase in conductance of the memristor. In contrast, when a train of positive spikes are interleaved in anti-phase with the negative spikes (red trace), the output voltage increases to a lesser extent. This demonstrates the competing effects positive voltages have on the memristor. The positive spikes are reversing the changes in conductance cause by the negative spikes. **(D)** Setup to demonstrate the competing effects of negative and positive polarities. Gaussian pulses with a full width at half maximum (FWHM) of 1.3 ms are generated by a signal generator. These are applied to the top contact of the memristor which is in a potential divider with a fixed $1M\Omega$ resistor. The output voltage, V_{pot}, is measured at the output of the potential divider. **(E)** Our circuit that determines the difference in frequency of two input spike trains. Both inputs generate pulse trains with a negative amplitude and a frequency proportional to their input value. Each input is connected to a single memristor. Both memristors then join at a common node. The output of the circuit, V_{out}, is taken at this common node. The amplitude of output spikes indicate the difference between the two input frequencies. Larger differences in frequency result in larger amplitudes at the output.

volatile and resets on the order of seconds, whereas the decrease in conductance is more persistent. In this work, we operate within only the first portion of the transient. Within this region, applying a negative bias to the top electrode creates an increase in the device conductance which can then be reversed with subsequent positive biases. This behavior is demonstrated in **Figure 1C**. A single device was placed in series with a fixed 1 MΩ resistor connected to ground, forming a potential divider - as shown in **Figure 1D**. Any change in device conductance is observed in the change of the potential divider's output voltage. An increase in device conductance will result in larger output amplitudes while a decrease in conductance will result in smaller output amplitudes. When a train of pulses with negative amplitudes is applied to the device (black trace) we observe an increase in the amplitude of output spikes over time, corresponding to an increase in device conductance. In contrast, when a positive pulse train is interleaved in anti-phase with the original negative train (red trace) the two processes begin to compete. The negative pulses increase conductivity while the positive pulses decrease it. Although this still leads to a small increase in conductivity, which we assume is due to some asymmetries, it is less than when the negative train does not face competition. It is this competing behavior between spike trains of opposite polarities that forms the basis of our circuit.

3. CIRCUIT DESIGN

To determine the gradient across two neighboring pixels we require a circuit that detects the difference in frequency between two spike trains. Our circuit achieves this by exploiting the opposing effects spike trains of opposite polarities have on our devices.

Both input sources are connected to each other through a combination of two memristors in series, as shown in **Figure 1E**. The memristors are in opposite orientations with their bottom contacts connected, forming a potential divider. The amplitudes of output spikes are therefore dependent on the conductances of both memristors. Sources produce spikes of negative polarity and are connected to the top contact of their respective memristor. For either memristor, when the source directly connected to it generates a spike and the neighboring source is grounded, it experiences a negative bias, causing an increase in conductance. In direct contrast, the second memristor, whose source is grounded, is in the opposite orientation and so experiences a bias in the opposite polarity. This causes its conductance to decrease. We have therefore introduced a form of competition between the two inputs. When a source fires it acts to increase the conductance of its attached memristor while decreasing the conductance of its neighbor.

If the two inputs are of the same frequency any increase in conductance caused by one input is swiftly canceled out by the opposing effect of the other. This will result in both memristors having a similar conductance with no change in output amplitude. Alternatively, when one input has a higher frequency than the other, the high frequency input will overpower the opposing effect of the second input. This will drive the memristor with the high frequency input to become more conductive while suppressing increases in the conductance of the low frequency input. Given the potential divider arrangement, the amplitude of output spikes for the high frequency input will increase while those of the low frequency spikes are driven down to a minimum value. This behavior is shown in **Figure 2**. The inputs are initially both set to 50 Hz with no observed change in the amplitude of output spikes. The frequency of one input is then set to 100 Hz, resulting in a difference in frequency of 50 Hz. This causes the amplitude of the output spikes caused by the 100 Hz input to increase in amplitude, whereas the output spikes generated by the 50 Hz signal remain at their initial value. Therefore, as the difference in frequency between the two inputs increases, so does the amplitude of output spikes from the higher frequency input. These amplitudes can be used as an indicator for the difference in input frequencies.

Importantly, at no point should the two inputs be allowed to fire at the same time. If this occurs the circuit would no longer behave as a potential divider due to neither of the inputs being grounded. As a result the output voltage will merely follow the voltage of both inputs producing an erroneous output. In this work we avoid conflicts by inhibiting the latter spike when two spikes do happen to overlap. We chose this approach because it was considered the simplest to implement in a physical system. Each spike source would be designed with an enable/disable input, which, when triggered, inhibits any output. The output of a source would then connect to its neighbor's enable/disable input. Thus, when a source fires and produces a spike, it is simultaneously inhibiting its neighbor from firing. Crucially, this implementation uses only local signals, avoiding issues with the routing of control signals.

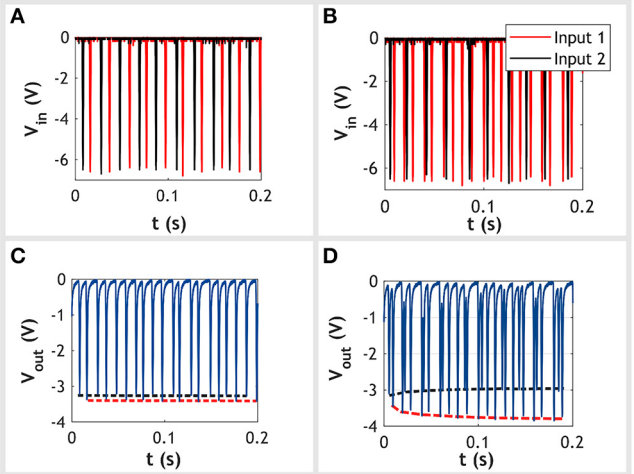


FIGURE 2 | Experimental data demonstrating the circuit's ability to detect differences in frequency between two input spike trains. Two scenarios are presented: the first with no difference in input frequency and the second with a difference of 50 Hz. Spikes are Gaussian shaped with a full width at half maximum (FWHM) of 1.3 ms. Each Gaussian pulse has been cropped to a width 2 ms. **(A,C)** These show the input and output signals, respectively, for the case of no difference in input frequency. Both input 1 (red trace) and input 2 (black trace) are set to 50 Hz. In the plot beneath we see the amplitude of spikes at the output remain approximately constant for both inputs. For clarity we have included an envelope tracking the output spikes caused by input 1 (dotted red trace) and input 2 (dotted black trace). **(B,D)** Shows the input and output signals, respectively, for the case with a 50 Hz difference in input frequency between the two inputs. Input 1 is set to 100 Hz while input 2 remains at 50 Hz. For clarity, we have again overlaid two envelopes tracking the output spikes caused by input 1 and input 2. This time, we observe at the output that spikes caused by input 1 increase in amplitude over time, whereas those from input 2 remain constant.

Edges manifest as sharp changes in pixel values across the image, equivalent to large differences between the frequencies of neighboring pixels. By connecting our circuit between two neighboring pixels, as shown in **Figure 3A**, we detect these differences and produce output spikes with amplitudes proportional to the differences in frequency. Large amplitude output spikes correspond to sharp changes in pixel values, indicating potential edges.

4. METHODOLOGY

4.1. Simulation

In order to simulate the circuit's performance when applied to an image, we require a model approximating its behavior. We constructed a look-up table describing the circuit's response. Given two input frequencies, the look-up table returns the average amplitude of spikes at the circuit's output. These

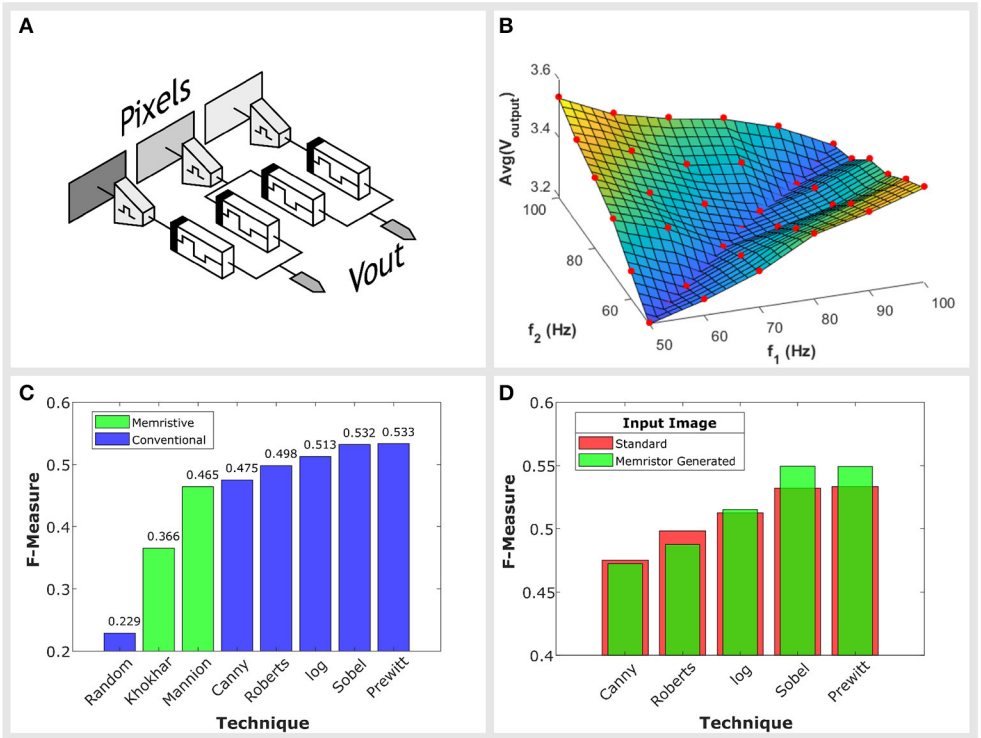


FIGURE 3 | (A) An illustration of how edge detection would be implemented. The circuit is placed between two neighboring pixels. Large differences in pixel values will produce output spikes with larger amplitudes. **(B)** The look-up map describing our circuit's behavior. The average amplitude of output spikes above a threshold is plotted along the *z* axis. We use this look-up table during simulations. It approximates the circuit's output for any given pair of input frequencies. The sampling points from which this map was interpolated from are illustrated with red circles. **(C)** Benchmarking results on the BSDS500 dataset. The distribution of F-Measures, defined in Arbeláez et al. (2011) are plotted for memristive techniques (green) and standard operators (blue). The results are obtained from the set of 200 test images provided by BSDS500. **(D)** Comparison of F-Measure scores for a set of operators using both the original test images and images produced by our own circuit as their input. An improvement in performance is observed over the Prewitt, Sobel, and log operators.

measurements were taken after the circuit was allowed to settle, always $\leq$ 500 ms after inputs were first applied. Spikes take the same form as those used in **Figure 2**, they are Gaussian in shape with a full width at half maximum (FWHM) of 1.5 ms and are trimmed to a width of 2 ms. The model was constructed using data obtained from a physical implementation of the circuit with input frequencies ranging from 50 to 100 Hz. We characterize the circuit with a sampling resolution of 10 Hz. Each of the points sampled are illustrated with a red circle in **Figure 3B** and are used to form the dataset for interpolation. The look-up table produced as a result of this process is shown in **Figure 3B**.

Input frequencies are generated from images such as those presented in **Figure 4A**. Pixel values are first converted from color to grayscale using MATLAB's `rgb2gray` function. The function uses the following equation to combine the three components: $0.2989R + 0.5870G + 0.1140B$ where R, G, and B correspond to the red, green and blue components of the pixel. The grayscale values are then linearly mapped from 0 to 255 to frequencies between 50 and 100 Hz. Example outputs for each simulation are shown in **Figure 4B**. In this figure we have combined the results from both a horizontal and vertical edge detection. Each pixel represents a single circuit placed between two neighboring pixels. The value of the pixel is proportional to the average amplitude of output spikes that are above a defined threshold - the same quantity as that returned from the look-up table. Brighter pixels correspond to larger output amplitudes, which are caused by larger differences in input frequencies and therefore indicate potential edges.

4.2. Benchmarking

Benchmarking is a useful tool in comparing solutions to a computational problem. Of the previous memristive edge detection studies, only one makes use of benchmarking (Khokhar and Khalid, 2018), with the BSDS500 dataset (Arbeláez et al., 2011). The BSDS dataset provides 500 images for testing edge/boundary detection algorithms combined with a benchmarking script to standardize comparisons between algorithms. Although the authors make use of the dataset, they do not use the associated benchmarking script. Instead using their own custom analysis. We make use of both the dataset as well as its benchmarking script in the hope that future studies can compare effectively against our work. The authors of Khokhar and Khalid (2018) have kindly made their data available for us to put through the standard benchmark as a comparison.

The performance of each memristor implementation is compared against a number of standard operators: Prewitt, Sobel,

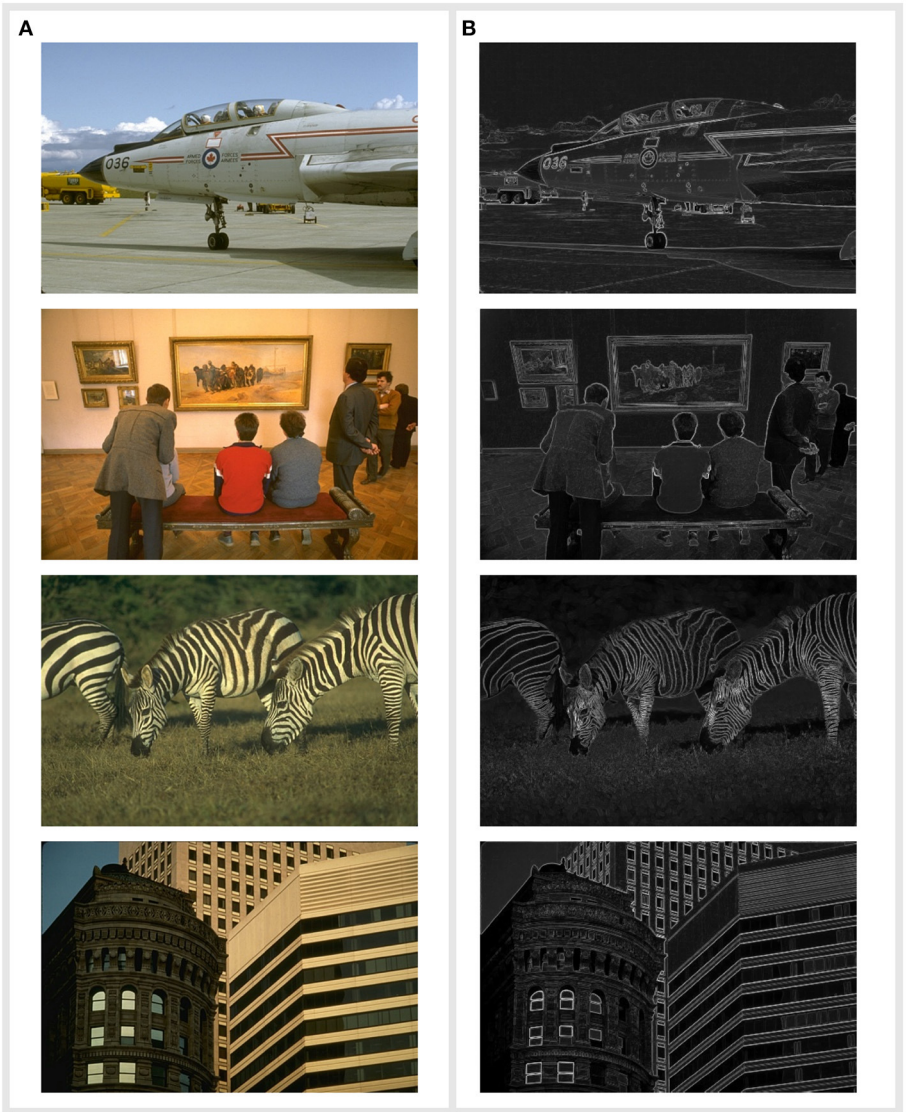


FIGURE 4 | (A) A sample of the original input images presented to the circuit. Source: Arbeláez et al. (2011). In simulation, there exists an edge detection circuit between each neighboring pixel. Pixels are mapped from their 0–255 value to a frequency range between 50–100 Hz. **(B)** The corresponding output images of the simulation. Each pixel represents the output of an edge detection circuit placed between two neighboring pixels. The average amplitude of output spikes above a threshold is mapped from the voltage to a pixel value from 0 to 255 and is plotted in this image. Brighter pixels indicate edges. We have combined the simulations of edge detection in both the vertical and horizontal plane.

log, Roberts and Canny. Each technique is awarded a score, named the F-Measure. This score is related to the probability of a pixel being an edge and the probability of a false positive with more details on its derivation given in Arbeláez et al. (2011). The larger the F-Measure the more effective the edge detection. In **Figure 3C** we have plotted the F-Measure scores for each technique, in addition to a random approach which merely classifies pixels as edges with a 50% probability.

In addition to this, we also characterize the use of our circuit as an input to standard edge detection operators, quantifying whether or not it improves performance. For each operator, we begin by running the benchmarking script using the original dataset images as inputs to form a set of control data. We then run a second test but instead now use the output image generated by our circuit as the input image to the conventional operator. The performances of these two cases are then compared to identify any improvements in performance.

5. RESULTS AND DISCUSSION

5.1. Performance

Examples of the circuit's output are presented in **Figure 4B**. Unfortunately, it is not possible to quantitatively compare our circuit against the techniques of Li et al. (2018) and Yakopcic

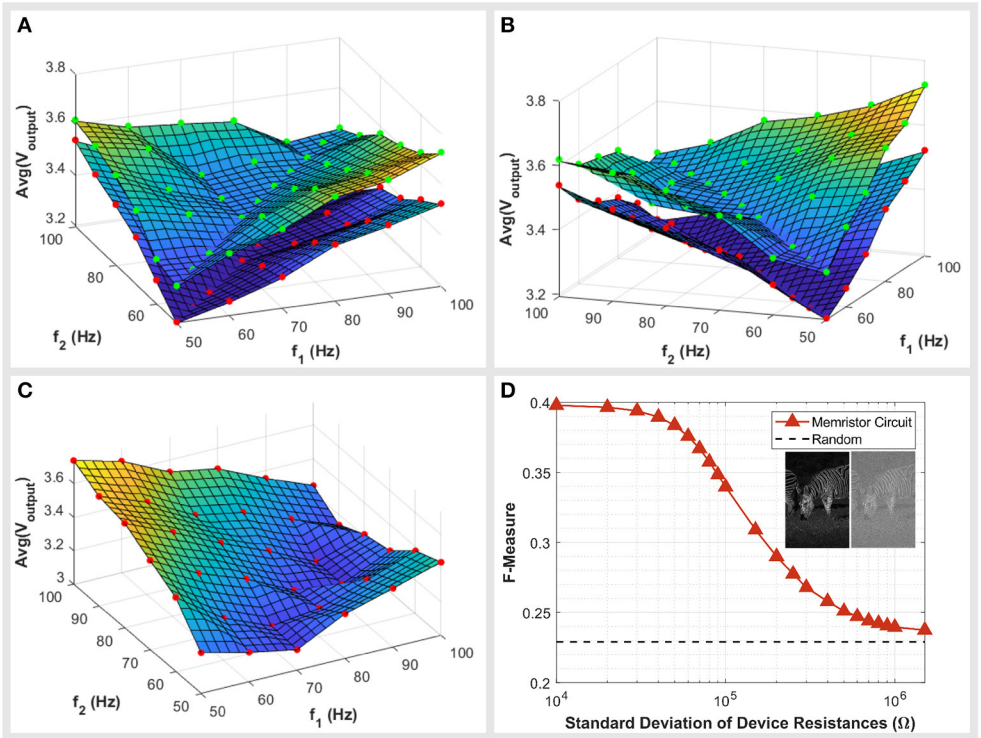


FIGURE 5 | (A,B) The circuit response plotted in two different perspectives for clarity. The response for two circuits are presented in both figures, a potential divider of two high resistance devices (red markers) and a potential divider of two low resistance devices (green markers). Markers represent the inputs sampled for the circuit. **(C)** The circuit response when devices have asymmetric resistances. The red markers indicate the points sampled from the circuit. **(D)** The effect of device variance on circuit performance. The circuit's benchmark score (F-Measure) is plotted for a range of simulations where variations in device resistances were introduced. Device variations were distributed randomly and with a Gaussian distribution. The standard deviation of device resistances were increased with the effect of a reduction in performance. For reference the score of a system randomly categorizing pixels as edges is also plotted.

and Taha (2017). Both studies use different images and in the case of Li et al. (2018), their input has purposefully been made to exhibit noise. However, in using the BSDS500 dataset we are able to compare the circuit's performance against other conventional operators as well the memristor based work of Khokhar and Khalid (2018) who also make use of this dataset.

Figure 3C shows our circuit's benchmark performance against other techniques, where we achieve an F-Measure of 0.465. This places our performance at the bottom end of conventional operators, on par with the Canny operator. However, in comparison to other memristive techniques, such as Khokhar and Khalid (2018) who achieve an F-measure of 0.366, we present a jump in performance.

An alternative approach would to consider our circuit an accelerator, for example, as the input to one of the standard operators. This technique leads to an improvement in performance for the Prewitt, Sobel and log operators as shown in **Figure 3D**. On the other hand, algorithms that score lower F-Measures on the BSDS500 dataset, such as the Roberts and Canny detectors, do not improve through the use of our circuit as an input.

5.2. Variance

Variability in device performance is a common issue with memristive devices. In this work, we are primarily concerned with the variance in device resistances leading to offsets in voltage at the potential divider's output and in turn define a maximum tolerable variability.

Our device resistances varied from 0.77 to 2.17 $M\Omega$ when sampled across 16 devices, with 50% of devices falling within the resistance range of 0.99–1.66 $M\Omega$. We found these variations were spatially distributed, neighboring devices would have similar resistances while those separated were likely to vary.

We consider two scenarios when assessing the effect of variance on circuit performance. The first is when both devices of the potential divider have a similar resistance. The second is when the two devices have different resistances. When the two devices making up the potential divider have similar resistances, we find little differences in the circuit behavior other than an offset in spike amplitudes. **Figure 5A** shows the circuit's response for two instances, a pair of 1.24 and 1.32 $M\Omega$ devices and pair of 2.17 and 2.10 $M\Omega$ devices. **Figure 5B** shows the same plot from a different perspective for clarity. The shape of the circuit's response does not vary significantly between each case,

whereas there is a noticeable offset. Alternatively, when pairs of devices are not equal and instead have asymmetric resistances the circuit has an asymmetric response, as shown in **Figure 5C**. Fortunately, the asymmetry of the circuit's response does not have a significant impact on the circuit performance, with benchmark scores dropping from an F-Measure of 0.474 to 0.459 when we account for such effects.

On the other hand, symmetric variances affect the circuit in a more significant way. The voltage offsets caused by such variances interfere with the classification of edges. This is because a pixel is deemed an edge if the output voltage is above a threshold, however, voltage offsets cause a blurring of this threshold. Some non-edges are raised above the threshold and some edge pixels are dragged beneath the threshold, introducing errors in the output image. Such offsets can be caused by additive thermal noise, investigated in the **Supplementary Material**, or by device variances which we will now detail.

To quantify the effect of variances on the circuit performance, we simulate the circuit with varying device resistances. Devices exhibit a gaussian distribution of resistances allocated randomly across the image. These variances lead to voltage offsets at the outputs of each circuit, an example of which is included as an inset of **Figure 5D**. We simulated and benchmarked the system for a range of distributions with varying standard deviations. This resulted in a drop of performance as shown in **Figure 5D**. The system performs no better than the random control beyond approximately a standard deviation of 250 kΩ. Through this we can define a maximum acceptable standard deviation by using the score of other memristive studies as a threshold. Taking Khokhar and Khalid (2018) as the threshold with a score of 0.366, we can determine a maximum allowable standard deviation of 50 kΩ. The standard deviation of our current devices can be approximated to be 472 kΩ, although this should be treated with caution seeing as we have only 16 samples to characterize hence the approach is not statistically significant. At this stage, it appears the variance in our devices is too large for the system to be realized. Although we cannot comment on the specific cause of such variances, if this were the result of sample fabrication, then it is a matter of refining fabrication processes. However, a more detailed study would have to be carried out to identify the causes of such variances and the ultimate limitations.

5.3. Scalability

Our chosen figure of merit to compare the scalability of techniques considers the number of components required for each additional raster/pixel added to the circuit. This includes both the number of additional memristors as well as any periphery circuitry included at output or intermediate layers. This quantifier allows for a quick comparison regardless of whether a scanned or parallel approach is taken. We do not consider the input circuitry. **Table 1** compiles the component count per raster for each of the studies cited in this paper. Although data could be collected for most techniques it was not possible to fully assess (Khokhar and Khalid, 2018). We approximate their output to require a single comparator per raster in order to threshold outputs as stated in their paper. However, they also require peripheral circuitry to regularly update memristor weights, which they do not document. This will act to increase both the component count and circuit footprint, hence, we can say with some confidence it is a more complex circuit than the others presented here.

TABLE 1 | Comparison of the increase in component count required for each additional implemented raster.

Study	Memristors per raster/Pixel	Operational amplifiers per raster/Pixel	Size of raster
Mannion et al. (this study)	+2	+1 (Comparator)	1 × 2
Li et al. (2018)	+25	+1 (Transconductance)	5 × 5
Yakopcic and Taha (2017)	+461	+40 (20 Summing, 20 Unity Gain)	3 × 3
Khokhar and Khalid (2018)	+225	Not available	1 × 2

Of the remaining two techniques by Li et al. (2018) and Yakopcic and Taha (2017), the approach of Li et al. is by far superior with respect to component count. This is not surprising considering their approach features a single crossbar array with one transimpedance amplifier per column whereas Yakopcic's neural network consists of 10 input neurons, 20 hidden layer neurons and 2 memristors for each synapse connection to represent +/- weights. Equally, Yakopcic's hidden neurons consist of two operational amplifiers, further increasing circuit size.

When comparing our circuit to these three techniques we must consider how the system will be implemented. Either a scanned approach may be taken, whereby only a single kernel is physically implemented and then scanned across the image, or a parallel approach is taken, with multiple copies of the circuit operating in parallel, each on their respective section of the image. The scanned approach favors scenarios where latency is less of a concern and small footprints are desired, whereas the parallel approach suits scenarios requiring the real time processing of images. Our circuit requires approximately a twelfth of the memristors required by Can Li yet the same number of output operational amplifiers, albeit in a different configuration. That said, our circuit should not be considered the better technique solely on this basis. As detailed in the following section on limitations, our circuit has a finite settling time due to the memory properties of our devices. Therefore, if a scanning implementation is being used we require a finite relaxation time between the presentation of inputs to avoid mixing. As a result, a scanning approach would favor a crossbar technique such as Li et al. (2018) whereas our circuit is better suited to a parallel implementation.

5.4. Limitations

Although our circuit provides the advantage of a potentially reduced component count, we identify some limitations. The first applies to any technique determining the gradient across neighboring pixels and it concerns the resolution of the image. For high resolution images, a sharp edge may occur across a number of pixels. The change in intensity associated with the edge is now spread across the group of pixels, thereby

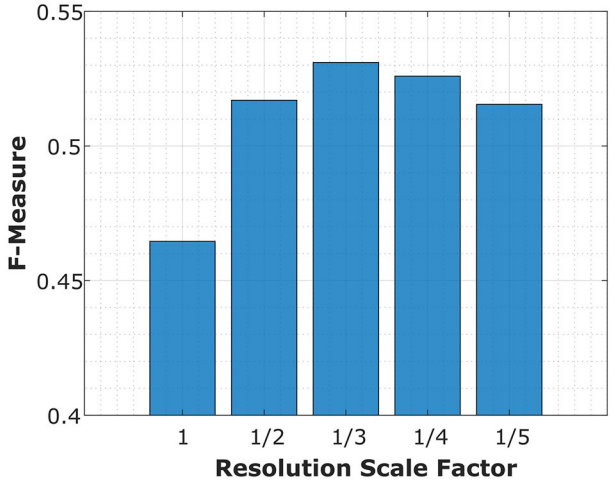


FIGURE 6 | The effect of image resolution on circuit performance. The benchmarking score (F-Measure) is plotted against the scale factor of the image resolution. An improvement in performance is observed for lower resolution images. For example, the highest score occurred when the images were at a 1/3 of their original resolution. It was not possible to investigate the effect of increasing image resolution as there were no high resolution images available for the dataset.

reducing the change experienced by each individual pair of pixels, essentially smoothing out the edge. The higher the resolution, the worse this effect will be. A simple solution is to down-sample the image. However, although this will help in some cases, the image resolution will always play a role in limiting which edges can be detected. We investigate this limitation by reducing the resolution of the benchmark images, processing the image and then scaling the image back to its original resolution for benchmarking. The scores of the circuit for different resolutions are plotted in **Figure 6**. When the image is reduced in resolution and then applied to the circuit the performance is generally better than at the benchmark's original resolution. However, if the resolution is reduced beyond a third of its original, the performance drops owing to the loss in information that is occurring.

The second limitation is specific to our approach. Once inputs are applied to the circuit, the output has a finite settling time after which it is then stable. The output should not be read before this time to avoid incorrect readings. This limits the operating frequency of the circuit. The configuration used in this work has a settling time consistently $\leq$ 500 ms. This time can be adjusted through changes in a number of parameters including the amplitudes or widths of input spikes and the chosen operating frequencies.

6. CONCLUSION

We have shown how a potential divider of two memristors can indicate the difference in frequencies between two spike trains. We confirmed this behavior experimentally and applied the circuit to the problem of edge detection successfully achieving a jump in performance compared to other memristive techniques. The circuit requires no external control signals, training signals or power supply, instead operating exclusively on input signals. This proves an advantage for scalability. Without the need for these external signals, as required with DPEs or neural networks, we have reduced the complexity of routing signal paths and computational overhead. Equally, its passive nature combined with spike operation makes it well suited for low power applications. Besides edge detection, the circuit may also have broader applications. Its fundamental behavior is the detection of differences in frequency between two input spike trains. This may prove useful in other computational schemes.

Furthermore, in showing an alternative operating region devoid of switching has computational uses, we have demonstrated yet another function resistance switching devices can provide. With the very same devices able to implement arrays of memory and both analog and discrete computations, we envisage reconfigurable networks of these devices having real potential in delivering flexible hardware accelerators.

DATA AVAILABILITY STATEMENT

The datasets generated for this study can be found in UCL's data repository: doi: 10.5522/04/9741722.

AUTHOR CONTRIBUTIONS

DM carried out the experimental work, built simulations and prepared the manuscript. AM devised experiments and prepared the manuscript. WN fabricated the devices used in the characterization of our circuit. AK supervised the project and prepared the manuscript.

FUNDING

DM and AK acknowledge the funding provided by EPSRC for the studentship enabling the work. This studentship is part of the Doctoral Training Partnership, Grant Code: EP/N509577/1, with the project reference: 1917855. AM acknowledges the funding provided by the Royal Academy of Engineering under the Research Fellowship Scheme.

ACKNOWLEDGMENTS

We would like to acknowledge and thank Dr. Salman Khokhar from the University of Central Florida for providing output images generated by their memrsitive technique.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2019.01386/full#supplementary-material

REFERENCES

Alibart, F., Zamanidoost, E., and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training. *Nat. Commun.* 4:2072. doi: 10.1038/ncomms3072

Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 898–916. doi: 10.1109/TPAMI.2010.161

Chua, L. (1971). Memristor-The missing circuit element. *IEEE Trans. Circ. Theory* 18, 507–519.

Covi, E., Brivio, S., Serb, A., Prodromakis, T., Fanciulli, M., and Spiga, S. (2016). Analog memristive synapse in spiking networks implementing unsupervised learning. *Front. Neurosci.* 10:482. doi: 10.3389/fnins.2016.00482

Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Duda, R. O. (1973). "Chapter 7.3: Spatial differentiation," in *Pattern Classification and Scene Analysis*, eds R. O. Duda and P. E. Hart (Chichester; New York, NY: Wiley-Interscience), 267–272.

Fatahalian, K., Sugerman, J., and Hanrahan, P. (2004). "Understanding the efficiency of GPU algorithms for matrix-matrix multiplication," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics hardware - HWWS '04*, (New York, NY: ACM Press), 133.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., et al. (2019). Event-based vision: a survey. *CoRR* abs/1904.08405.

Hu, M., Williams, R. S., Strachan, J. P., Li, Z., Grafals, E. M., Davila, N., et al. (2016). "Dot-product engine for neuromorphic computing," in *Proceedings of the 53rd Annual Design Automation Conference on - DAC '16* (New York, NY: ACM Press), 1–6.

Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.* 17, 211–221. doi: 10.1109/TNN.2005.860850

Joubert, A., Belhadj, B., Temam, O., and Heliot, R. (2012). "Hardware spiking neurons design: analog or digital?" in *The 2012 International Joint Conference on Neural Networks (IJCNN)* (Brisbane, QLD: IEEE), 1–5. doi: 10.1109/IJCNN.2012.6252600

Kenyon, A. J., Singh Munde, M., Ng, W. H., Buckwell, M., Joksas, D., and Mehonic, A. (2019). The interplay between structure and function in redox-based resistance switching. *Faraday Discuss.* 213, 151–163. doi: 10.1039/c8fd00118a

Khokhar, S., and Khalid, A. (2018). "Nanoscale memristive crossbar circuits for approximate edge detection in smart cameras," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (Vancouver, BC), 749–754.

Kueng, B., Mueggler, E., Gallego, G., and Scaramuzza, D. (2016). "Low-latency visual odometry using event-based feature tracks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 16–23.

Li, C., Hu, M., Li, Y., Jiang, H., Ge, N., Montgomery, E., et al. (2018). Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* 1, 52–59. doi: 10.1038/s41928-017-0002-z

Maan, A. K., Kumar, D. S., Sugathan, S., and James, A. P. (2015). Memristive threshold logic circuit design of fast moving object detection. *IEEE Trans. Very Large Scale Integrat. Syst.* 23, 2337–2341. doi: 10.1109/TVLSI.2014.2359801

Mehonic, A., Joksas, D., Ng, W. H., Buckwell, M., and Kenyon, A. J. (2019). Simulation of inference accuracy using realistic rram devices. *Front. Neurosci.* 13:593. doi: 10.3389/fnins.2019.00593

Mehonic, A., and Kenyon, A. J. (2016). Emulating the electrical activity of the neuron using a silicon oxide rram cell. *Front. Neurosci.* 10:57. doi: 10.3389/fnins.2016.00057

Mehonic, A., Munde, M., Ng, W., Buckwell, M., Montesi, L., Bosman, M., et al. (2017). Intrinsic resistance switching in amorphous silicon oxide for high performance siox reram devices. *Microelect. Eng.* 178, 98–103. doi: 10.1016/j.mee.2017.04.033

Merrikh-Bayat, F., Bagheri Shouraki, S., and Merrikh-Bayat, F. (2014). Memristive fuzzy edge detector. *J. Real Time Image Process.* 9, 479–489. doi: 10.1007/s11554-012-0254-9

Meyer, R., Liedtke, R., and Waser, R. (2005). Oxygen vacancy migration and time-dependent leakage current behavior of ba0.3sr0.7tio3 thin films. *Appl. Phys. Lett.* 86:112904. doi: 10.1063/1.1874313

Munde, M. S., Mehonic, A., Ng, W. H., Buckwell, M., Montesi, L., Bosman, M., et al. (2017). Intrinsic resistance switching in amorphous silicon suboxides: the role of columnar microstructure. *Sci. Rep.* 7:9274. doi: 10.1038/s41598-017-09565-8

Pajouhi, Z., and Roy, K. (2018). "Image edge detection based on swarm intelligence using memristive networks," *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* 37, 1774–1787. doi: 10.1109/TCAD.2017.2775227

Saha, S., and Krupanidhi, S. B. (2001). Transient analysis in al-doped barium strontium titanate thin films grown by pulsed laser deposition. *J. Appl. Phys.* 90, 1250–1254. doi: 10.1063/1.1384493

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

Wang, W., Pedretti, G., Milo, V., Carboni, R., Calderoni, A., Ramaswamy, N., et al. (2018). Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses. *Sci. Adv.* 4:eaat4752. doi: 10.1126/sciadv.aat4752

Werner, T., Vianello, E., Bichler, O., Garbin, D., Cattaert, D., Yvert, B., et al. (2016). Spiking neural networks based on oxram synapses for real-time unsupervised spike sorting. *Front. Neurosci.* 10:474. doi: 10.3389/fnins.2016.00474

Yakopcic, C., and Taha, T. M. (2017). "Memristor crossbar based implementation of a multilayer perceptron," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)* (Dayton, OH: IEEE), 38–43. doi: 10.1109/NAECON.2017.8268742

Zafar, S., Jones, R. E., Jiang, B., White, B., Chu, P., Taylor, D., et al. (1998). Oxygen vacancy mobility determined from current measurements in thin ba0.5sr0.5tio3 films. *Appl. Phys. Lett.* 73, 175–177.

Zarudnyi, K., Mehonic, A., Montesi, L., Buckwell, M., Hudziak, S., and Kenyon, A. J. (2018). Spike-timing dependent plasticity in unipolar silicon oxide rram devices. *Front. Neurosci.* 12:57. doi: 10.3389/fnins.2018.00057

Zhong, N., Shima, H., and Akinaga, H. (2010). Transient current study on pt/TiO2-x/pt capacitor. *Japan. J. Appl. Phys.* 49:04DJ15. doi: 10.1143/JJAP.49.04DJ15

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Mannion, Mehonic, Ng and Kenyon. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 22 January 2020
doi: 10.3389/fnins.2019.01429

LAO-NCS: Laser Assisted Spin Torque Nano Oscillator-Based Neuromorphic Computing System

Hooman Farkhani[1], Tim Böhnert[2], Mohammad Tarequzzaman[2], José Diogo Costa[2], Alex Jenkins[2], Ricardo Ferreira[2], Jens Kargaard Madsen[1] and Farshad Moradi[1]*

[1] Integrated Circuits and Electronics Laboratory, Department of Engineering, Aarhus University, Aarhus, Denmark,
[2] International Iberian Nanotechnology Laboratory, Braga, Portugal

Dealing with big data, especially the videos and images, is the biggest challenge of existing Von-Neumann machines while the human brain, benefiting from its massive parallel structure, is capable of processing the images and videos in a fraction of second. The most promising solution, which has been recently researched widely, is brain-inspired computers, so-called neuromorphic computing systems (NCS). The NCS overcomes the limitation of the word-at-a-time thinking of conventional computers benefiting from massive parallelism for data processing, similar to the brain. Recently, spintronic-based NCSs have shown the potential of implementation of low-power high-density NCSs, where neurons are implemented using magnetic tunnel junctions (MTJs) or spin torque nano-oscillators (STNOs) and memristors are used to mimic synaptic functionality. Although using STNOs as neuron requires lower energy in comparison to the MTJs, still there is a huge gap between the power consumption of spintronic-based NCSs and the brain due to high bias current needed for starting the oscillation with a detectable output power. In this manuscript, we propose a spintronic-based NCS (196 × 10) proof-of-concept where the power consumption of the NCS is reduced by assisting the STNO oscillation through a microwatt nanosecond laser pulse. The experimental results show the power consumption of the STNOs in the designed NCS is reduced by 55.3% by heating up the STNOs to 100°C. Moreover, the average power consumption of spintronic layer (STNOs and memristor array) is decreased by 54.9% at 100°C compared with room temperature. The total power consumption of the proposed laser assisted STNO-based NCS (LAO-NCS) at 100°C is improved by 40% in comparison to a typical STNO-based NCS at room temperature. Finally, the energy consumption of the LAO-NCA at 100°C is expected to reduce by 86% compared with a typical STNO-based NCS at the room temperature.

Keywords: neuromorphic computing system, laser, power efficient, COMSOL multiphysics, spin torque nano-oscillators

OPEN ACCESS

Edited by:
Hadi Heidari,
University of Glasgow,
United Kingdom

Reviewed by:
Siming Zuo,
University of Glasgow,
United Kingdom
Mostafa Rahimi Azghadi,
James Cook University, Australia

****Correspondence:***
Hooman Farkhani
farkhani@eng.au.dk;
h_farkhani@yahoo.com

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *12 July 2019*
Accepted: *17 December 2019*
Published: *22 January 2020*

Citation:
Farkhani H, Böhnert T, Tarequzzaman M, Costa JD, Jenkins A, Ferreira R, Madsen JK and Moradi F (2020) LAO-NCS: Laser Assisted Spin Torque Nano Oscillator-Based Neuromorphic Computing System. Front. Neurosci. 13:1429. doi: 10.3389/fnins.2019.01429

INTRODUCTION

The grand challenge of exascale computing, 10^{18} operations/second, calls for a dramatic change in hardware of the current petascale supercomputers. A paradigm shift is needed to tackle the issue of processing the explosively growing Big Data from different sources, which are mostly images and videos as the most time and power-consuming task for the existing Von-Neumann

computing machines (VNCs). Filling the gap between the performance of the current computing systems and the brain requires development of a computing system with similar features as the brain; brain-inspired computing systems, so-called neuromorphic computing systems (NCSs). Such systems overcome the limitation of the word-at-a-time thinking of the VNCs by massive parallel data processing similar to the brain (U.S. Department of energy, 2015; DeepMind, 2018; Hbp, 2018; Ibm, 2018; SpiNNaker, 2018). An NCS includes many parallel processors (neurons) communicating using simple messages (spikes) through programmable memory units (synapses). Although significant progress has been made in the CMOS implementation of NCSs, there are some fundamental limits to the simultaneous improvement of area and power in CMOS-based NCS (Fong et al., 2016). Such limits have driven a significant effort to investigate beyond-CMOS NCSs. The spin-based devices integrated with electronics (i.e., spintronics) have opened a door for designers to implement low-power high-density NCSs. In spintronic-based NCSs, magnetic switching in magnetic tunnel junction (MTJ) (Fong et al., 2016) or magnetic oscillation in spin-torque nano-oscillator (STNO) (Yogendra et al., 2015, 2016; Kurenkov et al., 2019) is used to mimic neuron firing. While using oscillation of magnetic moment decreases the power consumption by an order of magnitude compared with the magnetic moment switching [critical current: $\sim 10^{6}$ Acm^{-2}(Costa et al., 2017) vs. $\sim 10^{-7} Acm^{-2}$ (Fukami et al., 2016)], still there is a huge gap between spintronic-based NCSs and the brain in terms of power consumption and speed. This is due to the fact that the traditional way of oscillating the magnetic moment through the bias current consumes high power and it is done at low speeds. Hence, there is a crucial need for eliminating or decreasing the bias current in spintronic-based NCSs.

Magnetic tunnel junctions and STNOs can be used to perform Bayesian computation in networks inspired by cortical microcircuits of pyramidal stochastic neurons. This type of neurons spikes stochastically, observed in the cortex (Sengupta et al., 2016). The membrane voltage of a cell can change from the rest potential to oscillatory mode as a result of bifurcation (Bose, 2014). This is very similar to what happen inside STNOs, where the magnetization of FL starts to oscillate by increasing the current passing through the STNO to the currents higher than critical current (Hopf bifurcation). On the other hand, STNOs can show different precession modes based on their bias current (out-of-plane precession and in-plane precession with small or large angle), which are as the result of different bifurcation types, e.g., Hopf bifurcation causes in plane precession and heteroclinic bifurcation leads to out-of-plane precession (Nakada and Miura, 2016). However, in this work, the STNOs with in-plane precession have been used and in order to mimic neuron firing the transition from the magnetization resting state (non-oscillating) to the magnetization oscillation is utilized. It should be noted that the STNOs cannot be used to mimic all bifurcations, for example STNOs unable to mimic SNIC (saddle node on an invariant circle) bifurcation where the f-I curve is continuous (Bose, 2014). In neural networks inspired by biological behavior, the activation function represents the rate of action potential firing in the cell (Hodgkin and Huxley, 1952). In this manuscript, STNOs are used to implement the binary activation function, which is widely used to implement the linear perceptrons in neural networks. The weakness of this type of activation function is that the number of neurons needed for achieving a certain amount of accuracy increases. However, the main goal of this manuscript is to investigate the proof-of-the-concept of improving the performance of the STNO-based systems by elevating the temperature of the STNOs using laser illumination. The STNO-based NCS is used as an application to explore the effectiveness of the proposed idea.

In this manuscript, for the first time to our knowledge, we propose to design a laser-assisted STNO-based NCS (LAO-NCS) to improve power consumption of the state-of-the-art NCSs by at least 40%; narrowing the gap of power efficiency between the Brain and the current NCSs.

Spin Torque Nano-Oscillators Basics

The schematic of an STNO is shown in **Figure 1A**, which consists of a Pinned Layer (PL) with fixed magnetization and a free layer (FL) with changeable magnetization direction separated by a tunneling oxide layer e.g., MgO or Al_2O_3. **Figure 1B** shows the magnetization direction of the free layer (m) and different torques acting on it (Yogendra et al., 2015). T_P describes the precession torque that leads to the oscillation of m. T_D is the damping torque that aligns m with H_{eff} and T_{STT} is the spin-transfer torque caused by a bias current (Yogendra et al., 2016). The interaction of T_{STT} and T_D determines the oscillatory orbit of m. As T_{STT} increases, m will be placed in an orbit farther than H_{eff}, which will lead to a lower frequency of oscillation of m as shown in **Figure 1B** (Csaba and Porod, 2013). It is shown experimentally and through simulation that the frequency of the STNO can be locked to the frequency of an RF current passing through it (Rippard et al., 2005, 2013) or an external oscillating RF field (Slavin et al., 2010). Moreover, the frequency of two STNOs can lock if they are close to each other (Kaka et al., 2005). In STNO-based NCSs, the frequency locking of the STNO and comparing its output power with a threshold power are two mechanisms used to implement neuron firing. However, in all cases, a very high DC current (i.e., bias current) is needed flowing through the STNO to generate the required torque (i.e., T_{STT}).

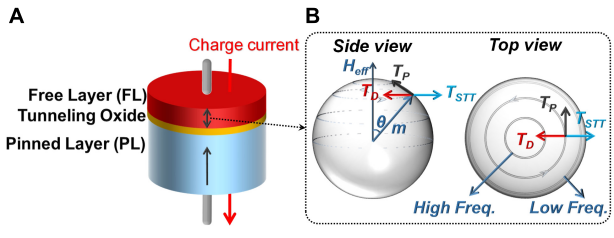


FIGURE 1 | (A) The schematic view of a MTJ as spin torque nano-oscillators (STNO) and **(B)** the magnetization direction of MTJ free layer (FL) and torques acting on it.

Effect of Raising Temperature on Spin Torque Nano-Oscillators

The dynamic behavior of the FL magnetic moment is modeled using Landau-Lifshitz-Gilbert-Slonczewski (LLGS) equation as follows (Sengupta et al., 2016):

$$\frac{(1+\alpha^2)}{|\gamma|}\frac{\partial \hat{m}}{\partial \tau} = -\hat{m}\times\vec{H}_{EFF} - \alpha\hat{m}\times\hat{m}\times\vec{H}_{EFF} + \frac{1}{qN_s}(\hat{m}\times I_s\times\hat{m}) \quad (1)$$

where, α, and $\hat{m}$ are the gyromagnetic ratio, Gilbert damping factor and magnetization of the FL, respectively. $\vec{H}_{EFF}$ is the effective magnetic field acting on FL described by $\vec{H}_{EFF} = \vec{H}_{UA} + \vec{H}_{ex} + \vec{H}_{TF}$, where $\vec{H}_{UA}$, $\vec{H}_{ex}$, and $\vec{H}_{TF}$ are uniaxial anisotropy field, external magnetic field and thermal fluctuations field, respectively. $N_s = \frac{M_S V}{\mu_B}$ is the number of spins in the FL of volume V (M_S is the saturation magnetization and μ_B is Bohr magneton) and I_s is the input spin current. The first term in (1) represents the precession torque (T_P) that makes $\hat{m}$ precess around the easy axis. The second term is the damping term (T_D) that tries to align $\hat{m}$ with easy axis. The third term represents the transverse component of spin current being absorbed by $\hat{m}$ (T_{STT}). In the absence of third term (no current passing through STNO), and in the equilibrium, $\hat{m}$ is aligned with easy axis. By increasing the current, the third term starts to increase, and m starts to oscillate around the easy axis. Higher currents will make the $\hat{m}$ to be placed on orbit farther than easy axis (i.e., higher output power). Increasing the temperature affects the dynamic behavior of the FL through decreasing the saturation magnetization (M_S) of it, decreasing the resistance of the STNO and increasing the dispersion of the initial deviation of the magnetic moment from easy axis due to higher thermal fluctuations.

Saturation Magnetization

It is shown theoretically (Ashcroft and Mermin, 1976) and experimentally (Alzate et al., 2014) that the dependency of M_S can be well described by Bloch's law as follows:

$$M_S(T) = M_S(0)\left(1-(T/T^*)^{\frac{3}{2}}\right) \quad (2)$$

where T is the absolute temperature in Kelvin and $M_S(0)$ is the saturation magnetization at 0K, and T^* is a fitting factor. Equation (2) shows that increasing the temperature decreases the M_S(T). This will lead to a degradation of the uniaxial anisotropy field, which decreases the minimum current required for the FL magnetic oscillation.

Resistance

Two tunneling mechanisms contribute to the STNO resistance including electron spin-polarized direct elastic tunneling and spin independent tunneling. The total conductance of the STNO can be described as (Teixeira et al., 2010).

$$G(\theta) = G_T\left[1+P_1P_2\cos\theta\right] + G_{ST} \quad (3)$$

where θ is the angle between the magnetization of the FL and the PL. P_1 and P_2 are the effective tunneling spin polarization of the magnetic layers. G_T is the pre-factor for direct elastic tunneling. All these parameters are temperature-dependent. Elevating the temperature will increase G_T and reduces P_1 and P_2 (Teixeira et al., 2010). As a result, R_P is almost independent of temperature while R_{AP} reduces approximately linearly with temperature. This has been experimentally shown in Teixeira et al., 2010, Takeuchi et al., 2015, and Hu et al., 2016.

Thermal Fluctuations

The effect of temperature on random fluctuating field can be modeled by $\vec{H}_{TF}$ while its x, y, and z components have uncorrelated Gaussian distribution with zero mean and $\sqrt{(2\alpha k_B T)/(\gamma M_S V \triangle t)}$ standard deviation (Brown, 1963; Sankey et al., 2005; Yogendra et al., 2017). α, k_B, γ, V, and Δt are the Gilbert damping parameter, the Boltzmann's constant, the gyromagnetic ratio, the volume of the FL and the integration time step. Elevating the temperature will increase the dispersion of $\vec{H}_{TF}$, which leads to an easier oscillation of the FL magnetic moment. In order to explore the mentioned effects on oscillation behavior of the STNO at elevated temperatures, different characteristics of the STNO (e.g., resistance, TMR, and output power of the oscillation) have been measured at different temperatures from 27°C up to 100°C in section "Memristor Behavior at Elevated Temperatures."

Memristor Behavior at Elevated Temperatures

Tantalum-oxide (TaO_x) memristors are one of the best candidate in memory and NCS applications due to their unique characteristics such as CMOS compatibility (Diokh et al., 2013), low power operation (Strachan et al., 2011), high endurance (Lee et al., 2011), and long retention of states (Ninomiya et al., 2013). The conduction mechanism of the TaO_x memristors can be modeled by two parallel conduction mechanisms including hopping conduction and Schottky thermionic emission (Graves et al., 2017) as follows:

$$I_{total} = \underbrace{2elv_{ph}Nk_BTe^{\left(-\frac{W}{k_BT}\right)}e^{\left(-\frac{2I}{\zeta}\right)}sinh\left(\frac{qlF}{k_BT}\right)}_{I_{Hop}} + \underbrace{AT^2e^{\left(-\frac{\phi_{B0}-\beta\sqrt{F}}{k_BT}\right)}}_{I_{Sch}} \quad (4)$$

where, k_B is the Boltzmann constant, I is the hopping distance, W is the hopping energy, ζ is the wave function localization, F is the applied field (converts from V), T is the temperature, v_{ph} is the vibrational phonon frequency, A is the reduced effective Richardson constant multiplied by active device area, ϕBo is the barrier height, and β is the barrier lowering factor. N is proportional to the density of electrons in the conduction path multiplied by the relevant conducting area. Based on this model, which is well fitted with experimental results, the temperature dependence of TaO_x memristor resistance can be divided into two regions called cold and hot regions (Graves et al., 2017). In the cold region ($T \leq 350$K), the state-dependent hopping conduction is dominant and the resistance of memristor is almost temperature insensitive. In the hot region, however, the Schottky emission of electrons determines the hot current, and the memristor's resistance decreases with raising the temperature,

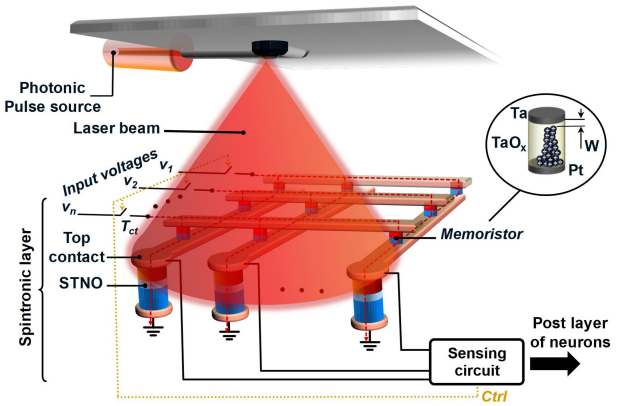


FIGURE 2 | The schematic view of the novel LAO-NCS. The STNO and memristor act as neuron and synapse, respectively. The STNOs will be heated to 100°C by illuminating a laser pulse.

rapidly. Note that, the amount of resistance change of memristor in hot region depends on the memristor's initial resistance.

Proposed Laser Assisted Neuromorphic Computing System

Our novel envisioned LAO-NCS is shown in **Figure 2**, which is a crossbar array of programmable TaO_x memristors as synapses and the STNOs as neurons assisted thermally by a narrow laser-pulse. Considering the fact that in many applications, size of the memristor array is much larger than the area of the STNO-based neurons, there is no significant area improvement in stacking the memristor array on top of the STNOs. Hence, the memristor array and the STNOs are supposed to be next to each other in the proposed LAO-NCS. Moreover, this structure allows direct laser illumination on the STNOs' top contacts. The resistance of the memristors can be tuned using an electric signal flowing through them. The NCS operation starts with a calibration phase in which the temperature of the STNOs will be elevated to 100°C and stabilized. Then, the NCS is ready for operation and the processing phase will start. The processing phase can be divided into two steps including stimulation and recovery, which will be repeated in sequence. In the stimulation step, the crossbar array sums the weighted input currents passing them to the STNOs, which are already set in AP-state (the magnetization direction of the FL and the PL are anti-parallel). In case, the weighted input currents are sufficiently large, the FL magnetic moment of the STNO starts to oscillate that will be detected by a sensing circuit immediately, and translated to neuron firing. The sensing circuit should use track and terminate method (Farkhani et al., 2017, 2018; Torrejon et al., 2017) in order to minimize the energy consumption of the NCS. Immediately after detecting the STNO oscillation, the recovery step begins. In the recovery step, the input corresponding to the fired neuron will be activated in the post-synaptic neuronal layer. Note that one of the advantages of using oscillation instead of magnetic moment switching is that there is no need for switching back the FL magnetization. Hence, the recovery step can be done in a very short time (∼600 ps) compared with the magnetic moment switching (∼2 ns) without extra energy consumption for switching back the magnetic moment. In our approach, the energy consumption needed for starting the STNO oscillation will be lowered significantly by increasing the temperature of the STNO using a nanosecond laser pulse. In fact, increasing the temperature of the STNO will decrease its energy barrier, which leads to a lower bias current needed for starting the oscillation in the STNO. On the other hand, in case, the temperature of TaO_x memristor array increases to temperatures above 350K due to heat propagation, the resistance of memristors will decrease, as discussed in previous section. However, it seems unlikely that the memristor array temperature reaches above 350K due to limited laser power. Moreover, in order to keep the memristor temperature below 350K, a thermal insulator layer can be placed between the memristor array and the STNOs. Considering the fact that the STNO current passes through the memristor array, the total power consumption of the memristor array will be reduced, significantly. As a result, the power consumption of the LAO-NCS decreases compared with typical spintronic-based NCSs. Considering the fact that the control transistors (T_{ct}) act as switches, heating them up has no significant impact on the overall performance of the LAO-NCS.

Interaction Between Laser and the Spin Torque Nano-Oscillators

The on-chip laser can be achieved through vertical cavity surface emitting laser (VCSEL) (Chen et al., 2014; Zhou et al., 2015; Kozlov and Carusone, 2016). VCSEL's unique specification is that, in contrast to the conventional edge-emitting semiconductor lasers, its laser beam is emitted perpendicular to its surface, which makes it a proper candidate for on-chip laser applications including the LAO-NCS. The output power of VCSEL can be tuned through changing the supply voltage of its driver (Kozlov and Carusone, 2016). Hence, in order to control the output power of the laser, a CMOS interface circuit is designed, which is described below.

CMOS Interfacing Circuit

Figure 3A shows the block diagram of the proposed LAO-NCS. The spintronic layer includes a memristors array, STNOs, T_c, and a sensing circuit. The CMOS interface circuit adjusts the output laser power by manipulating the supply voltage of the laser diode driver (LDD). In this way, the CMOS interfacing block can control the STNO temperature in the spintronic layer. **Figures 3B,C** show the circuit design of the CMOS interfacing block and its timing diagram, respectively. As mentioned before, the LAO-NCS operating time can be divided to *calibration* and *processing* phases. In the calibration phase, the temperature of the STNO is increased from 27°C to 100°C (first laser pulse with high power), and stabilized at this temperature (second laser pulse with low power). In the processing phase, the STNO temperature will be kept at 100°C with a sequence of low power laser illuminations as shown in **Figure 3C**. The operation of the CMOS interfacing circuit is as follows. The counter is clocked with a 500 MHz clock and generates the b0, b1, and b2 signals.

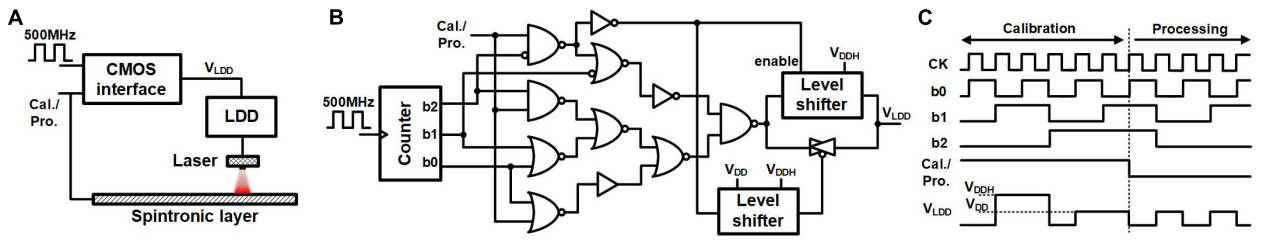


FIGURE 3 | (A) The block diagram of the proposed LAS-NCS including CMOS interfacing block, VCSEL array, LDD, and the spintronic layer. **(B)** CMOS interfacing circuit design. **(C)** Timing diagram of the CMOS interfacing block.

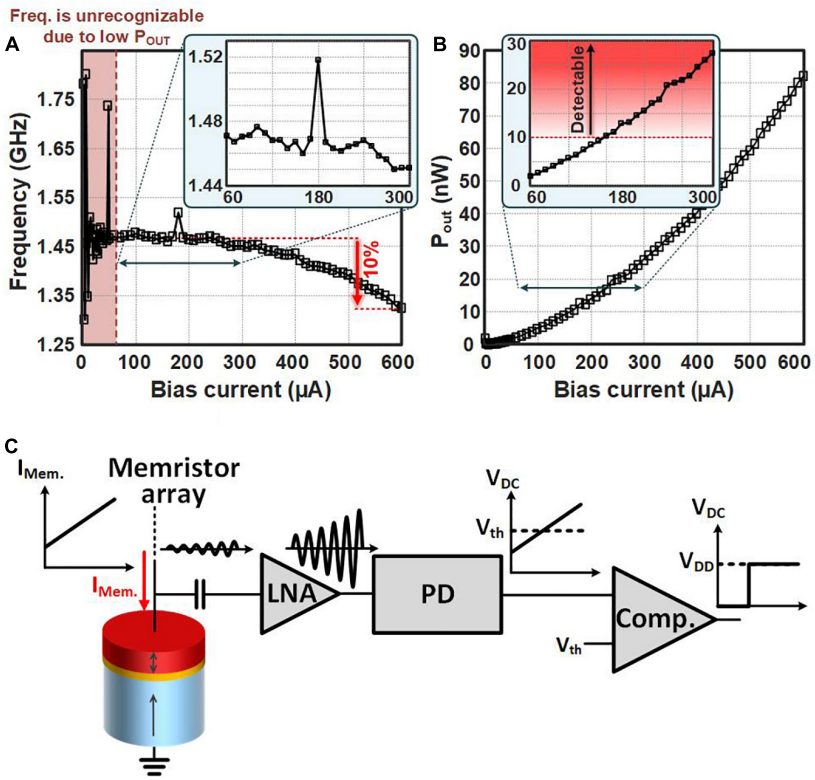


FIGURE 4 | The measured **(A)** frequency and **(B)** output power of the STNO versus different bias current from 0 to 600 μA. The frequency of oscillation is unrecognizable from noise at I_{Bias} <60 μA due to low output power of oscillation. The maximum frequency change is 10% @ I_{Bias}=600 μA. Output powers higher than 10 nW are detectable by sensing circuits. **(C)** The schematic view of the neuron firing detection approach.

Then, the logic circuit generates the V_{LDD} signal from the output of the counter. During the first pulse of V_{LDD}, the level shifter is enabled by a logic circuit and the voltage of V_{LDD} will be set at V_{DDH} that leads to a high output power laser pulse. During the next pulses, the transmission gate is enabled and the level shifter is disabled by the logic circuit. Hence, the voltage of the V_{LDD} node is at V_{DD} and the laser output power will be lower.

Neurons' Readout Approach

The sensing circuit is to sense the magnetization oscillation of the STNOs (neurons) in order to find the fired neuron(s) and activate the corresponding input(s) in the post-synaptic neuronal layer. This can be done either by sensing the frequency or the output power of the oscillating signal across the STNOs, and comparing it with a threshold frequency or a threshold output power. **Figures 4A,B** shows the measured frequency and the output power of our STNO samples in response to different bias currents. At bias currents lower than 60 μA, the output power of oscillation is very low. As a result, the frequency of oscillation is not detectable. By increasing the bias current, the frequency of oscillation decreases. However, the frequency reduction rate is slow (just 10% frequency reduction at 600 μA). Hence, it is difficult to detect the fired neuron by comparing the frequency of oscillation with a reference frequency. In contrast, thanks to the advances in power detector (PD) circuits, signals with few nano-Watt output power are

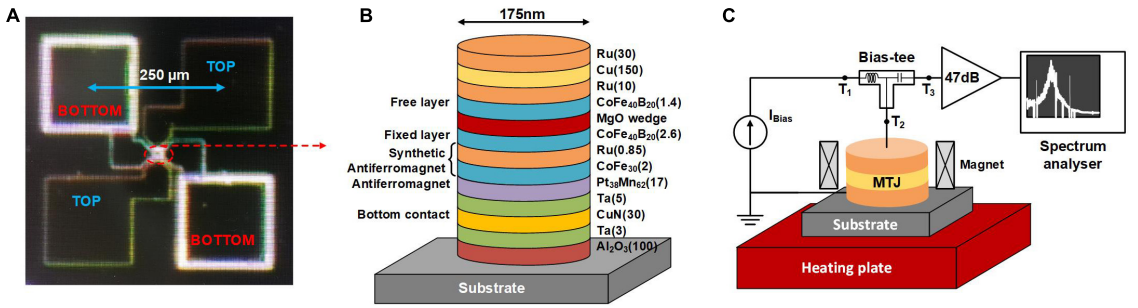


FIGURE 5 | (A) The microscopic image and **(B)** schematic view of the MTJ stack as STNO. **(C)** Schematic view of the experimental setup used for characterization of the STNO at different temperatures.

detectable within few nano-seconds and with micro-Watts power consumption (Li et al., 2010; Qayyum and Negra, 2017). Hence, the output power of oscillation is used to detect the oscillating STNO.

The schematic view of the sensing approach is shown in **Figure 4C**. The current of memristor array passing through the STNO leads to its resistance oscillation. As a result, a weak signal with milli-Volt amplitude oscillating at GHz frequency will appear across the STNO. This weak AC signal, first, will be amplified by a low noise amplifier (LNA). Then, the output signal of LNA will be converted to a DC voltage by the PD. The output voltage of the PD will be compared with a threshold voltage by the comparator. In case, $I_{Mem.}$ passing through the STNO will be high enough, the DC output voltage of the PD becomes higher than the threshould voltage. Hence, the output voltage of the comparator switches from "0" to "1" and it will be considered as neuron firing.

RESULTS

In order to evaluate the power efficiency of the LAO-NCS, first, the effect of elevating the temperature on the STNO characteristics is measured. Then, based on the measured results, a behavioral model of the STNO is extracted. For TaO_x memristor, a behavioral model for the temperature dependency of its resistance is extracted based on the data of Graves et al. (2017). Finally, both models are used to measure the power consumption of LAO-NCS in MATLAB simulator. The CMOS interface circuit is simulated and validated by HSPICE simulator in 65 nm CMOS technology. The thermal interaction between the laser pulses and the STNO is simulated in COMSOL simulator.

Experimental Measurement

In order to explore the effect of rising temperature on the STNO characteristics, we used the STNO stack structure of Substrate/(100) Al_2O_3/(3) Ta/(30) CuN/(5) Ta/(17) $Pt_{38}Mn_{62}$/(2) $CoFe_{30}$/(0.85) Ru/(2.6) $CoFe_{40}B_{20}$/MgO wedge/(1.4) $CoFe_{40}B_{20}$/(10) Ru/(150) Cu/(30) Ru (thicknesses in nm). The CoFeB FL has in-plane magnetization. The stack has the circular shape with diameter of 175 nm. The microscopic image of the STNO sample and the schematic view of the deposited layer stack are shown in **Figures 5A,B**.

To evaluate the output power of the STNO at different temperatures, the experimental setup of **Figure 5C** is utilized. The bias current is injected to the STNO through T_1 and T_2 terminals of the bias-tee. In case, the bias current will be high enough, it will lead to the oscillation of the STNO resistance. This resistance oscillation will provide a micro-volt oscillation at T_3 terminal of the bias-tee. Finally, the micro-volt oscillation of the STNO is amplified by an amplifier (47 dB) and will be injected to a spectrum analyzer in order to measure the oscillation characteristics of the STNO. The heating plate is used to set the temperature of the STNO at different temperatures above the room temperature. **Figure 6A** shows the PSD measured at different temperature from 27°C to 100°C for 230 μA bias current (the curves are offset by 10 μV^2 along the vertical axis for clarity). Note that the impedance mismatch in the acquired spectrum needs to be considered. The input impedance of the amplifier is 50Ω. Hence, considering the resistance mismatch between the amplifier and the STNO, the measured output power is only a fraction of actual emitted power of the STNO. In order to eliminate the effect of impedance mismatch, the integrated matched output power (P_{out}) of each device is calculated as follows (Costa et al., 2017):

$$P_{out} = P_{measured}\left(\frac{(R_{STNO} + R_{Amp})^2}{4R_{STNO}.R_{Amp}}\right) \quad (5)$$

where R_{STNO} and R_{Amp} are the resistance of the STNO and input resistance of the amplifier, respectively. $P_{measured}$ is the measured output power based on the spectrum analyzer output. **Figure 6B** shows the measured STNO resistance in P- and AP-state at different temperatures from 27°C to 100°C. The AP-state resistance is decreased with increasing the temperature and the P-state resistance is almost constant as predicted by equation (3), and shown experimentally before (Teixeira et al., 2010; Takeuchi et al., 2015; Hu et al., 2016). As a result, the TMR ratio decreases by increasing the temperature (**Figure 6C**) that shows the typical behavior of MTJs as a function of the bias current. The matched output power (P_{out}) of the STNO versus the bias current at different temperatures from 27°C to 100°C is shown in **Figure 6D**. By applying sufficient positive bias current, the oscillation will start, and by further increasing the bias current, the amplitude of the oscillation increases, which

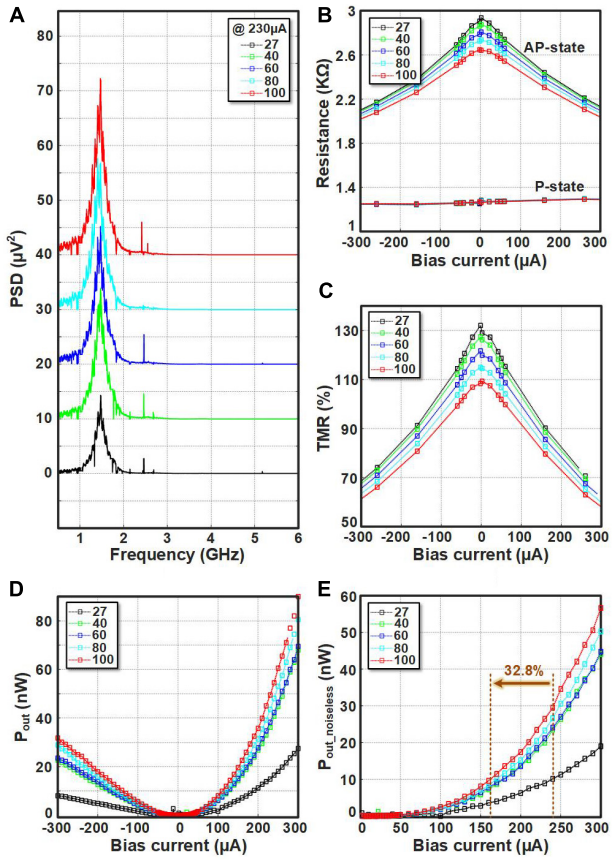


FIGURE 6 | (A) The PSD measured at different temperature from 27°C to 100°C for 230 μA bias current, the curves are offset by 10 μV^2 along the vertical axis for clarity. **(B)** The AP and P states resistance, **(C)** the TMR ratio, **(D)** the integrated matched output power (P_{out}), and **(E)** the noiseless output power of STNO sample.

leads to a higher output power. Although the decrease in TMR with the bias current and temperature give an adverse result, the total power increases due to the fact that the input power increase dominates. It should be noted that applying a negative bias current will not cause oscillation, but increases the noise power, which leads to a higher P_{out}. In order to eliminate the effect of noise on P_{out}, the output power of the negative bias currents are deducted from the output power of the positive bias currents as shown in spin Hall nano-oscillators (SHNOs) (Tarequzzaman et al., 2019). As a result, the minimum bias current needed to detect the STNO oscillation of the fired neuron by the sensing circuit will decrease. This decreases the total energy consumption of the LAO-NCS as will be discussed in section "Hand-Written Digit Recognition Application".

Laser-Spin Torque Nano-Oscillators Interaction

The laser-STNO heat transfer is simulated in the COMSOL multiphysics simulator for the STNO stack (Böhnert et al., 2017). The shape, material and sizing of each layer is exactly similar to the STNO stack used in the experimental measurements. The laser beam is illuminated on the top electrode of the STNO stack to heat up the overall temperature of it. Hence, the top electrode should absorb the maximum laser energy in order to achieve the maximum efficiency. The bottom electrode is made of CuN with a thickness of 30 nm, while the top electrode is made of AlSiCu. Hence, a nanosecond laser with 355 nm wavelength is used to decrease the transmissivity of electrode. The optical transmittance and reflectance of the electrode are around 0.13 and 0.25, respectively (Maruyama and Morishita, 1998). Note that, the transmitted laser will be absorbed by lower layers in the STNO stack and increases its overall temperature. Hence, the energy loss is just related to the reflected laser. This is considered for calculating the total energy consumption described in the next sections. This energy loss can be reduced by engineering the material and surface of the top electrode.

Figure 7A shows the temperature distribution in the STNO stack at the end of calibration phase, which shows a uniform temperature distribution in all parts of the STNO. **Figure 7B** shows the laser power distribution. The power and the diameter of the laser beam are 71 μW and 350 nm, respectively. **Figure 6C** shows the normalized laser power in each laser radiation during the calibration and processing phases. The first two consecutive laser pulses do the calibration phase. The first laser pulse is illuminated for 4 ns with 100% power (71 μW) to heat the MTJ stack above 100°C. Then, the laser beam is cut off for 2 ns. The second laser pulse is applied for 4 ns with 30% power (21.3 μW) in order to stabilize the STNO temperature.

As mentioned before, the processing phase includes the stimulation and recovery steps. In the stimulation step, the inputs will be applied to the NCS and their corresponding response will be calculated by the NCS. The laser is cut off during the stimulation step. Then, the recovery step will be started and the input corresponding to the fired neuron will be activated in the post-synaptic neuronal layer. Moreover, the laser will be illuminated on the STNO with 30% power for 2 ns during the recovery step in order to compensate the heat-loss during the stimulation step. During the recovery step, the NCS inputs are disconnected and the power consumption of the STNOs and the memristor crossbar array is almost zero. This will continue repeatedly to keep the STNO temperature around 100°C. **Figure 7D** shows the temperature of the FL in the STNO stack, which is almost stabilized around 100°C (∼±5°C). It should be noted that the low temperature variations across the MgO barrier at the STNO stack prevents the reliability issues.

Power Consumption of Spin Torque Nano-Oscillators and Memristor

In this section, the effect of elevating the temperature on power consumption of the STNO and memristor is explored. **Figure 8A** shows the power consumption of an STNO at different temperatures. In order to calculate the STNO power consumption, the STNO current is supposed to be the minimum current required for starting the oscillation with a detectable

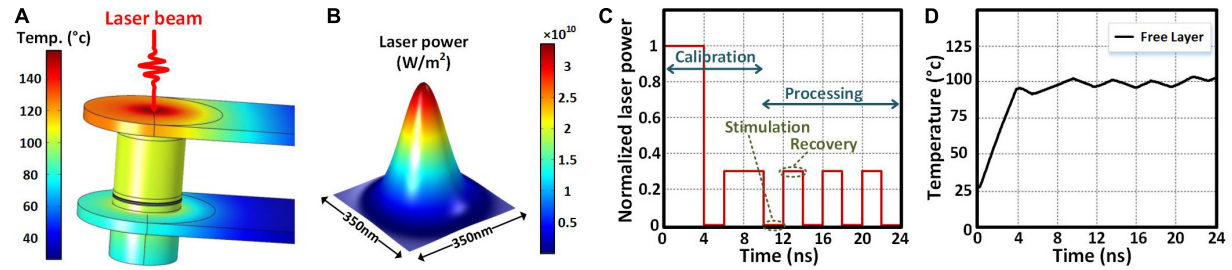


FIGURE 7 | (A) The temperature distribution in MTJ stack at the end of calibration phase. **(B)** The laser power distribution. **(C)** The laser power pulse normalized to maximum laser power versus time. **(D)** The maximum and minimum temperature of MTJ stack and the FL temperature versus time.

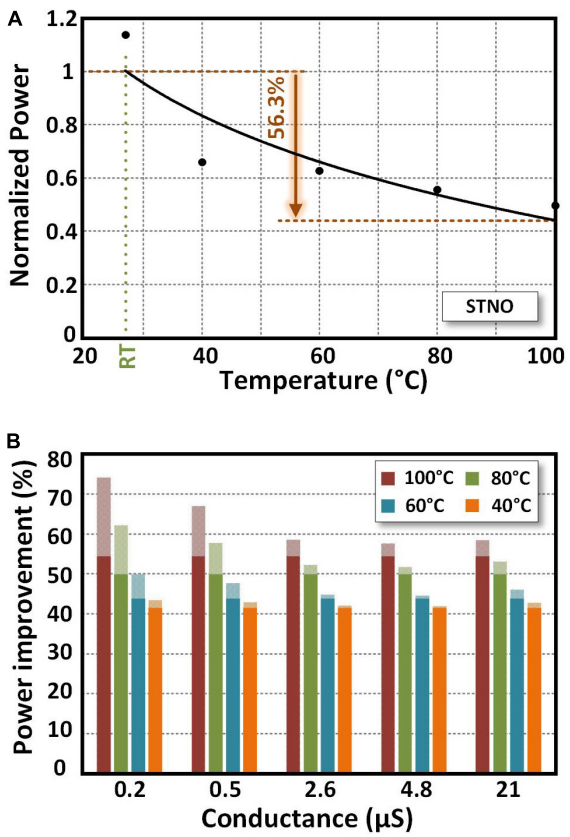


FIGURE 8 | (A) The power consumption of an STNO vs. temperature. Power consumption is normalized to its room temperature value. **(B)** The power consumption improvement of memristor with different initial values at different temperature from 40°C to 100°C compared with room temperature (27°C). The solid part is related to memristor current reduction due to STNO current reduction. The dotted part is related to resistance reduction of memristor at elevated temperatures. In order to calculate the power consumption, the STNO and memristor current is supposed to be the minimum current required for starting oscillation (162 μA @ 100°C and 241 μA @ 27°C).

output power, ranging from 241 μA at 27°C to 162 μA at 100°C (**Figure 6E**). The power consumption of STNO at the recovery step is zero. Hence, the calculated power consumption for the STNO is related to the stimulation step. The power consumption of the STNO decreases by 56.3% (127 μW @ 27°C to 55.5 μW @ 100°C) while increasing the temperature to 100°C. This is due to the fact that heating up the STNO reduces its magnetization saturation and effective anisotropy field that tends to keep the magnetization direction of the FL aligned with the easy axis. Note that in real applications, the STNO current depends on the input voltages of the memristor array and the initial resistance of the memristors. Hence, the real power consumption improvement is application-dependent.

In order to calculate the power consumption of TaO_x memristors at elevated temperatures, the conductance of the memristors and the current passing through them should be measured at different temperatures. However, the amount of conductance increase depends not only on the temperature, but also is a function of the initial resistance of the memristor (weights) and the applied voltage (inputs). **Figure 8B** shows the power consumption improvement of the TaO_x memristors with different initial conductance from 0.2 μS to 21 μS at different temperatures from 40°C to 100°C compared with the room temperature (27°C). The solid part is related to memristor current reduction due to the STNO current reduction and the dotted part is related to the resistance reduction of the memristor at elevated temperatures. The conductance range used in **Figure 8B** is aligned with the experimental data of Graves et al. (2017) that is used for benchmarking. Considering the fact that the STNO current will be passed through the memristors, the memristor current is considered equal with the STNO current at different temperatures when calculating the memristor power consumption.

As illustrated in **Figure 8B**, by increasing the memristor temperature to 100°C, the power consumption reduction of 59% for memristors with initial conductance values equal or lower than 2.6 μS is expected. However, for memristors with higher initial conductance values, the conductance increase rate due to the increased temperature is higher which leads to a larger power reduction. Note that most of the power consumption improvement of the memristors is due to the lower current passing through them (e.g., 4.8 μS at 100°C: 54.4% power improvement due to lower current versus 3% power improvement due to resistance reduction), especially at conductance values equal or lower than 2.6 μS. On the other hand, heating up the memristor array requires a laser pulse with higher output power, which reduces the power efficiency of the

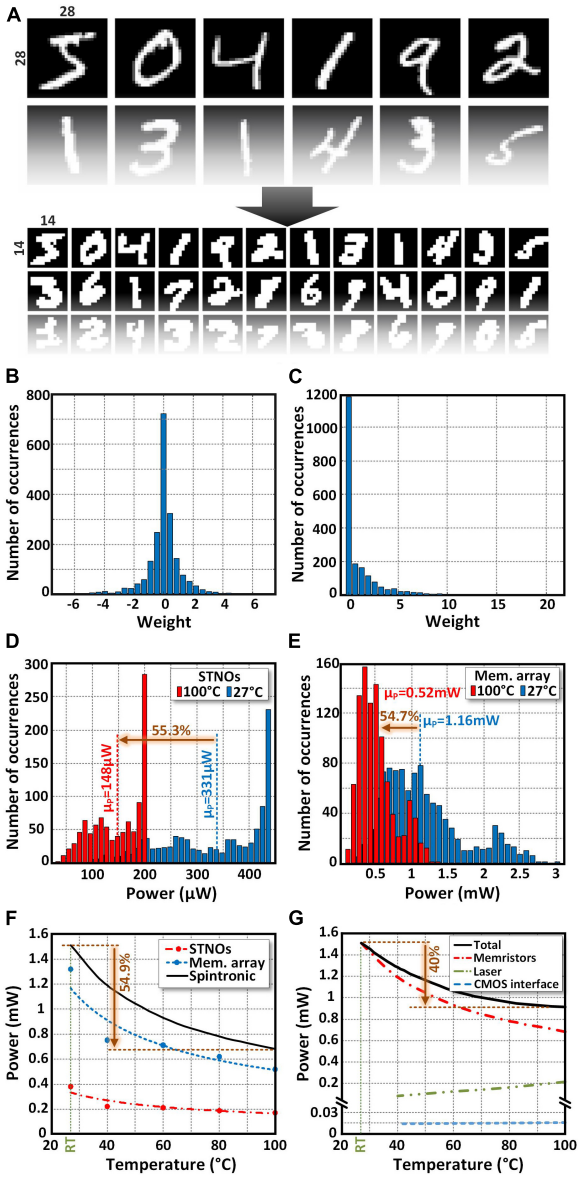


FIGURE 9 | (A) The size of pictures in MNIST database is reduced to 14 × 14 and used to train the 196 × 10 NCS. **(B)** The original weights after training. **(C)** Weights after setting the negative ones to zero and multiplying the positive ones by 3. The power consumption distribution of **(D)** STNOs and **(E)** memristor array in 196 × 10 NCS for 1000 test images at 27°C and 100°C, respectively. The average power consumption of **(F)** STNOs, memristor array, **(G)** laser, CMOS interface circuit, and the whole NCS at different temperatures from 27°C to 100°C.

proposed LAO-NCS. Hence, in the LAO-NCS, the laser is used to heat up the STNOs only and the temperature of memristor array leaved unchanged.

Hand-Written Digit Recognition Application

Considering the fact that independent studies of spintronic elements cannot accurately reflect the performance of the whole NCS, the effectiveness of the proposed LAO-NCS is evaluated by the hand-written digit recognition application. To do that, a 196 × 10 NCS is designed to recognize the handwritten digits in MATLAB. Then, the MNIST handwritten digits database (LeCun et al., 1998) is used to train the NCS and the weights are extracted. The network is trained by 1000 training images using the Scaled Conjugate Gradient (SCG) method for a fully connected feedforward neural network. The size of the images is reduced to 14 × 14 (**Figure 9A**). Considering the facts that the negative weights cannot be implemented by the memristors, the negative weights are considered as zero (**Figures 9B,C**). This will reduce the accuracy of system (89% → 54%). In order to compensate the accuracy reduction partially, the positive weights are multiplied by three. This will increase the accuracy from 54% to 71.3%. In the next step, the weights are mapped to the resistance of memristors in the array (the zero weights are considered as open circuit). In order to model the effect of temperature increase on the power consumption of the STNOs, equations are fitted to the experimental results of section "Interaction between Laser and the STNOs" (**Figure 6**). Then, the fitted equations are used in MATLAB to model the power consumption of the LAO-NCS at different temperatures from 27°C to 100°C. Considering the fact that the laser just illuminated on the STNOs, the temperature of the memristor array is assumed to be lower than 350K (the memristor resistance is constant with respect to its temperature). However, in case, the temperature of the memristor array increases due to heat propagation from the STNOs, the resistance of the memristors slightly reduces that improves the power efficiency of the LAO-NCS. Finally, 1000 test images have been applied to the modeled NCS in MATLAB and the power consumption reduction of the STNOs (**Figure 9D**) and the memristor array (**Figure 9E**) is calculated for each test image. The accuracy of the system is 71.3%. This is due to the fact that the negative weights cannot be implemented using memristors. The accuracy can be improved by adding a hidden layer to the system or adding the number of neurons in each layer, which comes with a higher complexity in hardware implementation. However, the main focus of this manuscript is on investigating the effect of raising temperature on the power consumption of the STNO-based NCS.

As illustrated in **Figures 9D,E**, increasing the temperature from 27°C to 100 has reduced the average power consumption of the memristor array and the STNOs by 54.7% and 55.3%, respectively. Hence, the total power consumption of the spintronic layer is reduced by 54.9% as shown in **Figure 9F**. This is due to the smaller resistance of the STNOs and the lower bias current passing through the memristors and the STNOs at elevated temperatures. The average power consumption improvement of the STNOs (54.9% in for a single STNO (56.3% in **Figure 8A**) in previous section. This is due to the fact that the calculated power consumption is the average of power consumption of all STNOs (the oscillating one and the non-oscillating STNOs).

The total power consumption of the LAO-NCS includes the power consumption of the spintronic layer (the memristors and the STNOs), the CMOS interfacing circuit, the CMOS sensing circuit and the laser. However, it should be noted that the

sensing circuit is common between the LAO-NCS and typical STNO-based NCS. Hence, it has similar effect on the total power consumption of both circuit. Due to the fact that the calibration phase is done just one time at the beginning of the NCS operation, its power consumption's contribution to the total power consumption of the LAO-NCS is negligible. Hence, the power consumption of the LAO-NCS is calculated only for the processing phase. The power consumption of the CMOS interfacing circuit and the laser are shown in **Figure 9G** at different temperatures. The power consumption of the CMOS interfacing circuit can be further decreased using low voltage circuit techniques. However, due to its very low power consumption (∼15μW), its effect on the total power consumption is negligible. To achieve a higher temperature, a higher laser power is required. Note that, in stimulation step, the laser is turned off and it has no power consumption. In recovery step, the laser is illuminated for 2 ns with 213 μW power to keep the temperature of the STNOs around 100°C. Considering the fact that the power consumption improvement in the spintronic layer is significantly higher than the power consumption of the laser and the CMOS interfacing circuit, the total power consumption of the LAO-NCS decreases by 40% at 100°C compared with the room temperature.

The power consumption of the CMOS sensing circuit can be estimated as the total power consumption of LNA, PD, and comparator. The power consumption of LNA at required frequency range (1.3–1.5 GHz, **Figure 4A**) can be estimated around 160 μW (Parvizi et al., 2016). However, this is a wideband LNA and the power consumption of narrowband LNA can be lower (Kargaran et al., 2018). Moreover, PD circuits with power consumption lower than 100 μW are realized in the literatures (Li et al., 2010; Qayyum and Negra, 2017). Finally, the power consumption of the comparator (@500 MHz) can be estimated 200 μW (Khorami and Sharifkhani, 2018). All in all, the total power consumption of the CMOS sensing circuit is estimated to be lower than 400 μW for this specific application.

Technology Scaling Effect on LAO-NCS

Technology scaling will lead to a lower laser power consumption due to the smaller size of the STNOs. As a result, the power efficiency of the LAO-NCS is expected to improve further. As an example, for the STNO samples of Monteblanco et al. (2017), the STNO area is 60 nm × 70 nm = 4200 nm^2. Comparing with the STNO samples used in this manuscript with 24052 nm^2, the STNO samples of Monteblanco et al. (2017) need a laser pulse with ∼ 38 μW output power in order to increase their temperature up to 100°C. Decreasing the laser power from 213 μW to 38 μW reduces the total power consumption of the NCS at 100°C from 911 μW to 736 μW in **Figure 9G**. As a result, the power consumption improvement of the LAO-NCS will be increased from 40% to 51.3%.

Energy Consumption

Elevating the temperature reduces the switching time of the MTJs (Farkhani et al., 2019b). Since the oscillation mechanism of STNOs is similar with switching, similar trend is expected for the delay before starting the oscillation. Hence, at 100°C, 77% delay reduction can be expected (Farkhani et al., 2019b). The energy consumption can be calculated from multiplication of power consumption and delay. Hence, considering 77% delay reduction and 40% power consumption reduction, 86% lower energy consumption of the LAO-NCS at 100°C can be expected compared with a typical STNO-based NCS at room temperature.

Comparison With CMOS-Based NCS

Considering the fact that there is no fully-implemented and integrated spintronic-based NCS, it is hard to perform an accurate comparison between spintronic-based NCS and the CMOs-based NCS. Hence, it is tried to give a general perspective. Synaptic memristors (130×10^{15}) shows 10–100 times better performance (operation/sec/Watt/cm^2) over CMOS-based synapses ($\sim 2 \times 10^{15}$) (Mandal et al., 2014). Furthermore, 2–3 orders of magnitude improvement by MTJ neurons and their sensing circuit (1.2×10^8) is achieved over fully-CMOS implementations (2.3×10^5) (Mizrahi et al., 2018). Neurons and their sensing circuits contribute the most to the overall performance. Therefore, 2–3 orders of magnitude performance improvement is expected using MTJ-Memristor NCSs compared with the CMOS-based NCSs. The use of nano-oscillators specified for NCSs, one order of magnitude improvement in performance compared to the use of MTJ neuron, where full switching is used [critical current density: $\sim 10^6$ A/cm^2 (Costa et al., 2017) vs. $\sim 10^7$ A/cm^2 (Fukami et al., 2016)], is expected. Finally, thermally assisting STNOs using laser can improve the power consumption by 40%. However, the spin-based devices are suffering from high process variation and relatively high cost compared with their CMOS counterpart.

Comparison With the Other Heating Methods

Thermally assisted MTJ switching in STT-RAMs is widely used to decrease the bias current (Walter et al., 2011; Prejbeanu et al., 2013; Bender and Tserkovnyak, 2016; Dai et al., 2017; Safranski et al., 2017). Heating up the MTJ is used to improve the FL switching in two different ways including creating a temperature gradient between FL and PL, called Seebeck effect (Walter et al., 2011; Bender and Tserkovnyak, 2016; Safranski et al., 2017) and heating the MTJ above FL blocking temperature (T_B) to reduce its switching current, called Thermally Assisted Switching – TAS (Prejbeanu et al., 2013; Bandiera and Dieny, 2016; Dai et al., 2017). In the first method, using a temperature gradient across the MTJ, a pure spin current will be injected to FL. This pure spin current acts as an anti-damping thermal spin torque (also called spin Seebeck torque) and decreases the bias current (Safranski et al., 2017). Note that in NCS application, heating up the MTJs should not lead to their switching, but heating should ease the switching. This means that the switching should happen by the current flowing from crossbar array with the help of heating. Hence, the Seebeck effect, in original form, cannot be used in NCSs. Moreover, considering the fact that temperature gradient is the source of Seebeck effect, specific time should be allocated for MTJ cooling before starting the next stimulation step that lowers the general speed of the NCS.

In TAS, a modified type of MTJ is needed, where the FL consists of a ferromagnetic layer pinned with a low T_B antiferromagnet (AF), such as FeMn (90–160°C) or IrMn (120–260°C). The PL is a SyF pinned with a high T_B antiferromagnet, such as PtMn (350°C) (Bandiera and Dieny, 2016). In standby mode, the FL presents a very high thermal stability, because it is pinned by the low T_B antiferromagnet. Then, during stimulation phase, the stack heats up in order to ease the FL switching (Prejbeanu et al., 2013). In TAS, Joule heating is used to heat the MTJ junction above the blocking temperature of antiferromagnetic layer by passing an extra current through it. Then, a magnetic field or a spin polarized current switches the FL magnetization. Finally, while keeping the magnetic field or spin polarized current, the MTJ stack is cooled down.

Energy Consumption

In TAS, in order to heat the FL layer by 200°C, a current density of 2–4 $\times$ 10^6 A.cm^{-2} with a bias voltage of 1.1V is needed for different materials (Prejbeanu et al.; 2013). Hence, the power consumption of the TAS is estimated as 0.53 mW to 1.06 mW for the MTJ stack with a cross section area similar to the MTJ stack used in our simulations [$\pi \times (87.5\ \text{nm})^2 = 24.052 \times 10^{-15}\text{m}^2$]. The power consumption of LAO for a 200°C temperature increase of the STNO is estimated as 0.4 mW, which shows 1.3X-2.6X lower power consumption compared with the TAS. In addition, the use of LAS has the following advantages over TAS:

(1) The TAS, in its original form, cannot be used in an NCS application because heating all the MTJs above T_B and passing current through them leads to FL switching in all MTJs.
(2) There is no need for an antiferromagnetic layer close to the FL and the proposed laser assisted method can be applied to typical MTJs.
(3) No heating current line is required, which improves the density.
(4) In contrast with the TAS, which needs a bipolar select transistor in order to inject two bipolar current into the MTJ, LAS can be used with CMOS select transistor.
(5) In TAS, the minimum heating time is limited to 500 ps (Bandiera and Dieny, 2016) due to the fact that the MTJ voltage should not exceed the MTJ breakdown voltage. However, the heating of the ferromagnetic material above Curie temperature by a femto-second laser pulse has been shown experimentally (Walowski, 2012).

In terms of complexity, considering the extra layer (photonic layer) needed in the implementation of the LAO, LAO comes with a higher complexity compared to the Joule heating approach.

CONCLUSION

To reduce the power consumption of future STNO-based NCSs, a microwatt-nanosecond laser pulse is utilized for the first time to ease the magnetic oscillation of the STNO through heating. The power consumption of the spintronic layer and the total power consumption of the proposed LAO-NCS are improved by 54.9% and 40% at T = 100°C compared with operation at the room temperature. Moreover, 86% lower energy consumption can be expected for the LAO-NCA at 100°C compared with a typical NCS at the room temperature. It should be noted that scaling the technology and increasing the temperature above 100°C leads to further improvement of the power consumption.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

HF and FM designed and performed the research, and wrote the manuscript together with TB and JM. JC, MT, and RF designed and fabricated the STNO samples for testing and characterization that was done by HF, TB, MT, AJ, and RF.

FUNDING

This work was supported by the Marie Skłodowska Curie Individual Fellowship (IF) under contract number 751089.

ACKNOWLEDGMENTS

This manuscript is an extension of a previously published conference manuscript (Farkhani et al., 2019a).

REFERENCES

Alzate, J. G., Khalili Amiri, P., Yu, G., Upadhyaya, P., Katine, J. A., and Langer, J. (2014). Temperature dependence of the voltage-controlled perpendicular anisotropy in nanoscale MgO|CoFeB|Ta magnetic tunnel junctions. *Appl. Phys. Lett.* 104:112410. doi: 10.1063/1.4869152

Ashcroft, N. W., and Mermin, N. D. (1976). *Solid states physics*. Rochester, NY: Saunders college.

Bandiera, S., and Dieny, B. (2016). "Thermally assisted MRAMs," in *Handbook of Spintronics*, eds Y. Xu, D. Awschalom, and J. Nitta, (Dordrecht: Springer).

Bender, S. A., and Tserkovnyak, Y. (2016). Thermally driven spin torques in layered magnetic insulators. *Phys. Rev. B.* 93, 064418.

Böhnert, T., Guisan, S. S., Paz, E., Lacoste, B., Ferreira, R., and Freitas, P. P. (2017). Magnetic tunnel junctions with integrated thermometers for magnetothermopower measurements. *Journal of Physics: Condensed Matter*. 29, 185303. doi: 10.1088/1361-648X/aa63ab

Bose, A. (2014) . "Bifurcations Dynamics of Single Neurons and Small Networks," in *Encyclopedia of Computational Neuroscience*, eds D. Jaeger, and R. Jung, (New York, NY: Springer/pubname).

Brown, W. F. (1963). Thermal fluctuations of a single-domain particle. *Physical review*. 130, 1677. doi: 10.1103/physrev.130.1677

Chen, C., Zhang, T., Contu, P., Klamkin, J., Coskun, A. K., and Joshi, A. (2014). "Sharing and Placement of On-chip Laser Sources in Silicon-Photonic NoCs," in *Proceedings of the 18th IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, (Ferrara), 88–95.

Costa, J. D., Serrano-Guisan, S., Lacoste, B., Jenkins, A. S., Böhnert, T., Tarequzzaman, M., et al. (2017). High power and low critical current density spin transfer torque nano-oscilltors using MgO barriers with intermediate thickness. *Nature Sci. Rep* 7, 7237. doi: 10.1038/s41598-017-07762-z

Csaba, G., and Porod, W. (2013). "Computational Study of Spin-Torque Oscillator Interactions for Non-Boolean Computing Applications," in *Proceedings of the IEEE Transactions on Magnetics*, (Piscataway, NJ).

Dai, B., Guo, Y., Zhu, J., Kato, T., Iwata, S., Tsunashima, S., et al. (2017). Spin transfer torque switching in exchange coupled amorphous GdFeCo/TbFe bilayers for thermally assisted MRAM application. *Journal of Physics D: Applied Physics* 50, 135005. doi: 10.1088/1361-6463/aa5bca

DeepMind (2018). *DeepMind*. Available at: https://deepmind.com/ (accessed October 1, 2018).

Diokh, T., Le-Roux, E., Jeannot, S., Cagli, C., Jousseaume, V., Nodin, J.-F., et al. (2013). Study of resistive random access memory based on TiN/TaOx/TiN integrated into a 65 nm advanced complementary metal oxide semiconductor technology. *Thin Solid Films* 533, 24–28. doi: 10.1016/j.tsf.2012.11.124

Farkhani, H., Böhnert, T., Tarequzzaman, M., Costa, D., Jenkins, A., Ferreira, R., et al. (2019a). "Spin-Torque-Nano-Oscillator based neuromorphic computing assisted by laser," in *Proceedings of the International conference on Design & Technology of Integrated Systems in nanoscale era (DTIS)*, (Braga).

Farkhani, H., Prejbeanu, I. L., and Moradi, F. (2019b). LAS-NCS: A Laser-Assisted Spintronic Neuromorphic Computing System. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 66, 838. doi: 10.1109/tcsii.2019.2908077

Farkhani, H., Tohidi, M., Farkhani, S., Madsen, J. K., and Moradi, F. (2018). A low-power high-speed spintronics-based neuromorphic computing system using real time tracking method. *IEEE Journal on emerging and selected topics in circuit and systems (JETCAS)*. 8, 627–638. doi: 10.1109/jetcas.2018.2813389

Farkhani, H., Tohidi, M., Farkhani, S., Madsen, J. K., Moradi, F., et al. (2017). "An energy efficient neuromorphic computing system using real time sensing method," in *Proceedings of the 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, (Turin).

Fong, X., Kim, Y., Yogendra, K., Fan, D., Sengupta, A., Raghunathan, A., et al. (2016). "Spin-transfer torque devices for logic and memory: prospects and perspectives," in *Proceedings of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, (Piscataway, NJ).

Fukami, S., Anekawa, T., Zhang, C., and Ohno, H. (2016). A spin-orbit torque switching scheme with collinear magnetic easy axis and current configuration. *Nature Nanotechnology* 11, 621–625. doi: 10.1038/nnano.2016.29

Graves, C. E., Davila, N., Grafals, E. J. M., Lam, S.-T., Strachan, J. P., and Williams, R. S. (2017). Temperature and field-dependent transport measurements in continuously tunable tantalum oxide memristors expose the dominant state variable. *Applied Physics Letters*. 110, 123501. doi: 10.1063/1.4978757

Hbp (2018). *Europe's Human Brain Project (HBP)*. Available at: https://www.humanbrainproject.eu/ (accessed October 1, 2018).

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Hu, B., Moges, K., Honda, Y., Liu, H., Uemura, T., and Yamamoto, M. (2016). Temperature dependence of spin-dependent tunneling conductance of magnetic tunnel junctions with half-metallic Co2MnSi electrodes. *Phys. Rev. B.* 94, 094428.

Ibm (2018). *Research: Brain-inspired chip*. Available at: http://www.research.ibm.com/articles/brain-chip.shtml (accessed October 1, 2018).

Kaka, S., Pufall, M. R., Rippard, W. H., Silva, T. J., Russek, S. E., and Katine, J. A. (2005). Mutual phase-locking of microwave spin torque nanooscillators. *Nature Lett.* 437, 389–392. doi: 10.1038/nature04035

Kargaran, E., Manstretta, D., and Castello, R. (2018). Design and Analysis of 2.4 GHz 30 μW CMOS LNAs for Wearable WSN Applications. *IEEE Tran. On Circuits and Systems-I: Regular Papers* 65, 891–903. doi: 10.1109/tcsi.2017.2771940

Khorami, A., and Sharifkhani, M. (2018). A Low-Power High-Speed Comparator for Precise Applications. *IEEE Trans. on Very Large Scale Integration* 26, 2038–2049. doi: 10.1109/tvlsi.2018.2833037

Kozlov, V., and Carusone, A. C. (2016). Capacitively-Coupled CMOS VCSEL Driver Circuits. *IEEE Journal of Solid-State Circuits*. 51, 2077. doi: 10.1109/jssc.2016.2584641

Kurenkov, A., Gupta, S. D., Zhang, C., Fukami, S., Horio, Y., and Ohno, H. (2019). Artificial Neuron and Synapse Realized in an Antiferromagnet/Ferromagnet Heterostructure Using Dynamics of Spin–Orbit Torque Switching. *Advanced Materiaæs Communication*. 31, 1970167. doi: 10.1002/adma.201970167

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*. 86, 2278–2324. doi: 10.1109/5.726791

Lee, M.-J., Lee, C. B., Lee, D., Lee, S. R., Chang, M., Hur, J. H., et al. (2011). A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta2O5-x/TaO2-x bilayer structures. *Nature Materials* 10, 625–630. doi: 10.1038/nmat3070

Li, C., Gong, F., and Wang, P. (2010). A low-power ultrawideband CMOS power detector with an embedded amplifier. *IEEE Trans. on Instrumentation and Measurement*. 59, 3270–3278. doi: 10.1109/tim.2010.2047131

Mandal, S., El-Amin, A., Alexander, K., Rajendran, B., and Jha, R. (2014). Novel synaptic memory device for neuromorphic computing. *Nature Scientific Reports* 4, 5333.

Maruyama, T., and Morishita, T. (1998). Copper nitride and tin nitride thin films for write-once optical recording media. *Applied Physics Letters*. 69, 890–891. doi: 10.1063/1.117978

Mizrahi, A., Hirtzlin, T., Fukushima, A., Kubota, H., Yuasa, S., Grollier, J., et al. (2018). Neural-like computing with populations of superparamagnetic basis functions. *Nature communications* 9, 1533. doi: 10.1038/s41467-018-03963-w

Monteblanco, E., Garcia-Sanchez, F., Gusakova, D., Buda-Prejbeanu, L. D., and Ebels, U. (2017). Spin transfer torque nano-oscillators based on synthetic ferrimagnets: Influence of the exchange bias field and interlayer exchange coupling. *Journal of applied physics* 121, 013903. doi: 10.1063/1.4973525

Nakada, K., and Miura, K. (2016). "Pulse-coupled Spin Torque Nano Oscillators with Dynamic Synapses for Neuromorphic Computing," in *Proceedings of the 16th International Conference on Nanotechnology*, (Piscataway, NJ: IEEE).

Ninomiya, T., Wei, Z., Muraoka, S., Yasuhara, R., Katayama, K., and Takagi, T. (2013). Conductive Filament Scaling ofTaOxBipolar ReRAM for Improving Data Retention Under Low Operation Current. *IEEE Trans. on Electron Devices* 60, 1384–1389. doi: 10.1109/ted.2013.2248157

Parvizi, M., Allidina, K., and Ei-Gamal, M. N. (2016). Short Channel Output Conductance Enhancement Through Forward Body Biasing to Realize a 0.5 V 250 μW 0.6–4.2 GHz Current-Reuse CMOS LNA. *IEEE Journal of Solid-State Circuits* 51, 574–586. doi: 10.1109/jssc.2015.2504413

Prejbeanu, I. L., Bandiera, S., Alvarez-Hérault, J., Sousa, R. C., Dieny, B., and Nozières, J.-P. (2013). Thermally assisted MRAMs: ultimate scalability and logic functionalities. *Journal of physics D: Applied Physics*. 46, 074002. doi: 10.1088/0022-3727/46/7/074002

Qayyum, S., and Negra, R. (2017). "0.8 mW, 0.1-110 GHz RF power detector with 6 GHz video bandwidth for multigigabit software defined radios," in *Proceedings of the IEEE MTT-S International Microwave Symposium (IMS)*, (Honolulu, HI).

Rippard, W., Pufall, M., and Kos, A. (2013). Time required to injection-lock spin torque nanoscale oscillators. *Applied Physics Letters* 103, 182403. doi: 10.1063/1.4821179

Rippard, W. H., Pufall, M. R., Kaka, S., Silva, T. J., and Russek, S. E. (2005). Injection Locking and Phase Control of Spin Transfer Nano oscillators. *Physical Review Letters* 95, 067203.

Safranski, C., Barsukov, I., Lee, H. K., Schneider, T., Jara, A. A., Smith, A., et al. (2017). Spin caloritronic nano-oscillator. *Nature communications* 8, 117. doi: 10.1038/s41467-017-00184-5

Sankey, J. C., Krivorotov, I. N., Kiselev, S. I., Braganca, P. M., Emley, N. C., Buhrman, R. A., et al. (2005). Mechanisms limiting the coherence time of spontaneous magnetic oscillations driven by dc spin-polarized currents. *Physical review B*. 72, 224427.

Sengupta, A., Panda, P., Wijesinghe, P., Kim, Y., and Roy, K. (2016). Magnetic Tunnel Junction Mimics Stochastic Cortical Spiking Neurons. *Nature Scientific Reports* 6, 30039. doi: 10.1038/srep30039

Slavin, A., Tyberkevych, V., Tabor, P., and Urazhdin, S. (2010). *Phase locking of a spin-troque nano-oscillator to a strong microwave magnetic field*. College Park, MD: APS.

SpiNNaker (2018). Available at: http://apt.cs.manchester.ac.uk/projects/SpiNNaker/ (accessed October 1, 2018).

Strachan, J. P., Torrezan, A. C., Riberio, G. M., and Williams, R. S. (2011). Measuring the switching dynamics andenergy efficiency of tantalum oxide memristors. *Nanotechnology* 22, 505402. doi: 10.1088/0957-4484/22/50/505402

Takeuchi, Y., Sato, H., Fukami, S., Matsukura, F., and Ohno, H. (2015). Temperature dependence of energy barrier in CoFeB-MgO magnetic tunnel junctions with perpendicular easy axis. *Applied Physics Letters* 107, 152405.

Tarequzzaman, M., Böhnert, T., Decker, M., Costa, J. D., Borme, J., and Lacoste, B. (2019). Spin torque nano-oscillator driven by combined spin injection from tunneling and spin Hall current. *Nature Communications Physics* 2, 20.

Teixeira, J. M., Ventura, J., Araujo, J. P., Sousa, J. B., Fernández-García, M. P., Wisniowski, P., et al. (2010). Evidence of spin-polarized direct elastic tunneling and onset of superparamagnetism in MgO magnetic tunnel junctions. *Physical review B* 81, 134423.

Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., et al. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature Letters.* 547, 428–431. doi: 10.1038/nature23011

U. S. Department of energy (2015). *Neuromorphic Computing: From Materials to Systems Architecture*. Washington, DC: U. S. Department of energy.

Walowski, J. (2012). *Physics of laser heated ferromagnets: Ultrafast demagnetization and magneto-Seebeck effect, PhD. Thesis*. Göttingen: Niederschsische Staats-und Universittsbibliothek.

Walter, M., Walowski, J., Zbarsky, V., Münzenberg, M., Schäfers, M., and Ebke, D. (2011). Seebeck effect in magnetic tunnel junctions. *Nature materials.* 10, 742–746. doi: 10.1038/nmat3076

Yogendra, K., Fan, D., Jung, B., and Roy, K. (2016). "Magnetic Pattern Recognition Using Injection-Locked Spin-Torque Nano-Oscillators," in *Proceedings of the IEEE Transactions On Electron Devices*, (Piscataway, NJ).

Yogendra, K., Fan, D., and Roy, K. (2015). "Coupled Spin Torque Nano Oscillators for Low Power Neural Computation," in *Proceedings of the IEEE Transactions on Magnetics* (Piscataway, NJ).

Yogendra, K., Liyanagedera, C., Fan, D., Shim, Y., and Roy, K. (2017). Coupled spin-torque nano-oscillator-based computation: a simulation study. *ACM journal on emerging technologies in computing systems.* 13, 56.

Zhou, Z., Yin, B., and Michel, J. (2015). On-chip light sources for silicon photonics. *Nature Light: Science & Applications* 4, e358. doi: 10.1038/ncomms15535

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Farkhani, Böhnert, Tarequzzaman, Costa, Jenkins, Ferreira, Madsen and Moradi. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 26 February 2020
doi: 10.3389/fnins.2020.00094

On-Chip TaO_x-Based Non-volatile Resistive Memory for *in vitro* Neurointerfaces

Maksim Zhuk[1], Sergei Zarubin[1], Igor Karateev[2], Yury Matveyev[3], Evgeny Gornev[4], Gennady Krasnikov[4], Dmitiry Negrov[5] and Andrei Zenkevich[1]**

[1] Laboratory of Functional Materials and Devices for Nanoelectronics, Moscow Institute of Physics and Technology, Moscow, Russia, [2] National Research Center, Kurchatov Institute, Moscow, Russia, [3] Deutsches Elektronen-Synchrotron, Hamburg, Germany, [4] Molecular Electronics Research Institute (MERI), Moscow, Russia, [5] Laboratory of Neurocomputing Systems, Moscow Institute of Physics and Technology, Moscow, Russia

The development of highly integrated electrophysiological devices working in direct contact with living neuron tissue opens new exciting prospects in the fields of neurophysiology and medicine, but imposes tight requirements on the power dissipated by electronics. *On-chip* preprocessing of neuronal signals can substantially decrease the power dissipated by external data interfaces, and the addition of embedded non-volatile memory would significantly improve the performance of a co-processor in real-time processing of the incoming information stream from the neuron tissue. Here, we evaluate the parameters of TaO_x-based resistive switching (RS) memory devices produced by magnetron sputtering technique and integrated with the 180-nm CMOS field-effect transistors as possible candidates for on-chip memory in the hybrid neurointerface under development. The electrical parameters of the optimized one-transistor–one-resistor (1T-1R) devices, such as the switching voltage (approx. ±1 V), uniformity of the R_{off}/R_{on} ratio (∼10), read/write speed (<40 ns), and the number of the writing cycles (up to 10^{10}), are satisfactory. The energy values for writing and reading out a bit ∼30 and ∼0.1 pJ, respectively, are also suitable for the desired *in vitro* neurointerfaces, but are still far too high once the prospective *in vivo* applications are considered. Challenges arising in the course of the prospective fabrication of the proposed TaO_x-based RS devices in the back-end-of-line process are identified.

Keywords: neural tissue, *in vitro* neurointerfaces, high-density microelectrode arrays, non-volatile memory, resistive switching, tantalum oxide, 1T-1R device, back-end-of-line process

OPEN ACCESS

Edited by:
Erika Covi,
Politecnico di Milano, Italy

Reviewed by:
Zhongrui Wang,
University of Massachusetts Amherst, United States
Jean-Michel Portal,
Aix-Marseille Université, France

****Correspondence:***
Dmitiry Negrov
negrov.dv@mipt.ru
Andrei Zenkevich
zenkevich.av@mipt.ru

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *08 November 2019*
Accepted: *23 January 2020*
Published: *26 February 2020*

Citation:
Zhuk M, Zarubin S, Karateev I, Matveyev Y, Gornev E, Krasnikov G, Negrov D and Zenkevich A (2020) On-Chip TaO_x-Based Non-volatile Resistive Memory for in vitro Neurointerfaces. Front. Neurosci. 14:94. doi: 10.3389/fnins.2020.00094

INTRODUCTION

Modern electrophysiological techniques provide us with the versatile tools to study the inner workings of living neuronal circuits and open an opportunity to control them at the finest level. These methods, ranging from patch clamp to high-density microelectrode arrays, are of tremendous use in single-cell, neuronal culture, and brain studies (Bonifazi and Fromherz, 2002; Eversmann et al., 2011; Eickenscheidt et al., 2012; Szostak et al., 2017; Luan et al., 2018). Moreover, the advances in electrophysiology and neuroscience provide the possibility to implement novel medical devices, such as neuroprosthetics and brain–computer interfaces. In turn, the technological

development of microelectronics and microfabrication have made it possible to implement tiny devices that can simultaneously receive the data from tens of thousands of channels (Frey et al., 2010; Massobrio et al., 2015). When combined with modern data processing techniques, such as spike sorting algorithms, these capabilities can be used to handle the data describing the real-time behavior of thousands of neurons in living tissue.

Meanwhile, advanced neuron stimulation techniques, such as optogenetics, have emerged (Goncalves et al., 2017). Using such tools, one can create precise high-bandwidth bidirectional interfaces to neuronal tissue, which is of interest not only to fundamental neurophysiological studies but for pharmacology and medicine as well.

Direct contact with living tissue imposes tight requirements on the power dissipated by electronics. On the other hand, highly integrated electrophysiological devices provide tremendous amount of data, and in all cases, except for simple recording experiments, these data should be processed in real time. In this respect, off-loading raw data to remote processing equipment is not the best solution since, in high-density systems, the data acquisition rate can reach several gigabytes per second, and the data transfer circuitry will itself use a substantial amount of power. Moreover, this transfer and processing will introduce additional delays and, particularly in the case of medical applications, it is inconvenient to route such wide high-speed interfaces to the external processing devices. Today, it is possible to make processing devices with power requirements of less than 20 mW/GOPS (Reuther et al., 2019) and this paves the way to preprocess *on-chip* neuronal signals from several tens of thousands of channels. Considering a typical event rate of 1,000 events per second, observed by a probe of comparable area (Juavinett et al., 2019) and estimating the amount of computations required to classify a single event as 10^6 operations, one can detect and classify neuronal spikes in less than a 100-mW power envelope, which is much less than it would be required to transmit the raw data out (around 300–400 mW) for serial link with the required bandwidth (Hsiao et al., 2006).

Nevertheless, even using on-chip processing of the raw data, it is still desirable to lower the power consumption of the data processing circuitry further down. One way to achieve this is to replace the static memories used by data processors with some kind of the emerging non-volatile memory. Since static memories are responsible for a substantial fraction of the dissipated power, this approach looks appealing.

The above-mentioned approaches are currently being investigated, ultimately aiming at the development of a hybrid neurointerface for bidirectional communication with the living neuronal tissue in real time. The schematic diagram of such prospective neurointerface is shown in **Figure 1**. As pointed out above, the idea is to reduce the external input/output data rate and to enable online processing of neuronal activity. A substantial data rate reduction can be achieved by processing raw voltage waveforms to extract spiking activity from neurons in contact with the interface electrode array. Such procedure requires the application of data clustering algorithms, known as spike sorting, which work by matching raw data to pre-extracted encoded patterns and adapting to changes online. Although these patterns change slowly following the changes in the neuronal tissue as well as the electrical drift of electrodes, access to them is constantly required, and every extracted spike demands quite a large exchange with the memory. The detailed architecture of the neurointerface will be described elsewhere. The current work presumes that, potentially, the density of resistive random access memory (ReRAM) can be substantially higher than those of static RAM (SRAM). In addition, suggesting that the spiking activity of neurons occurs irregularly, the corresponding access to the memory is relatively rare, and since ReRAM in the retention mode does not require any (static) energy consumption, it would eventually be beneficial compared to SRAM. Therefore, here, we shall consider resistive switching (RS) non-volatile memory arrays, which can be integrated into the neurointerface chip. RS non-volatile memory has been previously used for processing-in-memory, particularly to simulate spiking networks (Pantazi et al., 2016; Wang et al., 2018), to accelerate vector–matrix multiplications (Prezioso et al., 2015) or to discriminate the recorded neuronal spiking events from the background activity and perform data compression of signals recorded by a multi-electrode array (Gupta et al., 2016).

Among the different non-volatile memory concepts, the one exploiting the reversible RS effect in thin films of transition metal oxides is a viable candidate (Rohde et al., 2005; Yang et al., 2010; Wong et al., 2012). The advantages of this kind of RS devices, once they are carefully optimized in terms of combination of materials, thickness of the functional layer, and switching pulses parameters, are as follows: good scalability (Govoreanu et al., 2011; Park et al., 2012), low power consumption (Goncalves et al., 2017), relatively high read/write speed (Lee et al., 2010, 2011), large number of writing cycles (Rohde et al., 2005; Kim et al., 2011), and, most importantly for the emerging neurointerfacing applications, the possibility to fabricate memory devices in the back-end-of-line (BEOL) process (Kim et al., 2012; Park et al., 2013; Goux et al., 2014; Li et al., 2018) of modern high-density mixed-signal fabrication flow. Dozens of papers have been published previously, describing the ReRAM devices employing mainly TiO_2 (Frey et al., 2010; Jeong et al., 2009), HfO_2 (Yang et al., 2010; Goncalves et al., 2017), and Ta_2O_5 (Rohde et al., 2005; Wedig et al., 2015) as functional layers. Over the last decade, very promising parameters have been demonstrated in terms of memory window, uniformity, endurance, and retention in RS devices integrated with the CMOS process, and ReRAM has been eventually successfully commercialized (see e.g. Fujitsu ReRAM memory data sheet)[1]. However, the variability of the electrical parameters for different transition metal oxide-based RS devices on the chip and from one switching cycle to another is still an issue, which is attributed to the inherent stochastic nature of the switching process (Fantini et al., 2014; Kim et al., 2014b). Also, the fact that such memory devices have been implemented does not necessarily imply they can be easily integrated into hybrid neurointerfaces under development. In particular, high-density embedded memory arrays should

[1]https://www.fujitsu.com/global/documents/products/devices/semiconductor/memory/reram/MB85AS4MT-DS501-00045-1v0-E.pdf

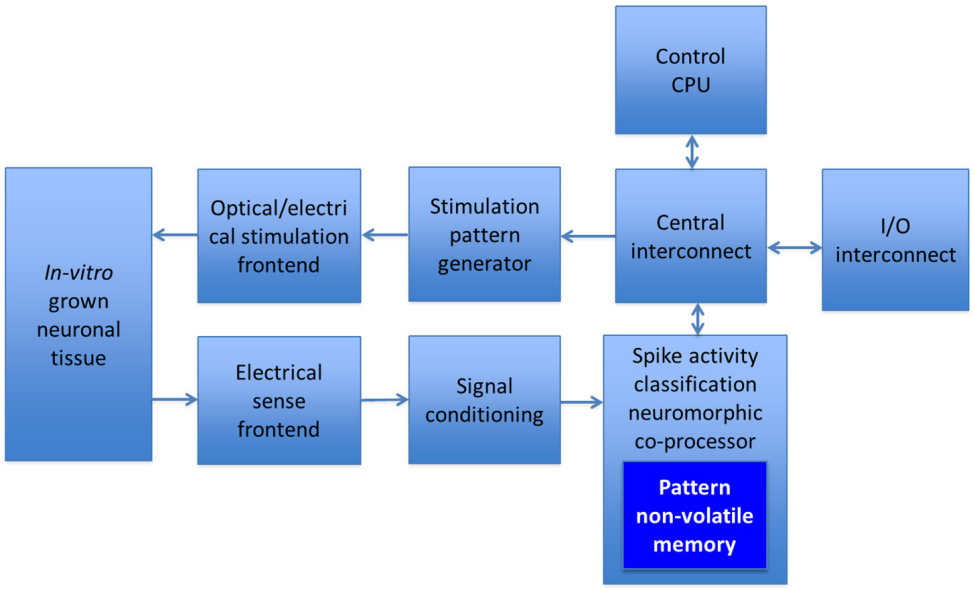


FIGURE 1 | Schematic diagram of a high-density *in vitro* hybrid neurointerface for bidirectional communication with neuronal culture.

have RS devices placed in the lowest possible metallization layers to increase the density and to lower routing congestion. Moreover, this approach decreases the parasitic capacitances of memory lines, thus lowering the overall energy consumption. Such placement requires the stability of the RS device parameters upon subsequent processing steps, which are performed at temperatures up to ∼400°C as part of the standard fabrication technology (Walczyk et al., 2011).

Tantalum oxide is a popular functional layer used to devise resistive memory devices, and indeed, there have been a large number of published papers describing the operation of RS devices employing TaO_x and different electrodes, including both 1-bit (Wong et al., 2012) and multi-bit ("analog") (Kim et al., 2014a) switching behavior. The functional properties of transition metal oxide-based RS devices integrated with CMOS transistors (so-called one transistor–one resistor, 1T-1R, memory device) are well documented either (Lee et al., 2010, 2011). Nevertheless, once the goal is to fabricate TaO_x-based 1T-1R RS devices to be used as built-in memory arrays for neurointerface applications, careful optimization of their parameters is needed so that they could fit the requirements, such as uniform switching voltage (in the range ±1–2 V), low energy consumption (∼10/0.1 pJ per write/read operation), modest retention time (several days), and high endurance ($>10^7$ writing cycles).

In this work, we describe the implementation of TaO_x-based resistive switching devices and their integration with the matrices of 180-nm CMOS transistors, ultimately aiming at the development of on-chip non-volatile memory arrays. Such memory can be used for the temporary storage of the data from the co-processor integrated on the bidirectional neurointerface chip and processing the information from the neuronal tissue in real time.

MATERIALS AND METHODS

Pt bottom electrode was deposited by magnetron sputtering. In order to form the metal–insulator–metal functional structure, windows ∼5 μm × 5 μm in size were first formed by dry plasma etching in a SiO_x layer grown plasma-enhanced chemical vapor deposition technique (the schematic is shown in **Figure 2A**). The TaO_x layer, 5–20 nm in thickness, was deposited by direct current reactive magnetron sputtering of pure metal Ta target in pure O_2. The top electrode (TE) Ta thin film with precise thickness in the nanometer range was further deposited in the same vacuum cycle from the same Ta target sputtered in an Ar atmosphere. To study the effect of the top electrode on the electrical properties of TaO_x-based RS devices, alternative TEs, such as W, TiN, Ag, and Al, were also deposited (see **Supplementary Table S1**). The TE was capped with a thick W film in the same vacuum cycle to ensure the conductivity across the electrode area and protect the active Ta layer from oxidation.

Hard X-ray photoemission spectroscopy (HAXPES) analysis of as-grown TaO_x films was performed at DESY synchrotron (endstation P22) at the excitation X-ray energy of E = 6 keV (an overall energy resolution of about 0.2 eV) with Specs 225-HV analyzer. The photoelectrons at such energies have the inelastic mean free path of ∼7 nm, thus increasing the probing depth up to ∼20–22 nm. Consequently, the relative contribution of the surface components is decreased, and true chemical composition across the bulk of the tantalum oxide layer can be revealed.

Sub-micron 1T-1R RS devices were fabricated by integrating the Pt/TaO_x/Ta structures described above with 180-nm CMOS field-effect transistors in a 1,024 × 1,024 matrix (**Figure 2B**). Combining optical and e-beam lithography patterning, the RS

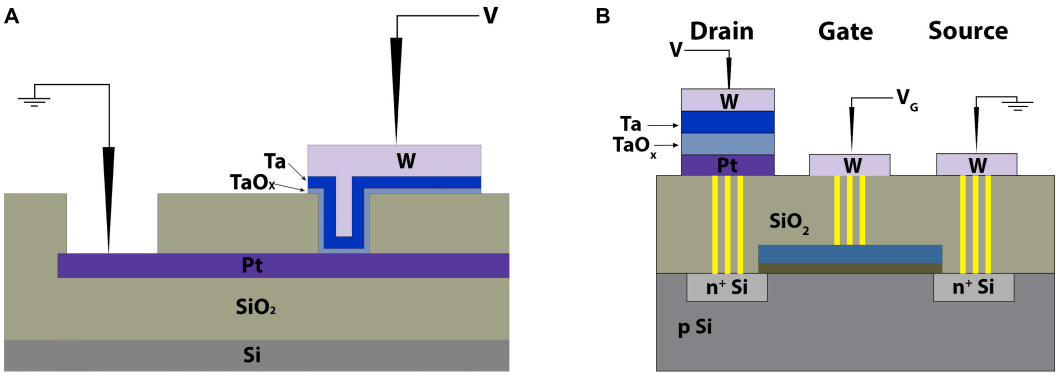


FIGURE 2 | Schematic of panel (**A**) the Pt/TaO_x/Ta 1R device topology with "cross-bar" geometry and (**B**) its integration with the field-effect transistor (FET).

devices were formed on top of remote W contacts to the drain of *n*-channel transistors (see **Supplementary Figure S1a**).

The electrical measurements were performed using a Keysight B1500A semiconductor device parameter analyzer in combination with a Cascade Microtech Summit 11000M probe station. The polarity of the voltage corresponds to the value on the top electrode. The forming voltage was derived from the first *I–V* curve. All endurance tests were performed by switching with square waveform voltage pulses in a vast time width range (40 ns–1 μs). For 1T-1R device characterization, an additional channel was used to control the gate voltage.

Transmission electron microscopy (TEM) study was performed with the S/TEM Titan 80–300 (Thermo Fisher Scientific) microscope equipped with a spherical aberration probe corrector, an energy-dispersive X-ray spectrometer (EDAX), and a high-angle annular dark-field detector (Fischione). The microscope was operated at 300 kV.

RESULTS AND DISCUSSION

The actual elemental composition of a few-nanometer-thick functional TaO_x layer (capped with Al) is O:Ta ~3.1, as revealed by Rutherford backscattering spectrometry (RBS) analysis (see **Supplementary Figure S2**). HAXPES analysis was used to confirm the overall super-stoichiometric elemental composition of the TaO_x layer up to O:Ta ~3.9, as compared to the stoichiometric Ta_2O_5 film grown by atomic layer deposition (spectra shown in **Figure 3**), implying a large excess of O atoms in the as-grown tantalum oxide layer. In addition, HAXPES data revealed two non-equivalent O states in the sputtered TaO_x layer: the lines with BE = 532.3 eV and BE = 533.2 eV, which are attributed to the stoichiometric Ta_2O_5 (equivalent to that grown by ALD), and extra oxygen trapped during the sputtering process, respectively. By taking the relative Ta4d and O1s peak areas and using the corresponding photoeffect cross-sections, we calculated the overall composition to be $Ta_1O_{3.9}$, in reasonable agreement with the RBS results.

The fabricated Ta/TaO_x/Pt RS cells were characterized in quasi-direct current (DC) mode by recording the *I–V* sweeps. Using the compliance current set at $I_c = 10^{-4}$ A, the first switching cycle (called "electroforming") was similar, within 0.5 V, to the subsequent ones, indicating the "forming-free" operation (**Figures 4A,B**). However, the variability of the switching parameters from cycle to cycle evident from the presented *I–V* curves is quite significant.

The thickness of the functional layer was further varied in the range of 3–24 nm in order to minimize the forming voltage while maintaining the maximal number of switching cycles. The data for $U_{form.}$ vs. the thickness of the TaO_x layer are given in **Figure 5**. The number of the switching cycles for the devices with different thicknesses is given in the inset.

Thus, an optimized functional structure was further used to fabricate devices in "cross-bar" geometry (**Supplementary Figure S1b**) for endurance tests with short (<50 ns) pulses. The results of such tests using $U_{on} = 0.8$ V/$U_{off} = -1.0$ V and $t = 40$ ns are presented in **Figure 6**.

The retention test was further conducted for the same RS devices to examine their long-term memory functionality. After the SET process at room temperature, when all devices are in the low-resistance state (LRS), the chip is subjected to heating up to $T = 200$°C. The change of LRS R_{on} value by 10% was chosen as

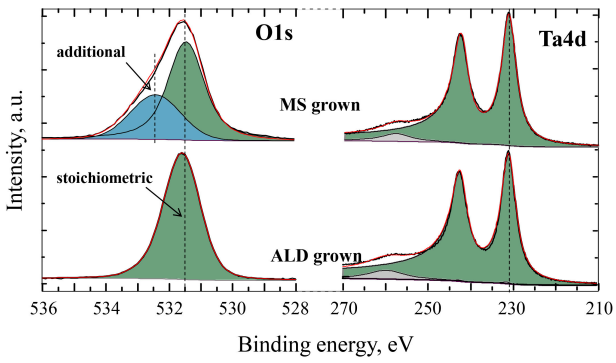


FIGURE 3 | Core-level photoemission spectra of Ta 4d and O 1s lines of the 10-nm-thick TaO_x layer as grown by magnetron sputtering obtained by the HAXPES technique.

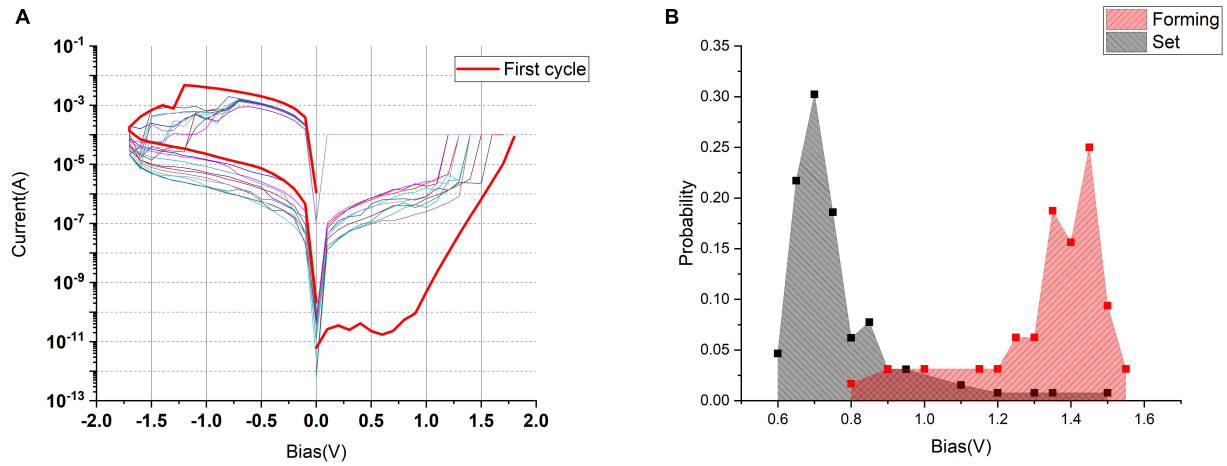


FIGURE 4 | (A) Bipolar RS characteristics of the single Ta/TaO_x/Pt device. **(B)** Normalized distribution of the forming (*red*) and SET (*black*) voltages for the Pt/TaO_x/Ta 1T1R devices.

a criterion for the device unacceptable degradation. According to the Arrhenius plot of the measured data (**Figure 7**), the devices successfully pass the retention time of 10 years at T = 85°C. However, this temperature obviously cannot be reached in our application since on-chip memory matrix basically contacts the living neuron cells. Therefore, the operating temperature should be less than 40°C, which will be ensured by the heat removal in the current version of the *in vitro* chip. Direct simulation of the heating balance in the entire system with realistic contributions has not been carried out so far.

Let us now describe the electrical properties of the 1T-1R RS devices employing the optimized 1R devices described above and the factory 180-nm CMOS transistors. While setting the compliance current by the gate voltage on the transistor at $I_c = 3 \times 10^{-2}$ mA, the average electroforming voltage of the 1T-1R devices in DC mode was less than 1.5 V, with stable I–V form during 100 DC cycles (**Figure 8**). The endurance test was performed by applying 100-ns-long switching pulses of selected memory cells. In order to maintain the best switching uniformity during the endurance test, the voltage pulse parameters were chosen to provide $R_{on}/R_{off} \sim 10$ and were set as $U = +1.7$ V/−2.1 V, t = 100 ns. The fabricated sub-micrometer 1T-1R devices survive more than 10^{10} switching cycles without any signs of degradation (**Figure 9**).

The use of 1T-1R built-in memory devices prepared in the BEOL process implies that they are fabricated early BEOL flow

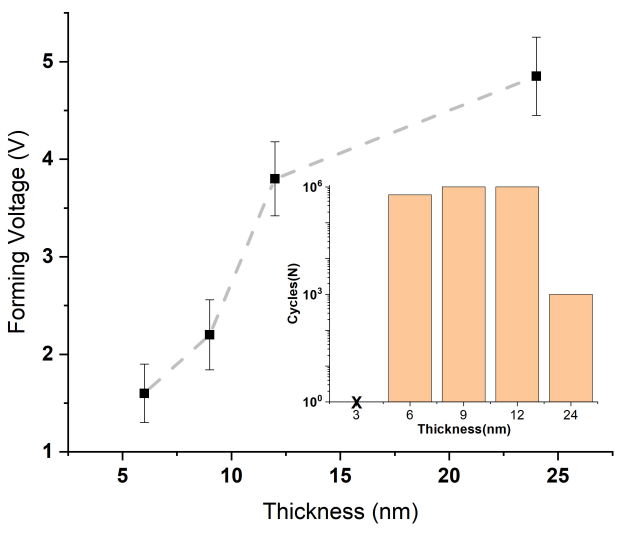


FIGURE 5 | Plot of the electroforming voltage vs. the thickness of the functional TaO_x layer in the Ta/TaO_x/Pt RS 1R devices (*inset*: number of switching cycles for different TaO_x layer thicknesses).

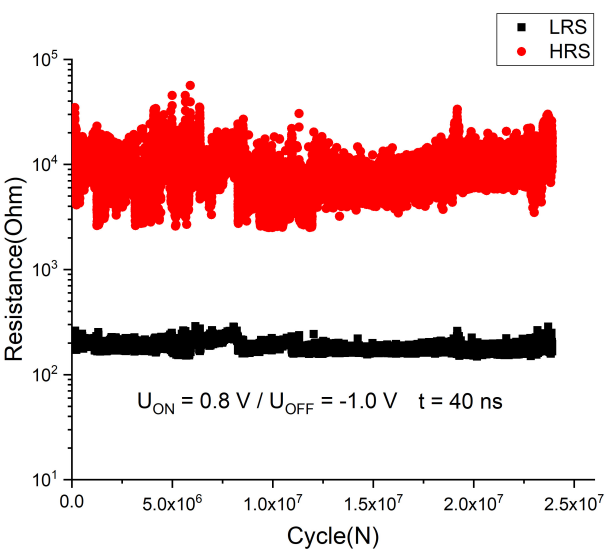


FIGURE 6 | Endurance test of the "cross-bar" Ta/TaO_x/Pt devices, switched by U_{switch}. ~ ± 1 V and t = 40 ns pulses.

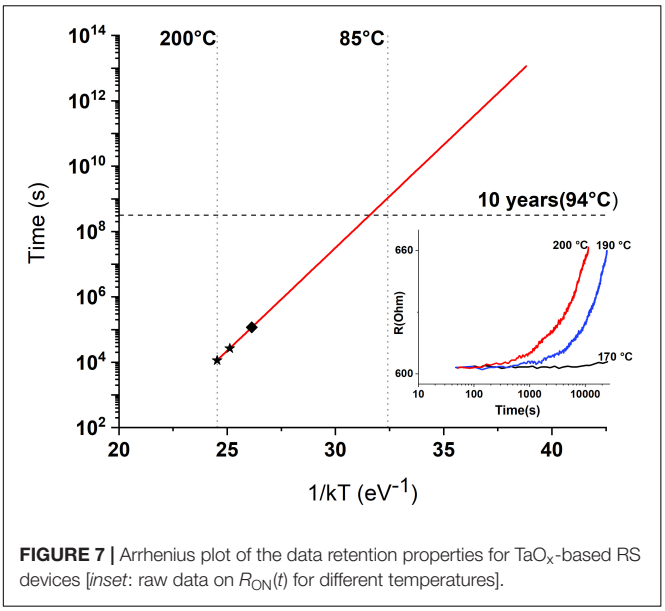


FIGURE 7 | Arrhenius plot of the data retention properties for TaO_x-based RS devices [*inset*: raw data on $R_{ON}(t)$ for different temperatures].

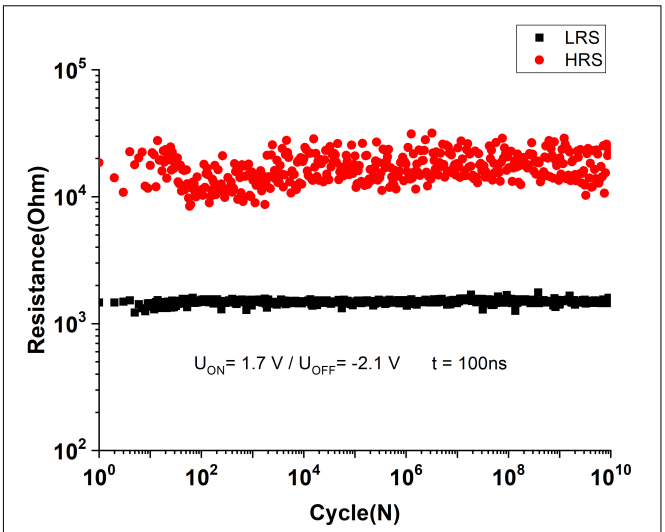


FIGURE 9 | Number of writing cycles for the sub-micrometer Ta/TaO_x/Pt crossbar RS devices.

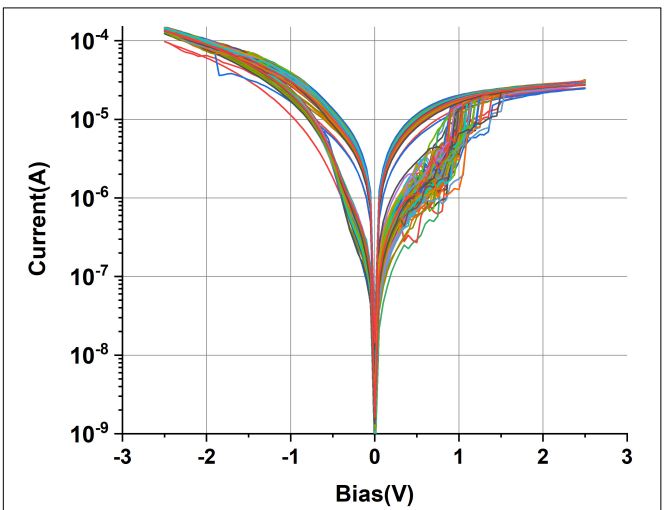


FIGURE 8 | Sequential DC *I–V* curves of 1T-1R devices based on the Ta/TaO_x/Pt stack.

in a second–third metallization layer and should maintain their characteristics following the subsequent annealing at $T = 400°C$, which is part of the upper Al metallization layers technology. In order to investigate the compatibility of the fabricated 1T-1R memory devices with the 180-nm CMOS technology, they were post-annealed in vacuum (10^{-6} Torr) at $T = 400°C$ for ∼30 min. It was found that, upon such annealing step, both pristine and electroformed devices have degraded, yielding very high leakage current and no resistive switching (**Supplementary Figure S3**). In order to investigate the degradation mechanism to possibly improve the thermal stability of the device parameters, we have used transmission electron microscopy analysis of the device stack cross-section before and after annealing. The images shown in **Figure 10** indicate that the crystalline structure of the Ta layer has vanished upon annealing, while the relative thickness of the Ta vs. TaO_x layers has changed. This suggestion is confirmed by comparing fast Fourier transform (FFT) pictures of the Ta layer before and after annealing, shown in the insets in **Figures 10A,C**. The oxygen concentration profile across the stack obtained using energy-dispersive X-ray (EDX) analysis with a sub-nanometer exciting electron beam reveals the redistribution of oxygen atoms in the stack (**Figures 10B,D**), implying the redox reaction at the Ta/TaO_x interface. The reduction of tantalum oxide may eventually result in the dramatic decrease of its resistivity, which is the cause of the degradation.

In order to overcome the latter problem of degradation of the 1T-1R device properties during the BEOL process, magnetron sputtering of the tantalum oxide layer was performed at $T = 400°C$ (prior to Ta layer deposition at room temperature). Such step results in the "normalizing" of the stoichiometry of the TaO_x layer to O/Ta ≈ 2.5, as confirmed by RBS analysis (not shown), and excludes further oxidation of the Ta layer on top. Thus, the prepared Ta/TaO_x/Pt-based 1T-1R devices successfully survive the annealing at $T = 400°C$ for 30 min, to yield at least $\sim 10^7$ of the switching cycles (tests are still in progress) (**Supplementary Figure S4**). However, it comes at the price of increasing the electroforming voltage up to $U_{form.} = +2.5$ V (as compared to $U_{form.} = +1.4$ V for non-annealed devices).

In conclusion, on-chip non-volatile memory may significantly improve the performance of the co-processor to be used for real-time processing of the information stream received from the neuron tissue in neurointerfaces under development. Among several candidates, resistive memory (ReRAM) is a viable option. We have evaluated the use of Ta/TaO_x/Pt-based resistive devices produced by magnetron sputtering and integrated with the 180-nm CMOS field-effect transistors as a possible candidate for on-chip memory. While the electrical parameters of the optimized 1T-1R devices, such as switching voltage (approx. ± 1 V), uniformity of the R_{off}/R_{on} ratio (∼10), read and write speed

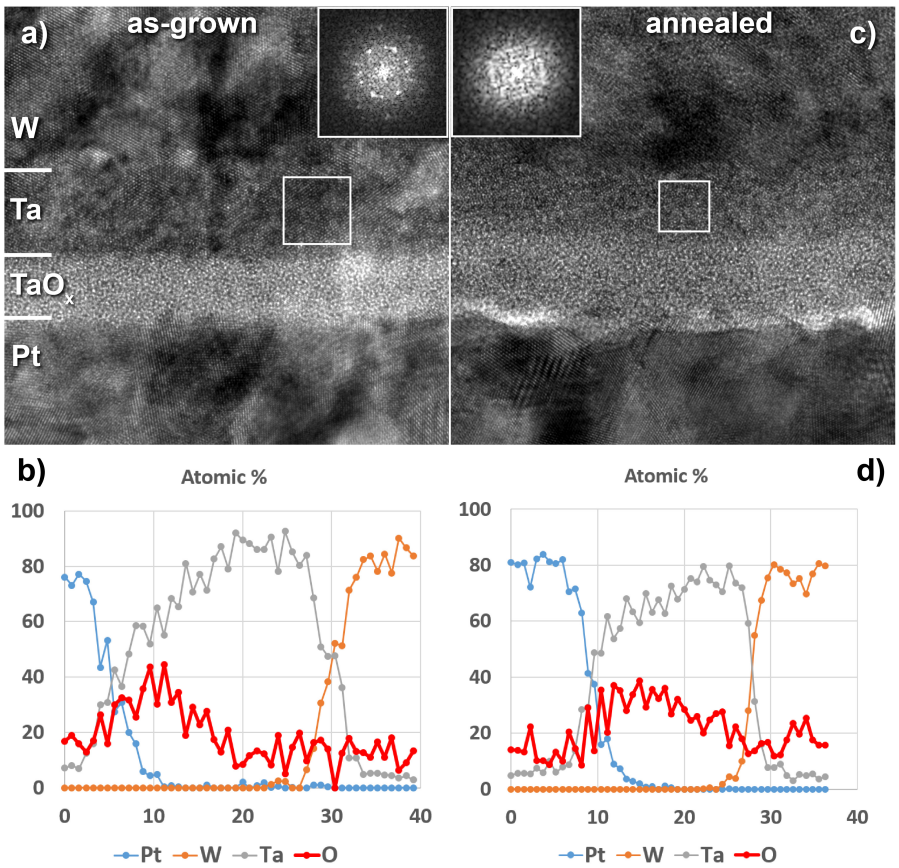


FIGURE 10 | High-resolution TEM images of the Ta/TaO_x/Pt device stack cross-section **(a)** and the same device following the annealing at T = 400°C, 30 min **(c)** (FFT of the Ta metallic layer is shown in the *inset*). **(b,d)** Elemental profiles across the stack as revealed by EDX analysis of the as-prepared and annealed stacks, respectively.

(<40 ns), and the number of the writing cycles ($\sim 10^{10}$), are encouraging, there are still challenges to overcome. In particular, the energy per write operation is ~30 pJ, which is still much too high for use *in vivo* applications, where the power consumption and heat dissipation are critical constraints. Also, the perspective to fabricate memory on-chip in the BEOL process implies the metallization annealing steps (at T = 400°C), which affects the operation of the fabricated devices. Further work is necessary to optimize the device stack and fabrication technology to enable TaO_x-based non-volatile memory matrices to be used in hybrid neurointerfaces under development.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/**Supplementary Material**.

AUTHOR CONTRIBUTIONS

MZ fabricated devices and performed electrical characterization and wrote the draft of the manuscript. SZ prepared samples and performed preliminary TEM analysis. IK analyzed with HR TEM/EDX the composition/structure of RS devices. YM analyzed with HAXPES the chemical composition of TaOx layers. EG and GK provided wafers with CMOS transistors pre-fabricated for further integraion of memory devices under investigation. DN motivated the work, analyzed the data and partially wrote the manuscript. AZ posed the problem, directed the experiments and contributed to the writing of the manuscript.

FUNDING

This work was performed using equipment of MIPT Shared Facilities Center with the financial support from the National Technology Initiative Program "Artificial Intelligence" (Russia).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.00094/full#supplementary-material

REFERENCES

Bonifazi, P., and Fromherz, P. (2002). Silicon chip for electronic communication between nerve cells by non-invasive interfacing and analog–digital processing. *Adv. Mater.* 14, 1190–1193. doi: 10.1002/1521-4095

Eickenscheidt, M., Jenkner, M., Thewes, R., Fromherz, P., and Zeck, G. (2012). Electrical stimulation of retinal neurons in epiretinal and sub retinal configuration using a multi capacitor array. *J. Neurophysiol.* 107, 2742–2755. doi: 10.1152/jn.00909.2011

Eversmann, B., Lambacher, A., Gerling, T., Kunze, A., Fromherz, P., and Thewes, R. (2011). "A neural tissue interfacing chip for in-vitro applications with 32k recording / stimulation channels on an active area of 2.6 mm2," in *Proceedings of the European Solid-State Circuits Conference*, Piscataway, NJ.

Fantini, A., Goux, L., Redolfi, A., Degraeve, R., Kar, G., Chen, Y. Y., et al. (2014). "Lateral and vertical scaling impact on statistical performances and reliability of 10nm TiN/Hf(Al)O/Hf/TiN RRAM devices," in *Proceedings of the 2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*, Honolulu, HI.

Frey, U., Sedivy, J., Heer, F., Pedron, R., Ballini, M., Mueller, J., et al. (2010). Switch-matrix-based high-density microelectrode array in CMOS technology. *IEEE J. Solid State Circ.* 45, 467–482. doi: 10.1109/JSSC.2009.2035196

Goncalves, S. B., Ribeiro, J. F., Silva, A. F., Costa, R. M., and Correia, J. H. (2017). Design and manufacturing challenges of optogenetic neural interfaces: a review. *J. Neural Eng.* 14:041001. doi: 10.1088/1741-2552/aa7004

Goux, L., Fantini, A., Redolfi, A., Chen, C. Y., Shi, F. F., Degraeve, R., et al. (2014). "Role of the Ta scavenger electrode in the excellent switching control and reliability of a scalable low-current operated TiN\Ta2O5\Ta RRAM device," in *Proceedings of the 2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*, Honolulu, HI.

Govoreanu, B., Kar, G. S., Chen, Y., Paraschiv, V., Kubicek, S., Fantini, A., et al. (2011). "10×10nm2 Hf/HfOx crossbar resistive RAM with excellent performance, reliability and low-energy operation," in *Proceedings of the 2011 International Electron Devices Meeting*, Washington, DC.

Gupta, I., Serb, A., Khiat, A., Zeitler, R., Vassanelli, S., and Prodromakis, T. (2016). Real-time encoding and compression of neuronal spikes by metal-oxide memristors. *Nat. Commun.* 7:12805. doi: 10.1038/ncomms12805

Hsiao, C. H., Kao, M. S., Jen, C. H., Hsu, Y. H., Yang, P. L., Chiu, C. T., et al. (2006). "A 3.2 GBIT/S CML transmitter with 20: 1 multiplexer in 0.18 CMOS technology," in *Proceedings of the International Conference Mixed Design of Integrated Circuits and System, 2006. MIXDES 2006*, (Piscataway, NJ: IEEE), 179–183. doi: 10.1109/MIXDES.2006.1706563

Jeong, H. Y., Lee, J. Y., Choi, S.-Y., and Kim, J. W. (2009). Microscopic origin of bipolar resistive switching of nanoscale titanium oxide thin films. *Appl. Phys. Lett.* 95:62108. doi: 10.1063/1.3251784

Juavinett, A. L., Bekheet, G., and Churchland, A. K. (2019). Chronically implanted neuropixels probes enable high-yield recordings in freely moving mice. *eLife* 8:47188. doi: 10.7554/eLife.47188

Kim, K.-H., Gaba, S., Wheeler, D., Cruz-Albrecht, J. M., Hussain, T., Srinivasa, N., et al. (2012). A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. *Nano Lett.* 12, 389–395. doi: 10.1021/nl203687n

Kim, S., Choi, S., Lee, J., and Lu, W. D. (2014a). Tuning resistive switching characteristics of tantalum oxide memristors through Si doping. *ACS Nano* 8, 10262–10269. doi: 10.1021/nn503464q

Kim, S., Choi, S., and Lu, W. (2014b). Comprehensive physical model of dynamic resistive switching in an oxide memristor. *ACS Nano* 8, 2369–2376. doi: 10.1021/nn405827t

Kim, Y., Lee, S. R., Lee, D., Lee, C. B., Chang, M., Hur, J. H., et al. (2011). "Bi-layered RRAM with unlimited endurance and extremely uniform switching," in *Proceedings of the 2011 Symposium on VLSI Technology - Digest of Technical Papers*, Honolulu, HI.

Lee, H. Y., Chen, Y. S., Chen, P. S., Gu, P. Y., Hsu, Y. Y., Wang, S. M., et al. (2010). "Evidence and solution of over-RESET problem for HfOX based resistive memory with sub-ns switching speed and high endurance," in *Proceedings of the 2010 International Electron Devices Meeting*, San Francisco, CA.

Lee, M.-J., Lee, C. B., Lee, D., Lee, S. R., Chang, M., Hur, J. H., et al. (2011). A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta2O5-x/TaO2-x bilayer structures. *Nat. Mater.* 10, 625–630. doi: 10.1038/nmat3070

Li, C., Hu, M., Li, Y., Jiang, H., Ge, N., Montgomery, E., et al. (2018). Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* 1, 52–59. doi: 10.1038/s41928-017-0002-z

Luan, S., Williams, I., Maslik, M., Liu, Y., De Carvalho, F., Jackson, A., et al. (2018). Compact standalone platform for neural recording with real-time spike sorting and data logging. *J. Neural Eng.* 15:046014. doi: 10.1088/1741-2552/aabc23

Massobrio, P., Tessadori, J., Chiappalone, M., and Ghirardi, M. (2015). In vitro studies of neuronal networks and synaptic plasticity in invertebrates and in mammals using multielectrode arrays. *Neural Plast.* 2015:196195. doi: 10.1155/2015/196195

Pantazi, A., Woźniak, S., Tuma, T., and Eleftheriou, E. (2016). All-memristive neuromorphic computing with level-tuned neurons. *Nanotechnology* 27:355205. doi: 10.1088/0957-4484/27/35/355205

Park, S., Kim, H., Choo, M., Noh, J., Sheri, A., Jung, S., et al. (2012). "RRAM-based synapse for neuromorphic system with pattern recognition function," in *Proceedings of the 2012 International Electron Devices Meeting*, San Francisco, CA.

Park, S., Noh, J., Choo, M., Sheri, A. M., Chang, M., Kim, Y.-B., et al. (2013). Nanoscale RRAM-based synaptic electronics: toward a neuromorphic computing device. *Nanotechnology* 24:384009. doi: 10.1088/0957-4484/24/38/384009

Prezioso, M., Merrikh-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 61–64. doi: 10.1038/nature14441

Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., and Kepner, J. (2019). *Survey and benchmarking of machine learning accelerators*. Waltham, MA: IEEE High Performance Extreme Computing Conference (HPEC), 1–9. doi: 10.1109/HPEC.2019.8916327

Rohde, C., Choi, B. J., Jeong, D. S., Choi, S., Zhao, J.-S., and Hwang, C. S. (2005). Identification of a determining parameter for resistive switching of TiO2 thin films. *Appl. Phys. Lett.* 86:262907. doi: 10.1063/1.1968416

Szostak, K. M., Grand, L., and Constandinou, T. G. (2017). Neural interfaces for intracortical recording: requirements, fabrication methods, and characteristics. *Front. Neurosci.* 11:665. doi: 10.3389/fnins.2017.00665

Walczyk, C., Walczyk, D., Schroeder, T., Bertaud, T., Sowinska, M., Lukosius, M., et al. (2011). Impact of temperature on the resistive switching behavior of embedded HfO2-based RRAM devices. *IEEE Trans. Elec. Dev.* 58, 3124–3131. doi: 10.1109/TED.2011.2160265

Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., et al. (2018). Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* 1, 137–145. doi: 10.1038/s41928-018-0023-2

Wedig, A., Luebben, M., Cho, D.-Y., Moors, M., Skaja, K., Rana, V., et al. (2015). Nanoscale cation motion in TaOx, HfOx and TiOx memristive systems. *Nat. Nanotechnol.* 11:67. doi: 10.1038/nnano.2015.221

Wong, H. P., Lee, H., Yu, S., Chen, Y., Wu, Y., Chen, P., et al. (2012). Metal–Oxide RRAM. *Proc. IEEE* 100, 1951–1970. doi: 10.1109/JPROC.2012.2190369

Yang, J. J., Zhang, M.-X., Strachan, J. P., Miao, F., Pickett, M. D., Kelley, R. D., et al. (2010). High switching endurance in TaOx memristive devices. *Appl. Phys. Lett.* 97:232102. doi: 10.1063/1.3524521

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Zhuk, Zarubin, Karateev, Matveyev, Gornev, Krasnikov, Negrov and Zenkevich. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 26 February 2020
doi: 10.3389/fnins.2020.00088

Spatial Properties of STDP in a Self-Learning Spiking Neural Network Enable Controlling a Mobile Robot

Sergey A. Lobov[1,2], Alexey N. Mikhaylov[1], Maxim Shamshin[1], Valeri A. Makarov[1,3] and Victor B. Kazantsev[1,2]*

[1] Neurotechnology Department, Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia, [2] Neuroscience and Cognitive Technology Laboratory, Center for Technologies in Robotics and Mechatronics Components, Innopolis University, Innopolis, Russia, [3] Instituto de Matemática Interdisciplinar, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid, Madrid, Spain

Development of spiking neural networks (SNNs) controlling mobile robots is one of the modern challenges in computational neuroscience and artificial intelligence. Such networks, being replicas of biological ones, are expected to have a higher computational potential than traditional artificial neural networks (ANNs). The critical problem is in the design of robust learning algorithms aimed at building a "living computer" based on SNNs. Here, we propose a simple SNN equipped with a Hebbian rule in the form of spike-timing-dependent plasticity (STDP). The SNN implements associative learning by exploiting the spatial properties of STDP. We show that a LEGO robot controlled by the SNN can exhibit classical and operant conditioning. Competition of spike-conducting pathways in the SNN plays a fundamental role in establishing associations of neural connections. It replaces the irrelevant associations by new ones in response to a change in stimuli. Thus, the robot gets the ability to relearn when the environment changes. The proposed SNN and the stimulation protocol can be further enhanced and tested in developing neuronal cultures, and also admit the use of memristive devices for hardware implementation.

Keywords: spiking neural networks, spike-timing-dependent plasticity, learning, neurorobotics, neuroanimat, synaptic competition, neural competition, memristive devices

OPEN ACCESS

Edited by:
Stefano Brivio, Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Junxiu Liu, Ulster University, United Kingdom
Valerio Milo, Politecnico di Milano, Italy

****Correspondence:***
Sergey A. Lobov
lobov@neuro.nnov.ru

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *13 August 2019*
Accepted: *22 January 2020*
Published: *26 February 2020*

Citation:
Lobov SA, Mikhaylov AN, Shamshin M, Makarov VA and Kazantsev VB (2020) Spatial Properties of STDP in a Self-Learning Spiking Neural Network Enable Controlling a Mobile Robot. Front. Neurosci. 14:88. doi: 10.3389/fnins.2020.00088

INTRODUCTION

The adoption of brain-inspired spiking neural networks (SNNs) constitutes a relatively novel paradigm in neural computations with high potential, yet not fully discovered. One of the most intriguing and promising experimental illustrations of SNNs was the development of robots controlled by biological neurons, the so-called neuroanimates, proposed at the end of the XX century and currently attracting much attention (Meyer and Wilson, 1991; Potter et al., 1997; Reger et al., 2000; Izhikevich, 2002; Pamies et al., 2014; Dauth et al., 2016). In those experiments, neural networks self-organized in dissociated neuronal cultures, which was suggested to be used as a decision-making element in robotic systems. In the earlier 1990s, Meyer and Wilson introduced the term an animat, as a composition of words "animal" and "automat," referring to a robot exhibiting the behavior of an animal (Meyer and Wilson, 1991). Later, several research groups developed

prototypes of hybrid systems composed of a robot controlled by a living neural network. The main idea was to achieve adaptive learning in biological SNNs with a real physical embodiment.

Learning is inevitably linked with the interaction of an agent with its environment. Therefore, to implement learning *in vitro*, a neural network should be equipped with a "body" interacting with the environment. The first neuroanimat was proposed by Mussa-Ivaldi's group (Reger et al., 2000). To control a tiny wheeled robot Khepera, they used electric potentials recorded from brain slices of the sea lamprey fed by signals from light sensors. Almost in parallel with this study, Potter et al. (1997) suggested connecting a neuronal culture grown on a multielectrode array (MEA) to animate a roving robot (DeMarse et al., 2001). They succeeded in constructing a virtual neuroanimat capable of moving in the desired direction within 60° corridor after 2 h of "training" with a success rate of 80% (Bakkum et al., 2008). Shahaf et al. (2008) used ultrasonic sensors detecting the presence of an obstacle in the trajectory of a neuroanimat by stimulating a neuronal culture, which, in turn, controlled the movement. Obstacles located on the right or left side provoked population bursts with different spiking signatures. Then, a computer algorithm detected and classified the population bursts and moved the robot in the corresponding direction.

Despite extensive experimental studies conducted over the last decades, the high computational potential of SNNs has not been really achieved. The main problem faced by the researchers building "living computers" is the absence of robust learning algorithms. Unlike the backpropagation algorithm (Rumelhart et al., 1986) and deep learning approaches (Lecun et al., 1998), which revolutionized artificial neural networks (ANNs), SNNs still lack similar methodology. In a more general context, the learning principles of biological neural networks are not explored up to the level sufficient for designing engineering solutions (Gorban et al., 2019). Several attempts were made to adapt the backpropagation algorithm and its variations to SNNs (Hong et al., 2010; Xu et al., 2013). Within this approach, an ANN is subject to learning, and then the obtained weights are transferred with some limitations to a similar SNN (Esser et al., 2016). However, SNNs trained in such a way usually do not achieve a level of accuracy similar to their ANN counterparts. This can be explained both by the formulation of the recognition problem and by the nature of the tests (Tavanaei et al., 2019).

One of the intriguing brain features is the ability to associative learning. It is based on synaptic plasticity, most likely of a Hebbian type (Hebb, 1949). A classic example of associative learning is Pavlovian conditioning (Pavlov, 1927). Generally, it binds a conditional stimulus (CS) with an unconditional stimulus (US). The US always evokes a response in the nervous system, whereas the CS initially does not. After several presentations of the US and CS together, the nervous system starts responding to the CS alone. Hebbian associative learning can be extremely efficient, given that the neural input dimension is high enough (Gorban et al., 2019; Tyukin et al., 2019). Experimentally, associative learning is often achieved in the form of operant or instrumental conditioning, which is characterized by the presentation of stimuli to an animal depending on its behavior (Pavlov, 1927; Hull, 1943; Dayan and Abbott, 2001).

There are several approaches to implement associative learning in mathematical models. One is to incorporate US and CS events as spiking waves or patches of activity propagating in neural tissue and associate them through a spatiotemporal interaction. Learning underlying such a "spatial computation" can be implemented by using spike-timing-dependent plasticity (STDP) (Gong and van Leeuwen, 2009; Palmer and Gong, 2014). The STDP implements the Hebbian rule. In this case, repeated arrival of presynaptic spikes a few milliseconds before the generation of postsynaptic action potentials leads to potentiation of the synapse, whereas the occurrence of presynaptic spikes after postsynaptic ones provokes synaptic depression (Markram et al., 1997; Bi and Poo, 1998; Sjöström et al., 2001). A different approach to the conditioning paradigm uses reinforcement learning, e.g. on the basis of an eligibility trace and dopamine modulated STDP (Houk et al., 1995; Izhikevich, 2007). Based on this type of plasticity, a robot interacting with humans capable of associating color and touch patterns was recently designed (Chou et al., 2015). However, this approach is quite complicated and was implemented only in model neural networks.

Many attempts to implement learning features in neuroanimats have been made in cultured neural networks grown *in vitro*. The use of synaptic plasticity as a mechanism of reinforcement or control of functional connections was demonstrated only in the case of relatively simple adaptive changes in the network. It has been suggested that the network homogeneity (e.g. unstructured connectivity) precludes the emergence of more complex forms of learning (Pimashkin et al., 2013, 2016). Earlier, we proposed an approach to explain the problems of learning in unstructured neural networks by the competition between different pathways conducting excitation to a neuron or set of neurons (Lobov S. A. et al., 2017; Lobov S. et al., 2017b). Recently, the possibility to structure the network geometry by directing axon growth was demonstrated experimentally (Malishev et al., 2015; Gladkov et al., 2017), which opens a new venue to build network architectures *in vitro*.

In this article, we study how spatial or topological properties of STDP can be used to implement associative learning in small SNNs. We show that the competition of spike-conducting pathways in a network plays an essential role in establishing the association of neural connections. In particular, on the network scale, STDP potentiates the shortest neural pathways and depresses alternative longer pathways. It permits replacing irrelevant associations by new ones in response to changes in the structure of external stimuli. We show that a roving robot controlled by an especially designed SNN can exhibit classical and operant conditioning. Application of the shortest-pathway rule allows the robot to relearn sensory-motor skills by rewiring the SNN on the fly when the environment changes. The developed SNN topology and the stimulation protocol can be adapted further for structured neural network cultured *in vitro* and for designing hardware SNNs based on, e.g. memristive plasticity.

MATERIALS AND METHODS

The SNN Model

To simulate the dynamics of a SNN, we adopt the approach described elsewhere (Lobov S. A. et al., 2017). Briefly, the dynamics of a single neuron is given by Izhikevich (2003):

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I(t), \quad (1)$$

$$\frac{du}{dt} = a(bv - u), \quad (2)$$

where v is the membrane potential, u is the recovery variable, and $I(t)$ is the external driving current. If $v \geq 30$, then $v \leftarrow c$, $u \leftarrow u + d$, which corresponds to generation of a spike. We set $a = 0.02$, $b = 0.2$, $c = -65$, and $d = 8$. Then, the neuron is silent in the absence of the external drive and generates regular spikes under a constant stimulus, which is a typical behavior of cortical neurons (Izhikevich, 2003, 2004). The driving current is given by:

$$I(t) = \xi(t) + I_{syn}(t) + I_{stml}(t), \quad (3)$$

where $\xi(t)$ is an uncorrelated zero-mean white Gaussian noise with variance D, $I_{syn}(t)$ is the synaptic current, and $I_{stml}(t)$ is the external stimulus. As a stimulus, we use a sequence of square electric pulses of the duration of 3 ms delivered at 10 Hz rate, with the amplitude sufficient to excite the neuron.

The synaptic current is the weighted sum of all synaptic inputs to the neuron:

$$I_{syn}(t) = \sum_j g_j w_j(t) y_j(t), \quad (4)$$

where the sum is taken over all presynaptic neurons, w_j is the strength of the synaptic coupling directed from neuron j, g_j is the scaling factor, in this paper we set them equal to 20 or -20 (Lobov S. A. et al., 2017) for excitatory and inhibitory neurons, respectively, and $y_j(t)$ describes the amount of neurotransmitters released by presynaptic neuron j.

To model the neurotransmitters, we use Tsodyks-Markram's model (Tsodyks et al., 1998) that accounts for short-term depression and facilitation. We use this model with the following parameters: the decay constant of postsynaptic currents τ_I = 10 ms, the recovery time from synaptic depression τ_{rec} = 50 ms, the time constant for facilitation τ_{facil} = 1 s.

The dynamics of the synaptic weight w_{ij} of coupling from an excitatory presynaptic neurons j to a postsynaptic neuron i is governed by the STDP with two local variables (Song et al., 2000; Morrison et al., 2008). Assuming that τ_{ij} is the time delay of spike transmission between neurons j and i, a presynaptic spike fired at time t_j and arriving to neuron i at $t_j + \tau_{ij}$ induces a weight decrease proportional to the value of the postsynaptic trace s_i. Similarly, a postsynaptic spike at t_i induces a weight potentiation proportional to the value of the presynaptic trace s_j. The weighting functions obey the multiplicative updating rule (Song et al., 2000; Morrison et al., 2008). Thus, the weight dynamics is given by:

$$\frac{ds_i}{dt} = -\frac{s_i}{\tau_S} + \sum_{t_i} \delta(t - t_i), \quad (5)$$

$$\frac{ds_j}{dt} = -\frac{s_j}{\tau_S} + \sum_{t_j} \delta(t - t_j - \tau_{ij}), \quad (6)$$

$$\frac{dw_{ij}}{dt} = \lambda \left[(1 - w_{ij}) s_j \delta(t - t_i) - \alpha w_{ij} s_i \delta\left(t - t_j - \tau_{ij}\right) \right], \quad (7)$$

where τ_S = 10 ms is the time constant of spiking traces, λ = 0.001 is the learning rate, and α = 5 is the asymmetry parameter.

We implemented the SNN model (see below) as custom software NeuroNet developed in QT C++ environment. For the axonal delays, we used τ_{ij} = 3 ms for parallel connections and τ_{ij} = 4.2 ms for diagonal coupling. The selected delays are proportional to the interneuron distances and thus take into account the network topology. The app supports SNNs with up to 10^4 neurons. On an Intel® Core™ i3 processor, the simulation can be performed in real time for a SNN with tens of neurons.

Mobile Robot and Unconditional Motor Response

We built a robotic platform from a LEGO® NXT Mindstorms® kit. **Figure 1A** shows the mapping of the robot sensors and motors to the sensory- and motoneurons, respectively. NeuroNet software was used to implement SNNs of different types controlling the robot behavior. **Figure 1B** illustrates the simplest SNN providing the robot with unconditional responses to touching events (see below). The software was run on a standalone PC connected to the robot controller through a Bluetooth interface.

The robot is equipped with two touch sensors and two ultrasonic sonars (**Figure 1C**). A sensitive bumper detects touch stimuli (collisions with obstacles) from the left and right side of the robot (**Figure 1B**). When a touch sensor is on, the corresponding sensory neuron (either N3 or N4) is stimulated by a train of pulses delivered at 10 Hz rate (**Figure 1C**, top-left panel). Such stimulation models signal processing in the sensory system of animals. The ultrasonic sonars are located above the bumper and are coupled to sensory neurons N1 and N2 (**Figure 1C**, bottom-left panel). A sonar sensor turns on if the distance to an obstacle is less than 15 cm. Then, the corresponding neuron is stimulated by a train of square pulses delivered at 10 Hz rate.

The SNN controls the robot movements through the activation of motoneurons. Motor neuron N7 produces tonic spiking with the mean frequency F, which is mapped simultaneously to the left and right motors. As a result, the robot moves straightforward with the velocity proportional to F. Neurons N5 and N6 are coupled to the right and left motors, respectively. The amount of

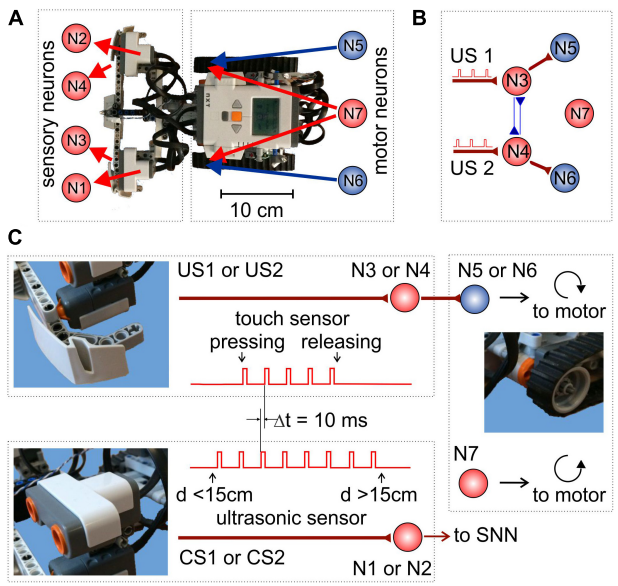


FIGURE 1 | Experimental setup. **(A)** Mapping of the sensory and motoneurons in the mobile LEGO robot. **(B)** Simple SNN controlling basic robot movements and providing unconditional responses to touch stimuli. **(C)** Signaling pathways. Touch (top) and sonar (bottom) sensory neurons receive stimulating trains of rectangular pulses from the corresponding sensors. Then, motoneurons drive the robot's motors.

neurotransmitters released by these neurons modulates the rotation velocity of the corresponding motor. When N5 (N6) fires, the right (left) motor slows down (or even rotates backward if, e.g. $F = 0$), and the robot turns to the right (left).

The robot also has three LEDs facilitating its recognition in the arena by a zenithal video camera. Video frames, captured at 29 Hz rate, were analyzed offline. Trajectory tracking was performed by employing a computer vision algorithm implemented in the OpenCV library. Robot detection is based on the fact that the robot image is a high gradient area. The LEDs turn off when a touch sensor is activated, which allows such events to be detected by analyzing the overall glow of the robot image.

The touch sensors mediate US (**Figures 1B,C**, top). When one of them is activated due to a collision with an obstacle, the corresponding sensory neuron (N3 or N4) starts firing and directly excites a motoneuron (N5 or N6, **Figure 1B**). As a result, the corresponding motor starts rotating backward, and the robot turns away from the obstacle and thus avoids the negative stimulus (**Supplementary Video S1**).

The sonars are connected to sensory neurons N1 and N2 and mediate CS. At the beginning of learning, the CS in the form of an approaching obstacle does not evoke any robot's response. The goal of learning is to associate CS with US to avoid the obstacles in advance without touching them. To provide stimulation of "sensory neurons", according to the STDP protocol, the stimulating pulses from the touch sensors have a 10-ms delay relative to the sonar pulses (**Figure 1C**).

RESULTS

The Shortest Pathway Rule

Let us consider a pair of unidirectionally coupled neurons driven by periodic stimuli applied to one of them (**Figure 2A**). Stimuli excite the first neuron, and then the activation propagates along the "chain" to the second cell, which fires, given that the coupling strength w_{21} is strong enough. Then, the presynaptic spikes precede the postsynaptic ones, and, as a result, the weight increases following the STDP rule (the first term in the right-hand side of Eq. 7). Such a situation can be extended into a chain of three or even more neurons (**Figure 2B**). Thus, STDP increases the corresponding synaptic weights.

However, if we add a new connection from the first neuron to the third one (**Figure 2C**), the weight dynamics changes crucially. Although all synapses are excitatory, the coupling directed from the second to the third neuron is depressed, while the other two are potentiated. This occurs because the axonal delay via the direct way N1–N3 (τ_{31}, **Figure 2C**) is significantly shorter than the delay via the pathway N1–N2–N3 ($\tau_{321} = \tau_{21} + \tau_{32}$, **Figure 2C**). Thus, the first neuron makes fire directly the third one (which is also postsynaptic for w_{32}), and its spikes appear ahead of the spikes coming from the second neuron (presynaptic for w_{32}). Such an inverse sequence (**Figure 2D**) forces depression of the coupling w_{32} according to the STDP rule (the second term in the right-hand side of Eq. 7). We thus can formulate the shortest pathway rule:

- *On the network scale, STDP potentiates the shortest neural pathways and depresses alternative longer pathways.*

SNN Exhibiting Non-trivial Associative Learning

Let us now employ the shortest-pathway rule to implement conditional learning in an SNN. **Figure 3A** shows a simple SNN consisting of four neurons, which can exhibit associative learning. The SNN receives two types of inputs: CS and US applied to neurons N1 and N3, respectively. To comply with the STDP protocol of paired stimulation, we assume that the US pulses arrive with a delay of 10 ms relative to CS pulses (see also **Figure 1C**).

At the beginning, the coupling between N1 and N3, w_{31}, is not sufficient to excite N3 through the CS pathway. However, under stimulation, it is potentiated due to the appropriate delay between US and CS. At the same time, the coupling between N2 and N3, w_{32}, is depressed due to the shortest pathway rule. Thus, after learning, we get the network shown in **Figure 3B** and the CS alone can activate neuron N3 and then the motoneuron. We also note that, similarly, if the CS is applied to N2 instead of N1, then w_{32} will be potentiated, while w_{31} depressed, and we get the same effect of associative learning.

SNN Driving Robot

The above-discussed SNN (**Figure 3**) has one motoneuron and hence can drive one motor channel. To process events coming from the right and left sensors of the robot, we need to extend the SNN to account for two motor channels. Thus, we duplicate

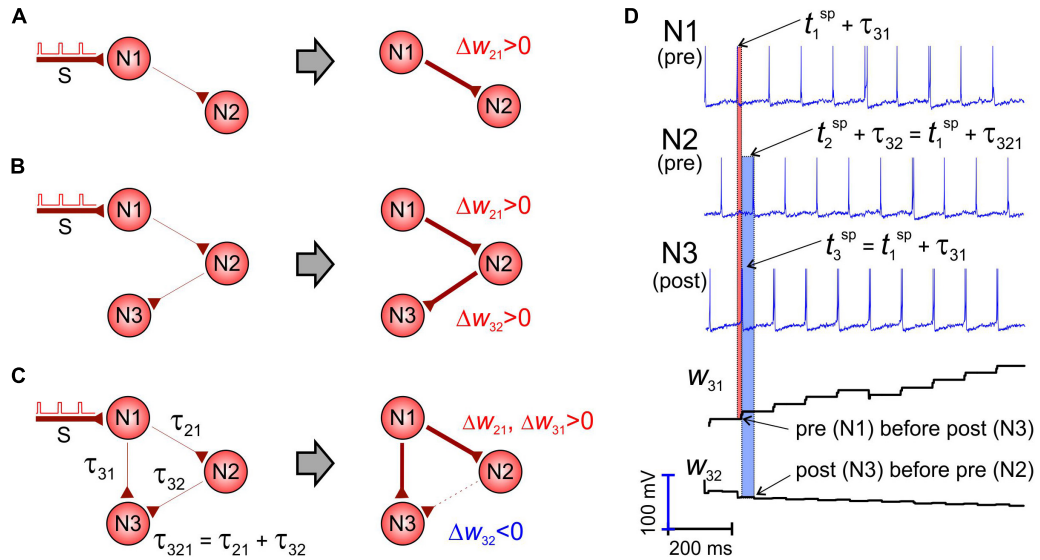


FIGURE 2 | The shortest pathway rule. STDP potentiates the shortest pathways and inhibits alternative connections (W_{ij}, τ_{ij} are the weight and axonal delay of the coupling from neuron *j* to neuron *i*). **(A,B)** *Left*: Initial situation. *Right*: After STDP. The link width corresponds to the synaptic strength. Presynaptic spikes in a unidirectional chain precede postsynaptic spikes and STDP potentiates synaptic couplings. **(C)** The shortcut from neuron N1 to N3 makes the coupling from N2 to N3 "unnecessary" and STDP depresses it. **(D)** Spikes in the network and evolution of synaptic weights.

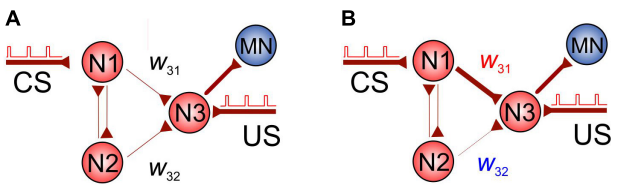


FIGURE 3 | Associative learning based on the spatial properties of STDP. **(A)** The initial SNN. **(B)** Potentiation of the coupling w_{31} and depression of w_{32} during simultaneous stimulation of neuron N3 and N1 (US pulses are applied with a delay of 10 ms relative to CS pulses in order to comply with the STDP protocol).

the SNN shown in **Figure 3** but, at the same time, share some of the neurons between two copies of the SNN (**Figure 4A**). The resulting SNN contains four sensory neurons (N1, N2 for CS and N3, N4 for US, **Figure 4A**) and two motoneurons N5, N6 modulating the rotation velocities of the left and right motors, respectively (see also **Figure 1**). Neurons N3 and N4 are mutually inhibitory coupled with fixed synaptic weights ($w_{34} = w_{43} = 1$).

The pair of neurons receiving CS (N1, N2) can be connected to the pair of sonars in an arbitrary order (left–right or right–left). Depending on the connection, there can be two types of associations between the stimuli and motors: either with strong "parallel" (PA) or strong "diagonal" (DA) pathways (**Figure 4B**). Such freedom ensures that there is no *a priori* chosen structure in the complete SNN. Instead, the SNN adapts to the stimuli coming from the environment. Thus, the mutual exchange of the CS sources can simulate a situation with a change in the environment, which should induce relearning in the SNN and adaptation to novel conditions. Note that the bidirectional coupling between neurons N1 and N2 plays a fundamental role by providing synaptic competition while training couplings to neurons N3 and N4.

Classical or Pavlovian Conditioning

To implement Pavlovian (classical) conditioning, let us, for a moment, deactivate neuron N7 responsible for forward movement. If an object approaches the robot from one side, the corresponding touch sensor is activated, and we get an unconditional response (**Figure 4C** and **Supplementary Video S1**). At the same time, the corresponding sonar is also triggered on, and paired trains of stimuli innervate sensory neurons with a time delay of 10 ms.

We repeated such a stimulation alternately on the left and right sides of the robot. This protocol led to the potentiation of two associations for the left and right sides. Five stimulating cycles applied to the right and left sides were sufficient to achieve robust learning. After switching the connections of the sonars between sensory neurons N1 and N2, the SNN was able to relearn the associations (i.e. to switch between PA and DA, **Figure 4B**) after about 10–15 stimulus cycles.

In practice, to avoid obstacles successfully, the robot should gain high selectivity of the right and left channels. Then, in the presence of an obstacle on the left side, neuron N5 fires while neuron N6 is silent, which occurs in part due to inhibitory connections between neurons N3 and N4. Experimentally, the channel selectivity can be monitored by measuring the ratio of synaptic weights of "parallel" and "diagonal" connections:

$$w_P = (w_{31} + w_{42})/2,\ w_D = (w_{41} + w_{32})/2. \quad (8)$$

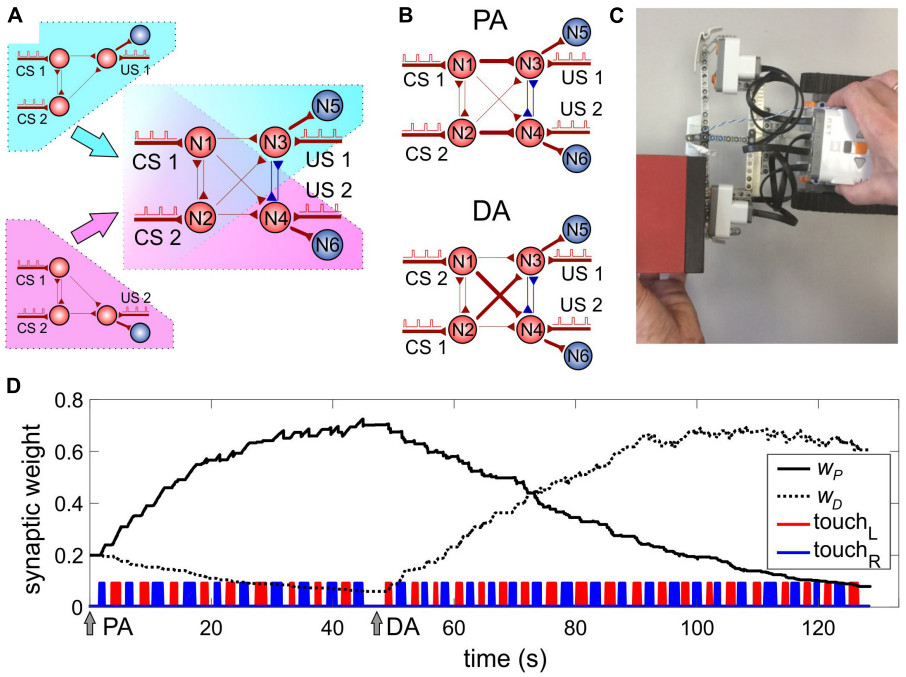


FIGURE 4 | Model of classical conditioning. **(A)** The design of a two-channel SNN by duplicating the single-channel SNN (**Figure 3**) sharing some neurons. The neural circuit includes neurons N1–N4 involved in learning. Motoneurons N5 and N6 provide turning the robot away from an obstacle. **(B)** The SNN after learning. PA, parallel association: N1 (N2) is associated with N3 (N4), couplings w_{31} and w_{42} are potentiated. DA, diagonal association: N2 (N1) is associated with N3 (N4), couplings w_{32} and w_{41} are potentiated. **(C)** Application of a stimulus to the touch and sonar sensors. **(D)** Evolution of the average weights of parallel (w_P) and diagonal (w_D) couplings under classical conditioning. Arrows PA and DA denote the time instants of the beginning of learning with correspondent scheme of the US mapping; touchL (touchR) is the time course of triggering the left (right) touch sensor.

Figure 4D shows the dynamics of these connections when simulating classical conditioning. Note that in the case of PA, the parallel connection w_P is potentiated, while the diagonal connection w_D is depressed. This happens due to simultaneous potentiation/depression of the pairs (w_{31}, w_{42}) and (w_{41}, w_{32}), according to the shortest pathway rule. After switching the CS inputs (**Figure 4D**, DA arrow), the opposite effect is observed, which leads to relearning in the SNN.

To achieve a high learning rate, our experiments show that the SNN should satisfy the following conditions:

1. Intermediate noise variance ($D = 5.5$ in experiments).
2. Bidirectional coupling between CS neurons (N1 and N2, **Figure 4A**).
3. Couplings between CS and US neurons are STDP-driven.
4. Inhibitory connections between US neurons (N3 and N4, **Figure 4A**).

Condition (1) agrees with our previous findings showing that the network rearrangement under stimulation takes place in a certain interval of the noise intensity (Lobov S. A. et al., 2017). At low noise intensity, the neuronal activation may not reach the level necessary for STDP-ordered pre and post-synaptic spiking. At high noise intensity, random STDP events dominate and break learning (see **Supplementary Figure S1**). Condition (2) expresses competition between the synapses involved in the associations increasing the SNN selectivity. Thus, competition plays a positive role in learning, unlike the case study reported previously (Lobov S. et al., 2017b). Condition (3) implies a reduction of the SNN selectivity due to a negative effect that STDP can have on the synaptic couplings between CS neurons (w_{21} and w_{12}). Condition (4) leads to competition between neurons "for the right" to be activated and, as a result, to an increase in the selectivity of the connections of the right and left channels.

Operant or Instrumental Conditioning

Animals learn behaviors through active interaction with the environment. To model such natural learning, we use operant (or instrumental) conditioning. To implement it, we activated motoneuron N7 (**Figures 1B,C**) responsible for forward movement and introduced the robot in an arena with several obstacles (**Figure 5A**).

In the beginning, the robot could avoid obstacles only after touching them due to US (**Figure 5A**). Then, learning progressively established associations between approaching obstacles (sonars, CS) and touching events (US). Thus, the robot learned to avoid obstacles in advance, without touching them (**Figure 5B** and **Supplementary Videos S2, S3**). We then switched sonars. Similarly to classical conditioning, the robot was able to relearn the associations (**Figure 5C**, PA arrow).

The learning rate depends on the total time of activation of the touch sensors. In turn, this time depends on the configuration of the arena, i.e. the arena size and the number of obstacles.

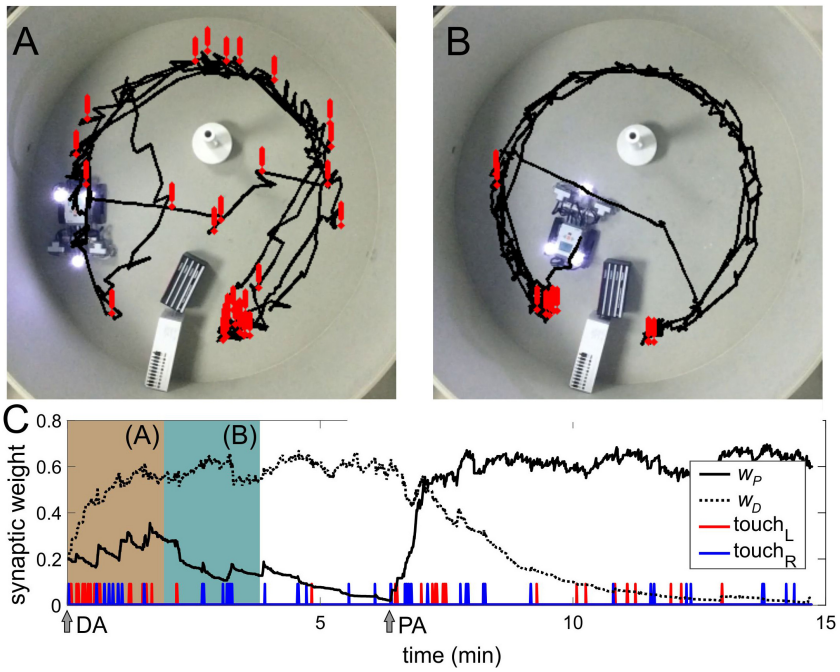


FIGURE 5 | Operant conditioning. **(A)** Trajectory of the robot in the first 2 min of the experiment. Exclamation marks indicate the positions of collisions with obstacles. **(B)** Same as in **(A)** but after learning. **(C)** Evolution of the weights of parallel (w_P) and diagonal (w_D) couplings (compare to **Figure 4D**). Beige and green-blue bars correspond to periods **(A,B)**, respectively.

In the Morris water maze (**Figure 5A**, 1 m^2), learning takes about 2 min. In a larger room (50 m^2) with a few obstacles, the learning time increases to 10–20 min. Relearning takes about twice a longer time.

In the operant conditioning, the SNN selectivity did not reach the value achieved in classical conditioning (compare **Figures 4D**, **5C**). It occurs due to the fact that in the arena, the robot can approach objects in front. In this case, both sonars detect them, which leads to a simultaneous generation of stimuli on the left and right sides and competition between two connections from the same sensory neuron. Technical constraints, such as a narrow sensing angle of the sonars, also affect the correct implementation of the obstacle-avoidance task negatively. All these factors diminish the learning quality. Therefore, the robot sometimes collides with obstacles. Thus, in a real environment, learning does not reach 100% collision avoidance.

DISCUSSION

Competition is a universal paradigm well-extended both in neurophysiology, e.g. in the form of lateral inhibition (Kandel et al., 2000) and the ANN studies, e.g. in the form of competitive learning in Kohonen networks (Kohonen, 1982) or imitation learning (Calvo Tapia et al., 2018). In this work, we have proposed an SNN model implementing associative learning through an STDP protocol and temporal coding of sensory stimuli. To achieve successful learning, the SNN makes use of two mechanisms of competition. The first type is neuronal competition, i.e. different neurons compete to be the first to get excited. In our case, this mechanism was provided by inhibitory connections between US neurons.

The second type of mechanism is synaptic competition; i.e. different synaptic inputs to a single neuron compete to be the one exciting the neuron. This mechanism has been less addressed in the literature on learning. Earlier, it was shown that in unstructured networks, synaptic competition leads to negative consequences for learning (Lobov S. A. et al., 2017; Lobov S. et al., 2017b). We have shown that the proposed structured architecture of the SNN, together with synaptic competition implementing the STDP-mediated rule of the shortest pathway, can ensure learning. We also note that the proposed mechanism of synaptic competition works well in the case of temporal coding of stimuli. Stimulus coding by the firing rate may require the development of a different approach. For example, in our recent study (Lobov et al., 2020), we implemented synaptic competition using synaptic forgetting, depending on the activity of the postsynaptic neuron. This allowed performing a mixed type of coding (temporal and rate) in the problem of recognition of electromyographic signals.

To test the SNN, we used it for controlling a mobile robot. We have shown that indeed, the robot exhibits successful learning at the behavioral level in the form of classical and operant conditioning. During navigation in an arena, the SNN self-organizes in such a way that after learning, the robot avoids obstacles without collisions, relying on CS only. Moreover, it can also relearn if the connection of CS sensors is switched between the corresponding sensory neurons, and a network rewiring, widely observed in biological neural networks, is

required (Calvo Tapia et al., 2020). The mechanism of relearning can be considered as a model of the animals' ability to adapt to changes in the environment. In the SNN, it is possible due to synaptic competition. Our experiments have also shown that learning is robust. The robot can operate in environments of different sizes and with varying densities of obstacles.

The proposed SNN implements a model with two associations: left and right sensors "coupled" to the right and left turns. In general, such associative learning can be extended to multiple inputs and outputs. Thus, the proposed architecture can be considered as a perceptron composed of spiking neurons with two inputs and two outputs, where logical 1 or 0 at an input corresponds to the presence or absence of a CS, respectively. Then, the US provides a learning mechanism on how to excite the target neuron in the output layer, i.e. how to obtain the desired output. Thus, we get a simple mechanism for supervised learning, i.e. a replacement of the backpropagation algorithm for SNNs. However, the question of how many neurons such a spiking perceptron can contain and, hence, how many classes can be discriminated in this way requires additional studies.

We note that the parameters of sensory stimuli play a crucial role in the learning of behaviors. For example, longer delays between stimuli or their inverse order (CS after US) can impair learning. In this sense, the temporal coding in SNNs requires fine-tuning of the neuronal circuits and maybe not robust. The rate coding using, e.g. the triplet-based STDP rule (Pfister and Gerstner, 2006), voltage-based STDP with homeostasis (Clopath et al., 2010), or STDP together with BCM rule (Wade et al., 2008; Liu et al., 2019) is likely to increase the reliability of robot control. However, in this case, we may end up with a mixed type of coding (temporal and rate).

Due to structural simplicity, the proposed SNN and the learning algorithm admit a hardware implementation by, e.g. using memristors, which are adaptive circuit elements with memory. Memristors change their resistance depending on the history of electrical stimulation (Wang et al., 2019). Since the first experiments and simulations (Linares-Barranco et al., 2011), significant progress has been achieved in the implementation of excitatory and inhibitory STDP by using resistive-switching devices (RRAM), which are a particular class of memristors with two-terminal metal–insulator–metal structure. Although most of STDP demonstrations still rely on a time overlap of pre- and postsynaptic spikes (Yu et al., 2011; Kuzum et al., 2013; Emelyanov et al., 2019), the rich internal dynamics of higher-order memristive devices related to multi-time-scale microscopic transport phenomena provides timing- and frequency-dependent plasticity in response to non-overlapping input signals in a biorealistic fashion (Du et al., 2015; Kim et al., 2015). Memristive plasticity can be realized at different time scales, in particular with STDP windows of the order of microseconds (Kim et al., 2015), which is essential for the development of fast spike encoding systems.

Upon reaching the technology maturity, arrays of memristive synapses offer unique scalability being integrated with CMOS layers and showing spatiotemporal functions (Wang W. et al., 2018), as well as combined with artificial memristive neurons (Wang Z. et al., 2018) within a single network. Simple spiking architectures of Pavlov's dog association have been proposed on memristors (Ziegler et al., 2012; Milo et al., 2017; Tan et al., 2017; Minnekhanov et al., 2019). However, more sophisticated architectures are required to reproduce different types of associative learning to be adopted in advanced robotic systems. We anticipate that, soon, artificial neurons can be realized on the CMOS architecture, whereas the STDP can be implemented by incorporating memristors (Emelyanov et al., 2019). It seems convenient to have paired micro-scaled memristive devices to reproduce bipolar synaptic weights. They can be mounted in a standard package for easier integration into the SNN circuits.

Finally, we also foresee that the provided architecture can be implemented in biological neural networks grown in neuronal cultures *in vitro*. Modern technology of microfluidic channels permits building different network architectures (Gladkov et al., 2017). On the one hand, such a living SNN could verify if our understanding of the learning mechanism at the cell level is correct. From the other side, biological neurons have a much higher level of flexibility mediated by different molecular mechanisms that may shed light on how learning and sensory-motor control are organized in nature.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

SL, VM, and VK conceived and designed the research. SL developed the NeuroNet software, designed the robot configuration, and implemented the control of the robot by SNN. SL and MS carried out experiments with the robot and video tracking the movements of the robot. AM suggested the approach for emulating synaptic plasticity by memristive microdevices. All authors participated in the interpretation of the results and wrote the manuscript.

FUNDING

This work was supported by the Russian Science Foundation (project 19-12-00394, the SNN development and implementation), by the Russian Foundation for Basic Research (grant 18-29-23001, the emulation of synaptic plasticity by using memristive microdevices), and by the Spanish Ministry of Science, Innovation, and Universities (grant FIS2017-82900-P, the analysis of the robot dynamics).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.00088/full#supplementary-material

REFERENCES

Bakkum, D. J., Chao, Z. C., and Potter, S. M. (2008). Spatio-temporal electrical stimuli shape behavior of an embodied cortical network in a goal-directed learning task. *J. Neural Eng.* 5, 310–323. doi: 10.1088/1741-2560/5/3/004

Bi, G., and Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472. doi: 10.1523/jneurosci.18-24-10464.1998

Calvo Tapia, C., Makarov, V. A., and van Leeuwen, C. (2020). Basic principles drive self-organization of brain-like connectivity structure. *Commun. Nonlinear Sci. Numer. Simul.* 82:105065. doi: 10.1016/j.cnsns.2019.105065

Calvo Tapia, C., Tyukin, I. Y., and Makarov, V. A. (2018). Fast social-like learning of complex behaviors based on motor motifs. *Phys. Rev.* 97:052308. doi: 10.1103/PhysRevE.97.052308

Chou, T.-S., Bucci, L., and Krichmar, J. (2015). Learning touch preferences with a tactile robot using dopamine modulated STDP in a model of insular cortex. *Front. Neurorobot.* 9:6. doi: 10.3389/fnbot.2015.00006

Clopath, C., Büsing, L., Vasilaki, E., and Gerstner, W. (2010). Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nat. Neurosci.* 13, 344–352. doi: 10.1038/nn.2479

Dauth, S., Maoz, B. M., Sheehy, S. P., Hemphill, M. A., Murty, T., Macedonia, M. K., et al. (2016). Neurons derived from different brain regions are inherently different in vitro: a novel multiregional brain-on-a-chip. *J. Neurophysiol.* 117, 1320–1341. doi: 10.1152/jn.00575.2016

Dayan, P., and Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: The MIT Press.

DeMarse, T. B., Wagenaar, D. A., Blau, A. W., and Potter, S. M. (2001). The neurally controlled animat: biological brains acting with simulated bodies. *Auton. Robots* 11, 305–310.

Du, C., Ma, W., Chang, T., Sheridan, P., and Lu, W. D. (2015). Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics. *Adv. Funct. Mater.* 25, 4290–4299. doi: 10.1002/adfm.201501427

Emelyanov, A. V., Nikiruy, K. E., Demin, V. A., Rylkov, V. V., Belov, A. I., Korolev, D. S., et al. (2019). Yttria-stabilized zirconia cross-point memristive devices for neuromorphic applications. *Microelectron. Eng.* 215:110988. doi: 10.1016/j.mee.2019.110988

Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. Natl. Acad. Sci. U.S.A.* 113, 11441–11446. doi: 10.1073/pnas.1604850113

Gladkov, A., Pigareva, Y., Kutyina, D., Kolpakov, V., Bukatin, A., Mukhina, I., et al. (2017). Design of cultured neuron networks in vitro with predefined connectivity using asymmetric microfluidic channels. *Sci. Rep.* 7:15625. doi: 10.1038/s41598-017-15506-2

Gong, P., and van Leeuwen, C. (2009). Distributed dynamical computation in neural circuits with propagating coherent activity patterns. *PLoS Comput. Biol.* 5:e1000611. doi: 10.1371/journal.pcbi.1000611

Gorban, A. N., Makarov, V. A., and Tyukin, I. Y. (2019). The unreasonable effectiveness of small neural ensembles in high-dimensional brain. *Phys. Life Rev.* 29, 55–88. doi: 10.1016/j.plrev.2018.09.005

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York, NY: Wiley.

Hong, S., Ning, L., Xiaoping, L., and Qian, W. (2010). "A cooperative method for supervised learning in Spiking neural networks," in *Proceedings of the 14th International Conference on Computer Supported Cooperative Work in Design*, Shanghai.

Houk, J., Adams, J., and Barto, A. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. *Model. Inf. Process. Basal Ganglia* 13.

Hull, C. L. (1943). *Principles of Behavior: An Introduction to Behavior Theory*. New York, NY: Appleton-Century-Crofts.

Izhikevich, E. M. (2002). Resonance and selective communication via bursts in neurons having subthreshold oscillations. *Biosystems* 67, 95–102. doi: 10.1016/S0303-2647(02)00067-9

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* 15, 1063–1070. doi: 10.1109/TNN.2004.832719

Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb. Cortex* 17, 2443–2452. doi: 10.1093/cercor/bhl152

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2000). *Principles of Neural Science*, 4th Edn, New York, NY: McGraw-Hill.

Kim, S., Du, C., Sheridan, P. J., Ma, W., Choi, S., and Lu, W. D. (2015). Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity. *Nano Lett.* 15, 2203–2211. doi: 10.1021/acs.nanolett.5b00697

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69. doi: 10.1007/bf00337288

Kuzum, D., Yu, S., and Philip Wong, H.-S. (2013). Synaptic electronics: materials, devices and applications. *Nanotechnology* 24:382001. doi: 10.1088/0957-4484/24/38/382001

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Linares-Barranco, B., Serrano-Gotarredona, T., Camuñas-Mesa, L., Perez-Carrasco, J., Zamarreño-Ramos, C., and Masquelier, T. (2011). On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Front. Neurosci.* 5:26. doi: 10.3389/fnins.2011.00026

Liu, J., Mcdaid, L. J., Harkin, J., Karim, S., Johnson, A. P., Millard, A. G., et al. (2019). Exploring self-repair in a coupled spiking astrocyte neural network. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 865–875. doi: 10.1109/TNNLS.2018.2854291

Lobov, S. A., Chernyshov, A. V., Krilova, N. P., Shamshin, M. O., and Kazantsev, V. B. (2020). Competitive learning in a spiking neural network: towards an intelligent pattern classifier. *Sensors* 20:E500. doi: 10.3390/s20020500

Lobov, S. A., Zhuravlev, M. O., Makarov, V. A., and Kazantsev, V. B. (2017). Noise enhanced signaling in STDP driven spiking-neuron network. *Math. Model. Nat. Phenom.* 12, 109–124. doi: 10.1051/mmnp/201712409

Lobov, S., Balashova, K., Makarov, V. A., and Kazantsev, V. (2017b). "Competition of spike-conducting pathways in stdp driven neural networks," in *Proceedings of the 5th International Congress on Neurotechnology, Electronics and Informatics*, Setúbal.

Malishev, E., Pimashkin, A., Arseniy, G., Pigareva, Y., Bukatin, A., Kazantsev, V., et al. (2015). Microfluidic device for unidirectional axon growth. *J. Phys.* 643:12025. doi: 10.1088/1742-6596/643/1/012025

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275, 213–215. doi: 10.1126/science.275.5297.213

Meyer, J. A., and Wilson, S. W. (1991). *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*. Cambridge: MIT Press.

Milo, V., Ielmini, D., and Chicca, E. (2017). "Attractor networks and associative memories with STDP learning in RRAM synapses," in *Proceedings of the 2017 IEEE International Electron Devices Meeting (IEDM)*, San Francisco, CA.

Minnekhanov, A. A., Emelyanov, A. V., Lapkin, D. A., Nikiruy, K. E., Shvetsov, B. S., Nesmelov, A., et al. (2019). Parylene based memristive devices with multilevel resistive switching for neuromorphic applications. *Sci. Rep.* 9:10800. doi: 10.1038/s41598-019-47263-9

Morrison, A., Diesmann, M., and Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybern.* 98, 459–478. doi: 10.1007/s00422-008-0233-1

Palmer, J. H. C., and Gong, P. (2014). Associative learning of classical conditioning as an emergent property of spatially extended spiking neural circuits with synaptic plasticity. *Front. Comput. Neurosci.* 8:79. doi: 10.3389/fncom.2014.00079

Pamies, D., Hartung, T., and Hogberg, H. T. (2014). Biological and medical applications of a brain-on-a-chip. *Exp. Biol. Med.* 239, 1096–1107. doi: 10.1177/1535370214537738

Pavlov, I. P. (1927). *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Oxford: Oxford University Press.

Pfister, J.-P., and Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.* 26, 9673–9682. doi: 10.1523/JNEUROSCI.1425-06.2006

Pimashkin, A., Gladkov, A., Agrba, E., Mukhina, I., and Kazantsev, V. (2016). Selectivity of stimulus induced responses in cultured hippocampal networks on microelectrode arrays. *Cogn. Neurodyn.* 10, 287–299. doi: 10.1007/s11571-016-9380-6

Pimashkin, A., Gladkov, A., Mukhina, I., and Kazantsev, V. (2013). Adaptive enhancement of learning protocol in hippocampal cultured networks grown on multielectrode arrays. *Front. Neural Circuits* 7:87. doi: 10.3389/fncir.2013.00087

Potter, S. M., Fraser, S. E., and Pine, J. (1997). "Animat in a petri dish: cultured neural networks for studying neural computation," in *Proceedings of the 4th Joint Symposium on Neural Computation, UCSD*, San Diego, CA.

Reger, B. D., Fleming, K. M., Sanguineti, V., Alford, S., and Mussa-Ivaldi, F. A. (2000). Connecting brains to robots: an artificial body for studying the computational properties of neural tissues. *Artif. Life* 6, 307–324. doi: 10.1162/106454600300103656

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536. doi: 10.1038/323533a0

Shahaf, G., Eytan, D., Gal, A., Kermany, E., Lyakhov, V., Zrenner, C., et al. (2008). Order-based representation in random networks of cortical neurons. *PLoS Comput. Biol.* 4:e1000228. doi: 10.1371/journal.pcbi.1000228

Sjöström, P. J., Turrigiano, G. G., and Nelson, S. B. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32, 1149–1164. doi: 10.1016/S0896-6273(01)00542-6

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3:919. doi: 10.1038/78829

Tan, Z.-H., Yin, X.-B., Yang, R., Mi, S.-B., Jia, C.-L., and Guo, X. (2017). Pavlovian conditioning demonstrated with neuromorphic memristive devices. *Sci. Rep.* 7:713. doi: 10.1038/s41598-017-00849-7

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A. (2019). Deep learning in spiking neural networks. *Neural Netw.* 111, 47–63. doi: 10.1016/j.neunet.2018.12.002

Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. *Neural Comput.* 10, 821–835. doi: 10.1162/089976698300017502

Tyukin, I., Gorban, A. N., Calvo, C., Makarova, J., and Makarov, V. A. (2019). High-dimensional brain: a tool for encoding and rapid learning of memories by single neurons. *Bull. Math. Biol.* 81, 4856–4888. doi: 10.1007/s11538-018-0415-5

Wade, J. J., McDaid, L. J., Santos, J. A., and Sayers, H. M. (2008). "SWAT: an unsupervised SNN training algorithm for classification problems," in *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong.

Wang, F. Z., Li, L., Shi, L., Wu, H., and Chua, L. O. (2019). Φ memristor: real memristor found. *J. Appl. Phys.* 125:54504. doi: 10.1063/1.5042281

Wang, W., Pedretti, G., Milo, V., Carboni, R., Calderoni, A., Ramaswamy, N., et al. (2018). Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses. *Sci. Adv.* 4:eaat4752. doi: 10.1126/sciadv.aat4752

Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., et al. (2018). Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* 1, 137–145. doi: 10.1038/s41928-018-0023-2

Xu, Y., Zeng, X., Han, L., and Yang, J. (2013). A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks. *Neural Netw.* 43, 99–113. doi: 10.1016/j.neunet.2013.02.003

Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., and Wong, H. S. P. (2011). An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electron Devices* 58, 2729–2737. doi: 10.1109/TED.2011.2147791

Ziegler, M., Soni, R., Patelczyk, T., Ignatov, M., Bartsch, T., Meuffels, P., et al. (2012). An electronic version of pavlov's dog. *Adv. Funct. Mater.* 22, 2744–2749. doi: 10.1002/adfm.201200244

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Lobov, Mikhaylov, Shamshin, Makarov and Kazantsev. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 06 March 2020
doi: 10.3389/fnins.2020.00184

Topological Properties of Neuromorphic Nanowire Networks

*Alon Loeffler[1], Ruomin Zhu[1], Joel Hochstetter[1], Mike Li[2], Kaiwei Fu[1], Adrian Diaz-Alvarez[3], Tomonobu Nakayama[3], James M. Shine[2] and Zdenka Kuncic[1]**

[1] School of Physics, The University of Sydney, Sydney, NSW, Australia, [2] Central Clinical School, The University of Sydney, Sydney, NSW, Australia, [3] International Center for Materials Nanoarchitectonics (WPI-MANA), National Institute for Materials Science (NIMS), Tsukuba, Japan

Graph theory has been extensively applied to the topological mapping of complex networks, ranging from social networks to biological systems. Graph theory has increasingly been applied to neuroscience as a method to explore the fundamental structural and functional properties of human neural networks. Here, we apply graph theory to a model of a novel neuromorphic system constructed from self-assembled nanowires, whose structure and function may mimic that of human neural networks. Simulations of neuromorphic nanowire networks allow us to directly examine their topology at the individual nanowire–node scale. This type of investigation is currently extremely difficult experimentally. We then apply network cartographic approaches to compare neuromorphic nanowire networks with: random networks (including an untrained artificial neural network); grid-like networks and the structural network of *C. elegans*. Our results demonstrate that neuromorphic nanowire networks exhibit a small–world architecture similar to the biological system of *C. elegans*, and significantly different from random and grid-like networks. Furthermore, neuromorphic nanowire networks appear more segregated and modular than random, grid-like and simple biological networks and more clustered than artificial neural networks. Given the inextricable link between structure and function in neural networks, these results may have important implications for mimicking cognitive functions in neuromorphic nanowire networks.

Keywords: neuromorphic, atomic-switch networks, nanowires, topology, complex networks, structural connectivity, graph theory, artificial neural networks

OPEN ACCESS

Edited by:
Stefano Brivio, Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Martin Ziegler, Technische Universität Ilmenau, Germany
Garibaldi Pineda García, University of Sussex, United Kingdom

****Correspondence:***
Zdenka Kuncic
zdenka.kuncic@sydney.edu.au

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *30 November 2019*
Accepted: *19 February 2020*
Published: *06 March 2020*

Citation:
Loeffler A, Zhu R, Hochstetter J, Li M, Fu K, Diaz-Alvarez A, Nakayama T, Shine JM and Kuncic Z (2020) Topological Properties of Neuromorphic Nanowire Networks. Front. Neurosci. 14:184. doi: 10.3389/fnins.2020.00184

1. INTRODUCTION

1.1. Graph Theory Applications

Graph theory is a framework used to represent complex networks mathematically, whereby network components are represented as nodes (N) and connections between components are represented as edges (E) (Boccaletti et al., 2006). Since the 1950s, graph theory has been applied to networks in a variety of fields, including social networks (Harary and Norman, 1953), progression of disease (Eubank et al., 2004; Mason and Verwoerd, 2007), transport networks (Wakabayashi and Iida, 1992), the internet (Albert et al., 2011), and many others. Graph theory has largely been employed to study the structure of networks, known as structural connectivity. Measures such as the path length (PL), clustering coefficient ($CCoeff$), participant coefficient ($PCoeff$), within-module degree z-Score (MZ), degree and small worldness (see **Box 1** for definitions), are useful

BOX 1 | Graph Theory Terms

Clustering Coefficient (CCoeff): A measure of how much nodes in a graph tend to cluster together. This reflects the proportion of nodes connected to node *N* that are also connected to each other (Verweij et al., 2014).

Degree (DEG): The number of edges connected to a node, *N*.

Hubs: Areas through which large amounts of information flow to reach from one part of a network to another (Types of hubs and non-hub nodes are described in **Figure 6**).

Within-Module Degree z-Score (MZ): Measures how well connected a node is to other nodes in the same module (or cluster/community). This demonstrates whether the node is a hub in the network (i.e., much of the information flows through this node) (Guimerà and Amaral, 2005). Guimera and Nunes Amaral define MZ > 2.5 as hub-like nodes, and MZ < 2.5 as non-hub nodes.

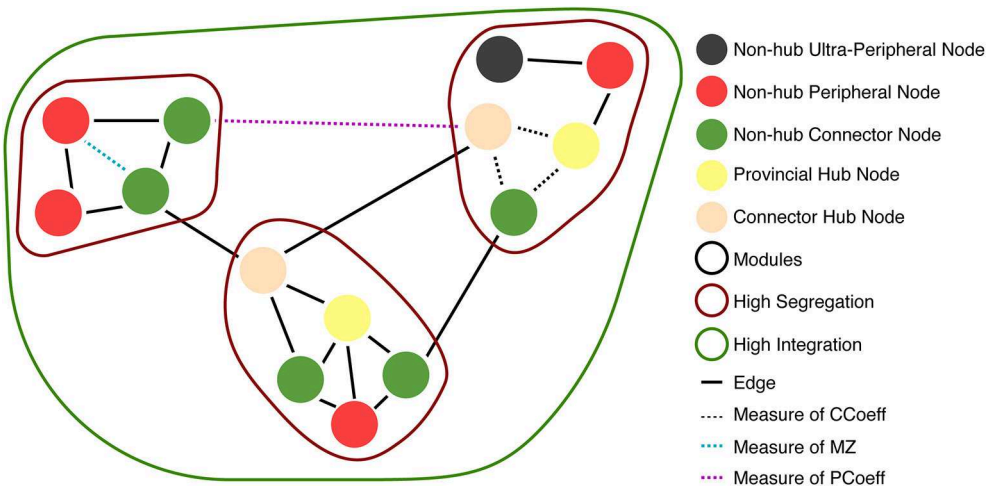
Modularity: A measure of network segregation into distinct modules (or clusters/communities) that have sparse connections between each module (Cohen and D'Esposito, 2016).

Participant Coefficient (PCoeff): Measures how homogeneous a node's edges are distributed across modules (or clusters/communities). Nodes are divided into two classes: (1) High PCoeff: connector nodes with many global edges across modules (strong between-module and weak within-module connectivity Rubinov and Sporns, 2010; Cohen and D'Esposito, 2016); and (2) Low PCoeff: provincial/local nodes with mostly edges that connect nodes within a module (strong within-module and weak between-module connectivity) (Joyce et al., 2010; Van Diessen et al., 2014; Bertolero et al., 2015).

Path Length (PL): Measures the minimal number of edges of all possible node connections in a network (Van Diessen et al., 2014; Verweij et al., 2014).

Small–worldness: A type of network architecture in which local clustering is combined with short path length. This architecture offers important advantages for network functionality, ranging from synchronizability to information flow (Oliveira et al., 2014; Muldoon et al., 2016).

Small–world Propensity: Introduced by (Muldoon et al., 2016), used to account for potential variations in connection strength in a network, by measuring how clustering and path length differ from random and grid-like networks.



characterizations of the structural properties of a network (Strogatz, 2001; Estrada and Hatano, 2008; Grayson et al., 2016). In many cases, analyzing the structure of a network is the first step to understanding its function (Strogatz, 2001).

Graph theory measures have been applied to the study of biological networks, including the brain structure of organisms such as the neural networks of *C. elegans* (Achacoso and Yamamoto, 1991; Yan et al., 2017) and Macaque monkeys (Achard et al., 2006), in attempt to better understand their function. Biological networks typically demonstrate a small–world architecture (see **Box 1** for definition). Small–worldness has been shown to allow for high efficiency of synchronized and parallel information transfer between neural regions (Bullmore and Sporns, 2009). Within such a system, shorter paths from node to node (with few longer sparse connections) may provide more efficient communication across an entire system, thereby facilitating dynamical processes that require global coordination and information flow (Watts and Strogatz, 1998; Strogatz, 2001). For instance, regions with short path length and high clustering coefficient confer an ability to transfer information quickly between a large number of nodes. Contrastingly, areas with long path lengths and low clustering may allow for sparse connections between individual clusters in a network, resulting in a slower spread of information over greater distance (Strogatz, 2001; Bullmore and Sporns, 2012; Muldoon et al., 2016). Understanding these distinct structural features within biological neural networks has allowed researchers to infer that such networks may utilize different structural properties to communicate under separate time scales (e.g., fast local synchronization within dense regions and slow global communication between dense regions; Chow and Kokotovic, 1985; Tahmassebi et al., 2017).

The commencement of the Human Connectome Project in 2005 (Sporns et al., 2005) has driven a surge in techniques and

studies to map the structure and function of the human brain network (Sporns et al., 2002; Bertolero et al., 2015; Farahani et al., 2019; Gilson et al., 2019). Many such studies apply graph theory to analyse the connectivity within and between regions of the brain (Bullmore and Sporns, 2009). While the networks of simple organisms such as *C. elegans* are composed of only 270–300 individual neurons (Yan et al., 2017), the human brain is a much larger network, composed of tens of billions of neurons (although the exact number is contested), each of which has around ten thousand synapses (Koch, 2004; Shepherd, 2004). The sheer number of neural components makes it extremely difficult to model human neural networks graphically. Therefore, much of the graph theory analysis on human neural networks is applied to large collection of neurons, or even entire regions of the brain (e.g., Bassett and Bullmore, 2006; Gilson et al., 2019).

1.2. Neuromorphic Systems: Mimicking the Brain in Hardware

In parallel to developments in neuroscience, the engineering community has spurred the development of neuromorphic systems that can mimic the function of human neurons in hardware (Vianello et al., 2017). Carver Mead's pioneering efforts to emulate biological information processing using analog circuits (instead of logic gates used in digital computing) and leveraging the inherent device physics of Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) established a new paradigm in computing hardware (Mead, 1990). Today, neuromorphic computing encompasses the use of novel nanotechnologies such as non-volatile memory devices and memristors (memory-resistors) that can mimic synapse-like memory and spiking temporal characteristics (Yang et al., 2013; Burr et al., 2017; Ziegler et al., 2018; Roy et al., 2019). Because of their unconventional "beyond von Neumann" architecture, which substantially reduces power requirements, such devices are also attractive for implementing Artificial Neural Network (ANN) algorithms, which require computationally-intensive training to learn input-output relationships, thereby mimicking neurons and synaptic connections in software (Xu et al., 2018).

Similarly, neuromorphic chips [e.g., IBM's TrueNorth (Merolla et al., 2014; Akopyan et al., 2015) and Intel's Loihi (Davies et al., 2018)] have been developed specifically as ANN accelerators, although their neuromorphic hardware attributes are limited to the integration of processing and memory to reduce power requirements. More generally, a limitation of neuromorphic in-memory computing hardware systems is their restriction to a regularized grid-like array structure that emphasizes the role of individual synapse-like elements (e.g., memristors), rather than the network architecture as a whole.

This limits potential advantages arising from structure–function integration in a distributed network, such as in a small–world architecture seen in biological neural networks (Bullmore and Sporns, 2009; Chialvo, 2010). It is likely that due to their conventional grid-like array structure, most neuromorphic systems lack the emergent dynamical properties that are characteristic of neural network circuitry (e.g., memory, learning, and even intelligence). Such emergent properties are attributed to the complexity of neural networks and the interplay between structure and function (Hagmann et al., 2008; Chialvo, 2010; Bassett and Gazzaniga, 2011). It is important to note that factors other than topology may influence emergent behavior (e.g., learning rules specifically designed for ability acquisition; Chollet, 2019). However, much of the literature exploring emergence in complex systems, including biological networks, emphasizes the role of topology, and structural properties as key to understanding emergence (Angeline, 1994; Chialvo, 2010; Pascual-García, 2016; Dumitrescu et al., 2017).

We previously introduced a novel neuromorphic system comprised of self–assembled nanowires whose structure and function (in response to electrical stimulation) mimic that of biological neural networks (Kuncic et al., 2018; Diaz-Alvarez et al., 2019). In these networks, each junction between nanowires provides a non-linear synaptic function in a similar manner as an atomic switch (Terabe et al., 2005; Ohno et al., 2011). Rather than focusing on the controllability of individual synapses like ANNs or other neuromorphic systems, our Atomic Switch-like Networks (ASNs) mimic the complex topology of biological neural networks, by mimicking biological self–assembly to form similarly complex networks comprised of nanowires (synthetic neurons) and junctions (synthetic synapses) (Stieg et al., 2012; Diaz-Alvarez et al., 2019).

Previous studies have shown that ASNs exhibit emergent properties such as non-linear dynamics, recurrence and capacity for learning, which arise from the complexity of the networks, as well as the properties of the atomic switch-like junctions (Terabe et al., 2005; Avizienis et al., 2012; Kuncic et al., 2018). Such properties are essential for brain-like function (Avizienis et al., 2012). However, due to the complexity of ASNs, it is highly difficult to understand or predict the impact and interactions of the networks' structure and functions from experimental data alone. Furthermore, due to the networks' self-assembled structure, it is experimentally difficult to control the topology to measure how it influences dynamics. It is also extremely difficult to use imaging-based techniques such as or electron microscopy (e.g., White et al., 1986; Eberle and Zeidler, 2018) reconstructions to unpack the structural connectivity of ASNs, as it is impossible to tell whether or not intersecting wires form a junction between them. We therefore have developed a computational model that simulates the structure experimental ASNs, based on functional, experimental validation (Kuncic et al., 2018; Diaz-Alvarez et al., 2019). For the purposes of the present study, we use this model solely to construct simulated self-assembled networks for structural analysis. ASNs are made of a fixed nanowire structure that does not change under electrical activation. Our simulations allow us to visualize each wire and connection individually in a graphical representation, and to easily alter them, either by changing the positioning and lengths of individual wires and junctions, or manipulating the density and dispersion of the networks. Consequently, our model enables us to examine the structural properties of specific sections of the network, which is currently impossible to do experimentally, as well as different realizations of nanowire networks.

Here, we apply graph theory measures to simulated ASNs with varying topologies. This allows us to examine the topological properties of ASNs, and compare them with a range of real-world networks. These include the simple organism *C. elegans*, as well as random and grid-like networks.

2. METHODS

2.1. Construction of Simulated ASNs

To explore the topology of ASNs, we generated multiple networks with different structural parameters (see **Figure 1** for visualizations). Hardware ASNs acquire a complex network structure through bottom-up self-assembly (Avizienis et al., 2012; Stieg et al., 2012; Diaz-Alvarez et al., 2019), similar to neural network growth in the brain. To simulate this self-assembly, we modeled nanowires as 1D objects of length uniformly drawn from a normal distribution of specified average wire length (mean of distribution, ranging from 6 to 9 μm) and wire dispersion (ratio of standard deviation to the mean, ranging from 0 to 50%). These wires were randomly placed within a 2D plane of fixed size (30 $\times$ 30 μm), with horizontal and vertical positions of the wire centers generated from a uniform spatial distribution. The angular orientation of each wire was generated from a uniform distribution on 2π. A junction was modeled at each intersection point between nanowires (Kuncic et al., 2018; Diaz-Alvarez et al., 2019). The connectivity was mapped to a graph adjacency matrix representation with nodes corresponding to nanowires and edges corresponding to junctions. In real networks, not every intersection between wires need necessarily form a junction. It is, however, practically difficult to determine where individual junctions exist in the self-assembled networks (Diaz-Alvarez et al., 2019). In our simulated networks, the simplifying assumption that all intersections result in junctions has negligible effect on network functionality when compared to experimental measurements of hardware ASNs (see **Supplementary Materials**).

For each of the networks, the following parameters were varied: number of nanowires (i.e., 100, 500, 1,000, or 2,000 nws), average nanowire length (6 – 9 μm), and dispersion of wire length (0, 10, 20, and 50% of average nanowire length). Using this process, we generated a total of 39 different combinations of networks. All simulated networks were constructed in Matlab v2018a and Python v3.7.3. All structural connectivity measures were taken from the open-source Brain Connectivity Toolbox (Rubinov and Sporns, 2010) and NetworkX (Hagberg et al., 2008) packages.

To contextualize the structural connectivity of our ASNs, we simulated the topology of Watts-Strogatz networks ranging from random to grid-like, and *C. elegans*. Graph theory measures were applied to the connectivity data of each ASN, as well as to each of the Watts-Strogatz and *C. elegans* networks (see **Figure 2** for graphical representations of all networks). We also included a fully-connected ANN similar to a random Watts-Strogatz network. Next, we compared global clustering coefficient and average path lengths (Watts and Strogatz, 1998). We also calculated the small–world propensity values for each network to establish an unbiased (see **Box 1**) measure of small–worldness in all networks (Muldoon et al., 2016). Finally, we mapped 100 and 500 nw ASNs, as well as *C. elegans* and correspondingly sized random and grid-like WS networks, on the Guimerà and Amaral (2005) cartographic plane to compare participant coefficient and within-module degree z-score. This allowed us to examine the modularity and integration of the networks (Guimerà and Amaral, 2005; Power et al., 2013; Bertolero et al., 2015).

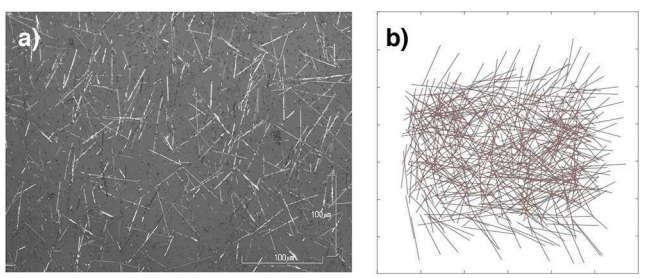


FIGURE 1 | Neuromorphic nanowire networks. **(a)** Optical microscopy image of an actual self–assembled network of nanowires. The length of wires varies from∼6 to 50μm in this image. **(b)** Simulated 500 nw (6,065 junction) network generated by our model. The length of wires in the simulated networks varies from 6 to 9μm.

2.2. Construction of Random and Grid-Like Watts-Strogatz Networks

To create a series of Watts-Strogatz networks, we first created a ring lattice with N nodes of mean degree $2k$, where $2k$ = mean degree of the corresponding ASN with N nodes. In the Watts-Strogatz networks, each node is connected to its k nearest neighbors on either side. For each edge, E, in the graph, we then rewire the target node to k other nodes in the network with probability β. When $\beta = 0$, no edges are rewired and the model returns a locally-clustered ring lattice. We term this network grid-like, as its non-graphical representation is formed from a grid-shaped lattice. In contrast, when $\beta = 1$, all of the edges are rewired and the ring lattice is transformed into a random graph (MathWorks, 2016). We varied β from 0 to 1 in steps of 0.05, leaving 21 networks ranging from completely Grid-Like ($\beta = 0$) to completely Random ($\beta = 1$), for each size N. A β of 0.2 is denoted as displaying "small–world" characteristics (Watts and Strogatz, 1998). Furthermore, to compare ASNs with a WS random-like ANN model, we constructed a 5-layer ANN, with 10 input nodes, 10 output nodes and 160 nodes in each middle layer. Every node in each layer is connected to every node in its parent and child layers (hence the term "fully-connected"). However, no nodes are connected within layers.

2.3. Construction of *C. elegans* Networks

Neuronal connectivity data of the simple nematode *C. elegans* (277 neurons and 2,105 synaptic connections) was adapted from Achacoso and Yamamoto (1991), and electron microscope reconstructions by White et al. (1986).

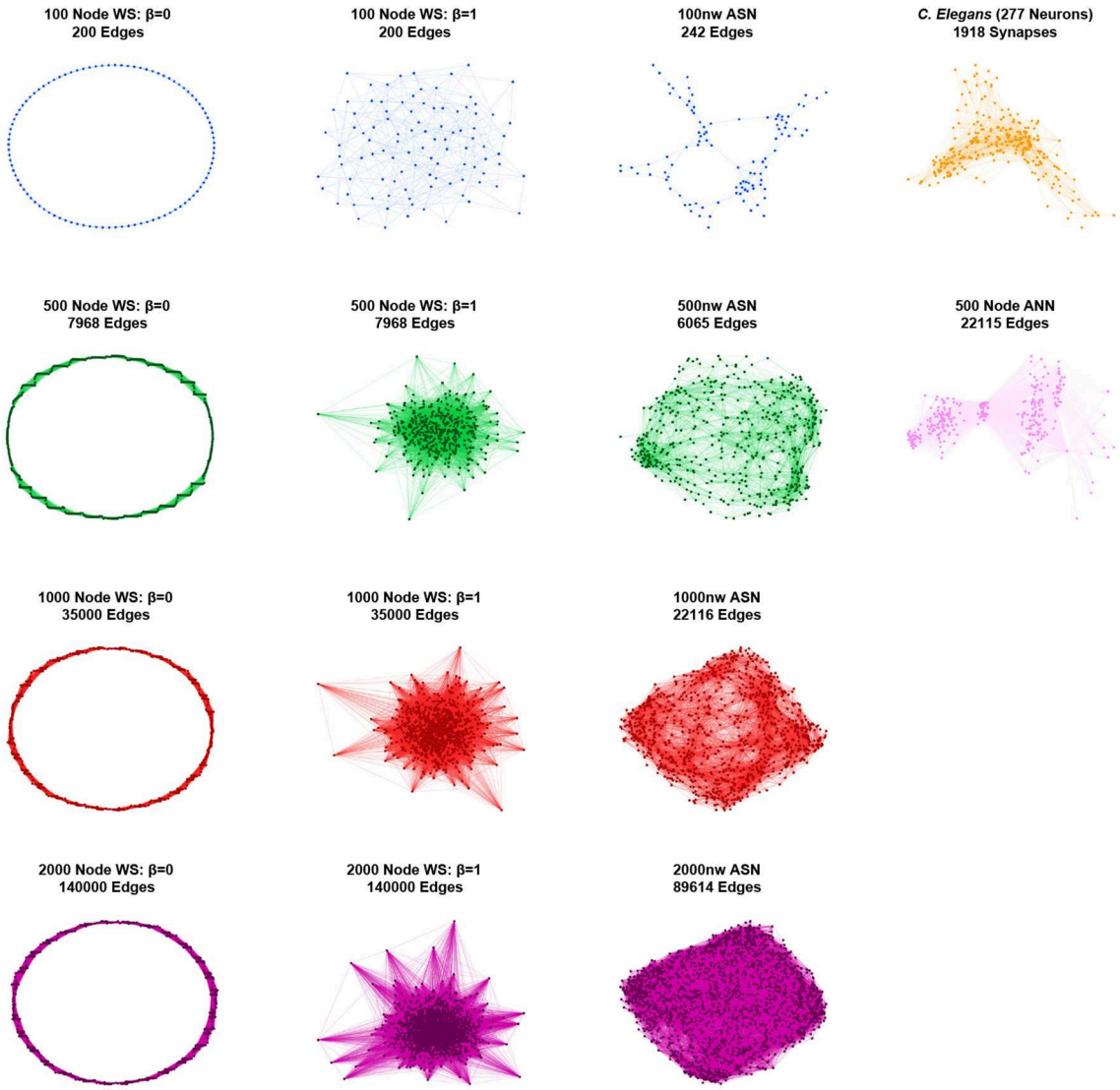


FIGURE 2 | Graphical representations of sample networks: Grid-like Watts-Strogatz (WS) networks (β=0; left); random Watts-Strogatz networks (β=1; center-left); ASNs (center-right); *C. elegans* and fully-connected ANN networks (right). Nodes represent nanowires (or neurons for *C. elegans* and ANN), while edges represent junctions (or synapses for *C. elegans* and virtual synapses for ANN).

3. RESULTS

3.1. Small–Worldness

We compared the structures of multiple unique ASNs across four sizes (a total of 39 networks comprised of 100, 500, 1,000, or 2,000 nanowires) with a fully-connected ANN, a *C. elegans* network, and Watts-Strogatz random/grid-like networks across four sizes and 21 varying β parameters (one network for each β, and for each size). See **Table 1** for a full statistical description of each network.

Figure 3 shows a comparison of path lengths and path distances between 100 and 500 nw ASNs, and a *C. elegans* network. **Figure 4** shows a comparison of global mean path length and global clustering coefficient for each of the networks studied. Larger ASNs have similar mean path length to *C. elegans*, but higher clustering. However, ASN networks of similar size to *C. elegans* have a higher average path length.

ASNs are also more clustered and have a longer mean path length than random WS networks ($\beta = 1$). Compared to grid-like WS networks ($\beta = 0$), ASNs tend to be less clustered with generally shorter path lengths. Compared to a fully–connected ANN of 500 nodes, ASNs display much higher clustering, and longer path lengths.

Using path length and clustering coefficient to estimate small–worldness Watts and Strogatz (1998), ASNs would fall in the small–world category, with relatively low path length and high clustering. Recently a measure called small–world propensity has been employed to consider potential drawbacks of the Watts–Strogatz method (see **Box 1**; Muldoon et al., 2016).

TABLE 1 | Mean and Standard Deviation (SD) for ASNs, WS random networks, WS grid-like networks of different sizes, and *C. elegans*.

	ASNs			WS $\beta = 1$ (Random)			WS $\beta = 0.5$			WS $\beta = 0$ (Grid-like)			*C. elegans*
Number of Nodes	100	500	1,000	100	500	1,000	100	500	1,000	100	500	1,000	277
Mean (PL)	3.65	2.96	2.77	2.74	2.05	1.94	2.8	2.11	1.95	8.76	8.27	7.64	2.69
SD (PL)	0.13	0.07	0.08	0.76	0.42	0.27	0.79	0.47	0.29	4.77	4.49	4.12	0.79
Mean (CCoeff)	0.39	0.39	0.4	0.05	0.06	0.07	0.1	0.13	0.14	0.6	0.73	0.74	0.2
SD (CCoeff)	0.01	0.01	0										
Mean (PCoeff)	0.22	0.29	0.31	0.57	0.85	0.87	0.52	0.7	0.7	0.2	0.16	0.17	0.41
SD (PCoeff)	0.23	0.22	0.22	0.17	0.03	0.02	0.18	0.09	0.07	0.21	0.2	0.2	0.25
Mean (MZ)	0	0	0	0	0	0	0	0	0	0	0	0	0
SD (MZ)	0.97	0.99	1	0.97	0.99	1	0.97	0.99	1	0.96	0.99	1	0.99
Mean (Small–world Prop)	0.69	0.68	0.67	0.29	0.29	0.29	0.29	0.29	0.29	0.37	0.37	0.37	0.55
SD (Small–world Prop)	0.04	0.02	0.01										

One-way ANOVAs were conducted to compare the small–world propensity of 100 nw ASNs with 100 node WS random-like, 100 node WS grid-like and *C. elegans* networks. There was a significant difference between groups [$F_{(3,10)} = 47.16$, $p < 0.001$] (where F is the ratio of mean square values of each group). *Post-hoc* analysis using the Bonferonni correction for multiple comparisons indicated that 100 nw ASNs had higher small–world propensity [*Mean* (M) $= 0.69$, *Standard Deviation* (SD) $= 0.04$] than random networks ($M = 0.29$, $SD = 0$) and grid-like networks ($M = 0.29$, $SD = 0$), but there was no significant difference between ASNs and *C. elegans* ($M = 0.55$, $SD = 0$; see **Supplementary Materials** for boxplots and multiple comparison graphs). We repeated these ANOVAs for 500 nw ASNs and 500 node WS networks [$F_{(3,9)} = 182.16$, $p < 0.001$]. *Post-hoc* analysis indicated that 500 nw ASNs ($M = 0.68$, $SD = 0.02$) had higher small–world propensity than random networks ($M = 0.29$, $SD = 0$), grid-like networks ($M = 0.29$, $SD = 0$) and *C. elegans* ($M = 0.55$, $SD = 0$). **Figure 5** shows a visual difference between ASNs and other networks.

3.2. Modularity and Integration

We used MZ and PCoeff measures to plot ASNs on a Guimerà and Amaral (2005) cartographic space (see **Figure 6A** for 100 and 500 nw ASN values, and **Supplementary Figure 5** for 1,000 and 2,000 nw ASN values). Briefly, this involves calculating the modular assignment of each node (see **Box 1**), and then estimating each nodes' topological role, relative to the modular assignment: high MZ = high within-node connection (segregation) and high PCoeff = high between-node connection (integration). When combined, these measures exhibit the modularity and hub characteristics of a network. Each region in this space classifies a node in a network as a specific type. Almost all the nodes in all sizes of ASNs were categorized as ultra-peripheral (PCoeff $= 0$), peripheral (MZ < 2.5, $0 <$ PCoeff < 0.62), and non-hub connector regions (MZ < 2.5, $0.62 <$ PCoeff < 0.80). There were also a very few nodes that fell in the provincial hub region (MZ > 2.5, and $0 <$ PCoeff < 0.30).

We compared the PCoeff and MZ of ASNs to both WS networks, and a biological system such as the *C. elegans* (see **Figure 6B**). One-way ANOVAs were conducted to compare the PCoeff and MZ of 100 nw ASNs with 100 node WS random-like, 100 node WS grid-like and *C. elegans* networks. There was a significant difference between groups [$F_{(3,1,517)} = 112.64$, $p < 0.001$] for PCoeff, but there was no significant difference for MZ. *Post-hoc* analysis using the Bonferonni correction for multiple comparisons indicated that 100 nw ASNs had lower PCoeff ($M = 0.22$, $SD = 0.23$) than *C. elegans* ($M = 0.41$, $SD = 0.21$) and random networks ($M = 0.57$, $SD = 0.17$), but there was no significant difference between ASNs and grid-like networks. We repeated these ANOVAs for 500 nw ASNs and 500 node WS networks, and the results were largely unchanged (see **Supplementary Materials**).

The structures of WS random-type networks tend to have higher PCoeff values (see **Table 1** for means and standard deviations), mainly in the PCoeff > 0.8 region (non-hub kinless

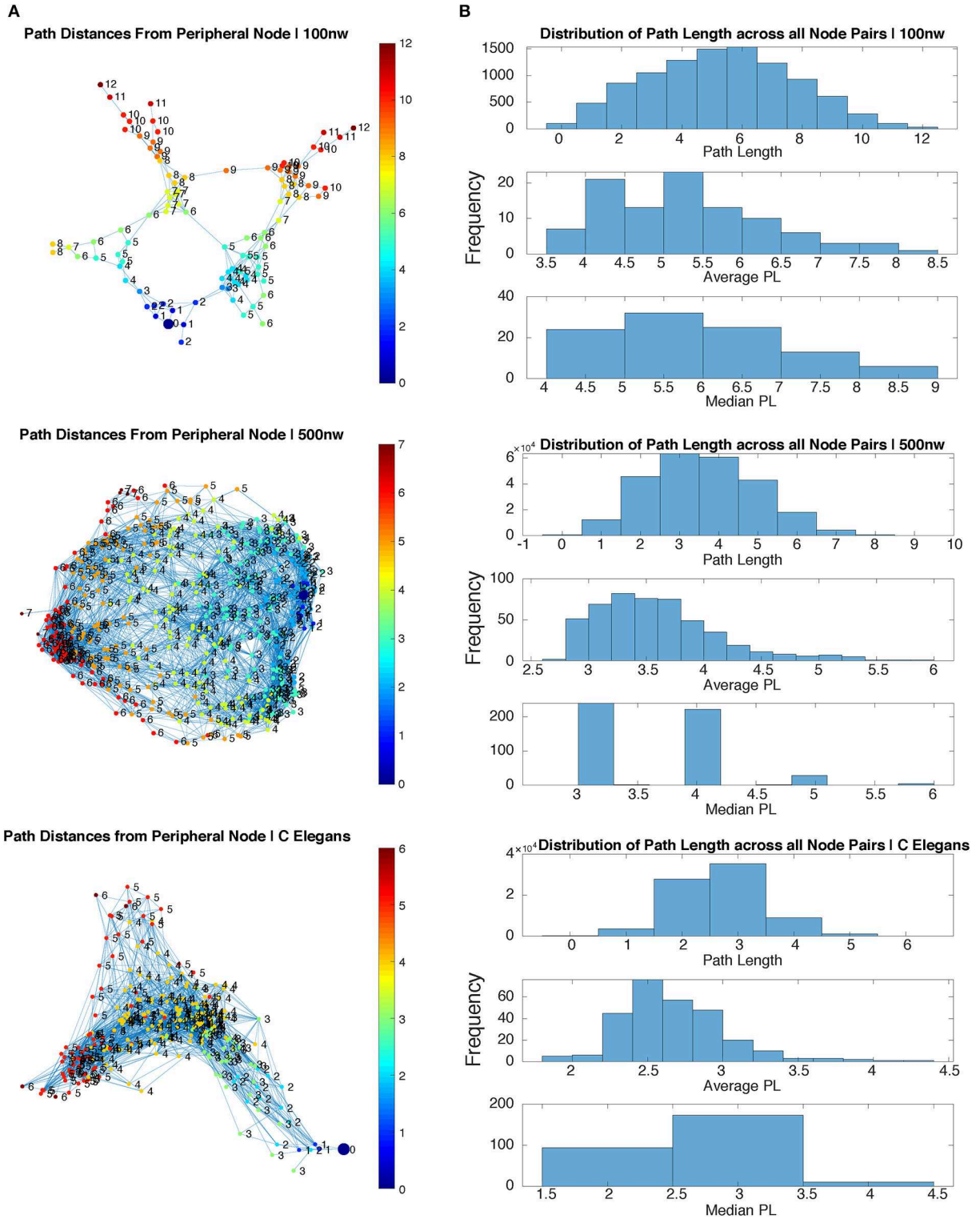


FIGURE 3 | Path length comparison of sample 100 and 500 nw ASNs with *C. elegans*. **(A)** Path length of each node from a randomly selected peripheral node for sample 100, 500 nw and *C. elegans* networks. **(B)** Distribution of path lengths from all node pairs in each sample network, including the average and median path length distributions, for 100, 500 nw and *C. elegans* networks.

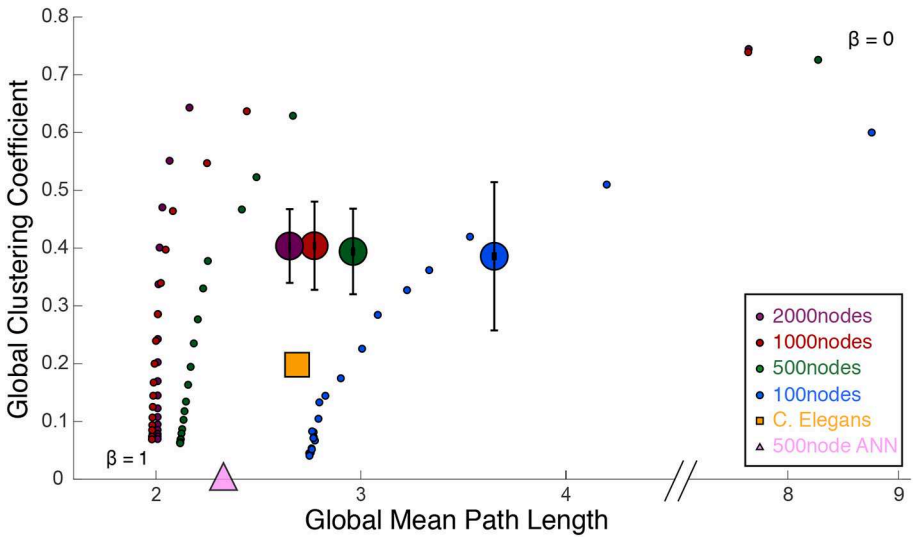


FIGURE 4 | Watts and Strogatz (1998) cartographic plane: global clustering coefficient and global mean path length. Each large dot represents 39 ASNs of varying parameters, with colors representing the network size (number of nanowire nodes). The small dots are Watts-Strogatz networks, rewired from completely random ($\beta = 1$) to grid-like ($\beta = 0$). Beta values decrease from bottom to top. The large yellow square measures the *C. elegans* network, and the large pink triangle is a 500-node fully-connected ANN.

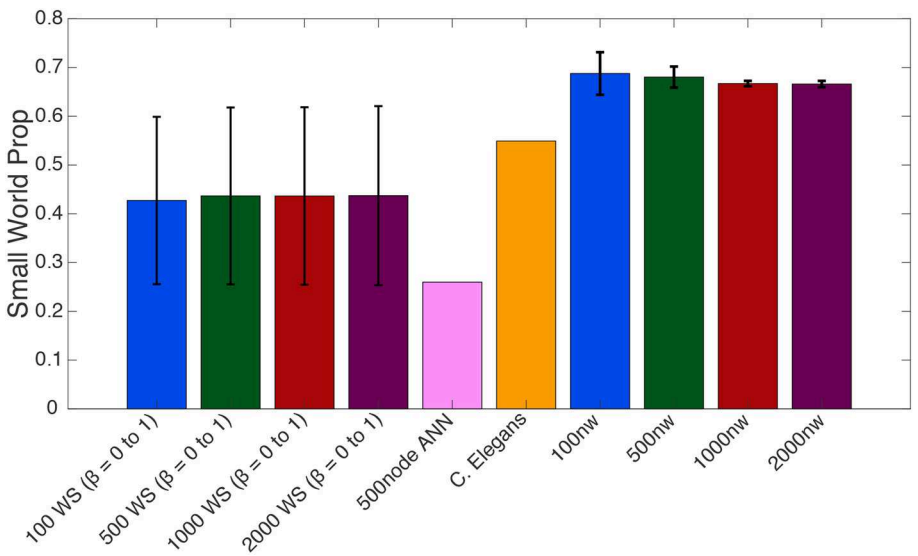


FIGURE 5 | Average small–world propensity values for Watts-Strogatz (WS), 500-node ANN, *C. elegans* and ASNs of varying sizes (number of nanowires, nw). Averages for WS were taken across all 21 β parameters from 0 to 1, with error bars reflecting standard deviation across β parameters. Averages for ASNs were taken from 39 networks with varying parameters as described in the methods section, with error bars reflecting standard deviation across network parameters.

nodes). They also have some examples of MZ $>$ 2.5 in the connector and kinless hub regions, but mainly MZ $<$ 2.5. WS grid-like networks have lower PCoeff values, typically limited to ultra-peripheral and peripheral regions. *C. elegans* networks cover a greater portion of the cartographic space, although most of the nodes tend to fall within the peripheral and non-hub regions (see Guimerà and Amaral, 2005 for more examples of biological PCoeff/MZ distributions).

4. DISCUSSION

ASNs exhibit a small–world structure, characterized by relatively short mean path length, alongside high clustering (Sporns et al., 2002; Sporns and Zwi, 2004). When compared with random or grid-like Watts–Strogatz networks, ASNs demonstrate more biological-like small–worldness features. While both random networks and ASNs have short path lengths, ASNs show higher

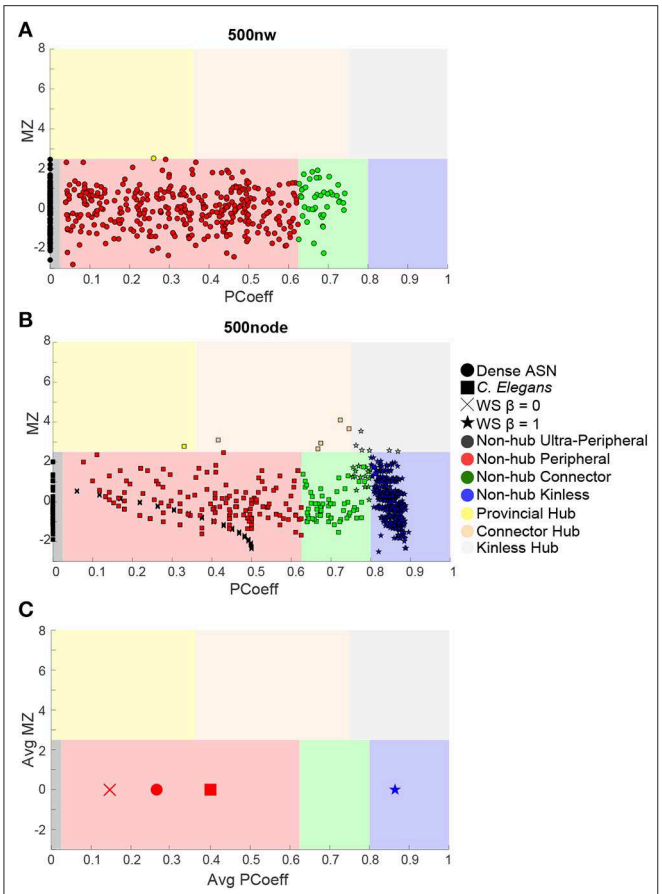


FIGURE 6 | Guimera-Amaral (2005) Cartographic-Space: Within-Module Degree z-Score (MZ) and Participant Coefficient (PCoeff). The dark-gray region (bottom left) refers to ultra-peripheral nodes (i.e., nodes with only one or few connections within a module); The light red region refers to peripheral nodes (i.e., nodes that are non-hubs and are only connected within a module); The green region refers to non-hub connector nodes (i.e., nodes that are connected to other modules but are not hubs); The blue region (bottom right) refers to non-hub kinless nodes (i.e., non-hub nodes not belonging to a particular module); The yellow region (top left) refers to provincial hubs (i.e., hub nodes that are highly connected within a module, but not between modules); The cream region refers to connector hubs (i.e., nodes that are highly connected between modules, but not necessarily within modules); the light-gray region (top right) refers to kinless hubs (i.e., hubs not belonging to a particular module). Regions are adapted from Guimerà and Amaral (2005). See **Box 1** for example nodes. **(A)** 500 nw ASN networks. Each colored dot is the mean MZ and PCoeff for the most dense (Avg DEG = 40.66) variations of ASNs. **(B)** 500 node WS networks and *C. elegans* networks. Squares represent the MZ and PCoeff for *C. elegans*. Stars and crosses represent the MZ and PCoeff values for WS networks with $\beta = 0$ and 1, respectively. **(C)** Average PCoeff and MZ scores for each type of network.

clustering. In studies on human neural networks, it has been suggested that a small–world network is ideal, for example, for synchronizing neural activity between brain regions (Latora and Marchiori, 2001; Reijneveld et al., 2007; Verweij et al., 2014). In turn, this reflects the capacity for high global efficiency of parallel information transfer between such regions (Bullmore and Sporns, 2009). ASNs may therefore have capacity for efficient, synchronized and parallel information transfer across the entire network, similar to that of biological systems.

However, the structure of wiring within and between regions/clusters, as highlighted by PCoeff and MZ measures, may be different from biological systems such as *C. elegans* (see **Supplementary Figure 3** for comparison with human node types). In biological neural networks, PCoeff and MZ are used to identify whether particular nodes play a hub-like role in the network. Hubs are central areas through which large amounts of information is trafficked to reach different parts of a network (van den Heuvel and Sporns, 2013). They are characterized by high connectivity to other network regions, as well as central positioning in the network. MZ scores have been used to denote hub status (e.g., z-Score < 2.5), while PCoeff values are used to classify the type of hub (Guimerà and Amaral, 2005; Joyce et al., 2010). Our results are consistent with previous studies showing that nodes in *C. elegans* have many peripheral and non-hub connector nodes, but also some hub-type provincial and connector nodes (Achacoso and Yamamoto, 1991; Guimerà and Amaral, 2005; Power et al., 2013). Such networks maintain a balance between integration and segregation of modules. In contrast, random WS networks are largely comprised of highly integrated, non-hub nodes, with a few hub-type nodes. This reflects a network with few modules. Grid-like WS networks are entirely comprised of non-hub, ultra-peripheral/peripheral type nodes, with very little integration or even modularity, as they have no connector nodes to connect between any modules that may exist.

How do ASNs fit within this space? Our results suggest that ASNs have a high proportion of peripheral, non-hub type nodes, similar to grid-like graphs. However, ASNs also have many non-hub connector nodes, which grid-like graphs lack. This means that ASNs are highly segregated, but also have many connections between modules, although they are weaker than within-module connections. Therefore they also have higher modularity than *C. elegans*. Random networks, on the other hand are highly integrated and have very few connector or peripheral nodes. Therefore ASNs have greater segregation than random networks, and higher modularity than both random and grid-like networks. ASNs involve a balance between integration and segregation, that is biased toward the presence of highly clustered, tight-knit modules with sparse inter-connectivity.

However the modularity and segregation of ASNs do not seem to reflect that of an organism like *C. elegans*. The nematode network has a greater balance between segregation and integration than ASNs, although likely with less modularity. Even ASNs that are highly dense only have a few hub-type nodes, meaning that most of the network's capacity to transfer information occurs in segregated modules, with sparse links between modules. Networks like the *C. elegans* would likely have fewer modules, with more central hub-type nodes that are responsible for directing information flow to and from the segregated modules of the network (hence the term hub).

Optimization of the structure of ASNs to represent biological-like networks may be desirable in the future, to allow for more biological-like capacities. For instance, increasing the size of the networks, and allowing for a greater balance between sparse and dense connections may allow for a more equivalent distribution of MZ and PCoeff scores, as well as increasing small–worldness even more. If these parameters are changed, it may be possible to

construct nanowire networks that are even more representative of a biological system. However, it may be that ASNs currently demonstrate similar functionality to biological organisms, but with a uniquely highly modular and segregated structure that has more emphasis on peripheral-type nodes. In such a case, nodes within a particular module or cluster may communicate more within that module than with nodes outside it, yet still produce dynamics that are similar to biological systems (Kuncic et al., 2018). Due to the high small–worldness that ASNs demonstrate, it may be possible that these types of networks place much less importance on hub-type nodes or regions, as many other small–world complex networks do (e.g., Guimerà and Amaral, 2005; van den Heuvel and Sporns, 2013; Verweij et al., 2014). We plan to investigate the functional connectivity of ASNs in a future study, to understand how similar the interplay between structure and function in these networks may be to biological systems and other real-world networks.

4.1. Conclusion

Neuromorphic nanowire networks demonstrate a small–world architecture that is similar to the biological system of *C. elegans*, and is distinct from random or grid-like networks (including untrained artificial neural networks). However, they also appear to be comprised of nodes that are equivalent to peripheral or non-hub nodes in a biological system, while being more segregated and modular, and less reliant on hubs of information flow. In future studies, investigating the functional properties of neuromorphic nanowire networks under electrical activation, coupled with altering the topology of these networks, will provide new insights into the interplay between structural and functional connectivity in a way that is extremely difficult experimentally. This may bring us closer to better understanding the physical components that may give rise to emergent dynamical behaviors of neural-network-like structures; behaviors that, in turn, enable cognitive functions such as learning and memory, or even intelligence.

DATA AVAILABILITY STATEMENT

The code used to generate the datasets analyzed for this study can be found on Github. Datasets are also available on request from the corresponding author.

AUTHOR CONTRIBUTIONS

Data collection, writing, and editing was conducted mainly by AL, with support from RZ, JH, ML, and KF. AD-A and TN helped with code and experimental verification of the simulation model. JS and ZK supervised the project and provided writing, editing, and feedback for the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.00184/full#supplementary-material

REFERENCES

Achacoso, T. B., and Yamamoto, W. S. (1991). *Ay's Neuroanatomy of C. elegans for Computation*, Vol. 1991. Boca Raton, FL: CRC Press.

Achard, S., Salvador, R., Whitcher, B., Suckling, J., and Bullmore, E. (2006). A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *J. Neurosci.* 26, 63–72. doi: 10.1523/JNEUROSCI.3874-05.2006

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., et al. (2015). "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 34, 1537–1557.

Albert, R., Jeong, H., and Barabási, A. L. (2011). "Diameter of the world-wide web," in *The Structure and Dynamics of Networks*, Vol. 9781400841356 (Princeton, NJ: Princeton University Press), 182.

Angeline, P. J. (1994). Genetic programming and emergent intelligence. *Adv. Genet. Programm* 1, 75–98.

Avizienis, A. V., Sillin, H. O., Martin-Olmos, C., Shieh, H. H., Aono, M., Stieg, A. Z., et al. (2012). Neuromorphic atomic switch networks. *PLoS ONE* 7:e42772. doi: 10.1371/journal.pone.0042772

Bassett, D. S., and Bullmore, E. (2006). Small-world brain networks. *Neuroscientist* 12, 512–523. doi: 10.1177/1073858406293182

Bassett, D. S., and Gazzaniga, M. S. (2011). Understanding complexity in the human brain. *Trends Cogn. Sci.* 15, 200–209. doi: 10.1016/j.tics.2011.03.006

Bertolero, M. A., Thomas Yeo, B. T., and D'Esposito, M. (2015). The modular and integrative functional architecture of the human brain. *Proc. Natl. Acad. Sci. U.S.A.* 112, E6798–E6807. doi: 10.1073/pnas.1510619112

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. U. (2006). Complex networks: Structure and dynamics. *Phys. Rep.* 424:175–308. doi: 10.1016/j.physrep.2005.10.009

Bullmore, E., and Sporns, O. (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nat. Rev. Neurosci.* 10, 186–198. doi: 10.1038/nrn2575

Bullmore, E., and Sporns, O. (2012). The economy of brain network organization. *Nat. Rev. Neurosci.* 13:336–349. doi: 10.1038/nrn3214

Burr, G. W., Shelby, R. M., Sebastian, A., Kim, S., Kim, S., Sidler, S., et al. (2017). Neuromorphic computing using non-volatile memory. *Adv. Phys. X* 2, 89–124. doi: 10.1080/23746149.2016.1259585

Chialvo, D. R. (2010). Emergent complex neural dynamics. *Nat. Phys.* 6, 744–750. doi: 10.1038/nphys1803

Chollet, F. (2019). On the Measure of Intelligence. *arXiv [Preprint]*. arXiv:1911.01547. Available online at: https://arxiv.org/abs/1911.01547

Chow, J. H., and Kokotovic, P. V. (1985). "Time scale modeling of sparse dynamic networks," in *IEEE Transactions on Automatic Control*, Vol. 30, 714–722.

Cohen, J. R., and D'Esposito, M. (2016). The segregation and integration of distinct brain networks and their relationship to cognition. *J. Neurosci.* 36, 12083–12094. doi: 10.1523/JNEUROSCI.2965-15.2016

Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

Diaz-Alvarez, A., Higuchi, R., Sanz-Leon, P., Marcus, I., Shingaya, Y., Stieg, A. Z., et al. (2019). Emergent dynamics of neuromorphic nanowire networks. *Sci. Rep.* 9:14920. doi: 10.1038/s41598-019-51330-6

Dumitrescu, A. T., Oltean, E., Merezeanu, D., and Dobrescu, R. (2017). "Emergence in hierarchical complex systems structured as social networks," in *Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017* (Bucharest: Institute of Electrical and Electronics Engineers Inc.), 426–431.

Eberle, A. L., and Zeidler, D. (2018). Multi-beam scanning electron microscopy for high-throughput imaging in connectomics research. *Front. Neuroanat.* 12:112. doi: 10.3389/fnana.2018.00112

Estrada, E., and Hatano, N. (2008). Communicability in complex networks. *Phys. Rev. E Stat.* 77:036111. doi: 10.1103/PhysRevE.77.036111

Eubank, S., Guclu, H., Kumar, V. S., Marathe, M. V., Srinivasan, A., Toroczkai, Z., et al. (2004). Modelling disease outbreaks in realistic urban social networks. *Nature* 429, 180–184. doi: 10.1038/nature02541

Farahani, F. V., Karwowski, W., and Lighthall, N. R. (2019). Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review. *Front. Neurosci.* 13:585. doi: 10.3389/fnins.2019.00585

Gilson, M., Kouvaris, N. E., Deco, G., Mangin, J. F., Poupon, C., Lefranc, S., et al. (2019). Network analysis of whole-brain fMRI dynamics: A new framework based on dynamic communicability. *NeuroImage* 201:116007. doi: 10.1016/j.neuroimage.2019.116007

Grayson, D. S., Bliss-Moreau, E., Machado, C. J., Bennett, J., Shen, K., Grant, K. A., et al. (2016). The rhesus monkey connectome predicts disrupted functional networks resulting from pharmacogenetic inactivation of the amygdala. *Neuron* 91, 453–466. doi: 10.1016/j.neuron.2016.06.005

Guimerà, R., and Amaral, L. A. (2005). Functional cartography of complex metabolic networks. *Nature* 433, 895–900. doi: 10.1038/nature03288

Hagberg, A., Swart, P., and Schult, D. A. (2008). "Exploring network structure, dynamics, and function using networkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, eds G. Varoquaux, T. Vaught, and J. Millman (Pasadena, CA), 11–15.

Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Van Wedeen, J., et al. (2008). Mapping the structural core of human cerebral cortex. *PLoS Biol.* 6:e159. doi: 10.1371/journal.pbio.0060159

Harary, F., and Norman, R. Z. (1953). *Graph Theory as a Mathematical Model in Social Science*. Ann Arbor, MI: University of Michigan, Institute for Social Research.

Joyce, K. E., Laurienti, P. J., Burdette, J. H., and Hayasaka, S. (2010). A new measure of centrality for brain networks. *PLoS ONE* 5:e12200. doi: 10.1371/journal.pone.0012200

Koch, C. (2004). *Biophysics of Computation: Information Processing in Single Neurons: Computational Neuroscience Series*. Oxford, UK:Oxford University Press, Inc. Available online at: https://dl.acm.org/doi/book/10.5555/1137840

Kuncic, Z., Marcus, I., Sanz-Leon, P., Higuchi, R., Shingaya, Y., Li, M., et al. (2018). "Emergent brain-like complexity from nanowire atomic switch networks: towards neuromorphic synthetic intelligence," in *Proceedings of the IEEE Conference on Nanotechnology*, Vol. 2018 (Cork: IEEE), 1–3.

Latora, V., and Marchiori, M. (2001). Efficient behavior of small-world networks. *Phys. Rev. Lett.* 87, 198701: doi: 10.1103/PhysRevLett.87.198701

Mason, O., and Verwoerd, M. (2007). Graph theory and networks in Biology. *IET Syst. Biol.* 1, 89–119. doi: 10.1049/iet-syb:20060038

MathWorks (2016). *Build Watts-Strogatz Small World Graph Model*. Available online at: https://au.mathworks.com/help/matlab/math/build-watts-strogatz-small-world-graph-model.html

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668– 673. doi: 10.1126/science.1254642

Muldoon, S. F., Bridgeford, E. W., and Bassett, D. S. (2016). Small-world propensity and weighted brain networks. *Sci. Rep.* 6:22057. doi: 10.1038/srep22057

Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J. K., and Aono, M. (2011). Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* 10, 591–595. doi: 10.1038/nmat3054

Oliveira, C. L., Morais, P. A., Moreira, A. A., and Andrade, J. S. (2014). Enhanced flow in small-world networks. *Phys. Rev. Lett.* 112:148701. doi: 10.1103/PhysRevLett.112.148701

Pascual-García, A. (2016). A topological approach to the problem of emergence in complex systems. *arXiv [Preprint]*. arXiv:1610.02448. Available online at: https://arxiv.org/abs/1610.02448

Power, J. D., Schlaggar, B. L., Lessov-Schlaggar, C. N., and Petersen, S. E. (2013). Evidence for hubs in human functional brain networks. *Neuron* 79, 798–813. doi: 10.1016/j.neuron.2013.07.035

Reijneveld, J. C., Ponten, S. C., Berendse, H. W., and Stam, C. J. (2007). The application of graph theoretical analysis to complex networks in the brain. *Clin. Neurophys.* 118, 2317–2331. doi: 10.1016/j.clinph.2007.08.010

Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. doi: 10.1038/s41586-019-1677-2

Rubinov, M., and Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage* 52, 1059–1069. doi: 10.1016/j.neuroimage.2009.10.003

Shepherd, G. M. (2004). *The Synaptic Organization of the Brain*. Oxford, UK: Oxford University Press.

Sporns, O., Tononi, G., and Edelman, G. M. (2002). Theoretical neuroanatomy and the connectivity of the cerebral cortex. *Behav. Brain Res.* 135, 69–74. doi: 10.1016/s0166-4328(02)00157-2

Sporns, O., Tononi, G., and Kötter, R. (2005). The human connectome: a structural description of the human brain. *PLoS Comput. Biol.* 1:e42. doi: 10.1371/journal.pcbi.0010042

Sporns, O., and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics* 2, 145–162. doi: 10.1385/NI:2:2:145

Stieg, A. Z., Avizienis, A. V., Sillin, H. O., Martin-Olmos, C., Aono, M., and Gimzewski, J. K. (2012). Emergent criticality in complex turing B-type atomic switch networks. *Adv. Mater.* 24, 286–293. doi: 10.1002/adma.201103053

Strogatz, S. H. (2001). Exploring complex networks. *Nature* 410, 268–276. doi: 10.1038/35065725

Tahmassebi, A., Pinker-Domenig, K., Wengert, G., Lobbes, M., Stadlbauer, A., Romero, F. J., et al. (2017). "Dynamical graph theory networks techniques for the analysis of sparse connectivity networks in dementia," in *Proceedings Volume 10216, Smart Biomedical and Physiological Sensor Technology XIV* (Anaheim, CA), 1021609.

Terabe, K., Hasegawa, T., Nakayama, T., and Aono, M. (2005). Quantized conductance atomic switch. *Nature* 433, 47–50. doi: 10.1038/nature03190

van den Heuvel, M. P. and Sporns, O. (2013). Network hubs in the human brain. *Trends Cogn. Sci.* 17, 683–696. doi: 10.1016/j.tics.2013.09.012

Van Diessen, E., Zweiphenning, W. J., Jansen, F. E., Stam, C. J., Braun, K. P., and Otte, W. M. (2014). Brain network organization in focal epilepsy: a systematic review and meta-analysis. *PLoS ONE* 9:e114606. doi: 10.1371/journal.pone.0114606

Verweij, I. M., Romeijn, N., Smit, D. J., Piantoni, G., Van Someren, E. J., and van der Werf, Y. D. (2014). Sleep deprivation leads to a loss of functional connectivity in frontal brain regions. *BMC Neurosci.* 15:88. doi: 10.1186/1471-2202-15-88

Vianello, E., Garbin, D., Bichler, O., Piccolboni, G., Molas, G., Salvo, B. D., et al. (2017). *Advances in Neuromorphic Hardware Exploiting Emerging Nanoscale Devices, Vol. 31, Cognitive Systems Monographs*. New York, NY: Springer India.

Wakabayashi, H., and Iida, Y. (1992). Upper and lower bounds of terminal reliability of road networks:an efficient method with boolean algebra. *J. Natural Disaster Sci.* 14, 29–44.

Watts, D. J., and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature* 393, 440–442. doi: 10.1038/30918

White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philos. Trans. R. Soc. B Biol. Sci.* 314, 1–340. doi: 10.1098/rstb.1986.0056

Xu, X., Ding, Y., Hu, S. X., Niemier, M., Cong, J., Hu, Y., et al. (2018). Scaling for edge inference of deep neural networks perspective. *Nat. Electr.* 1, 216–222. doi: 10.1038/s41928-018-0059-3

Yan, G., Vértes, P. E., Towlson, E. K., Chew, Y. L., Walker, D. S., Schafer, W. R., et al. (2017). Network control principles predict neuron function in the *Caenorhabditis elegans* connectome. *Nature* 550, 519–523. doi: 10.1038/nature24056

Yang, J. J., Strukov, D. B., and Stewart, D. R. (2013). Memristive devices for computing. *Nat. Nanotechnol.* 8, 13–24. doi: 10.1038/nnano.2012.240

Ziegler, M., Wenger, C., Chicca, E., and Kohlstedt, H. (2018). Tutorial: concepts for closely mimicking biological learning with memristive devices: principles to emulate cellular forms of learning. *J. Appl. Phys.* 124: 152003. doi: 10.1063/1.5042040

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Loeffler, Zhu, Hochstetter, Li, Fu, Diaz-Alvarez, Nakayama, Shine and Kuncic. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 24 March 2020
doi: 10.3389/fnins.2020.00240

Equilibrium Propagation for Memristor-Based Recurrent Neural Networks

Gianluca Zoppo, Francesco Marrone and Fernando Corinto*

Department of Electronics, Politecnico di Torino, Turin, Italy

Among the recent innovative technologies, memristor (memory-resistor) has attracted researchers attention as a fundamental computation element. It has been experimentally shown that memristive elements can emulate synaptic dynamics and are even capable of supporting spike timing dependent plasticity (STDP), an important adaptation rule that is gaining particular interest because of its simplicity and biological plausibility. The overall goal of this work is to provide a novel (theoretical) analog computing platform based on memristor devices and recurrent neural networks that exploits the memristor device physics to implement two variations of the backpropagation algorithm: recurrent backpropagation and equilibrium propagation. In the first learning technique, the use of memristor–based synaptic weights permits to propagate the error signals in the network by means of the nonlinear dynamics via an analog side network. This makes the processing non-digital and different from the current procedures. However, the necessity of a side analog network for the propagation of error derivatives makes this technique still highly biologically implausible. In order to solve this limitation, it is therefore proposed an alternative solution to the use of a side network by introducing a learning technique used for energy-based models: equilibrium propagation. Experimental results show that both approaches significantly outperform conventional architectures used for pattern reconstruction. Furthermore, due to the high suitability for VLSI implementation of the equilibrium propagation learning rule, additional results on the classification of the MNIST dataset are here reported.

Keywords: artificial neural network, biologically plausible learning rule, neuromorphic computing, recurrent neural network, associative memory, memristor

OPEN ACCESS

Edited by:
Elisa Donati,
ETH Zurich, Switzerland

Reviewed by:
Adrien F. Vincent,
Institut Polytechnique de Bordeaux, France
Toshiyuki Yamane,
IBM Research, Japan

****Correspondence:***
Gianluca Zoppo
gianluca.zoppo@polito.it

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *30 September 2019*
Accepted: *03 March 2020*
Published: *24 March 2020*

Citation:
Zoppo G, Marrone F and Corinto F (2020) Equilibrium Propagation for Memristor-Based Recurrent Neural Networks. Front. Neurosci. 14:240. doi: 10.3389/fnins.2020.00240

INTRODUCTION

In the last few decades, the search of innovative computing platforms that could offer new, ultra-low power processing methods and architectures has intensified. Neuromorphic computing approaches aim to go beyond the state-of-the-art in conventional digital processing by exploiting complex dynamics and nonlinear phenomena emerging from the physics of nonvolatile memory devices (e.g., memristors) (Chua, 1971; Strukov et al., 2008). The hallmark of this kind of devices is the peculiar analog signal storing capability that allows them to mimic the behavior of neural synapses. The processing is not only analog and different from current digital processors, but also enhances computing speed and power efficiency for large sets of sensor data. This has been achieved by combining memristor technology with advanced deep learning algorithms used to train neural networks. In supervised learning, one of the most popular

method used for training feedforward neural networks is the backpropagation algorithm. Although it is considered a powerful technique, it is computationally expensive and is commonly labeled as biologically implausible. The generalization of this rule to continuous-time recurrent networks was first introduced by Almeida (1987) and Pineda (1988) who independently obtained the same results. Recurrent backpropagation aims to iteratively adjust the weight matrix of the network in order to let the system converge, for fixed input and initial state, to a desired attractor. As for feedforward neural networks, this is achieved by minimizing a particular loss function associated to the system parameters with the difference that the error signal is now backpropagated by introducing an associated differential equation. This allowed to avoid the direct gradient's computations and reduced the large number of required multiplications. However, the necessity of a side network for the propagation of error derivatives makes this technique still highly different from emulating the brain complex computation. This hypothesis is further supported by the fact that there is no known mechanism that could explain how an error message is propagated backwards through the same pathway of the incoming signal. Recently, Scellier and Bengio (2017) proposed an alternative solution to the use of a side network by introducing Equilibrium Propagation, a learning technique used for energy-based models. The advantage of this approach is indeed the requirement of just one kind of neural computation for the training phase of the network. Firstly, inputs are clamped and the network relaxes to a fixed point which corresponds to a local minimum of the energy function. Secondly, after introducing a small external error signal, the network relaxes to a new but close-by fixed point which now corresponds to a rather lower cost value. Even though the two methods seem quite different, it is easy to observe that both share the same goal, finding low-energy configurations that have low cost values. The aim of this work is to propose a novel (theoretical) analog computing platform based on memristor devices and recurrent neural networks that exploits the memristor device physics to implement two variations of the backpropagation algorithm. In the first section, it is provided a brief introduction on memristors and their peculiar properties useful for the physical implementation. In the second section, a general introduction on biological algorithms is presented with particular attention on recurrent backpropagation and equilibrium propagation. In the last section, the two techniques are compared with the existing algorithms used in pattern reconstruction providing results of their compelling efficiency. Lastly, it is shown the application of a memristor-based recurrent neural network trained with equilibrium propagation used for the classification of a small subset of the MNIST dataset. The choice of using only this learning rule was mainly dictated by the fact that using a side network, as in the recurrent backpropagation approach, would at least double the required IC area.

MEMRISTOR–BASED RECURRENT NEURAL NETWORK

Massive progress has already been made with neuromorphic systems based on traditional analog and digital integrated circuits. Among all the recent alternatives which aim to emulate neurobiological components and functions, memristive devices have drawn particular attention (Jo et al., 2010). Memristors, often termed as Resistive Switching devices, are single-port electrical dynamical systems whose conduction properties depend on the history of applied input at the port (Chua and Sung Mo, 1976). The typical memristor physical implementation consists of two metal electrodes sandwiching a switching material. An intuitive connection links these two electrodes to the corresponding role of axons and dendrites and the switching layer to the variable interconnection weight of synapses. The crossbar architecture is probably the most commonly used computing structure exploiting the memristive behavior for mapping neural networks in hardware. Its basic working principle is the application of Kirchhoff's Current Law to compute the input to the i-*th* neuron as the algebraic sum of the weighted inputs $I_i = \sum_j G_{ij} v_j$. Here, I_i is the i-th input current, G_{ij} is the connecting memductance between the i-th and j-th neurons and v_j is the output voltage generated by the j-th neuron. This produces the vector-matrix multiplication *in situ* by a single read operation which eliminates the need for constant bidirectional data transfer from the memory to the computing unit (Sun et al., 2019).

The most peculiar characteristic of this kind of devices is the synaptic plasticity effect which is also observed in biological neural systems. Since conductances can be tuned by controlling the coordinated activity of pre- and post-synaptic neurons, memristor-based neural networks can consequently emulate neurobiological phenomena while mimicking the underlying learning process. From neurological studies, it turned out that the neural coding is highly dynamic, therefore recurrent neural networks seem well suited to model similar behavior and have been used to investigate the mechanisms adopted by neurons populations in solving various complex tasks.

For this reason, consider a recurrent neural network and let each synaptic weight be described by a generic memristor (see also Corinto et al., 2015; Leon, 2015) that satisfies the following equations:

$$\begin{cases} i = G(\mathbf{x})v \\ \frac{d\mathbf{x}}{dt} = f(\mathbf{x}, v) \end{cases} \tag{1}$$

where i is the current, v is the voltage, $G(\cdot)$ is the memductance and $\mathbf{x}$ is the internal state vector. Let the memristor–based synaptic weight be $G(\mathbf{x}) = w$, with the purpose of giving a formal description of the network's learning process, the state vector can be defined by the two following dynamics:

$$\begin{cases} i = wv \\ \frac{dw}{dt} = f_1(w, v, y) \\ \frac{dy}{dt} = f_2(w, v, y) \end{cases} \qquad \begin{cases} i = wv \\ \frac{dw}{dt} = g(w, v) \end{cases} \tag{2}$$

In the next sections, the derivation of the previously mentioned variants of the backpropagation algorithm for recurrent neural network is given in order to clarify the use of the different choice of the state vector $\mathbf{x}$. Further work is still needed to

find physical devices that approximate the proposed memristive synapse dynamics in (2). Many models of memristor devices (e.g., Phase Change Memory, Resistive NonVolatile Memory, etc.) have been presented during the last decade but unfortunately the existing mathematical representations are not suitable for this kind of investigation. The current approach available in literature is to embed memristor device in suitable synaptic circuit so that the dynamics of internal (state) variables can be controlled by appropriate pulses. Thus, the learning rules can be implemented by a series of discrete programming pulses that perform the weights update according to the learning rules defined by the recurrent backpropagation and the equilibrium propagation algorithms. This can be obtained by means of amplitude/duration modulation of a voltage (or current) pulse applied on a physical device via the 1T–1R (one transistor–one memristor) architecture (see Liu et al., 2015; Merced-Grafals et al., 2016). An alternative approach is based on the use of emulator of generic memristors (Ascoli et al., 2016; Assaf et al., 2019) such that the dynamics in (2) can be obtained. Although the physical realization of memristor synapses is a challenging problem, its investigation is out of the scope of the present work that aims to show how memristor–based recurrent neural networks with memristor synapses support equilibrium propagation algorithms. A further study will be devoted to tackle the implementation of proposed memristor synapses.

BIOLOGICALLY-PLAUSIBLE LEARNING ALGORITHMS

The rules that govern the learning process in the brain are poorly understood. Despite the great success of deep learning in a wide variety of complex tasks (LeCun et al., 2015), learning rules in the brain are most likely local and strictly feedforward. Theoretical analysis of biological neural networks showed indeed that connections between neurons are mostly strengthened depending on the coordinated activity of pre-synaptic and post-synaptic cells rather than computations of all downstream neurons (Hebb, 1949; Gerstner et al., 2014). Therefore nowadays, there is an increasing interest in machine learning and computational neuroscience in the study of neuron-like architecture with local learning rules that aim to approximate the surprising efficiency of the backpropagation training process. Many bio-plausible approaches include feedback allignment (Lillicrap et al., 2016), target propagation algorithms (Lee et al., 2015), membrane potential based backpropagation algorithm (Lee et al., 2016), equilibrium propagation (Scellier and Bengio, 2017), etc. See for example Whittington and Bogacz (2019) for an extensive review. Since neurological research suggests that the neural representation is highly dynamic, models based on recurrent neural networks seem well suited to capture similar behavior and therefore have been used to investigate the mechanisms by which neural populations solve various computational problems. In order to take advantage of the intrinsic nonlinear dynamics of the system, two learning techniques for continuous time recurrent neural networks were mainly considered: recurrent backpropagation and equilibrium propagation. Even though the latter shows a more suitable affinity for VLSI implementations (Scellier and Bengio, 2017), the former represents the first attempt in approaching energy-based models from a supervised point of view and therefore is worth being mentioned and compared. In the next subsections, it is provided a brief introduction to the construction and derivation of both algorithms.

Recurrent Backpropagation

Consider a Recurrent Neural Network (RNN) whose state vector $\mathbf{v}$ evolves according to:

$$\frac{dv_i}{dt} = -v_i + g_i\left(\sum_{j=1}^{N} w_{ij}v_j + I_i\right), \qquad i = 1, \ldots, N \tag{3}$$

where N is the number of neurons of the network and I_i is an external input to the i-th neuron. There is no restriction on the choice of the activation function g_i as long as it is monotone and differentiable (Pineda, 1988). In the most general case, neurons can be considered either as input, output or hidden units depending on the application. The goal of the algorithm is to adjust the weights w_{ij} so that, for a given initial condition $\mathbf{v}^0 = \mathbf{v}(t_0)$ and a given vector of input $\mathbf{I}$, the RNN (3) converges to a desired fixed point $\mathbf{v}^\infty = \mathbf{v}(t_\infty)$. This is obtained by minimizing a loss function E which measures the euclidean distance between the desired fixed point and the actual fixed point:

$$E = \frac{1}{2}\sum_{i=1}^{N} J_i^2 = \frac{1}{2}\sum_{i=1}^{N}(T_i - v_i^\infty)^2 \tag{4}$$

where T_i is the i-th desired output state component and J_i is the i-th component of the difference between the current fixed point v_i^∞ and the target point T_i. Observe that E depends on the weight matrix $\mathbf{W}$ through the fixed point $\mathbf{v}^\infty(\mathbf{W}, \mathbf{I})$. Therefore, one way to drive the system to converge to a desired attractor is to let it evolve in the weight parameter space along trajectories which have opposite direction of the gradient of E:

$$\frac{dw_{ij}}{dt} = -\eta\frac{\partial E}{\partial w_{ij}} = \eta\sum_{k=1}^{N} J_k\frac{\partial v_k^\infty}{\partial w_{ij}}, \qquad \eta > 0 \tag{5}$$

where η is the learning rate. The derivative of v_k^∞ with respect to w_{ij} is derived by observing that the fixed points of (3) must satisfy the nonlinear equation:

$$v_k^\infty = g_k\left(\sum_{s=1}^{N} w_{ks}v_s^\infty + I_k\right). \tag{6}$$

Differentiating (6) with respect to w_{ij} one obtains (for more details see the **Appendix**):

$$\frac{\partial v_k^\infty}{\partial w_{ij}} = (\delta_{ki} - g_k'(\hat{I}_k^\infty)w_{ki})^{-1}g_i'(\hat{I}_i^\infty)v_j^\infty \tag{7}$$

where δ_{ki} is the kronecker delta. Unfortunately, (7) requires the computation of a reciprocal for computing the weights'

update and therefore Pineda (1988) bypassed this problem by considering

$$y_i = g_i'(\hat{I}_i^{\infty}) \sum_{k=1}^{N} J_k(\delta_{ki} - g_k'(\hat{I}_k^{\infty}) w_{ki})^{-1} \tag{8}$$

which can be seen as the steady state of the following side network:

$$\frac{dy_k}{dt} = -y_k + g_k'(\hat{I}_k^{\infty}) \left(\sum_{i=1}^{N} w_{ik} y_i + J_k \right). \tag{9}$$

In conclusion, the weights' update rule is defined by:

$$\frac{dw_{ij}}{dt} = \eta y_i^{\infty} v_j^{\infty} \tag{10}$$

which is therefore dependent on the corresponding fixed points of the dynamical systems (3) and (9). Here is the summary of the whole learning process:

1) Firstly, (3) evolves starting from a random initial condition and converges to the corresponding fixed point $\mathbf{v}^{\infty}$;
2) Secondly, (9) evolves starting again from a random initial condition and converges to the corresponding fixed point $\mathbf{y}^{\infty}$;
3) Lastly, the weights of the matrix $\mathbf{W}$ are updated according to

$$\Delta w_{ij} = \eta y_i^{\infty} v_j^{\infty}, \qquad \eta > 0. \tag{11}$$

Equilibrium Propagation

Consider now the following energy function E:

$$E(\mathbf{v}) = \sum_{i=1}^{N} \frac{v_i^2}{2} - \frac{1}{2} \sum_{i,j=1}^{N} w_{ij} g_i(v_i) g_j(v_j) - \sum_{i=1}^{N} I_i g_i(u_i) \tag{12}$$

where N is the number of neurons of the network and I_i is an external input to the i-th neuron. Again, there is no restriction on the choice of the activation functions $g_i(\cdot)\ \forall i = 1, \ldots, N$ as long as they are differentiable and monotone. Assume that the time evolution of the state variable $\mathbf{v}$ is governed by the gradient dynamics:

$$\frac{dv_i}{dt} = -\frac{\partial E}{\partial v_i} = -v_i + g_i'(v_i) \left(\sum_{j=1}^{N} w_{ij} g_i(v_j) + I_i \right), \qquad i = 1, \ldots, N \tag{13}$$

Observe that, the network is recurrently connected with symmetric connections (i.e., $w_{ij} = w_{ji}$). Typically in the supervised learning framework, the output units aim to recreate their targets $\mathbf{T}$. The deviation of the fixed points $\mathbf{v}^{\infty}$, output values of the network, from the targets $\mathbf{T}$ is measured by the quadratic loss function:

$$C = \frac{1}{2} \sum_{i=1}^{N} (T_i - v_i)^2 \tag{14}$$

Observe that this function is defined for any state of $\mathbf{v}$. The central idea of Equilibrium Propagation is to introduce the augmented energy function:

$$F(\mathbf{v}, \mathbf{W}, \mathbf{T}) = E(\mathbf{v}, \mathbf{W}) + \beta C(\mathbf{v}, \mathbf{W}, \mathbf{T}) \qquad \mathbf{v}, \mathbf{T} \in \mathbb{R}^N, \mathbf{W} \in \mathbb{R}^{N \times N}, \beta \geq 0 \tag{15}$$

and replace the free dynamics with the augmented dynamics:

$$\frac{dv_i}{dt} = -\frac{\partial F}{\partial v_i} \tag{16}$$

Here, the second term $-\beta \frac{\partial C}{\partial v_i}$ gradually pushes $\mathbf{v}$ toward configurations that have lower cost values. This is done, as in the previous model, by simply adjusting $\mathbf{W}$ so as to minimize the cost value of the fixed point. Now, in order to derive the corresponding learning rule, let us introduce the following objective function

$$J(\mathbf{W}) = C(\mathbf{v}^{\infty}, \mathbf{W}, \mathbf{T}) \qquad \mathbf{v}, \mathbf{T} \in \mathbb{R}^N, \mathbf{W} \in \mathbb{R}^{N \times N} \tag{17}$$

Observe that $J(\mathbf{W})$ is the cost at the fixed point. The equilibrium propagation algorithm estimates the gradient $\frac{\partial J}{\partial \mathbf{W}}$ based on measures at the fixed points of the free and the augmented dynamics that we will set as $\mathbf{v}^{\infty}$ and $\mathbf{v}_{\beta}^{\infty}$. Scellier and Bengio (2017), indeed, proved the following statement:

$$\frac{\partial J}{\partial \mathbf{W}} = \lim_{\beta \to 0} \frac{\frac{\partial F}{\partial \mathbf{W}}(\mathbf{v}_{\beta}^{\infty}) - \frac{\partial F}{\partial \mathbf{W}}(\mathbf{v}^{\infty})}{\beta} \tag{18}$$

offering an alternative way to estimate the gradient of the objective function. Therefore, the network follows the following dynamics for the training phase:

1) Firstly, $\mathbf{T}$ is clamped and the network follows the free dynamics (13) relaxing to the free fixed point $\mathbf{v}^{\infty}$ where $\frac{\partial F}{\partial \mathbf{W}}(\mathbf{v}^{\infty})$ is measured (free phase);
2) Secondly, the influence parameter is introduced and the network relaxes to a new but nearby fixed point $\mathbf{v}_{\beta}^{\infty}$ where $\frac{\partial F}{\partial \mathbf{W}}(\mathbf{v}_{\beta}^{\infty})$ is measured (weakly clamped phase).
3) Lastly, the weights of the matrix $\mathbf{W}$ are changed according to (18) and updated as follows:

$$\Delta w_{ij} \propto + \eta [g_i(v_i^{\beta,\infty}) g_j(v_j^{\beta,\infty}) - g_i(v_i^{\infty}) g_j(v_j^{\infty})], \qquad \eta > 0. \tag{19}$$

IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this section, it is first provided an experimental evidence of the two models' efficiency in a pattern reconstruction task. Afterwards, an example on the classification of a subset of MNIST dataset is here reported using equilibrium propagation as learning rule.

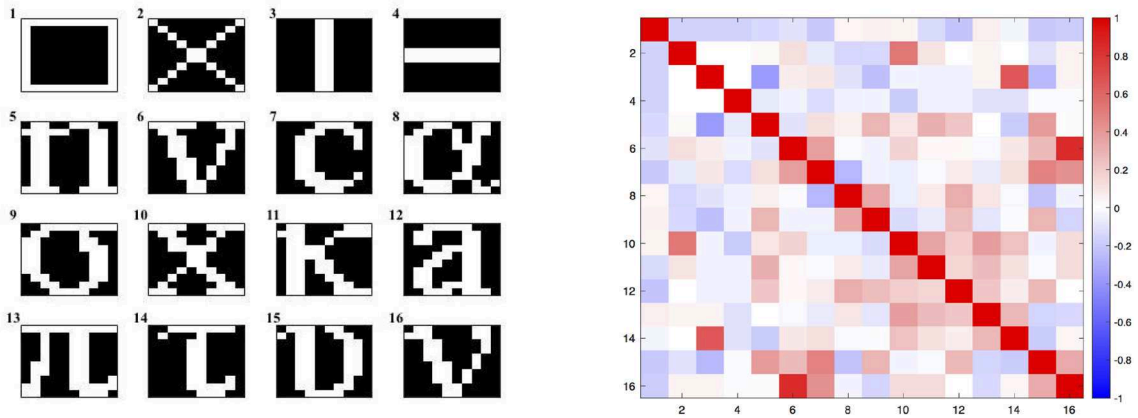


FIGURE 1 | In the left panel, the dataset of all the 16 patterns to be learnt. In the right panel, a graphical representation of the corresponding patterns' correlation matrix computed with the Pearson correlation coefficient.

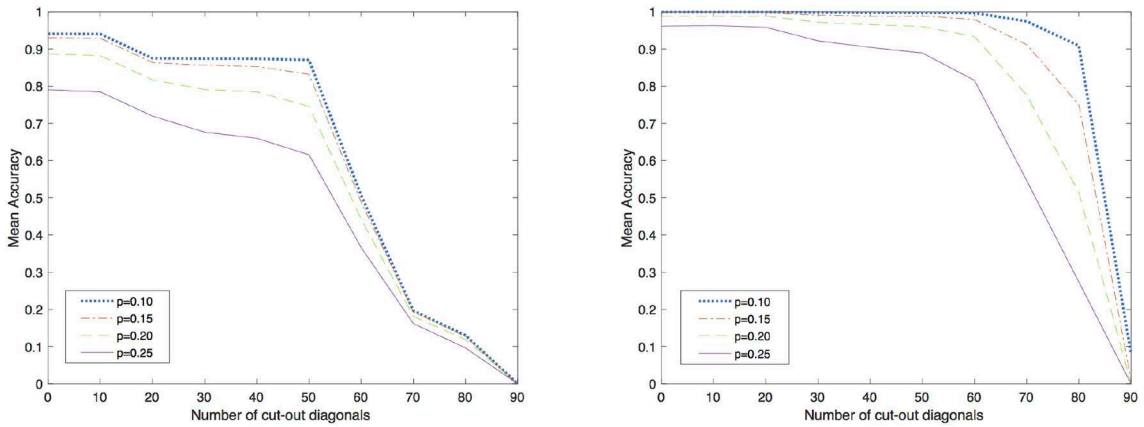


FIGURE 2 | Accuracy for different radius of connectivity. On the left, the results obtained by using Recurrent Backpropagation and on the right, the results obtained by using Equilibrium Propagation. *p* is the probability of flipping each pixel of the image from white to black and viceversa.

Pattern Reconstruction

In this section, a comparison between the two aforementioned training algorithms for pattern's reconstruction task is presented. For this kind of application, input units are chosen to be simultaneously output units and no hidden units are considered. Moreover, due to the construction of the gradient dynamics (13), symmetric weights were chosen for both methods. This condition also guarantees the convergence of the model (3). During the training phase, each image shown in **Figure 1** is repeatedly proposed to the network by means of a constant input $\mathbf{I}$ until it is memorized. In the case of multiple patterns to be learnt, the previous steps are performed for each single image of the dataset for different epochs. Here patterns were shown to the network in the same order for each epoch but this choice was not restrictive since similar performances were obtained even in the case the images were proposed in a random fashion. In order to train the network, the following hyperparameters and initial conditions were set for the training phase:

- Random initialization of the state variable $\mathbf{v}$;
- In the recurrent backpropagation case, each single time the first state variable converges, the second variable is reset to $\mathbf{y}(0) = (0.5, \ldots, 0.5)^T \in \mathbb{R}^N$;
- The matrix $\mathbf{W}$ is symmetric and initialized with uniform random values between $[-0.1; 0.1]$;
- The activation functions $g_i \; \forall i \; = \; 1, \ldots, N$ are hyperbolic tangent functions;
- The learning parameter $\eta = 0.01$;
- The number of epoch is 300.
- Time spans for the simulation of the dynamics systems are chosen in order to guarantee the convergence of the state variables.

With the aim of assessing the applicability of this method in VLSI implementation, a short analysis on the importance of local-global connections of the network's neurons was performed. For further details on the relation of the topology and the computational performance of attractor neural networks refer

to McGraw and Menzinger (2003), Hasler and Marr (2013) or Stauffer et al. (2003), Tanaka et al. (2019) for additional results on current approaches for enhancing the energy efficiency of hardware-level neural networks by means of sparse and less costly number of connections. Here, for sake of simplicity, a simpler investigation was carried out by increasingly disconnecting global connections arising from a full matrix by simply setting to zero all the elements that were located outside a band about the main diagonal. In order to test the network, corrupted patterns were created by flipping, with probability p, each pixel of the image from white to black and viceversa. The cut of K outer diagonals from the matrix reduces the number of synapses from N^2 to $N^2 - K(K+1)$. In this analysis, a corrupted pattern is recognized as reconstructed if the least square error with respect to the original images is equal to 0. The validation was carried out by testing the recovery capabilities of the network against 5000 corrupted patterns for each class shown in **Figure 1**. The results obtained by both methods are shown in **Figure 2** with different levels of test images' corruption (e.g., $p = 0.10, p = 0.15$, $p = 0.20$, and $p = 0.25$). It is easy to see that both methods seem to reach promising and equally meaningful results in the case of fully connected networks. However, Equilibrium Propagation is able to get better results even with a small amount of connections. This fact, together with the absence of a side network really motivates us to investigate this method as a solution worth to be considered for a VLSI implementation. This improvement might be induced by the noisy estimator of the gradient given by (19) that helps the network to efficiently explore the parameter space by avoiding to get stuck in local minima. This might be further seen in **Figure 3** where good values of accuracy are already obtained by Equilibrium Propagation in the first 50 epochs whereas Recurrent Backpropagation needs at least 300 epochs. In last analysis, in order to assess the efficiency of the two novel methods, it is additionally performed a comparison with two of the most used learning rules for training networks in associative memory's tasks. It is well known that a standard Hopfield model trained on uncorrelated patterns with the Hebbian rule has an approximate capacity of $0.14N$ (N is the number of units in the network) (McEliece et al., 1987). Unfortunately, this capacity decreases significantly if patterns are correlated. To overcome this problem, a novel learning method has been introduced by Storkey (1997). The Storkey learning rule presents indeed a significantly improved performance over the standard Hopfield model, both with correlated and uncorrelated data.
However, as shown in **Table 1** and in the examples of **Figure 4**, the results provide evidence that both recurrent backpropagation and the equilibrium propagation algorithms are perfectly able to reconstruct even in the presence of correlated patterns.

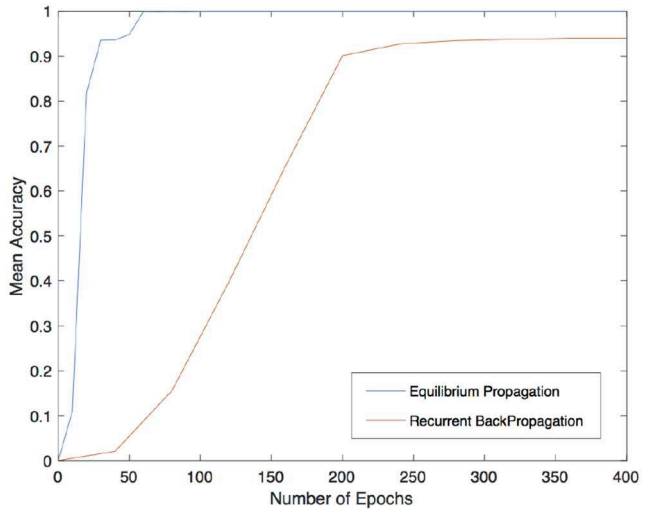


FIGURE 3 | Mean accuracy over 1000 reconstructed patterns for different number of epochs using Equilibrium Propagation (in blue) and Recurrent Backpropagation (in orange).

Pattern Classification

As a second experimental result, it is now provided an application of the model introduced by Scellier and Bengio (2017) in a pattern classification task of a subset of the MNIST dataset: 5 classes and 600 patterns for each class. The model used here is still a recurrent neural network with symmetric connections, 1 hidden layer, no skip-layer and no lateral connections. Following Scellier and Bengio (2017), a hard sigmoid was chosen as activation function and the training process was performed by iterating the successive steps:

1) Fix the pattern as a constant input;
2) Run the free phase until convergence of the hidden and the outputs units may be reached and collect $g(v_i^{\infty})g(v_j^{\infty})$;
3) Run the weakly clamped phase until convergence and collect $g(v_i^{\beta,\infty})g(v_j^{\beta,\infty})$;
4) Update the synaptic weights according to (19).

In order to perform the training process, (16) was first discretized into short time lapses of duration ϵ as follows:

$$v_{t+1} = v_t - \epsilon \frac{\partial F}{\partial v_i} \tag{20}$$

However, as suggested by Scellier and Bengio (2017), the state variable should be bounded between 0 and 1 and therefore a

TABLE 1 | Accuracy for each single learning rule over 1,000 corrupted images, with probability 0.1, for each of the 16 classes.

	Hebbian rule	Storkey rule	Recurrent BackProp rule	Equilibrium Propagation rule
Accuracy	0.1792	0.2663	0.9968	0.9971

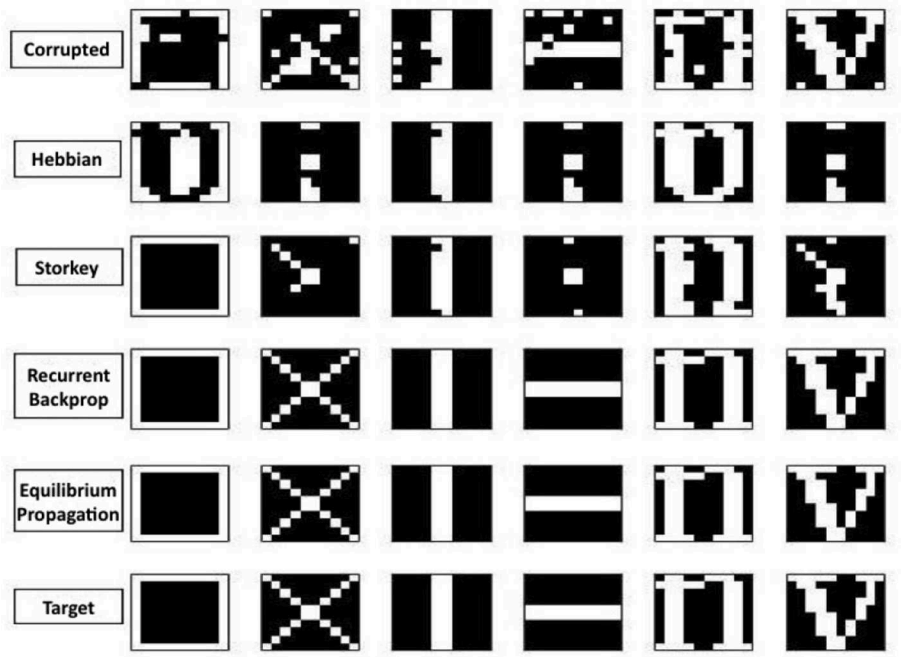


FIGURE 4 | From the top row: six corrupted patterns with probability $p = 0.1$, reconstructed pattern with hebbian rule, Storkey rule, recurrent backpropagation rule, Equilibrium Propagation rule and in the last row the target patterns.

slightly different update rule was used:

$$v_{t+1} = g\left(v_t - \epsilon \frac{\partial F}{\partial v_i}\right) \tag{21}$$

where $g(\cdot)$ is the hard sigmoid function. The predicted value corresponds to the index of the output units which reached the maximum value among all the others. All the hyperparameters chosen were in accordance with the suggestions proposed in Scellier and Bengio (2017): the learning rate $\epsilon = 0.5$ is used for the iterative inference, $\beta = 1$ is the value of the clamping factor in the second phase, $\alpha_1 = 0.1, \alpha_2 = 0.05$ are the two different learning rates for updating the parameters in the first and second layer. Observe that the authors were not considering a single learning rate η as in (19). However, instead of choosing a random sign for β for the second phase, the two learning parameters α_1, α_2 were decreased by half after each epoch. The results are shown in **Figure 5** and are consistent with the findings described in Scellier and Bengio (2017).

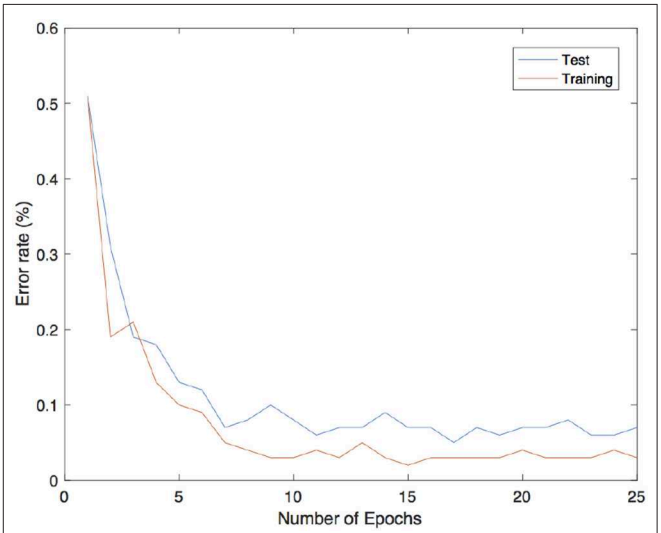


FIGURE 5 | Error rates of the trained neural network over 100 random patterns chosen among the training set (in orange) and 100 patterns from the test set (in blue) using Equilibrium Propagation learning rule.

CONCLUSIONS

In this paper, the dynamics of memristor–based recurrent neural networks has been analyzed. The network is trained by using two different generalizations of the backpropagation algorithm adapted to the continuous domain and energy-based models. Such *in situ* training learning rules permit to the memristor–based neural network to continuously adapt and adjust the synaptic weights without the direct computation of the loss function's gradient. Although, further work is still necessary to find physical memristor devices/emulators approximating the proposed memristive synapse dynamics, this manuscript provides two learning rules for the weights' update that can be implemented by a series of discrete programming pulses. Simulated results make clear that both methods significantly outperform conventional approach used for pattern reconstruction. In addition, promising results are

also obtained by using equilibrium propagation in performing classification tasks.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://yann.lecun.com/exdb/mnist/.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by the Ministero degli Affari Esteri e della Cooperazione Internazionale (MAECI) under the project n. PGR00823.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.00240/full#supplementary-material

REFERENCES

Almeida, L. B. (1987). "A learning rule for asynchronous perceptrons with feedback in a combinatorial environment," in *Proceedings, 1st First International Conference on Neural Networks*, Vol. 2 (San Diego, CA; New York, NY: IEEE), 609–618.

Ascoli, A., Corinto, F., and Tetzlaff, R. (2016). A class of versatile circuits, made up of standard electrical components, are memristors. *Int. J. Circ. Theory Appl.* 44, 127–146. doi: 10.1002/cta.2067

Assaf, H., Savaria, Y., and Sawan, M. (2019). "Memristor emulators for an adaptive dpe algorithm: Comparative study," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (IEEE), 13–17.

Chua, L. (1971). Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* 18, 507–519. doi: 10.1109/TCT.1971.1083337

Chua, L. O., and Sung Mo, K. (1976). Memristive devices and systems. *Proc. IEEE* 64, 209–223. doi: 10.1109/PROC.1976.10092

Corinto, F., Civalleri, P. P., and Chua, L. O. (2015). A theoretical approach to memristor devices. *IEEE J. Emerg. Select. Top. Circ. Syst.* 5, 123–132. doi: 10.1109/JETCAS.2015.2426494

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge, UK: Cambridge University Press.

Hasler, J., and Marr, H. B. (2013). Finding a roadmap to achieve large neuromorphic hardware systems. *Front. Neurosci.* 7:118. doi: 10.3389/fnins.2013.00118

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Approach*. New York, NY: Wiley, 62.

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521:436. doi: 10.1038/nature14539

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. (2015). "Difference target propagation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, eds A. Appice, et al. (Cham: Springer), 498–515.

Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508

Leon, C. (2015). Everything you wish to know about memristors but are afraid to ask. *Radioengineering* 24:319. doi: 10.13164/re.2015.0319

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* 7:13276. doi: 10.1038/ncomms13276

Liu, R., Mahalanabis, D., Barnaby, H. J., and Yu, S. (2015). Investigation of single-bit and multiple-bit upsets in oxide rram-based 1t1r and crossbar memory arrays. *IEEE Trans. Nucl. Sci.* 62, 2294–2301. doi: 10.1109/TNS.2015.2465164

McEliece, R., Posner, E., Rodemich, E., and Venkatesh, S. (1987). The capacity of the hopfield associative memory. *IEEE Trans. Informat. Theory* 33, 461–482. doi: 10.1109/TIT.1987.1057328

McGraw, P. N. and Menzinger, M. (2003). Topology and computational performance of attractor neural networks. *Phys. Rev. E* 68:047102. doi: 10.1103/PhysRevE.68.047102

Merced-Grafals, E. J., Da'vila, N., Ge, N., Williams, R. S., and Strachan, J. P. (2016). Repeatable, accurate, and high speed multi-level programming of memristor 1t1r arrays for power efficient analog computing applications. *Nanotechnology* 27:365202. doi: 10.1088/0957-4484/27/36/365202

Pineda, F. J. (1988). "Generalization of back propagation to recurrent and higher order neural networks," in *Neural Information Processing Systems* (American Institute of Physics), 602–611.

Scellier, B., and Bengio, Y. (2017). Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* 11:24. doi: 10.3389/fncom.2017.00024

Stauffer, D., Aharony, A., da Fontoura Costa, L., and Adler, J. (2003). Efficient hopfield pattern recognition on a scale-free neural network. *Eur. Phys. J. B Condens. Matt. Comp. Syst.* 32, 395–399. doi: 10.1140/epjb/e2003-00114-7

Storkey, A. (1997). "Increasing the capacity of a hopfield network without sacrificing functionality," in *Artificial Neural Networks – ICANN '97, 7th International Conference* (Lausanne), 451–456.

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453:80. doi: 10.1038/nature06932

Sun, Z., Pedretti, G., Ambrosi, E., Bricalli, A., Wang, W., and Ielmini, D. (2019). Solving matrix equations in one step with cross-point resistive arrays. *Proc. Natl. Acad. Sci. U.S.A.* 116, 4123–4128. doi: 10.1073/pnas.1815682116

Tanaka, G., Nakane, R., Takeuchi, T., Yamane, T., Nakano, D., Katayama, Y., et al. (2019). Spatially arranged sparse recurrent neural networks for energy efficient associative memory. *IEEE Trans. Neural Netw. Learn. Syst.* doi: 10.1109/TNNLS.2019.2899344

Whittington, J. C., and Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends Cogn. Sci.* 23, 235–250. doi: 10.1016/j.tics.2018.12.005

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Zoppo, Marrone and Corinto. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

PERSPECTIVE
published: 28 April 2020
doi: 10.3389/fnins.2020.00358

Neurohybrid Memristive CMOS-Integrated Systems for Biosensors and Neuroprosthetics

Alexey Mikhaylov[1], Alexey Pimashkin[1], Yana Pigareva[1], Svetlana Gerasimova[1], Evgeny Gryaznov[1], Sergey Shchanikov[2], Anton Zuev[2], Max Talanov[3], Igor Lavrov[4,5], Vyacheslav Demin[6], Victor Erokhin[3,6,7], Sergey Lobov[1,8], Irina Mukhina[1,9], Victor Kazantsev[1,8], Huaqiang Wu[10] and Bernardo Spagnolo[1,11,12]*

[1] Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia, [2] Department of Information Technologies, Vladimir State University, Murom, Russia, [3] Neuroscience Laboratory, Kazan Federal University, Kazan, Russia, [4] Department of Neurologic Surgery, Mayo Clinic, Rochester, MN, United States, [5] Laboratory of Motor Neurorehabilitation, Kazan Federal University, Kazan, Russia, [6] Kurchatov Institute, Moscow, Russia, [7] CNR-Institute of Materials for Electronics and Magnetism, Italian National Research Council, Parma, Italy, [8] Center for Technologies in Robotics and Mechatronics Components, Innopolis University, Innopolis, Russia, [9] Cell Technology Group, Privolzhsky Research Medical University, Nizhny Novgorod, Russia, [10] Institute of Microelectronics, Tsinghua University, Beijing, China, [11] Dipartimento di Fisica e Chimica-Emilio Segrè, Group of Interdisciplinary Theoretical Physics, Università di Palermo and CNISM, Unità di Palermo, Palermo, Italy, [12] Istituto Nazionale di Fisica Nucleare, Sezione di Catania, Catania, Italy

OPEN ACCESS

Edited by:
Stefano Brivio, Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Jessamyn Fairfield, National University of Ireland Galway, Ireland
Robert Rieger, University of Kiel, Germany

***Correspondence:**
Alexey Mikhaylov
mian@nifti.unn.ru

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *31 October 2019*
Accepted: *24 March 2020*
Published: *28 April 2020*

Citation:
Mikhaylov A, Pimashkin A, Pigareva Y, Gerasimova S, Gryaznov E, Shchanikov S, Zuev A, Talanov M, Lavrov I, Demin V, Erokhin V, Lobov S, Mukhina I, Kazantsev V, Wu H and Spagnolo B (2020) Neurohybrid Memristive CMOS-Integrated Systems for Biosensors and Neuroprosthetics. Front. Neurosci. 14:358. doi: 10.3389/fnins.2020.00358

Here we provide a perspective concept of neurohybrid memristive chip based on the combination of living neural networks cultivated in microfluidic/microelectrode system, metal-oxide memristive devices or arrays integrated with mixed-signal CMOS layer to control the analog memristive circuits, process the decoded information, and arrange a feedback stimulation of biological culture as parts of a bidirectional neurointerface. Our main focus is on the state-of-the-art approaches for cultivation and spatial ordering of the network of dissociated hippocampal neuron cells, fabrication of a large-scale cross-bar array of memristive devices tailored using device engineering, resistive state programming, or non-linear dynamics, as well as hardware implementation of spiking neural networks (SNNs) based on the arrays of memristive devices and integrated CMOS electronics. The concept represents an example of a brain-on-chip system belonging to a more general class of memristive neurohybrid systems for a new-generation robotics, artificial intelligence, and personalized medicine, discussed in the framework of the proposed roadmap for the next decade period.

Keywords: memristor, neuronal culture, spiking neural network, microfluidics, biosensor, neuroprosthetics

INTRODUCTION

The growing demand in miniature and energy-efficient electronic interface with bioelectrical activity for personalized medicine and other related products essentially depends on development of biohybrid electronic technologies (Vassanelli and Mahmud, 2016). The emergence of new technologies for creating thin-film sensors and non-invasive signal processing systems ensures the development of fundamentally new approaches to solve the problems of recording activity signals of brain, heart, and muscles, as well as skin condition in the form of wearable systems for processing and diagnostic. Such bio-compatible microelectronic systems, along with new biotechnologies, may provide a breakthrough in the field of neuroprosthetics with an important competitive advantage: a miniature bioelectrical sensor based on micro- and nanostructures with an option to store and

process signals in multiple manners, including feed-forward approach and feedback loops, may serve as an active neurointerface for intelligent control and management of neuronal structures.

A wide variety of neuroprosthetic technologies have emerged recently from prosthetic arms (Fukuma et al., 2016; Petrini et al., 2019b) and legs (Petrini et al., 2019a) to prosthetic hearing (Rouger et al., 2007) and vision (Fernandez, 2018). Some of limb prostheses were non-invasively controlled by electrical signals from the muscles electrical activity (myograms) or electrical activity of selected areas of the motor cortex. The most promising bionic technologies are aimed at creating prosthetic devices controlled by the electrical activity of neurons via specialized arrays of electrodes implanted into neural tissue. In order to provide sensory feedback, additional arrays can also be implanted in somatosensory areas of the cerebral cortex or afferent systems of the spinal cord. The implantation of sensor chips into the visual cortex or a retina of an eye is becoming a serial operation today (Chuang et al., 2014). Cochlear implants are used by hundreds of thousands of patients around the world (NIH, 2016). Another promising direction is neurohybrid computing systems with living neural cells cultured in a nutrient medium *in vitro* and, after the maturation and formation of a large number of synaptic connections between cells, implemented to control an external robotic device or solve the complex sensory-cognitive task (e.g., pattern recognition). These devices also called neuroanimats in the literature (Xydas et al., 2008).

Another modern technology, memristors, possess the unique property of non-linear resistive memory and could serve as analog information processing systems with a neuron-like structure, as well as an electrophysiological activity sensor with capacity of simultaneous accumulation and non-volatile storage. Further development of memory-embedded sensors (Tzouvadaki et al., 2015; Doucey and Carrara, 2019) and neurohybrid systems, including neuroprostheses based on the integration of memristive and microelectrode CMOS technologies, as well as spiking neural network (SNN) architectures, will ensure the processing and real-time classification of electrophysiological and other analog signals, related to the activity of biological neuronal networks. Potential applications of this technology may target *in vivo* testing of pharmacological effects, biosensors and detectors of electromyography (EMG) signals, as well as muscle force extraction for various technical systems (smart tissue, wearable electronics, smart wheelchairs, cyber-physical suits, and vehicles). Most challenging problems are currently related to the application of implantable and non-implantable machine-to-nervous-system interfaces and neuroprostheses for correcting and restoring cognitive abilities, complex motor patterns like locomotion, and vision.

In this perspective, we discuss the main challenges associated with development of compact multifunctional neurohybrid systems for the bidirectional interface of living biological systems and memristive electronics combined with microelectrode and microfluidic systems. As compared to the previous works (Vassanelli and Mahmud, 2016; Chiolerio et al., 2017) focused on general trends and approaches for interfacing between neuronal and extrinsic/intrinsic neuromorphic systems, here we provide a comprehensive analysis of the implementation of a CMOS-integrated hybrid system based on scalable memristive devices and arrays back-end-of-line or monolithically integrated with CMOS circuits, analog signal processing on CMOS chips with memristive and microelectrode arrays. Specialized memristive neural architectures are proposed to implement functional abilities of some regions of the brain and nervous system. A roadmap of research and development in the field of memristive neuromorphic and neurohybrid systems has been for the first time presented and discussed in this manuscript in the context from state-of-the-art tasks to future challenges (until 2030).

It is worth noting here that memristors provide only one of the possible options for creating biomimetic electronic systems for neural interfaces. In particular, the neuromorphic function has also been demonstrated in colloidal nanomaterials or networks of nanowires (O'Kelly et al., 2016; Manning et al., 2018) and organic electrochemical transistors (Gkoupidenis et al., 2015, 2017; Tarabella et al., 2015; Battistoni et al., 2019b). Certain advantages of such materials over CMOS architectures have been discussed in recent reviews (Inal et al., 2018; Rivnay et al., 2018; van De Burgt et al., 2018; Ling et al., 2020) and mainly related to the flexibility and mechanical property match with neural tissue, the lower impedance, and current densities. Nevertheless, they are outside the scope of this perspective, and we will limit ourselves only to the CMOS-compatible approaches that are ready for the integration into existing technological workflows dedicated to practical applications. The focus on metal–oxide memristive electronics will allow going beyond the traditional neuromorphic chips as parts of neurohybrid systems (Hogri et al., 2015; Boi et al., 2016; Buccelli et al., 2019).

MEMRISTIVE NEUROHYBRID CHIP: CONCEPT AND CHALLENGES

According to the general definition (Vassanelli and Mahmud, 2016; Chiolerio et al., 2017), the neurohybrid system provides an interaction between biological (neuronal) and artificial elements in the open- or closed-loop manner. Despite the large number of available examples, they usually reflect different sides of such interaction and primary confirm some level of connectivity between biological and artificial systems. A functional interface between simple living being (slime mold) and memristor devices has been reported (Adamatzky et al., 2012) and, recently, the possibility of direct synaptic coupling of neuron cells from the rat cortex through a memristive device has been demonstrated (Juzekaeva et al., 2019). Future implementation of this approach requires the development of interrelated solutions at all levels, using both existing and emerging technologies in a single conceptual map matching the requirements for compactness, performance, energy efficiency, speed, reliability, and safety. In this paper, we analyze such solutions within the framework of a single concept of a neurohybrid CMOS chip that implements a compact interface between the biological (neuronal) system and the electronic subsystem.

Figure 1A demonstrates a schematic representation of the proposed neurohybrid system, which consists of several functional layers combined in one CMOS-integrated chip. The top layer is a part of the neuronal system represented here by a culture of dissociated hippocampal cells grown on multielectrode array (MEA) and functionally ordered by a special layout of microfluidic channels indicated in **Figure 1B**. The MEA is used for extracellular registration and stimulation of neurons *in vitro* and is implemented on the top metallization layers of the CMOS layer together with an array of memristive devices (**Figure 1D**). The simplest task performed by memristive devices is the direct processing of spiking activity of the biological network (**Figure 1C**); however, self-learning neural network architectures based on fully connected cross-bar memristive arrays can be designed for adaptive decoding of spatiotemporal characteristics of bioelectric activity. The output of this artificial network (**Figure 1F**) can be used to control the cellular network via gradual modulation of extracellular stimulation (**Figure 1G**) according to the given protocol. This way, analog and digital circuits should be implemented in the main CMOS layer (**Figure 1E**) for accessing and controlling the MEA and memristive devices, amplifying, generating, and transmitting signals between layers. To create neurohybrid chip, joint design and optimization are required for all mentioned elements at the levels of materials, devices, architectures, and systems. Within the framework of this concept, the following subjects of interrelated research and developments should be considered at fundamental and applied levels:

1) Neural networks cultured *in vitro* with a given connectivity to implement a certain information function;
2) Microfluidic cell manipulation techniques on a chip;
3) CMOS- and bio-compatible technology for the MEA fabrication;
4) Scalable and CMOS-compatible memristive devices;
5) Microelectrode and memristive arrays integrated on-chip with CMOS electronics;
6) Analog/digital peripheral and control circuits on CMOS chip;
7) Specialized SNN based on memristive arrays and CMOS electronics;
8) Interconnection/integration solutions for connecting various functional modules.

The first two tasks are required only in the case of creating a neurohybrid device like the neuroanimat, with information processing by an ensemble of cultured living neuron cells. To create both implantable and non-implantable devices such as neuroprostheses, the implementation of this route apparently should start with the task 3.

Two main groups of challenges must be addressed for the successful development of this technology. From the biological side of neural integration, the main problems are related to biocompatibility and matching mechanical properties of MEA materials in contact with neuronal culture, device geometries and accessibility to neuronal culture, their scaling to brain activity *in vivo*, as well as the reaction of living neurons to electrical stimulation and power dissipation (including glial scarring). From the electronic engineering side, we should note the required high spatiotemporal resolution of MEA, transition from 2D to 3D electrode system, minimum size and high density of memristive devices needed for subsequent monolithic integration, area- and energy-efficient solutions for analog information processing by memristive circuits. Both groups of challenges, possible solutions, and trade-offs are considered in the corresponding sections below.

LIVING NEURAL NETWORK: BIOLOGICAL SIDE OF NEURAL INTEGRATION

The main problem of neuronal cultures *in vitro* is related to homogeneous network structure, which is developed in randomly patterned cells on the substrate. During the last decade, new methods of neuroengineering have been developed to control the position of cells and direction of axon and dendrite growth (le Feber et al., 2015; Na et al., 2016; Renault et al., 2016). Recently, it has been shown that the main feature of functional network topology as unidirectional synaptic connectivity between cell clusters can also be engineered using microfluidic technology (Gladkov et al., 2017; Poli et al., 2017; Forró et al., 2018). Being implanted in the damaged brain, such tools of network structure manipulation allow one to mimic brain areas, which are involved in reflex activity, pattern retrieval in multilayered unidirectional network (Brewer et al., 2013; Poli et al., 2017) for neural tissue recovery from brain injury (Shimba et al., 2019). Next, it could be combined with an array of non-invasive planar microelectrodes, which provide spiking activity registration and stimulation of isolated or multiple neurons. Spiking activity could be monitored and induced in several independent axonal pathways, which grow between subnetworks through the microchannels. Thus, the precise input and output could be implemented in engineered multilayered network with the designed connectivity, where the full potential of the proposed task can be solved in closed-loop conditions with memristive spiking network. First, such a system could be used for the stabilization of spontaneous activity, which slowly stochastically changes, and second, to classify patterns according to various input signals and induce spike-timing-dependent plasticity (STDP) in a living network, where pre- and postsynaptic neurons could be accessed independently.

Biocompatibility and mechanical matching of materials are the key problems that arise on the way to neural integration. They have been already addressed in many commercial MEA by using gold, platinum, indium tin oxide (ITO), and titanium nitride (TiN) as electrode materials. The signal-to-noise ratio (SNR) depends strongly on the biological part of the system, but can be increased by the small impedance of recording electrodes. In order to reduce the impedance and increase the charge transfer efficiency, the surface area of electrodes can be modified by covering with porous conductive materials, such as Pt-black, Au nanoparticles, carbon nanotubes (CNTs), and conductive polymers like poly(3,4-ethylenedioxythiophene) (PEDOT) (Obien et al., 2015).

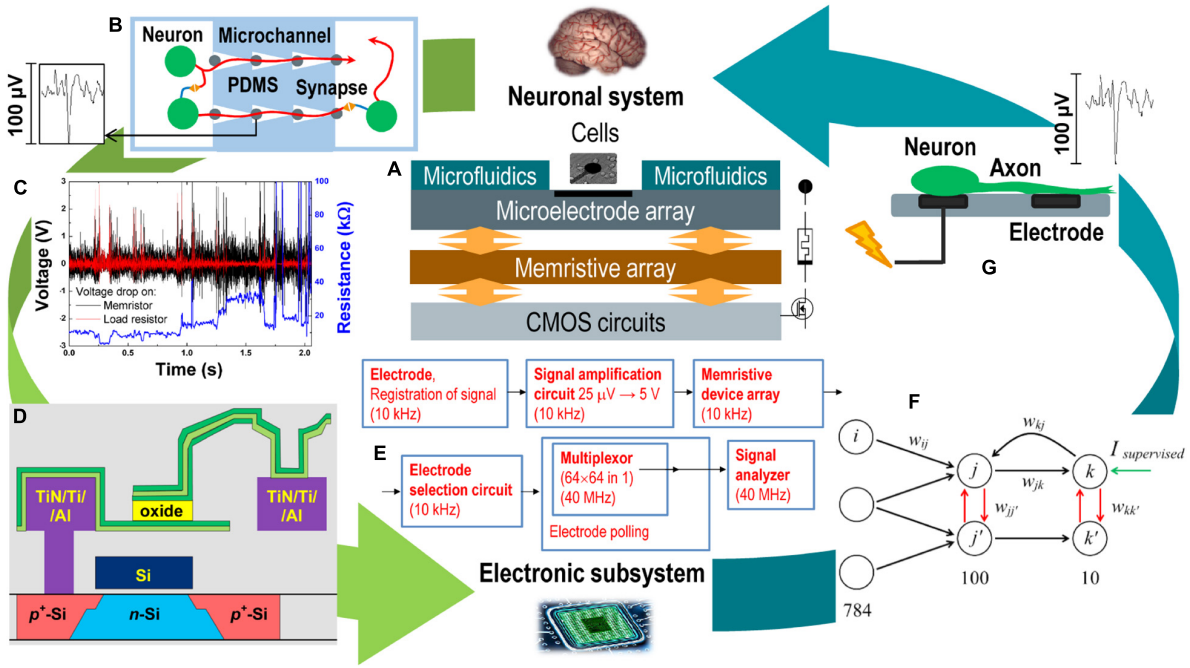


FIGURE 1 | Memristive neurohybrid chip. **(A)** Schematic representation of the neurohybrid chip composed of a neuronal system (the brain cellular culture grown on MEA) and an electronic subsystem represented by the mixed analog–digital circuits coupling microelectrode arrays, memristive devices, and intrinsic neuromorphic systems. **(B)** The sketch of a spatially ordered neuronal culture with individual axons grown in microfluidic channels. **(C)** The response of metal–oxide memristive device to spiking activity recorded in the culture. Black line—voltage drop on memristor, red line—voltage drop on load resistor as current sensor, and blue line—resistance of memristive device responding in a volatile or non-volatile manner to noise and spikes with different parameters. **(D)** The example of CMOS integration of metal–oxide memristive device based on thin $ZrO_2(Y)$ film sandwiched between top metal layers of CMOS circuit. **(E)** The typical diagram of registration, amplification, and analysis of bioelectric activity by using multielectrode/memristive arrays and embedded CMOS circuits. **(F)** The typical spiking neural architecture with competitive interneuron connections. **(G)** The scheme of extracellular electrical stimulation of living neurons modulated by the electronic subsystem to control their activity.

Moreover, the enhanced biocompatibility has been demonstrated for electrodes with a nanostructured porous surface in the form of laser-micropatterned PEDOT:PSS (Santoro et al., 2017). The next level of improved compatibility between electrodes and cells or living tissue relies on the use of extracellular matrix materials, which increase the adhesive properties of the electrodes and reduce the risk of inflammatory processes (Won et al., 2018).

An important problem of the registration of neuronal activity is associated with the geometry/topology and spatial resolution of microelectrode arrays. Conventional MEAs do not allow recording the activity of individual cells, because the step between electrodes (>30 µm) exceeds the neuron soma size (about 12–18 µm). Owing to the advanced CMOS technology, a new type of MEA has been commercialized, in which amplifiers and ADC are located on one chip with electrodes. This approach reduces the inter-electrode distance and consequently increases the spatial resolution of electrodes. The search for optimal solutions to combine high spatial resolution with a high SNR is currently underway (Ghane-Motlagh and Sawan, 2013; Müller et al., 2015). The proposed system concept presumes a 2D neuron interface on top of the MEA electronics. However, planar electrodes reach their limits when it comes to tissue slices or cell clusters. Although, a 3D-MEA with micron-size electrodes penetrating 40–100 µm deep into the tissue is already on the market[1], the lattice-like 3D electrode interface should be developed to really mimic or interface the brain.

All these problems are exacerbated when scaling the proposed technology to registration and stimulation of brain activity *in vivo*, especially taking into account high conductivity, inertness, biocompatibility, and stretchability required for the interaction with living tissue (Qi et al., 2017). Devices for detecting neural activity *in vivo* can be fabricated in the form of 2D or 3D arrays of electrodes combined on one substrate. Two types of 3D probes are widely used: electrodes placed on an array of silicon needles and neural probes, on which arrays of electrodes are located. Recently, densely arranged probes based on silicon-on-insulator (SOI) technology have been actively developed (Scholvin et al., 2015; Angotzi et al., 2017; Lopez, 2019). The dense arrangement of electrodes allows spatial oversampling of neural activity and accurate sorting of spikes. In the active neural probes, local amplification of the recorded signal near the electrode with microfabricated CMOS circuit improves the recording quality by reducing the electrode impedance and crosstalk between neighboring shank wires (Raducanu et al.,

[1]https://www.multichannelsystems.com/news/3d-meas-recording-inner-cell-layers

2017). Simultaneous recording of signals from a large number of electrodes (up to 1400) can be possible due to the time division multiplexing method. In addition, when developing neural probes, it is necessary to consider heating of tissue due to power dissipation, which is limited to a threshold of 1°C for chronic experiments (Kim et al., 2007). To reduce tissue damage and inflammatory response, the size of the shanks should be miniaturized and can reach 25 μm for the SOI technology. For the passive probes, one of the approaches to thin the shank can be based on materials with low stiffness like nanoelectrode filaments or poly(etherimide) fibers. On the other hand, such materials require additional support during implantation, particularly, control of localization and speed modes of insertion (Dryg et al., 2015), as well as special guides from soluble materials (Won et al., 2018).

Another important problem is the reaction of living cultures and tissues at the interface with the artificial electronic subsystem. Needless to say that purely electrical contact can serve non-invasively not affecting the cell in contrast to different methods of optical recordings or chemical manipulations. More complicated task is to provide correct stimulation of the target area in the brain or in the neuronal culture. Problems may appear in long-term implantations, when the neuronal system under stimulation (including electrical one) starts to adapt itself maintaining the network homeostasis and trying to escape the external perturbation (Middleton et al., 2010; Graczyk et al., 2018). Adaptation is based on the mechanism of homeostatic plasticity, which ensures the functional stability of neuronal system by equipoising intrinsic excitability and synaptic strength. It balances the network excitation and inhibition, and coordinates the changes in circuit connectivity (Tien and Kerschensteiner, 2018). In addition, any mechanical impact on the brain tissue, such as implantation of electrodes, may cause the appearance of a glial scar that restricts the area damaged by the electrodes. For instance, sharp electrodes implanted into a brain after some time are insulated by glial cells produced around the electrode hence decreasing the effect of stimulation (Sillay et al., 2013; Wu et al., 2013). The gliotic encapsulation problem can be mitigated by chemical functionalization of materials at the electrode–tissue interface. Coating of electrodes with extracellular matrix proteins, collagen and Matrigel films can reduce the astrogliotic scarring (He et al., 2006; De Faveri et al., 2014; Shen et al., 2015). Another efficient approach to mitigate the rejection is to reduce the electrode size, for example, by using the PEDOT-coated carbon fiber as a material of electrodes (Patel et al., 2016).

The limiting factor to resolution and functionality of the proposed neurohybrid concept may be the power consumption needed for a given SNR when basing the concept on CMOS (or any other semiconductor) technology. This problem depends on the type of interface with the neuronal system (*in vitro* or *in vivo*) and the energy efficiency of electronic subsystem. In the case of *in vitro* interface, the existing commercially available CMOS MEA has the overall power consumption of about 30 W[2], which is mainly determined by the off-chip interface electronics and does not include the data processing and analysis equipment. The *in vivo* interface systems have been studied previously in relation to neural prostheses for restoring and enhancing memory (Berger et al., 2011; Hampson et al., 2018; Song et al., 2018) also by using PC-controlled multichannel recording/stimulation closed-loop systems and special mathematical models. To the best of our knowledge, none of the mentioned systems has been implemented yet on a single chip. Although miniaturization is a general requirement to create such bioelectronic platforms (Birmingham et al., 2014), we believe that it can only be achieved using the area- and energy-efficient memristive electronics based on CMOS technology and shown below. Of course, this task should be reached hand in hand with the development of miniaturized wireless systems for energy harvesting and bi-directional communication that will definitely improve safety, access to anatomical sites, and enable ultra-minimally invasive delivery methods, reducing tissue trauma during implantation and immune response (Masius and Wong, 2020; Piech et al., 2020).

MEMRISTIVE DEVICES: TOWARD CMOS INTEGRATION

A memristor (memory resistor) was predicted by Chua (1971) as the fourth passive element of electrical circuits. For a long time, it was considered as a theoretical object. Only in 2008, the memristive effect was first correlated (Strukov et al., 2008) with the phenomenon of reversible resistive switching, which can occur in a simple thin-film metal–oxide–metal nanostructure and is associated with local rearrangement of the oxide atomic structure and composition under the action of inhomogeneous electric field, temperature, and concentration gradients (Ielmini and Waser, 2016). Currently, memristors and memristive systems are the basis of a new paradigm in electronics related to creation of brain-like network architectures by using the ability of memristive devices to emulate the most important functions of biological synapses and neurons. Since 2015, there has been an increase in the number of publications regarding a hardware implementation of the simplest artificial neural networks (ANNs) (most often in the form of a single-layer perceptron) based on a limited number of memristive connections (Prezioso et al., 2015; Serb et al., 2016; Yao et al., 2017). Larger integrated memristive 1T-1R or passive cross-bar arrays have been fabricated and shown to date (Cai et al., 2019; Kataeva et al., 2019; Zhou et al., 2019) to implement various multiplication operations and neuromorphic functionality on the basis of precise analog tuning the conductance of memristive devices. Although some higher functionalities of board-integrated systems like multilayer perceptron (Bayat et al., 2018; Li et al., 2018a; Mikhaylov et al., 2018) and the first fully memristive neural network with unsupervised learning (Wang et al., 2018) were demonstrated and revolutionized, the higher functionalities are still restricted with a practical size up to 64 × 128 of memristive arrays.

Thus, the necessary condition for the development of advanced functional electronic circuits based on memristors is their integration with mixed analog-digital CMOS transistor

[2]http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/CMOS-MEA5000-System_Manual.pdf

circuits. At the same time, the capabilities and functionality of traditional ANN architectures based on programmable memristive weights are limited by the size of the memristive array, the increase of which is constrained not by low scalability (the minimum size of the memristive element may be of the order of nanometer; Pi et al., 2019), but by insufficient reproducibility of device parameters due to the stochastic nature of resistive switching. For example, the widely used back propagation updating rule, which has been proved to be efficient for traditional supervised neural networks, often requires additional write-verification techniques (Yao et al., 2017) to modulate memristive devices into the desired states, incurring software/hardware overheads on memristive neurohybrid architectures.

The non-linear behavior of memristive devices in response to electrical pulses together with their unique scalability are the most important advantages that determine a unique possibility of hardware implementation of SNN (Demin and Nekhaev, 2018; Guo et al., 2019) based on the processes of self-organization in neural network architectures and qualitatively different from traditional neural networks (perceptrons). We believe that implementation of brain-like networks of future generations will be based on the stochastic dynamics of memristors and synchronization of neural oscillators. Such works are carried out at the most basic level (Ignatov et al., 2016; Gerasimova et al., 2017), demonstrate the possibility of implementing higher (cognitive) brain functions, but require the development of adequate models of neural synchrony based on stochastic memristive plasticity.

Nevertheless, such a rapid progress in the implementation of memristive neuromorphic systems makes it possible not only to expect in the nearest future the creation of brain-like networks with memristive plasticity for novel computing paradigms, but also to take the next step and develop memristive neurohybrid systems on the basis of intrinsic analogy in the properties of memristive and natural systems. It is important to note that compact memristor-based devices for real-time processing of bioelectric activity (threshold detection of spikes) can be created owing to the integrative change in their resistive state (Gupta et al., 2016). In this case, the metastable (volatile) behavior is an important property of memristive devices for continuous and energy-efficient encoding of large volumes of spiking activity of living biological cultures (Gupta et al., 2018). It should be mentioned that effective use of memristors in neurohybrid systems is dependent on the predictable behavior of memristive nanomaterials and devices, as well as on the ability to control the parameters of their non-linear response to complex electrical signals, which should be a subject of comprehensive study at the micro- and macro-levels.

Noise plays a very significant and constructive role in memristive devices, and only recently new investigations on the positive role of noise have been started (Mikhaylov et al., 2016; Filatov et al., 2019). Nowadays, there are many known examples, where the interplay of non-linearity and fluctuations can change the properties of a stochastic system in a counter-intuitive way, in classical and quantum physics (Fiasconaro et al., 2004; Chichigina et al., 2005, 2011; Valenti et al., 2008, 2015; Falci et al., 2013; Spagnolo et al., 2015, 2016, 2018a,b). Furthermore, internal and external noise can play a positive role in the switching dynamics of memristors, as in stochastic resonance phenomenon (La Barbera and Spagnolo, 2002; Valenti et al., 2004; Agudov et al., 2010). This paves the way for using the intensity of fluctuations as a control parameter for switching dynamics in memristive devices (Agudov et al., 2020).

CMOS CIRCUITS: ON-CHIP ANALOG AND DIGITAL SYSTEMS

As noted above, a significant progress has been demonstrated on the way toward integration of memristive arrays and CMOS circuits (Cai et al., 2019; Kataeva et al., 2019). The electronic subsystem required for the CMOS integration of memristive arrays includes peripheral and control circuitry. In Kataeva et al. (2019), large passive memristive cross-bars are accessed via on-chip CMOS interface circuits which are controlled by a custom FPGA board. To reduce latency and power consumption, a full set of mixed-signal interface blocks and a digital processor have been recently integrated together with memristive cross-bar array on a single chip (Cai et al., 2019), instead of using discrete components on printed board. On-chip integration of processor allows the neuron functions and network structures to be reprogrammed through simple software changes, enabling different algorithms to be mapped on the same hardware platform.

With respect to the electronic subsystem of the neurohybrid chip, a number of technical problems have to be solved to organize the optimal interaction of living neuronal culture with memristive arrays. Reading, processing, and reflection of the spiking activity of neural cells must be carried out with a duration of no more than a typical pulse (spike) in the areas of the neurons. At the prototyping stage, a separate reading amplifier and recording amplifier cannot be allocated to each electrode of contact with a living culture due to the limited area of the CMOS layer. This should be implemented in future with higher design standards or a smaller number of electrodes.

It is necessary to implement an array of reading and writing amplifiers in the CMOS layer, which allows transmitting pulses from a living culture through electrodes to memristive array and vice versa, simultaneously on a certain surface area. The reading and writing amplifiers must be tuned to the signal from living tissue amplified to the levels of active operating modes of memristors. In the CMOS layer, it is also necessary to implement access circuits for electrodes and memristors at row and column addresses.

Circuits for stimulation of living culture/tissue (by using the response of memristive network) are supposed to be implemented on the basis of the pulse-width modulation (PWM). If necessary, for the simultaneous reading of the electrode states in the CMOS layer, banks of buffer memory can be implemented. It is proposed to use ADC and DAC circuits to input and output information about the analog state of memristors, but the required bits of the ADC and DAC should be determined at the prototyping stage (8 bits are assumed in the layout). The initial input and subsequent output of information for a set of statistics

on the experiment and processing can be implemented on the basis of standard bidirectional interfaces (Cai et al., 2019).

Although the local resistive switching effect in memristive devices provides the unique compactness, fast and energy-efficient operation of passive memristive arrays (Xia and Yang, 2019), the active arrays integrated with peripheral and control electronics should be always a subject of explicit evaluation and benchmarking depending on the development/prototyping stage (Cai et al., 2019; Zhao et al., 2020). Recently, several reports on such benchmarking have shown potential advantages of memristive chips over conventional ones: 19.7, 6.5 times, and 2 orders of magnitude better energy efficiency compared to the Google's tensor processing unit (TPU), a highly optimized application-specific integrated circuit (ASIC) system, and the state-of-the-art graphics-processing unit (GPU), respectively (Sun et al., 2020; Yao et al., 2020). The performance benchmark of memristive neuromorphic computing system shows 110 times better energy efficiency and 30 times better performance density compared to Tesla V100 GPU. So, even rough estimates for memristive circuits considered in this article allow one to imagine their great potential from the viewpoint of speed, performance, power consumption, and compactness.

The issue of reliability of memristive neural networks is also currently in the eyeshot of many researchers and requires the use of system approach and comprehensive consideration (see Zhao et al., 2020 for review on the status of reliability studies in this field). An example of such a system approach to ensure the reliability of neural networks based on memristors is proposed by the authors (Danilin et al., 2019; Shchanikov et al., 2020). Another promising way is to use specialized algorithms for tuning (training) memristor-based neural networks, as it is proposed in Wang et al. (2019). This approach makes it possible to create a neural network that self-adapts to non-idealities of the 1T-1R memristive array, thereby providing the necessary level of reliability.

One more important limitation when creating electronic devices in contact with living cultures/tissues is to preserve a trade-off between performance and power dissipation. On-chip processing is more efficient than transmitting raw data to the external processing unit (Zhuk et al., 2020), but the power consumption of state-of-the-art digital processors is too high. The dissipated power of memristive chips, according to the estimates made by a number of research groups, does not exceed tens of mW: 13.7 mW (Li et al., 2018b), 7.438 mW (Yao et al., 2020), 6.62 mW (Wang et al., 2019), 42.1 mW (Lee et al., 2020), 64.4 mW (Cai et al., 2019). The power dissipation strongly depends on the amplitudes and frequencies of the signals and increases with increasing the values of these parameters, which is not necessary in principle when working with living neurons. In addition, in a traditional computing system, power is also dissipated in memory units (Horowitz, 2014) and even much more in data movement, while both data storage and computation can be combined in one memristive device. So, the use of memristors to create a system on chip seems to be much more efficient and safe for neural interfacing.

Therefore, one can argue that memristive CMOS circuits will outperform traditional digital computing tools (CPU, GPU, TPU, ASIC) in all key parameters for a wide range of data-intensive applications, one of which is the real-time on-chip processing of electrophysiological data in the frame of the proposed neurohybrid concept.

MEMRISTIVE NEURAL ARCITECTURES: TOWARD NEUROPROSTHETICS

Biological relevance should be ensured when developing substitutive (neuroprostheses, motorized prostheses) and assisting neuromorphic systems [computer interfaces (Lobov et al., 2016), exoskeletons (Mironov et al., 2017), wheelchairs, "neuromobiles" (Mironov et al., 2018)]. Here, if possible, the same neural "language" and the same principles of information processing should be used as in a biological brain. Only in this case, over time, we can expect the blurring of the boundary between living and artificial neural subsystems, which will ultimately lead to the expansion of human capabilities. On the other hand, in all the neurochip perspective applications discussed here, we have arrays of implantable or non-invasively attached electrodes that record in real time the electrical activity of ensembles of neurons and/or muscle fibers. It is clear that the more electrodes and more frequently the signal is taken from each of them, the higher is the spatial (topographic) and temporal resolution and, accordingly, the potentially higher is the accuracy of sensory recognition (vision, hearing) or motor control commands sent to an electromechanical prosthesis. In this manner, we get a huge amount of data that needs to be processed in real time. Currently, it is common to use an external processor, which performs this processing and provides an interface between an external part of a prosthesis (camera, microphone, artificial limb) and the microcontroller device from a living tissue side. However, the solution of such problems could be strongly optimized by exploiting a highly specialized processor with neural network architecture adapted for this specific kind of calculation and serving as if it is a natural extension of the biological nervous system (Boi et al., 2016). In this case, the computing device would be capable of processing a large input dimension (determined by the number of electrodes in the MEA) and performing the required real-time signal processing. In our opinion, the SNN architecture based on phenomenological models and integrated into the proposed hybrid system seems to be a good compromise in the sense of both biological similarity and computational/power cost.

Recently, the first steps have been taken toward EMG (Lobov et al., 2015, 2020a) and EEG (Goel et al., 2006; Tahtirvanci et al., 2018) interfaces based on SNNs. However, until now, no learning rule for SNNs has been proposed, which is equal in its universality and effectiveness to the back propagation algorithm for ANNs based on formal neurons. Several attempts were made to adapt the "backprop" and its variations to SNNs (Hong et al., 2010; Xu et al., 2013; Esser et al., 2016), but associative learning based on synaptic plasticity similar to that for living neurons seems to be a more "natural" way. Indeed, traditional formal neural networks contain artificial neurons with a static activation function as key computational elements, i.e. there is no dynamics

in such systems. Consequently, such systems are quite difficult to synchronize with time sequences of individual spikes recorded in a biological nervous system. Spiking artificial neurons, as well as their biological prototypes, generate spike sequences that could be synchronized with the biological pulse signal through a non-linear interface—an artificial analog of synapse, a memristor. Namely, it has been recently shown that a system consisting of several spiking pre-synaptic neurons connected via memristive devices to the one post-synaptic neuron can adapt their conductivities (synaptic weights) to the same distribution under STDP updates by the repeatable pre- and post-synaptic trains of pulses, independent of the initial resistances of memristors or their device-to-device variability (Emelyanov et al., 2020). This means that the non-linear memristive spiking system memorizes only useful information about millisecond-scale time intervals between spikes which could encode some real data about perceived objects from an environment or motor commands to actuators. In a neurohybrid interface, spikes from biological neurons could be transformed online (synchronized) to the trains of voltage pulses generated by artificial spiking units which, in turn, could be used for the informative update of memristive weights as described above.

Neural networks of a living brain appear to use both temporal and frequency coding (Clopath et al., 2010; Masquelier and Deco, 2013). Similar behavior is observed in memristive devices based electronic circuits (Battistoni et al., 2019a). Thus, the process of SNN learning should provide both types of coding. In addition, SNNs, unlike their formal counterparts, can be trained by bio-plausible, so-called local rules of synaptic weight change using information only about the activity of interconnected neurons and synaptic efficacy (weight magnitude) between them. These rules do not require information from the outside, as in the case of learning by error back propagation technique, and therefore can be the basis of self-learning computing systems, with a change in synaptic weight according to the rules of the Hebbian (Morris, 1999), STDP (Bi and Poo, 1998), BCM (Bienenstock et al., 1982), metabolic (Yousefzadeh et al., 2018), or homeostatic (van Rossum et al., 2000) types. In the case of frequency coding, it is necessary to use frequency-dependent varieties of STDP, such as the triplet-based rule STDP (Pfister and Gerstner, 2006) or voltage-based STDP (Clopath et al., 2010). Recent studies have shown the possibility of rate and temporal coding in SNN using a combination of Hebbian learning (through triplet-based STDP), synaptic and neuronal competition (Lobov et al., 2020a,b). Hebbian and other STDP rules have been demonstrated for a large number of different kinds of memristors (Kim et al., 2015; Ielmini and Waser, 2016; Emelyanov et al., 2019; Minnekhanov et al., 2019) that confirms their high potential to serve as the self-adjusting weights between neurons in SNN.

Moreover, on the basis of spiking architectures, it is possible to naturally train recurrent networks in which there are feedback connections from deeper layers of neurons to less deep layers, as well as the lateral connections between neurons of the same layer (Demin and Nekhaev, 2018). In general, such architectures cannot be reduced to a feedforward neural network, such as a long-short term memory (LSTM) "unrolled" in several consecutive modules (Hochreiter and Schmidhuber, 1997; Brownlee, 2019). Therefore, recurrent SNNs can potentially be trained on the basis of local rules to realize complex dynamic patterns corresponding to those in the biological part of neurointerface. This kind of training can take place in real time, continuously adapting to the individual characteristics of a user's behavior. This is a practically inaccessible task for the formal ANN architectures that require *a priori* training by error back propagation on a set of pre-recorded patterns.

Implementation of hardware SNN architectures based on memristors certainly requires additional wide studies: first, to identify a minimum set of local learning rules sufficient for the convergence of training the network to a solution of a given problem, second, to seek for the possibility of adapting local rules (like that of STDP type) to hardware implementation with memristive elements (either by appropriate selection of the memristive material or by engineering the temporal sequence and/or shape of spikes generated by artificial neurons), and, at last, to optimize (by energy efficiency, area and computing performance) the SNN architecture design and placing corresponding periphery systems on neurohybrid chip under development. Although the higher computational/power efficiency of SNN is one of the well-known advantages over traditional neural architectures (Lee et al., 2020), further improvement in this direction can be based on rich dynamics of memristive devices and avoiding special programming circuitry used for the implementation of learning rules.

The most interesting direction at the boundaries of neurotechnology and neuromorphic prosthetics has recently emerged thanks to the seminal paper (Juzekaeva et al., 2019), where the main principles and feasibility of a memristive prosthesis of a synapse connecting two not connected via natural synapses neurons of a rat brain slice are proposed. This work triggered the discussion of the option to use stochastic memristive devices of different nature as main building block of neuromorphic prosthesis relocating functions and topology of natural neuronal circuits. Some steps in this direction have been already presented (Talanov et al., 2018b) including blueprints of a memristive neuron circuit (Talanov et al., 2017a,b, 2018a). As the number of memristive neurons available grows and the technology of their fabrication becomes more and more mature, we could expect the rise of the number of spiking solutions for the reimplementation of neuronal structures as electronic memristive circuits with more and more bio-plausible functions. Possibly, the most promising and timely problem, due to the lesser number of neurons and synapses, is the spinal cord direction that seems to attract rising interest of the researchers community (Gill et al., 2018; Wagner et al., 2018). The current state of neurorehabilitation of patients with complete spinal cord injury including epidural spinal cord stimulation is mainly experimental (Lavrov et al., 2008; Gad et al., 2013; Moraud et al., 2016), and it seems that a memristive implementation of part of the spinal cord circuits could restore the walking patterns of patients with complete SCI. We should not limit ourselves with the reimplementation of the part of the nervous system for patients, we could envision the further development of augmented nervous systems with digital extensions using

memristive properties of self-adaptation for the bidirectional brain to machine interfaces (Musk and Neuralink, 2019) based on the proposed neurohybrid chip approach.

CONCLUSION AND OUTLOOK

Here the concept of a single neurohybrid chip is proposed based on existing and future solutions in the field of neural cells and microfluidic technologies, which allow spatial structuring of living neural network combined with CMOS MEA and memristive arrays for real-time recording, processing, and stimulation of bioelectric activity interfaced and controlled by mixed analog–digital circuits on the same chip. This concept paves the way toward the creation of compact biosensors and neuroprosthetics that cannot be realized on the basis of traditional neurointerface architectures. The functionality of the proposed neurohybrid chip is limited in several domains on the side of electronic subsystem, including challenges associated with power consumption and reliability of memristive circuits. Significant efforts should be made to further understand the basic principles of learning in living neural networks and development of universal learning algorithms for SNN, providing their biological relevance and compatibility with memristive arrays. The key challenge on the road toward neurohybrid systems still remains in the reliable interaction between living neurons and electronics. Although memristors can provide efficient recording and on-chip processing of the neural activity, a number of problems are still related to biocompatibility and mechanical impact, geometry, placement, and miniaturization of electrodes and probes, as well as the reaction of living cultures and tissues at the interface with the artificial electronic subsystem. The potential transition from the proposed 2D to the 3D electrode system could provide some solutions, opening further questions related to implantation into deeper regions of the brain without causing structural damage to the tissue.

We hope that future realizations of the proposed concept may go beyond the CMOS limitations and rely on a direct synapse-scale interface based on organic and stochastic memristive nano-networks. Such nano-neurointerface should provide network distributed stimulation, when each stimulation event will be at the level of small synaptic currents of physiological range, hence not affecting the self-protection mechanisms of the brain. Designed this way, recording, processing, and stimulation electronic networks can be "physiologically" integrated into different brain areas to compensate or enhance brain functions from sensory level to the level of cognition and memory. Integrated into neural tissue memristive networks can also shed the light on the fundamental questions of analog neuron information processing.

To illustrate the proposed approaches and related products in a foreseeable timeline, **Figure 2** shows a roadmap of memristive neuromorphic and neurohybrid systems focused on the specialized hardware based on the architecture and principles of biological neural networks to support the development and mass introduction of artificial intelligence technologies, machine learning, neuroprosthetics, and neural interfaces. The roadmap starts tentatively in 2008 with the beginning of the current wave of interest to memristors (Strukov et al., 2008) and includes long-lasting research in the broad fields of neurobiology and neurophysiology. The following product niches are provided at different stages of development in this roadmap:

1) Neuromorphic computing systems;
2) Non-invasive memristive neurointerfaces;
3) Neuroimplants, neuroprostheses, and invasive neurointerfaces.

There are the unique properties of memristive devices that determine their decisive importance in the development of neuromorphic and neurohybrid systems for computing systems, brain–computer interfaces, and neuroprosthetics. These products will occupy a significant part of the global high-tech market worth trillions of dollars by 2030, taking into account the speed of development and implementation of artificial intelligence technologies, the Internet of Things, technologies of big data, smart city, robotics. Targets of the near future are neuroprosthetics, instrumental adjustment/support/enhancement of human sensing and cognitive abilities.

Hardware support is not just necessary for these technologies—the further development of neurocognitive technology industry and artificial intelligence is impossible without it due to the pronounced inadequacy of the traditional von Neumann architecture of computers for solving anthropogenic problems requiring a neural network architecture. As a result, we have unsatisfactory performance with huge energy consumption by the existing ICT infrastructure in the processing of even current (ongoing) anthropogenic tasks. This trend with the spread of intelligent technologies will only worsen, and therefore the development of specialized hardware of neuromorphic and neurohybrid systems (discussed here and based on memristors in a priority) is a key condition for the development of high-tech industries as a whole.

The development of artificial hardware systems should be in line with the bio- and neurotechnologies shown on the roadmap in the form of critical milestones, when a clear decision should be made on the most appropriate solutions. Over the past two decades, some progress has been observed in the development of biocompatible materials with the aim of creating multi-channel recording devices for neuronal networks activity both *in vitro* and *in vivo*. The prototypes of such devices are already implanted in the brain of animals for a long time with minimal immune response. Further optimization involves minimizing damage during implantation into the brain by reducing the size and increasing the flexibility of the probes in conjunction with the electrodes scaling. At the same time, the development of new-generation neuorohybrid systems will require a lot of special not yet obtained tools and preliminary experiments on animal models. These are the further advancement of neural interfacing (in view of microelectrode biocompatibility, reliability and rejection problems, 2D to 3D transition, etc.), chronic

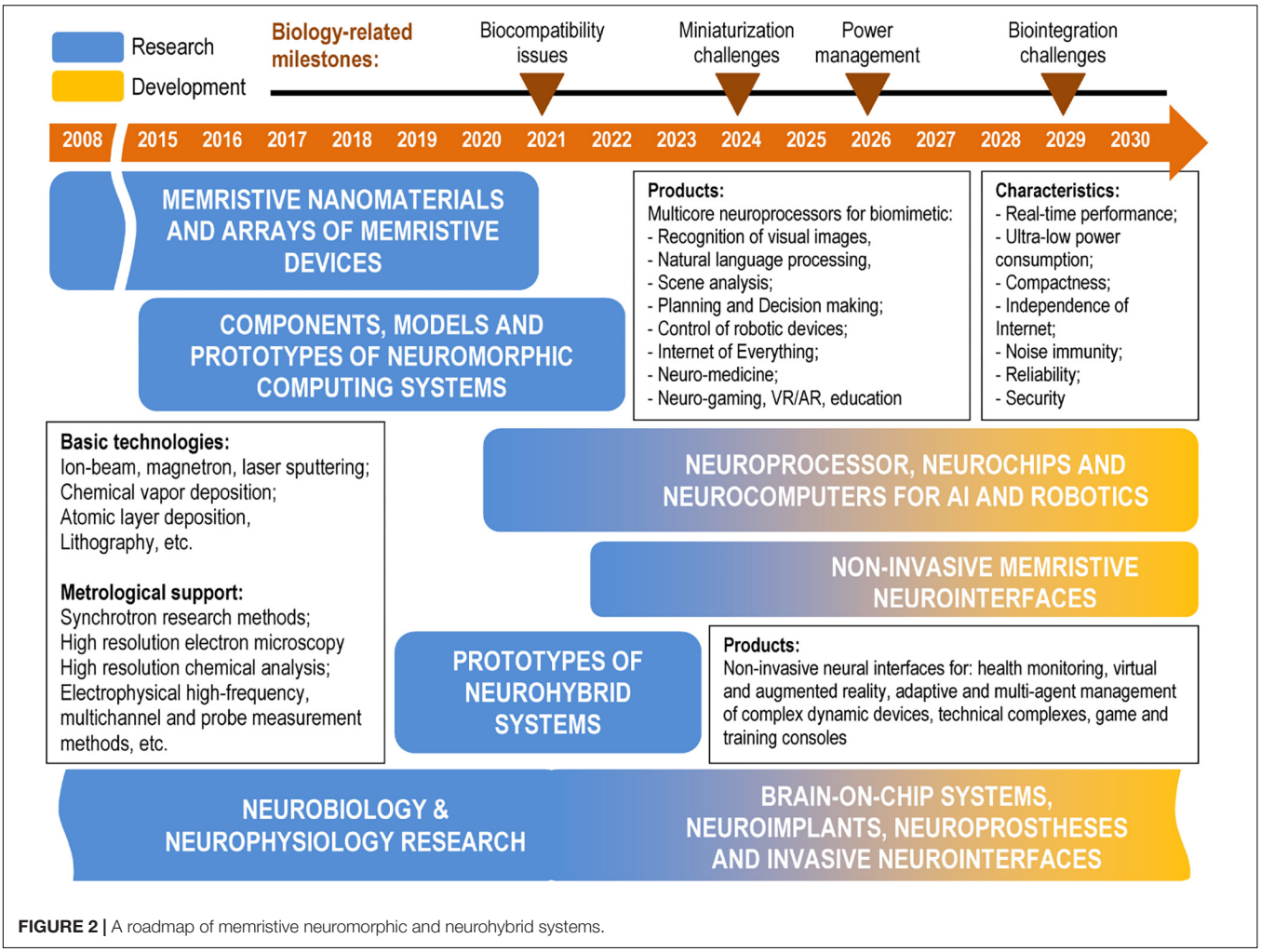


FIGURE 2 | A roadmap of memristive neuromorphic and neurohybrid systems.

neural pattern recognition and control devices (neurochips and algorithms for them), power management, signal processing, and data transfer in miniaturized platforms. Also, there is a problem with proof-of-principle investigation of which of neural circuits influence over disease progression in representative animal models. All these stages are crucial for the development of invasive neurointerfaces and memristive control systems for them. These neurobiology and neurophysiology involved investigations are still in progress and, due to the vast literature reports, could begin to develop mass products in 2 or 3 years.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

All authors gave substantial contribution to the development of this work equally, drafting, and revising it critically. Furthermore, all authors approved its final version for publication.

FUNDING

This work was supported by the Russian Science Foundation (Grant No. 16-19-00144) with regard to metal-oxide memristive arrays for bidirectional neurointerface, by the Russian Foundation for Basic Research (Grants Nos. 18-29-23001, 18-29-23041, 20-01-00368) with regard to brain-inspired memristive neural architectures and spiking neural networks, and by the Government of the Russian Federation [Agreement No. 074-02-2018-330 (2)] with regard to noise-induced phenomena in memristive materials, devices and networks. The studies of reliability and fault tolerance of memristor-based devices were partially supported by the Russian Foundation for Basic Research (Grants Nos. 18-38-00592, 19-07-01215), and the Grant of the President of the Russian Federation (MK-3927.2019.9). The reported research regarding the neuroprosthetics based on memristive schematics was partially funded by the Russian Foundation for Basic Research and the Government of the Republic of Tatarstan of the Russian Federation (Grant No. 18-44-160032), and by the Russian Foundation for Basic Research according to the research Project No. 19-29-03057. This work

was performed according to the Russian Government Program of Competitive Growth of Kazan Federal University. The reported study was also supported by the Foundation for Assistance to Small Innovative Enterprises in Science and Technology (FASIE) within the project "Neuromorphic neuroprosthesis for compensation of motor functions after spinal cord injury."

REFERENCES

Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., and Schumann, A. (2012). Physarum chip project: growing computers from slime mould. *Int. J. Unconvent. Comput.* 8, 319–323.

Agudov, N. V., Krichigin, A. V., Valenti, D., and Spagnolo, B. (2010). Stochastic resonance in a trapping overdamped monostable system. *Phys. Rev. E* 81:051123. doi: 10.1103/PhysRevE.81.051123

Agudov, N. V., Safonov, A. V., Krichigin, A. V., Kharcheva, A. A., Dubkov, A. A., Valenti, D., et al. (2020). Nonstationary distributions and relaxation times in a stochastic model of memristor. *J. Stat. Mech. Theory Exp.* 2020:24003.

Angotzi, G. N., Malerba, M., Maccione, A., Boi, F., Crepaldi, M., Bonanno, A., et al. (2017). "A high temporal resolution multiscale recording system for in vivo neural studies," in *Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, (Piscataway, NJ: IEEE), 1–4.

Battistoni, S., Erokhin, V., and Iannotta, S. (2019a). Frequency driven organic memristive devices for neuromorphic short term and long term plasticity. *Org. Electron.* 65, 434–438. doi: 10.1016/j.orgel.2018.11.033

Battistoni, S., Peruzzi, C., Verna, A., Marasso, S. L., Cocuzza, M., Erokhin, V., et al. (2019b). Synaptic response in organic electrochemical transistor gated by a graphene electrode. *Flex. Print. Electron.* 4:044002. doi: 10.1088/2058-8585/AB4DCE

Bayat, F. M., Prezioso, M., Chakrabarti, B., Nili, H., Kataeva, I., and Strukov, D. (2018). Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. Commun.* 9:2331. doi: 10.1038/s41467-018-04482-4

Berger, T. W., Hampson, R. E., Song, D., Goonawardena, A., Marmarelis, V. Z., and Deadwyler, S. A. (2011). A cortical neural prosthesis for restoring and enhancing memory. *J. Neural Eng.* 8:046017. doi: 10.1088/1741-2560/8/4/046017

Bi, G. Q., and Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472. doi: 10.1523/JNEUROSCI.18-24-10464.1998

Bienenstock, E., Cooper, L., and Munro, P. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2, 32–48. doi: 10.1523/JNEUROSCI.02-01-00032.1982

Birmingham, K., Gradinaru, V., Anikeeva, P., Grill, W. M., Pikov, V., McLaughlin, B., et al. (2014). Bioelectronic medicines: a research roadmap. *Nat. Rev. Drug Discov.* 13, 399–400. doi: 10.1038/nrd4351

Boi, F., Moraitis, T., De Feo, V., Diotalevi, F., Bartolozzi, C., Indiveri, G., et al. (2016). A bidirectional brain-machine interface featuring a neuromorphic hardware decoder. *Front. Neurosci.* 10:563. doi: 10.3389/fnins.2016.00563

Brewer, G. J., Boehler, M. D., Leondopulos, S., Pan, L., DeMarse, T., and Wheeler, B. C. (2013). Toward a self-wired active reconstruction of the hippocampal trisynaptic loop: DG-CA3. *Front. Neural Circuits* 7:165. doi: 10.3389/fncir.2013.00165

Brownlee, J. (2019). *A Gentle Introduction to RNN Unrolling*. Available online at: https://machinelearningmastery.com/rnn-unrolling/ (accessed October 14, 2019).

Buccelli, S., Bornat, Y., Colombi, I., Ambroise, M., Martines, L., Pasquale, V., et al. (2019). A neuromorphic prosthesis to restore communication in neuronal networks. *IScience* 19, 402–414. doi: 10.1016/j.isci.2019.07.046

Cai, F., Correll, J. M., Lee, S. H., Lim, Y., Bothra, V., Zhang, Z., et al. (2019). A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nat. Electron.* 2, 290–299. doi: 10.1038/s41928-019-0270-x

Chichigina, O., Valenti, D., and Spagnolo, B. (2005). A simple noise model with memory for biological systems. *Fluct. Noise Lett.* 05, L243–L250. doi: 10.1142/S0219477505002616

Chichigina, O. A., Dubkov, A. A., Valenti, D., and Spagnolo, B. (2011). Stability in a system subject to noise with regulated periodicity. *Phys. Rev. E* 84:021134. doi: 10.1103/PhysRevE.84.021134

Chiolerio, A., Chiappalone, M., Ariano, P., and Bocchini, S. (2017). Coupling resistive switching devices with neurons: state of the art and perspectives. *Front. Neurosci.* 11:70. doi: 10.3389/fnins.2017.00070

Chua, L. O. (1971). Memristor - the missing circuit element. *IEEE Trans. Circuit Theory* 18, 507–519. doi: 10.1109/tct.1971.1083337

Chuang, A. T., Margo, C. E., and Greenberg, P. B. (2014). Retinal implants: a systematic review. *Br. J. Ophthalmol.* 98, 852–856. doi: 10.1136/bjophthalmol-2013-303708

Clopath, C., Büsing, L., Vasilaki, E., and Gerstner, W. (2010). Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nat. Neurosci.* 13, 344–352. doi: 10.1038/nn.2479

Danilin, S., Shchanikov, S., Bordanov, I., and Zuev, A. (2019). Using simulation to define the tolerances for the information and physical parameters of memristors-based artificial neural networks. *J. Phys. Conf. Ser.* 1333:062026. doi: 10.1088/1742-6596/1333/6/062026

De Faveri, S., Maggiolini, E., Miele, E., De Angelis, F., Cesca, F., Benfenati, F., et al. (2014). Bio-inspired hybrid microelectrodes: a hybrid solution to improve long-term performance of chronic intracortical implants. *Front. Neuroeng.* 7:7. doi: 10.3389/fneng.2014.00007

Demin, V., and Nekhaev, D. (2018). Recurrent spiking neural network learning based on a competitive maximization of neuronal activity. *Front. Neuroinform.* 12:79. doi: 10.3389/fninf.2018.00079

Doucey, M.-A., and Carrara, S. (2019). Nanowire sensors in cancer. *Trends Biotechnol.* 37, 86–99. doi: 10.1016/j.tibtech.2018.07.014

Dryg, I. D., Ward, M. P., Qing, K. Y., Mei, H., Schaffer, J. E., and Irazoqui, P. P. (2015). Magnetically inserted neural electrodes: tissue response and functional lifetime. *IEEE Trans. Neural Syst. Rehabil. Eng.* 23, 562–571. doi: 10.1109/TNSRE.2015.2399856

Emelyanov, A. V., Nikiruy, K. E., Demin, V. A., Rylkov, V. V., Belov, A. I., Korolev, D. S., et al. (2019). Yttria-stabilized zirconia cross-point memristive devices for neuromorphic applications. *Microelectron. Eng.* 215:110988. doi: 10.1016/j.mee.2019.110988

Emelyanov, A. V., Nikiruy, K. E., Serenko, A. V., Sitnikov, A. V., Presnyakov, M. Y., Rybka, R. B., et al. (2020). Self-adaptive STDP-based learning of a spiking neuron with nanocomposite memristive weights. *Nanotechnology* 31:045201. doi: 10.1088/1361-6528/AB4A6D

Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. Natl. Acad. Sci. U.S.A.* 113, 11441–11446. doi: 10.1073/pnas.1604850113

Falci, G., La Cognata, A., Berritta, M., D'Arrigo, A., Paladino, E., and Spagnolo, B. (2013). Design of a Lambda system for population transfer in superconducting nanocircuits. *Phys. Rev. B* 87:214515. doi: 10.1103/PhysRevB.87.214515

Fernandez, E. (2018). Development of visual Neuroprostheses: trends and challenges. *Bioelectron. Med.* 4:12. doi: 10.1186/s42234-018-0013-8

Fiasconaro, A., Valenti, D., and Spagnolo, B. (2004). Nonmonotonic behavior of spatiotemporal pattern formation in a noisy Lotka-Volterra system. *Acta Phys. Polon. B* 35, 1491–1500.

Filatov, D. O., Vrzheshch, D. V., Tabakov, O. V., Novikov, A. S., Belov, A. I., Antonov, I. N., et al. (2019). Noise-induced resistive switching in a memristor based on $ZrO_2(Y)/Ta_2O_5$ stack. *J. Stat. Mech. Theory Exp.* 2019:124026. doi: 10.1088/1742-5468/ab5704

Forró, C., Thompson-Steckel, G., Weaver, S., Weydert, S., Ihle, S., Dermutz, H., et al. (2018). Modular microstructure design to build neuronal networks of defined functional connectivity. *Biosens. Bioelectron.* 122, 75–87. doi: 10.1016/j.bios.2018.08.075

Fukuma, R., Yanagisawa, T., Saitoh, Y., Hosomi, K., Kishima, H., Shimizu, T., et al. (2016). Real-time control of a neuroprosthetic hand by

magnetoencephalographic signals from paralysed patients. *Sci. Rep.* 6:21781. doi: 10.1038/srep21781

Gad, P., Lavrov, I., Shah, P., Zhong, H., Roy, R. R., Edgerton, V. R., et al. (2013). Neuromodulation of motor-evoked potentials during stepping in spinal rats. *J. Neurophysiol.* 110, 1311–1322. doi: 10.1152/jn.00169.2013

Gerasimova, S. A., Mikhaylov, A. N., Belov, A. I., Korolev, D. S., Gorshkov, O. N., and Kazantsev, V. B. (2017). Simulation of synaptic coupling of neuron-like generators via a memristive device. *Tech. Phys.* 62, 1259–1265. doi: 10.1134/S1063784217080102

Ghane-Motlagh, B., and Sawan, M. (2013). Design and implementation challenges of microelectrode arrays: a review. *Mater. Sci. Appl.* 4, 483–495. doi: 10.4236/msa.2013.48059

Gill, M. L., Grahn, P. J., Calvert, J. S., Linde, M. B., Lavrov, I. A., Strommen, J. A., et al. (2018). Neuromodulation of lumbosacral spinal networks enables independent stepping after complete paraplegia. *Nat. Med.* 24, 1677–1682. doi: 10.1038/s41591-018-0175-7

Gkoupidenis, P., Koutsouras, D. A., and Malliaras, G. G. (2017). Neuromorphic device architectures with global connectivity through electrolyte gating. *Nat. Commun.* 8:15448. doi: 10.1038/ncomms15448

Gkoupidenis, P., Schaefer, N., Strakosas, X., Fairfield, J. A., and Malliaras, G. G. (2015). Synaptic plasticity functions in an organic electrochemical transistor. *Appl. Phys. Lett.* 107:263302. doi: 10.1063/1.4938553

Gladkov, A., Pigareva, Y., Kutyina, D., Kolpakov, V., Bukatin, A., Mukhina, I., et al. (2017). Design of cultured neuron networks in vitro with predefined connectivity using asymmetric microfluidic channels. *Sci. Rep.* 7:15625. doi: 10.1038/s41598-017-15506-2

Goel, P., Liu, H., Brown, D. J., and Datta, A. (2006). "Spiking neural network based classification of task-evoked EEG signals," in *Knowledge-Based Intelligent Information and Engineering Systems*, eds B. Gabrys, R. J. Howlett, and L. C. Jain (Berlin: Springer).

Graczyk, E. L., Delhaye, B. P., Schiefer, M. A., Bensmaia, S. J., and Tyler, D. J. (2018). Sensory adaptation to electrical stimulation of the somatosensory nerves. *J. Neural Eng.* 15:046002. doi: 10.1088/1741-2552/aab790

Guo, Y., Wu, H., Gao, B., and Qian, H. (2019). Unsupervised learning on resistive memory array based spiking neural networks. *Front. Neurosci.* 13:812. doi: 10.3389/fnins.2019.00812

Gupta, I., Serb, A., Khiat, A., Zeitler, R., Vassanelli, S., and Prodromakis, T. (2016). Real-time encoding and compression of neuronal spikes by metal-oxide memristors. *Nat. Commun.* 7:12805. doi: 10.1038/ncomms12805

Gupta, I., Serb, A., Khiat, A., Zeitler, R., Vassanelli, S., and Prodromakis, T. (2018). Sub 100 nW volatile nano-metal-oxide memristor as synaptic-like encoder of neuronal spikes. *IEEE Trans. Biomed. Circuits Syst.* 12, 351–359. doi: 10.1109/TBCAS.2018.2797939

Hampson, R. E., Song, D., Robinson, B. S., Fetterhoff, D., Dakos, A. S., Roeder, B. M., et al. (2018). Developing a hippocampal neural prosthetic to facilitate human memory encoding and recall. *J. Neural Eng.* 15:036014. doi: 10.1088/1741-2552/aaaed7

He, W., McConnell, G. C., and Bellamkonda, R. V. (2006). Nanoscale laminin coating modulates cortical scarring response around implanted silicon microelectrode arrays. *J. Neural Eng.* 3, 316–326. doi: 10.1088/1741-2560/3/4/009

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hogri, R., Bamford, S. A., Taub, A. H., Magal, A., Del Giudice, P., and Mintz, M. (2015). A neuro-inspired model-based closed-loop neuroprosthesis for the substitution of a cerebellar learning function in anesthetized rats. *Sci. Rep.* 5:8451. doi: 10.1038/srep08451

Hong, S., Ning, L., Xiaoping, L., and Qian, W. (2010). "A cooperative method for supervised learning in Spiking neural networks," in *Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design*, Dalian, 22–26. doi: 10.1109/CSCWD.2010.5472007

Horowitz, M. (2014). 1.1 Computing's energy problem (and what we can do about it," in *Proceedings of the IEEE International Solid-State Circuits Conference on Digest of Technical Papers*, Montgomery Village, MD. doi: 10.1109/ISSCC.2014.6757323

Ielmini, D., and Waser, R. (2016). *Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications*. Weinheim: WILEY-VCH, 2016.

Ignatov, M., Hansen, M., Ziegler, M., and Kohlstedt, H. (2016). Synchronization of two memristively coupled van der Pol oscillators. *Appl. Phys. Lett.* 108:084105. doi: 10.1063/1.4942832

Inal, S., Rivnay, J., Suiu, A. O., Malliaras, G. G., and McCulloch, I. (2018). Conjugated polymers in bioelectronics. *Acc. Chem. Res.* 51, 1368–1376. doi: 10.1021/acs.accounts.7b00624

Juzekaeva, E., Nasretdinov, A., Battistoni, S., Berzina, T., Iannotta, S., Khazipov, R., et al. (2019). Coupling cortical neurons through electronic memristive synapse. *Adv. Mater. Technol.* 4:1800350. doi: 10.1002/admt.201800350

Kataeva, I., Ohtsuka, S., Nili, H., Kim, H., Isobe, Y., Yako, K., et al. (2019). "Towards the development of analog neuromorphic chip prototype with 2.4M integrated memristors," in *Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, Sapporo. doi: 10.1109/ISCAS.2019.8702125

Kim, S., Du, C., Sheridan, P., Ma, W., Choi, S., and Lu, W. D. (2015). Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity. *Nano Lett.* 15, 2203–2211. doi: 10.1021/acs.nanolett.5b00697

Kim, S., Tathireddy, P., Normann, R. A., and Solzbacher, F. (2007). Thermal impact of an active 3-D microelectrode array implanted in the brain. *IEEE Trans. Neural Syst. Rehabil. Eng.* 15, 493–501. doi: 10.1109/TNSRE.2007.908429

La Barbera, A., and Spagnolo, B. (2002). Spatio-temporal patterns in population dynamics. *Phys. A Stat. Mech. Appl.* 314, 120–124.

Lavrov, I., Dy, C. J., Fong, A. J., Gerasimenko, Y., Courtine, G., Zhong, H., et al. (2008). Epidural stimulation induced modulation of spinal locomotor networks in adult spinal rats. *J. Neurosci.* 28, 6022–6029. doi: 10.1523/JNEUROSCI.0080-08.2008

le Feber, J., Postma, W., de Weerd, E., Weusthof, M., and Rutten, W. L. (2015). Barbed channels enhance unidirectional connectivity between neuronal networks cultured on multi electrode arrays. *Front. Neurosci.* 9:412. doi: 10.3389/fnins.2015.00412

Lee, S. H., Zhu, X., and Lu, W. D. (2020). Nanoscale resistive switching devices for memory and computing applications. *Nano Res.* doi: 10.1007/s12274-020-2616-0

Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., et al. (2018a). Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* 9:2385. doi: 10.1038/s41467-018-04484-2

Li, C., Hu, M., Li, Y., Jiang, H., Ge, N., Montgomery, E., et al. (2018b). Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* 1, 52–59. doi: 10.1038/s41928-017-0002-z

Ling, H., Koutsouras, D. A., Kazemzadeh, S., Van De Burgt, Y., Yan, F., and Gkoupidenis, P. (2020). Electrolyte-gated transistors for synaptic electronics, neuromorphic computing, and adaptable biointerfacing. *Appl. Phys. Rev.* 7:011307. doi: 10.1063/1.5122249

Lobov, S., Mironov, V., Kastalskiy, I., and Kazantsev, V. (2015). A spiking neural network in SEMG feature extraction. *Sensors* 15, 27894–27904. doi: 10.3390/s151127894

Lobov, S., Krilova, N., Kastalskiy, I., Kazantsev, V., and Makarov, V. A. (2016). A human-computer interface based on electromyography command-proportional control. in *Proceedings of the 4th International Congress on Neurotechnology, Electronics and Informatics*. Available online at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85006379951&partnerID=40&md5=15efb57b2728f028e8cce9d4e2051f3, 57–64.

Lobov, S. A., Chernyshov, A. V., Krilova, N. P., Shamshin, M. O., and Kazantsev, V. B. (2020a). Competitive learning in a spiking neural network: towards an intelligent pattern classifier. *Sensors* 20:500. doi: 10.3390/s20020500

Lobov, S. A., Mikhaylov, A. N., Shamshin, M., Makarov, V. A., and Kazantsev, V. B. (2020b). Spatial properties of STDP in a self-learning spiking neural network enable controlling a mobile robot. *Front. Neurosci.* 14:88. doi: 10.3389/fnins.2020.00088

Lopez, C. M. (2019). Unraveling the brain with high-density CMOS neural probes: tackling the challenges of neural interfacing. *IEEE Solid State Circuits Mag.* 11, 43–50. doi: 10.1109/mssc.2019.2939338

Manning, H. G., Niosi, F., da Rocha, C. G., Bellew, A. T., O'Callaghan, C., Biswas, S., et al. (2018). Emergence of winner-takes-all connectivity paths in random nanowire networks. *Nat. Commun.* 9:3219. doi: 10.1038/s41467-018-05517-6

Masius, A. A., and Wong, Y. C. (2020). On-chip miniaturized antenna in CMOS technology for biomedical implant. *AEU Int. J. Electron. Commun.* 115:153025. doi: 10.1016/j.aeue.2019.153025

Masquelier, T., and Deco, G. (2013). "Learning and coding in neural networks," in *Principles of Neural Coding*, eds R. Q. Quiroga and S. Panzeri (Boca Raton, FL: CRC Press), 513–526. doi: 10.1201/b14756-30

Middleton, J. W., Kinnischtzke, A., and Simons, D. J. (2010). Effects of thalamic high-frequency electrical stimulation on whisker-evoked cortical adaptation. *Exp. Brain Res.* 200, 239–250. doi: 10.1007/s00221-009-1977-0

Mikhaylov, A. N., Gryaznov, E. G., Belov, A. I., Korolev, D. S., Sharapov, A. N., Guseinov, D. V., et al. (2016). Field- and irradiation-induced phenomena in memristive nanomaterials. *Phys. Status Solidi Curr. Top. Solid State Phys.* 13, 870–881. doi: 10.1002/pssc.201600083

Mikhaylov, A. N., Morozov, O. A., Ovchinnikov, P. E., Antonov, I. N., Belov, A. I., Korolev, D. S., et al. (2018). One-board design and simulation of double-layer perceptron based on metal-oxide memristive nanostructures. *IEEE Trans. Emerg. Top. Comput. Intell.* 2, 371–379. doi: 10.1109/tetci.2018.2829922

Minnekhanov, A. A., Emelyanov, A. V., Lapkin, D. A., Nikiruy, K. E., Shvetsov, B. S., Nesmelov, A. A., et al. (2019). Parylene based memristive devices with multilevel resistive switching for neuromorphic applications. *Sci. Rep.* 9:10800. doi: 10.1038/s41598-019-47263-9

Mironov, V., Kastalskiy, I., Lobov, S., and Kazantsev, V. (2017). "A biofeedback control system of the exoskeleton trainer for lower limbs motor function recovery," in *Proceedings of the 5th International Congress on Neurotechnology, Electronics and Informatics (NEUROTECHNIX 2017)*, Funchal, 54–59. doi: 10.5220/0006584700540059

Mironov, V. I., Lobov, S. A., Krylova, N. P., Gordleeva, S. Y., Kaplan, A. Y., Buylova, T. V., et al. (2018). Development of a neurally-controlled vehicle — neuro-mobile — for driving by individuals with motor deficiency. *Sovrem. Tehnol. Med.* 10, 49–59. doi: 10.17691/stm2018.10.4.06

Moraud, E. M., Capogrosso, M., Formento, E., Wenger, N., DiGiovanna, J., Courtine, G., et al. (2016). Mechanisms underlying the neuromodulation of spinal circuits for correcting gait and balance deficits after spinal cord injury. *Neuron* 89, 814–828. doi: 10.1016/j.neuron.2016.01.009

Morris, R. G. M. (1999). D.O. Hebb: the organization of behavior, Wiley: New York; 1949. *Brain Res. Bull.* 50:437. doi: 10.1016/s0361-9230(99)00182-3

Müller, J., Ballini, M., Livi, P., Chen, Y., Radivojevic, M., Shadmani, A., et al. (2015). High-resolution CMOS MEA platform to study neurons at subcellular, cellular, and network levels. *Lab Chip* 15, 2767–2780. doi: 10.1039/c5lc00133a

Musk, E., and Neuralink. (2019). An integrated brain-machine interface platform with thousands of channels. *J. Med. Internet Res.* 21:e16194. doi: 10.1101/703801

Na, S., Kang, M., Bang, S., Park, D., Kim, J., Sim, S. J., et al. (2016). Microfluidic neural axon diode. *Technology* 4, 240–248. doi: 10.1142/s2339547816500102

NIH (2016). *Cochlear Implants | NIDCD*. Available online at: https://www.nidcd.nih.gov/health/cochlear-implants (accessed October 14, 2019).

Obien, M. E. J., Deligkaris, K., Bullmann, T., Bakkum, D. J., and Frey, U. (2015). Revealing neuronal function through microelectrode array recordings. *Front. Neurosci.* 8:423. doi: 10.3389/fnins.2014.00423

O'Kelly, C. J., Fairfield, J. A., McCloskey, D., Manning, H. G., Donegan, J. F., and Boland, J. J. (2016). Associative enhancement of time correlated response to heterogeneous stimuli in a neuromorphic nanowire device. *Adv. Electron. Mater.* 2:1500458. doi: 10.1002/aelm.201500458

Patel, P. R., Zhang, H., Robbins, M. T., Nofar, J. B., Marshall, S. P., Kobylarek, M. J., et al. (2016). Chronic in vivo stability assessment of carbon fiber microelectrode arrays. *J. Neural Eng.* 13:066002. doi: 10.1088/1741-2560/13/6/066002

Petrini, F. M., Bumbasirevic, M., Valle, G., Ilic, V., Mijoviæ, P., Èvanèara, P., et al. (2019a). Sensory feedback restoration in leg amputees improves walking speed, metabolic cost and phantom pain. *Nat. Med.* 25, 1356–1363. doi: 10.1038/s41591-019-0567-3

Petrini, F. M., Mazzoni, A., Rigosa, J., Giambattistelli, F., Granata, G., Barra, B., et al. (2019b). Microneurography as a tool to develop decoding algorithms for peripheral neuro-controlled hand prostheses. *Biomed. Eng.* 18:44. doi: 10.1186/s12938-019-0659-9

Pfister, J.-P., and Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.* 26, 9673–9682. doi: 10.1523/JNEUROSCI.1425-06.2006

Pi, S., Li, C., Jiang, H., Xia, W., Xin, H., Yang, J. J., et al. (2019). Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension. *Nat. Nanotechnol.* 14, 35–39. doi: 10.1038/s41565-018-0302-0

Piech, D. K., Johnson, B. C., Shen, K., Ghanbari, M. M., Li, K. Y., Neely, R. M., et al. (2020). A wireless millimetre-scale implantable neural stimulator with ultrasonically powered bidirectional communication. *Nat. Biomed. Eng.* 4, 207–222. doi: 10.1038/s41551-020-0518-9

Poli, D., Thiagarajan, S., DeMarse, T. B., Wheeler, B. C., and Brewer, G. J. (2017). Sparse and specific coding during information transmission between co-cultured dentate gyrus and CA3 hippocampal networks. *Front. Neural Circuits* 11:13. doi: 10.3389/fncir.2017.00013

Prezioso, M., Merrikh-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 61–64. doi: 10.1038/nature14441

Qi, D., Liu, Z., Liu, Y., Jiang, Y., Leow, W. R., Pal, M., et al. (2017). Highly stretchable, compliant, polymeric microelectrode arrays for in vivo electrophysiological interfacing. *Adv. Mater.* 29:1702800. doi: 10.1002/adma.201702800

Raducanu, B. C., Yazicioglu, R. F., Lopez, C. M., Ballini, M., Putzeys, J., Wang, S., et al. (2017). Time multiplexed active neural probe with 1356 parallel recording sites. *Sensors* 17:2388. doi: 10.3390/s17102388

Renault, R., Durand, J. B., Viovy, J. L., and Villard, C. (2016). Asymmetric axonal edge guidance: a new paradigm for building oriented neuronal networks. *Lab Chip* 16, 2188–2191. doi: 10.1039/c6lc00479b

Rivnay, J., Inal, S., Salleo, A., Owens, R. M., Berggren, M., and Malliaras, G. G. (2018). Organic electrochemical transistors. *Nat. Rev. Mater.* 3:17086. doi: 10.1038/natrevmats.2017.86

Rouger, J., Lagleyre, S., Fraysse, B., Deneve, S., Deguine, O., and Barone, P. (2007). Evidence that cochlear-implanted deaf patients are better multisensory integrators. *Proc. Natl. Acad. Sci. U.S.A.* 104, 7295–7300. doi: 10.1073/pnas.0609419104

Santoro, F., van de Burgt, Y., Keene, S. T., Cui, B., and Salleo, A. (2017). Enhanced cell–chip coupling by rapid femtosecond laser patterning of soft PEDOT: PSS biointerfaces. *ACS Appl. Mater. Interfaces* 9, 39116–39121. doi: 10.1021/acsami.7b12308

Scholvin, J., Kinney, J. P., Bernstein, J. G., Moore-Kochlacs, C., Kopell, N., Fonstad, C. G., et al. (2015). Close-packed silicon microelectrodes for scalable spatially oversampled neural recording. *IEEE Trans. Biomed. Eng.* 63, 120–130. doi: 10.1109/TBME.2015.2406113

Serb, A., Bill, J., Khiat, A., Berdan, R., Legenstein, R., and Prodromakis, T. (2016). Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses. *Nat. Commun.* 7:12611. doi: 10.1038/ncomms12611

Shchanikov, S., Zuev, A., Bordanov, I., Danilin, S., Korolev, D., Belov, A., et al. (2020). Design and simulation of memristor-based artificial neural network for bidirectional adaptive neural interface. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2004.00154 (accessed April 02, 2020).

Shen, W., Karumbaiah, L., Liu, X., Saxena, T., Chen, S., Patkar, R., et al. (2015). Extracellular matrix-based intracortical microelectrodes: toward a microfabricated neural interface based on natural materials. *Microsyst. Nanoeng.* 2015:15010. doi: 10.1038/micronano.2015.10

Shimba, K., Chang, C. H., Asahina, T., Moriya, F., Kotani, K., Jimbo, Y., et al. (2019). Functional scaffolding for brain implants: engineered neuronal network by microfabrication and iPSC technology. *Front. Neurosci.* 13:890. doi: 10.3389/fnins.2019.00890

Sillay, K. A., Rutecki, P., Cicora, K., Worrell, G., Drazkowski, J., Shih, J. J., et al. (2013). Long-term measurement of impedance in chronically implanted depth and subdural electrodes during responsive neurostimulation in humans. *Brain Stimul.* 6, 718–726. doi: 10.1016/j.brs.2013.02.001

Song, D., Robinson, B. S., Hampson, R. E., Marmarelis, V. Z., Deadwyler, S. A., and Berger, T. W. (2018). Sparse large-scale nonlinear dynamical modeling of human hippocampus for memory prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 272–280. doi: 10.1109/TNSRE.2016.2604423

Spagnolo, B., Carollo, A., and Valenti, D. (2018a). Enhancing metastability by dissipation and driving in an asymmetric bistable quantum system. *Entropy* 20:226. doi: 10.3390/e20040226

Spagnolo, B., Carollo, A., and Valenti, D. (2018b). Stabilization by dissipation and stochastic resonant activation in quantum metastable systems. *Eur. Phys. J. Spec. Top.* 227, 379–420. doi: 10.1140/epjst/e2018-00121-x

Spagnolo, B., Guarcello, C., Magazzù, L., Carollo, A., Persano Adorno, D., and Valenti, D. (2016). Nonlinear relaxation phenomena in metastable condensed matter systems. *Entropy* 19:20. doi: 10.3390/e19010020

Spagnolo, B., Valenti, D., Guarcello, C., Carollo, A., Persano Adorno, D., Spezia, S., et al. (2015). Noise-induced effects in nonlinear relaxation of condensed matter systems. *Chaos Solitons Fractals* 81, 412–424. doi: 10.1016/j.chaos.2015.07.023

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

Sun, Z., Pedretti, G., Bricalli, A., and Ielmini, D. (2020). One-step regression and classification with cross-point resistive memory arrays. *Sci. Adv.* 6:eaay2378. doi: 10.1126/sciadv.aay2378

Tahtirvanci, A., Durdu, A., and Yilmaz, B. (2018). "Classification of EEG signals using spiking neural networks," in *Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU),* Izmir, 1–4. doi: 10.1109/SIU.2018.8404470

Talanov, M., Gerasimov, Y., and Erokhin, V. (2018a). Electronic schematic for bio-plausible dopamine neuromodulation of eSTDP and iSTDP. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1806.04703 (accessed April 01, 2020).

Talanov, M., Lavrov, I., and Erokhin, V. (2018b). Modelling reflex arc for a memristive implementation. *Eur. J. Clin. Investig.* 48, 223–223.

Talanov, M., Zykov, E. Y., Erokhin, V., Magid, E., and Distefano, S. (2017a). "The memristive artificial neuron high level architecture for biologically inspired robotic systems," in *Proceedings of the 2017 International Conference on Mechanical, System and Control Engineering (ICMSC)* (Piscataway, NJ: IEEE), 196–200.

Talanov, M., Zykov, E., Erokhin, V., Magid, E., Distefano, S., Gerasimov, Y., et al. (2017b). "Modeling inhibitory and excitatory synapse learning in the memristive neuron model," in *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017)*, Madrid, 514–521.

Tarabella, G., D'Angelo, P., Cifarelli, A., Dimonte, A., Romeo, A., Berzina, T., et al. (2015). A hybrid living/organic electrochemical transistor based on the Physarum polycephalum cell endowed with both sensing and memristive properties. *Chem. Sci.* 6, 2859–2868. doi: 10.1039/c4sc03425b

Tien, N. W., and Kerschensteiner, D. (2018). Homeostatic plasticity in neural development. *Neural Dev.* 13, 1–7. doi: 10.1155/2014/128631

Tzouvadaki, I., Puppo, F., Doucey, M.-A., De Micheli, G., and Carrara, C. (2015). Computational study of the electrical behavior of silicon nanowire memristive biosensors. *IEEE Sens. J.* 15, 6208–6217. doi: 10.1109/jsen.2015.2456336

Valenti, D., Fiasconaro, A., and Spagnolo, B. (2004). Pattern formation and spatial correlation induced by the noise in two competing species. *Acta Phys. Polon. B* 35, 1481–1489.

Valenti, D., Magazzù, L., Caldara, P., and Spagnolo, B. (2015). Stabilization of quantum metastable states by dissipation. *Phys. Rev. B* 91:235412. doi: 10.1103/PhysRevB.91.235412

Valenti, D., Tranchina, L., Brai, M., Caruso, A., Cosentino, C., and Spagnolo, B. (2008). Environmental metal pollution considered as noise: Effects on the spatial distribution of benthic foraminifera in two coastal marine areas of Sicily (Southern Italy). *Ecol. Modell.* 213, 449–462. doi: 10.1016/j.ecolmodel.2008.01.023

van De Burgt, Y., Melianas, A., Keene, S. T., Malliaras, G., and Salleo, A. (2018). Organic electronics for neuromorphic computing. *Nat. Electron.* 1, 386–397. doi: 10.1038/s41928-018-0103-3

van Rossum, M. C. W., Bi, G. Q., and Turrigiano, G. G. (2000). Stable Hebbian Learning from Spike Timing-Dependent Plasticity. *J. Neurosci.* 20, 8812–8821. doi: 10.1523/JNEUROSCI.20-23-08812.2000

Vassanelli, S., and Mahmud, M. (2016). Trends and challenges in neuroengineering: toward "intelligent" neuroprostheses through brain-"brain inspired systems" communication. *Front. Neurosci.* 10:438. doi: 10.3389/fnins.2016.00438

Wagner, F. B., Mignardot, J.-B., Le Goff-Mignardot, C. G., Demesmaeker, R., Komi, S., Capogrosso, M., et al. (2018). Targeted neurotechnology restores walking in humans with spinal cord injury. *Nature* 563, 65–71. doi: 10.1038/s41586-018-0649-2

Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., et al. (2018). Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* 1, 137–145.

Wang, Z., Li, C., Lin, P., Rao, M., Nie, Y., Song, W., et al. (2019). In situ training of feed-forward and recurrent convolutional memristor networks. *Nat. Mach. Intell.* 1, 434–442. doi: 10.1038/s42256-019-0089-1

Won, S. M., Song, E., Zhao, J., Li, J., Rivnay, J., and Rogers, J. A. (2018). Recent advances in materials, devices, and systems for neural interfaces. *Adv. Mater.* 30:1800534. doi: 10.1002/adma.201800534

Wu, C., Evans, J. J., Skidmore, C., Sperling, M. R., and Sharan, A. D. (2013). Impedance variations over time for a closed-loop neurostimulation device: early experience with chronically implanted electrodes. *Neuromodulation* 16, 46–50. doi: 10.1111/j.1525-1403.2012.00529.x

Xia, Q., and Yang, J. J. (2019). Memristive crossbar arrays for brain-inspired computing. *Nat. Mater.* 18, 309–323. doi: 10.1038/s41563-019-0291-x

Xu, Y., Zeng, X., Han, L., and Yang, J. (2013). A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks. *Neural Netw.* 43, 99–113. doi: 10.1016/j.neunet.2013.02.003

Xydas, D., Norcott, D. J., Warwick, K., Whalley, B. J., Nasuto, S. J., Becerra, V. M., et al. (2008). Architecture for neuronal cell control of a mobile robot. *Springer Tracts Adv. Robot.* 44, 23–31. doi: 10.1007/978-3-540-78317-6_3

Yao, P., Wu, H., Gao, B., Eryilmaz, S. B., Huang, X., Zhang, W., et al. (2017). Face classification using electronic synapses. *Nat. Commun.* 8:15199. doi: 10.1038/ncomms15199

Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., et al. (2020). Fully hardware-implemented memristor convolutional neural network. *Nature* 577, 641–646. doi: 10.1038/s41586-020-1942-4

Yousefzadeh, A., Stromatias, E., Soto, M., Serrano-Gotarredona, T., and Linares-Barranco, B. (2018). On practical issues for stochastic STDP hardware with 1-bit synaptic weights. *Front. Neurosci.* 12:665. doi: 10.3389/fnins.2018.00665

Zhao, M., Gao, B., Tang, J., Qian, H., and Wu, H. (2020). Reliability of analog resistive switching memory for neuromorphic computing. *Appl. Phys. Rev.* 7:011301. doi: 10.1063/1.5124915

Zhou, Y., Wu, H., Gao, B., Wu, W., Xi, Y., Yao, P., et al. (2019). Associative Memory for Image Recovery with a High-Performance Memristor Array. *Adv. Funct. Mater.* 29:1900155. doi: 10.1002/adfm.201900155

Zhuk, M., Zarubin, S., Karateev, I., Matveyev, Y., Gornev, E., Krasnikov, G., et al. (2020). On-Chip TaO*x*-Based Non-volatile Resistive Memory for *in vitro* Neurointerfaces. *Front. Neurosci.* 14:94. doi: 10.3389/fnins.2020.00094

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Mikhaylov, Pimashkin, Pigareva, Gerasimova, Gryaznov, Shchanikov, Zuev, Talanov, Lavrov, Demin, Erokhin, Lobov, Mukhina, Kazantsev, Wu and Spagnolo. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 30 April 2020
doi: 10.3389/fnins.2020.00379

Bio-Inspired Techniques in a Fully Digital Approach for Lifelong Learning

Stefano Bianchi†, *Irene Muñoz-Martin*† *and Daniele Ielmini**

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milan, Italy

Lifelong learning has deeply underpinned the resilience of biological organisms respect to a constantly changing environment. This flexibility has allowed the evolution of parallel-distributed systems able to merge past information with new stimulus for accurate and efficient brain-computation. Nowadays, there is a strong attempt to reproduce such intelligent systems in standard artificial neural networks (ANNs). However, despite some great results in specific tasks, ANNs still appear too rigid and static in real life respect to the biological systems. Thus, it is necessary to define a new neural paradigm capable of merging the lifelong resilience of biological organisms with the great accuracy of ANNs. Here, we present a digital implementation of a novel mixed supervised-unsupervised neural network capable of performing lifelong learning. The network uses a set of convolutional filters to extract features from the input images of the MNIST and the Fashion-MNIST training datasets. This information defines an original combination of responses of both trained classes and non-trained classes by transfer learning. The responses are then used in the subsequent unsupervised learning based on spike-timing dependent plasticity (STDP). This procedure allows the clustering of non-trained information thanks to bio-inspired algorithms such as neuronal redundancy and spike-frequency adaptation. We demonstrate the implementation of the neural network in a fully digital environment, such as the Xilinx Zynq-7000 System on Chip (SoC). We illustrate a user-friendly interface to test the network by choosing the number and the type of the non-trained classes, or drawing a custom pattern on a tablet. Finally, we propose a comparison of this work with networks based on memristive synaptic devices capable of continual learning, highlighting the main differences and capabilities respect to a fully digital approach.

Keywords: brain-inspired computing, supervised learning, unsupervised learning, spike-timing-dependent plasticity (STDP), neuronal redundancy, lifelong learning, continual learning, FPGA

OPEN ACCESS

Edited by:
Stefano Brivio,
Institute for Microelectronics and Microsystems (CNR), Italy

Reviewed by:
Jean-Michel Portal,
Aix-Marseille Université, France
Vyacheslav Demin,
Kurchatov Institute, Russia

***Correspondence:**
Daniele Ielmini
daniele.ielmini@polimi.it

†*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *13 January 2020*
Accepted: *30 March 2020*
Published: *30 April 2020*

Citation:
Bianchi S, Muñoz-Martin I and Ielmini D (2020) Bio-Inspired Techniques in a Fully Digital Approach for Lifelong Learning. Front. Neurosci. 14:379. doi: 10.3389/fnins.2020.00379

1. INTRODUCTION

In biology, systems consolidate and integrate information through neuropsychological processes that regulate synaptic and homeostatic plasticity (Friedemann Zenke and Ganguli, 2017; Power and Schlaggar, 2017). These mechanisms provide both plasticity for resilience and stability for protecting the previously learned information. Adaptation, retention and learning mechanisms have been recognized by the neuromorphic community as key tools for developing architectures capable of reproducing low-power, bio-inspired and robust intelligent computation.

Nowadays, Machine Learning (ML) empowers many aspects in our daily life, from virtual personal assistants to product recommendation and online fraud detection (Awoyemi et al., 2017). In particular, deep learning (DL) methods have dramatically enhanced classification and recognition capabilities of artificial networks by exploiting general-purpose learning algorithms with multiple processing layers (LeCun et al., 2015). Hardware architectures have been proposed for implementing deep multi-layer networks using CMOS technology and Field Programmable Gate Arrays (FPGAs) (Camuñas-Mesa et al., 2012; Gokhale et al., 2014; Indiveri and Liu, 2015). A further improvement of this trend is related to the non-Von Neumann hardware implementation of backpropagation algorithms using non-volatile memories (NVMs) such as phase-change-memory (PCM) and resistive switching memory (RRAM) (Burr et al., 2015; Merrikh-Bayat et al., 2017; Ambrogio et al., 2018).

Accurate DL generally relies on large stationary batches of training data for supervised algorithms, whereas autonomous agents should be capable of continually learning throughout their lifetime. In particular, lifelong learning refers to the capability of plastically accommodating new knowledge and stabilizing previous learnt information (Parisi et al., 2019). In computing processing systems, these two concepts define a trade-off which is studied as stability-plasticity dilemma (Martial Mermillod and Bonin, 2013; Ditzler et al., 2015). Neuromorphic functions based on DL algorithms lose the previously acquired information when the available data are incremental and not constant. This "catastrophic forgetting" is typical of artificial neural networks (ANNs) and can be prevented in biological systems by complex neurocognitive mechanisms (Cichon and Gan, 2015). Several solutions have been proposed for achieving continual learning in ANNs mainly by developing training methods able to overcome catastrophic forgetting, such as: (i) replacing old redundant information, useless for achieving better accuracy, with new one (Rebuffi et al., 2017), (ii) task-specific synaptic consolidation (Kirkpatrick et al., 2017) or (iii) allocating additional neural resources (Rusu et al., 2016).

However, all these attempts have only partially enabled continual learning mainly because they lack an intimate link with bio-inspired techniques. In fact, bio-inspired learning algorithms like spike-timing-dependent plasticity (STDP), neuronal redundancy and spike-frequency adaptation appear as key elements for achieving continual incremental learning in various neural networks (Takiyama and Okada, 2012; Chicca et al., 2014; Bianchi et al., 2019, 2020; Munoz-Martin et al., 2019).

In this paper, we demonstrate that the implementation of bio-inspired techniques in ANNs is a key element to achieve continual learning in a fully digital environment. In particular, we propose a new kind of supervised-unsupervised neural network that is able to merge the stability of backpropagation algorithm with the flexibility introduced by bio-inspired plasticity. We have implemented the whole network in a fully digital environment using the Xilinx Zynq-7000 system-on-chip (SoC). The blocks that configure the network have been designed into the programmable logic of the chip using the VHDL hardware descriptive language.

We propose an interactive setup including user-friendly peripherals for creating an interface with the external world. In this way, it is possible to select the dataset to be tested (e.g., MNIST or Fashion-MNIST) and challenge the network by drawing an original pattern on a touch screen. The evolution of the winner-take-all synapses over time, the real-time classification accuracy and the intermediate results at every layer of the network are monitored in real time on an LCD controlled by the FPGA. We show accurate inference by the network that is able to correctly classify up to 5 non-trained classes of the MNIST and Fashion MNIST datasets, only relying on the transfer learning of the trained information. Finally, we propose a comparison on efficiency, area and energy consumption of the network using non-volatile memories.

This work highlights the relevance of plausible implementations of neural functions inside standard neural networks and demonstrates the relevance of bio-inspired techniques for achieving lifelong learning in artificial intelligence systems.

2. ENABLE CONTINUAL LEARNING IN ARTIFICIAL NEURAL NETWORKS

Catastrophic forgetting is a relevant problem in machine learning, for which the network cannot plastically manage new information while maintaining the ability of performing previous learnt tasks. This behavior is opposite to what, actually, is observed in the human brain. In biology, the theory of complementary learning systems introduces a framework to understand the mutual effort of hippocampus and neocortex to accept new information at the same time in which the previous knowledge is progressively consolidated (Kumaran et al., 2016; Kirkpatrick et al., 2017). In particular, the hippocampal system is responsible for a continuous adaptation to new incoming information whereas the task of the neocortex is essentially specialized in consolidating previous knowledge.

In our supervised-unsupervised neural network, we essentially merge two approaches, i.e., (i) the accurate supervised learning of the convolutional neural network (CNN), and (ii) the plasticity provided by the STDP. The supervised part accounts for the neocortex, while the unsupervised part accounts for the hippocampal system. The bio-inspired neural redundancy and the spike frequency adaptation of the post-neurons (POSTs) used for classification further optimize the continual learning capability of the system. The merging of supervised artificial algorithms and bio-inspired approaches enables the solution of the so called "stability-plasticity" dilemma, which, so far, has prevented the achievement of lifelong learning in intelligent artificial systems (Martial Mermillod and Bonin, 2013).

The bio-inspired algorithms provide resilience to ANNs since they use previously stored knowledge to cluster non-trained input classes. In fact, the network dynamically evolves as a function of the evolving environment (i.e., increasing the number of non-trained classes), and enables plasticity in ANNs. This sort of transfer learning is different respect to what, actually, is performed in standard ANNs. Generally,

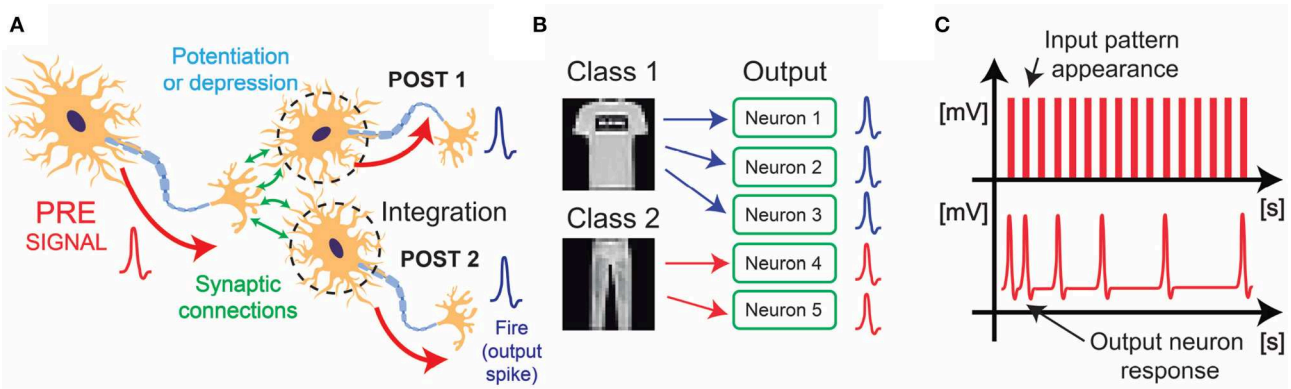


FIGURE 1 | (A) Schematic representation of biological STDP. Learning comes out by potentiation or depression of the synaptic connections, whose strength is defined by the temporal relationship between PRE-signal, and POST-signals (i.e., output spikes, "Fire," coming from the POST neurons, POST 1 and POST 2). **(B)** Schematic representation of the neural redundancy, where several output neurons specialize in just one input class for mutual aid. This mechanism has been studied in biology, and it is useful for increasing classification accuracy in WTA networks. **(C)** Schematic representation of spike-frequency adaptation in the POSTs, for optimizing the STDP algorithm and enable efficient specialization.

transfer learning refers to the use of previously acquired knowledge in one domain to solve a problem in a novel domain (Barnett and Ceci, 2002; Pan and Yang, 2010). Standard optimized approaches to transfer learning refer to the use of a large domain of data that share invariant relational information for further classification capabilities (Doumas et al., 2008), such as in the frameworks of zero and one shot learning (Palatucci et al., 2009; Vinyals et al., 2016). However, differently from standard approaches, transfer learning is here used to enable a continual and resilient evolution of the network by classifying new patterns during the unsupervised learning procedure. In this way, the network dynamically changes its hardware relying on bio-inspired algorithms like neuronal redundancy and spike frequency adaptation and provides on-line plasticity respect to the standard neural approach.

2.1. Neuronal Redundancy for STDP in Winner-Take-All Architecture

In winner-take-all (WTA) neural networks, groups of spiking neurons compete for improving the specialization capability using both inhibitory and excitatory synapses (Binas et al., 2014). These networks efficiently perform unsupervised learning of multiple patterns by exploiting bio-inspired algorithms such as STDP (**Figure 1A**) (Diehl and Cook, 2015; Ferré et al., 2018). STDP is a biological process in which synapses adjust their conductive strength as a function of the timing relationship between spikes coming from a PRE-neuron (PRE) and a POST-neuron (POST) (Markram et al., 1997; Abbott and Nelson, 2000). This behavior results into a long-term potentiation (LTP) or long-term depression (LTD) of the synaptic weights, enabling the plastic storage of useful correlated information in the synaptic connections (Abbott et al., 1997; Zucker and Regehr, 2002).

Due to their useful characteristics, WTA networks have been modeled by a computational point of view (Oster et al., 2009) and successively implemented in hardware CMOS (Chicca et al., 2014), with memristive devices (Ambrogio et al., 2016) and realizing digital designs in FPGAs (Ou et al., 2012).

We found an interesting improvement of the accuracy when a redundancy of neurons is provided to the WTA part of the network. Indeed, neuronal redundancy has been demonstrated to cover an important role in several biological aspects like the learning speed in the motor cortex (Takiyama and Okada, 2012). In our network, the system uses trained convolutional filters to find a certain shape within input images. For transfer learning, these features can be also recognized even in non-trained classes. In particular, a set of combinations of features can univocally define an input object even if the network has never been trained with the task of recognizing that type of image. Thus, for enabling continual learning, it is essential to prepare additional, or redundant, output neurons that can plastically adapt their synapses in a WTA framework in order to accept new input classes (**Figure 1B**).

2.2. Spike-Frequency-Adaptation for Optimizing STDP

In order to further optimize the classification system provided by the WTA part of the network, we have introduced the spike-frequency adaptation of the POSTs (**Figure 1C**). Spike-frequency adaptation is a bio-inspired technique (Connors et al., 1982; Stefan and Ernst, 2009), that provides stability to the unsupervised block of the neural architecture. In particular, when the synaptic window between the pattern and the background reaches a reference level, that specific POST increases its neuronal threshold, thus reducing its frequency activity in time. This is essential for power saving and specialization, because each POST tends to fire only when a specific pattern appears at the input.

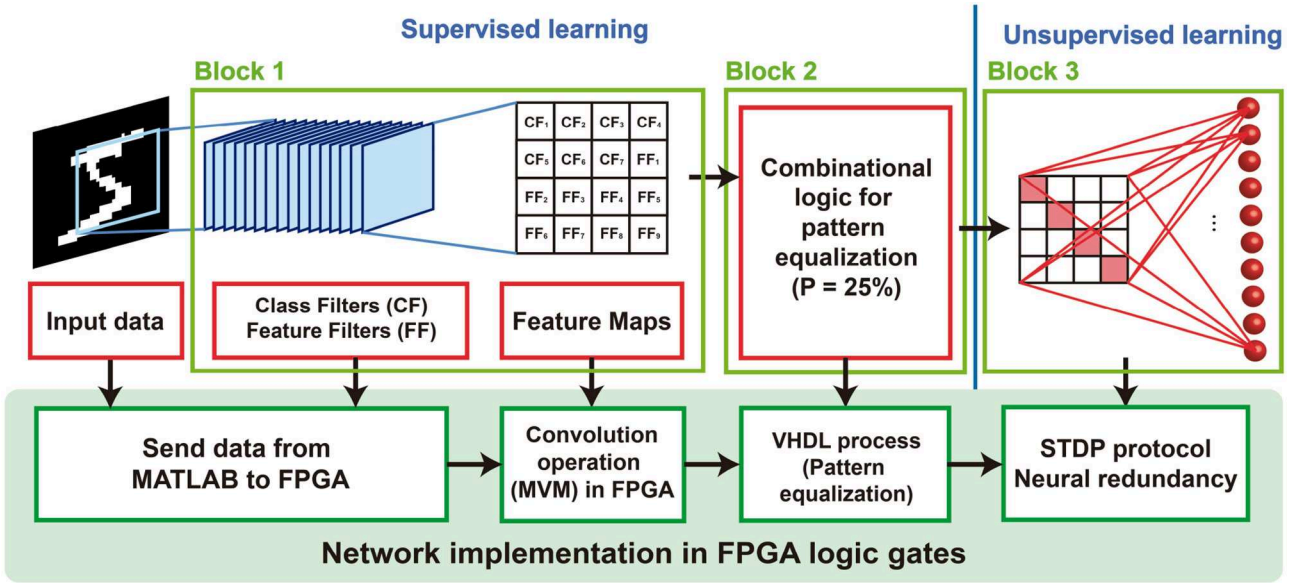


FIGURE 2 | Schematic representation of the hybrid supervised/unsupervised network for solving continual learning. Each one of the blocks have been implemented in an FPGA using logic gates.

3. THE HYBRID CONVOLUTIONAL-SPIKING NETWORK

The hybrid supervised-unsupervised neural network is shown in **Figure 2**. The network is divided into three main blocks that are described in the following sections.

3.1. Block 1: Convolutional Neural Network (CNN) for Recognition

The first part of the system is constituted by a set of custom convolutional filters that extract features from the input images. Two kinds of filters are used, namely the class filters (CFs) and the feature filters (FFs). Both the topologies have been trained using a fully convolutional approach (Long et al., 2015), for obtaining filters with dimensions 20 × 20. During inference, the convolution of the filters with the input 28 × 28 image creates a new matrix with dimension 9 × 9. After a max-pooling operation, the maximum value of the responses is selected. If this maximum is higher than the threshold, the response of the filter to that image is a digital "1," otherwise the response is taken as a digital "0."

Class filters and feature filters play different roles in the hybrid neural network and they are obtained by two different custom training algorithms.

1. **Class Filters** are designed to recognize only one specific class of the dataset. To determine these class-selected filters we used a fully convolutional approach as described in Munoz-Martin et al. (2019). Thanks to the training procedure, the system yields a positive response on the output neuron ("1" or "VDD") when only that specific class is detected.
2. **Feature Filters** have been extracted from the first layer of a custom fully convolutional neural network (FCNN), as described in Bianchi et al. (2019). The purpose of these filters is to extract generic features (angles, curves...) within the training dataset.

By keeping a constant total number of filters, e.g., 16, the number of FF varies as a function of the non-trained classes, namely those classes of patterns that are not presented during the preliminary training phase. For instance, if we train 7 classes over 10, we could accordingly train 7 CFs and 9 FFs. The splitting of the training procedure concerning the different subsets of filters is one of the key elements for performing lifelong learning. This is due to the following reasons:

1. Non-trained classes should not be confused with trained ones due to the high specialization of CFs that contain a very specific correlation of features related to only one specific class.
2. The dimension of the filters is higher with respect to a standard convolutional approach (Krizhevsky et al., 2012). This gives the possibility of visually mapping the feature in the filter.
3. The combination of digital responses ("feature map") after convolution defines a set of original clusters of the patterns belonging to a particular class.

In the digital design of the network presented in this paper, training is performed using MATLAB or Python codes. We have considered two datasets (MNIST and Fashion-MNIST) and all the possible combinations with up to 5 non-trained classes (637 different sets of filters). The implementation in the FPGA just performs testing operations. Input 28x28 images and filters (16

filters of 20x20) are sent from Matlab to the SoC by UART communication. The convolution has been implemented in the FPGA as a matrix-vector-multiplication (MVM). More technical details can be found in section 4.

3.2. Block 2: Combinational Logic for Pattern Equalization

The responses from the convolutional filters are binary (i.e., VDD or GND). Their combination configures a set of "feature maps" which is unique for each class of the dataset. These feature maps are classified in the third block.

Note that from class to class the pattern density P of the feature map, namely the number of responses equal to VDD with respect to the overall number of responses (Pedretti et al., 2017), can change. This results in an unfair competition between the feature maps presented to the WTA network, since the internal spiking threshold of every POST is initially fixed to a nominal value. To prevent spurious spiking activity due to the varying pattern density and unfair fire excitability of the POSTs, we assigned to each feature map an "equalized pattern" according to the combinational logic circuit of **Figure 3**. In this combinational logic, every particular set of responses after convolution of the inputs with the filter selects a different equalized feature map. The patterns are previously stored in proper registers of the FPGA and consist of the complete group of 4×4 patterns with uniform pattern density $P = 25$ %. We stored only those patterns which have, at most, 2 pixels in common with the others. Note that FFs are ignored in the combinational logic if a CF has given a VDD response. If none of the CFs gives a VDD response, the logic takes the combination of the responses from the FFs for selecting another equalized pattern.

3.3. Block 3: Winner-Take-All Network for Plastic Adaptation

The third block is formed by a WTA network that performs STDP learning and classification of the equalized patterns originating from the combinational logic. As shown in **Figure 4A**, the feature maps of the trained classes 0, 1, 2, 4, 7, 8, and 9 always have a VDD response for a particular CF. On the other hand, **Figure 4B** shows that, on average, non-trained classes 3, 5, and 6 give a GND response to all the CFs, while they are characterized by more than one combinations depending on the responses to the FFs. **Figure 4C** shows the three most probable combinations of responses from the average study of **Figure 4B**. Every combination of responses is associated by the combinational logic to an original equalized pattern that is classified on a further output neuron of the WTA network. The feature maps shown in **Figures 4A–C** have been extracted from the SoC operation. We have modeled digitally the STDP integration, the inhibition among neurons and the timing operation. In the digital model, synaptic weights are represented with a counter from 0 to 255.

To better clarify the role of redundancy ìn our network, **Figure 5A** shows the evolution of the pattern and background conductances for the non-trained class "5." Non-trained classes can generate different equalized patterns due to different combinations of responses from the FFs. However, the generated feature maps have different probabilities of appearance R_P, which could complicate the learning procedure (Pedretti et al., 2017). In fact, as shown in **Figure 5B**, the first pattern, which has a $R_P = 46\%$, achieves a good separation between pattern and background average conductance, while the second and the third ones (28% and 15%, respectively) show a smaller window. These three patterns are thus taken to represent the same non-trained class, i.e., "5" in this case.

4. DIGITAL IMPLEMENTATION OF THE NETWORK

This supervised-unsupervised neural network has been implemented in the Xilinx Zynq-7000 SoC, using both the Processor System (PS) and the Programmable Logic (PL). The PS consists of a dual core ARM Cortex-A9, while the PL part is configured by an FPGA Series 7. The SoC was mounted into the Zedboard, a low-cost development board (**Figure 6A**).

The digital implementation performs inference operations, allowing the assessment of the continual learning capabilities as a function of the dataset, the number and the type of non-trained classes. In particular, we have developed an interactive digital system where an external user can track the evolution of the system. For instance **Figure 6B** shows the learning procedure of the unsupervised layer of the network while **Figure 6C** shows the real-time evolution of the classification confusion matrix. Furthermore, it is possible to monitor the evolution of the intermediate layers, as in **Figure 6D** to study the results of the convolution between the input and the filters. In addition, an external touch screen is connected to the computer to allow the drawing of a custom digit (**Figure 6E**) and directly track its evolution.

4.1. Communication Setup of the SoC

Since continual learning is active during inference, we initially performed the training on a classical Von Neumann machine using Python or Matlab environments. Thus, in order to execute the inference operations, the SoC needs to receive the convolutional filters and the input images to test. We also create a communication line between the computer and the SoC using the UART-USB bridge. UART peripherals are connected to the PS part of the SoC through an AXI (Advanced eXtensible Interface) bus. Data are received or sent asynchronously. The PS part is programmed by a C++ code using the application program interface, API.

The input data from the datasets are grayscale images of 28×28 pixels. The convolutional filters use analog weights, which can be both positive and negative. To provide a digital implementation in the programmable logic of the SoC, we have transformed the grayscale input images and the weights of the convolutional filters into 8-bits integers.

Firstly, we send the data related to the convolutional filters and the equalized patterns, which are stored in the PL part of the SoC. The equalized patterns are defined by software simulations. A pattern density of 25% with, at maximum, 2 pixels in common, improves the multi-pattern learning in WTA networks based on

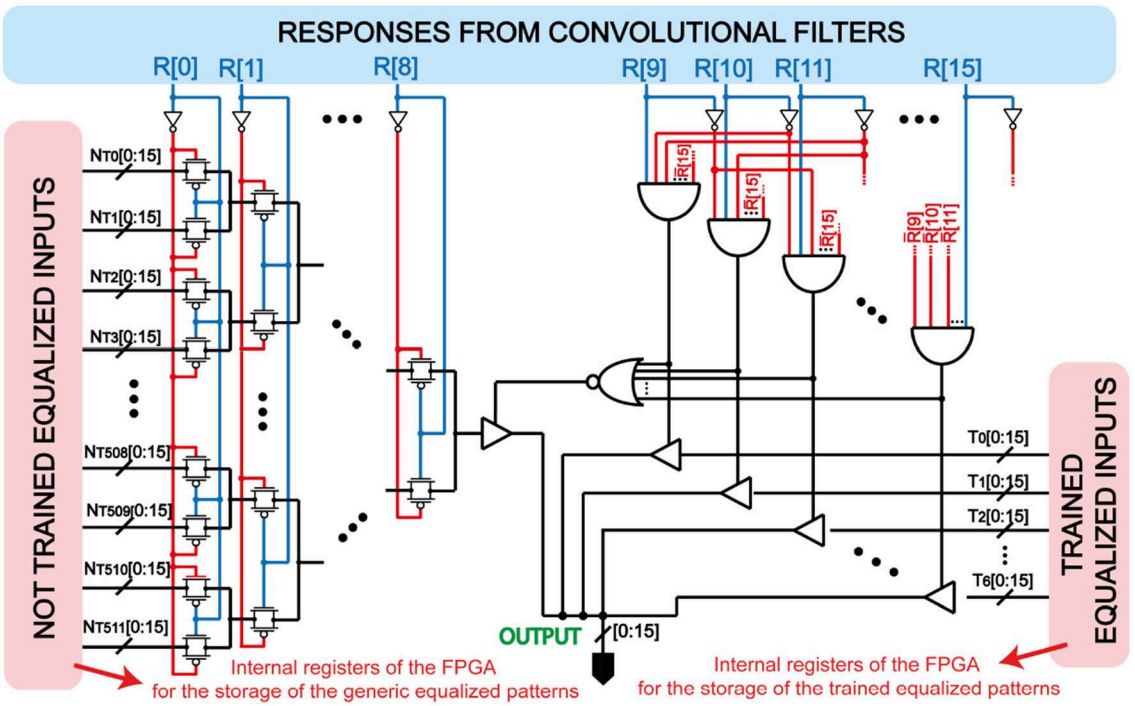


FIGURE 3 | Implemented combinational logic for equalizing the responses coming from the convolution between the input image and the filters. Every bus line has been associated with a certain equalized pattern. Note that the class filters have higher priority respect to the feature filters, which are selected only if all the class filters give a GND response. In this way, the trained classes are univocally selected by their own class filter.

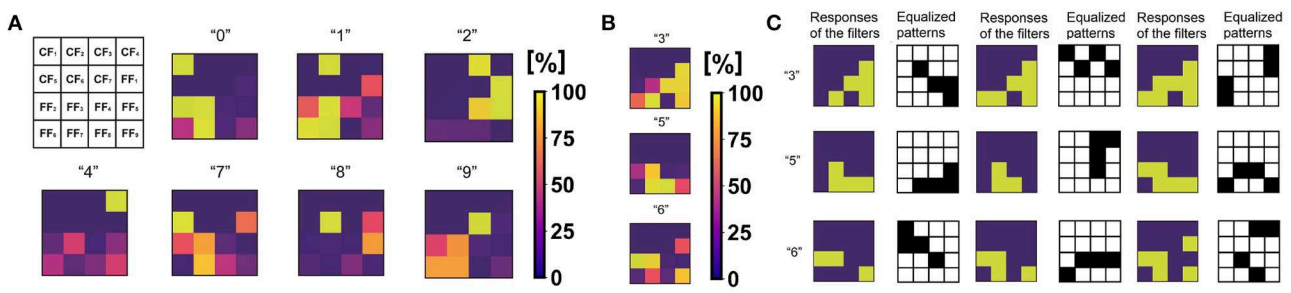


FIGURE 4 | (A) Average feature maps for the trained classes 0, 1, 2, 4, 7, 8, and 9. **(B)** Average feature maps for the non-trained classes 3, 5, and 6. **(C)** Extraction for the study of the three most probable feature maps for each one of the non-trained classes. Each one of these feature maps has a different equalized pattern. The neural redundancy allows to assign an equalized pattern to each feature map in order to improve the classification accuracy of the WTA network.

STDP (Pedretti et al., 2017). Once this data has been correctly sent and stored, the MNIST images are transferred one by one.

One inference cycle includes all the operations needed for classifying just one input pattern of MNIST or Fashion-MNIST datasets. As the master clock has a frequency of 50 MHz, and the UART baud-rate for sending and receiving data is equal to 230400 bps, the communication operation is the slowest one, as it takes 40 ms to send one pattern and its corresponding label (**Figure 7**). During this period, the system must perform sequentially the operations included in blocks (1) and (2) following a pipelined approach. The digital implementation of the STDP (4) takes 20 ms to be performed, as it follows the biological STDP timing. Thus, the calculation of the features map (1) and their equalization (2) must finish in less than 20 ms. Since the convolutions referred to each one of the filters are performed in parallel, the total required time is much smaller than 20 ms, so this timing limitation is not a constraint.

4.2. Real-Time Tracking of the Digital Synaptic Evolution

A graphical interface has been implemented for observing the real-time plastic adaptation of the system when performing continual learning of non-trained classes. The display is connected to the Zedboard by a VGA connection with a refresh

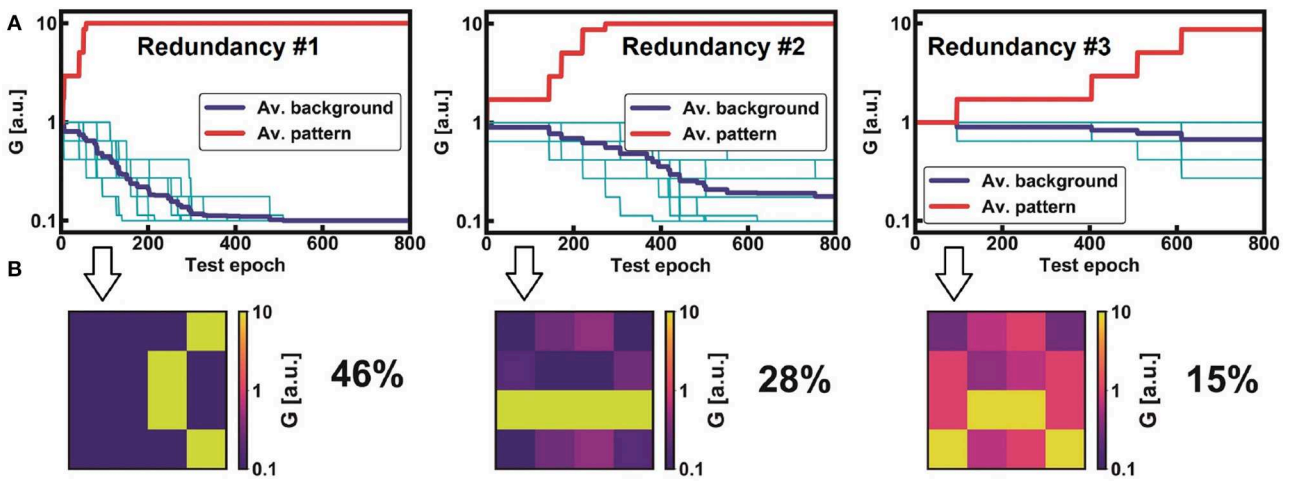


FIGURE 5 | Digital STDP learning activities of three equalized patterns for the non-trained class "5" **(A)**. The evolution of pattern and background average conductances depends on the appearance rate R_P. Note that R_P affects the opening of the average synaptic window between pattern and background, as evident by the corresponding synaptic images: the more a pattern appears at the input, the wider the window **(B)**. Note that the three learning activities are stable and do not present any errors. This efficiency has been achieved thanks to spike frequency adaptation, which optimizes the specialization of each post-neuron.



FIGURE 6 | (A) The low-cost Zedboard used for the implementation of the system. The Zedboard is connected to Matlab and to the screen. **(B)** Visual representation of the digital STDP learning procedure, **(C)** the confusion matrix for classification, and the **(D)** responses after convolving the input with the filters. **(E)** An example of a digit drawn by a user on the tabet connected to the system.

frequency of 60 Hz. By selecting the proper switches on the board, it is possible to study the different layers of the network, i.e., (i) the evolution of the digital synapses of STDP, (ii) the average feature maps and (iii) the real-time changes of the confusion matrix used for tracking the classification accuracy.

4.3. Computation of the Feature Maps and Equalization Logic

To enable the high recognition accuracy of CNNs, the input pattern must be convolved with the filters. Data is transferred from the PS to the PL through the AXI connection, while convolution is performed as a sequence of operations. The input image is divided into different subsets of matrices, all with dimension 20x20 (**Figure 8A**). The number of split buses follows the equation $(P_{SIDE} - F_{SIDE} + 1)^2$, where P_{SIDE} (pattern side) is equal to 28 and F_{SIDE} (filter side) is equal to 20. For each subset of matrices, the pattern is multiplied by the filter and then summed up, in order to get a 9×9 output matrix whose maximum value is selected by the system (max pooling operation). If the maximum is higher than the threshold set during the training procedure, the response of the filter is a "1," otherwise a "0." All the matrices are managed in parallel by serial bus communication in VHDL. Several multiply and accumulate cores are developed inside the programmable logic in order to speed up the system. At the same time, the system provides a further output bus that enables the visual tracking of the evolution of the system on the LCD monitor.

Referring to **Figure 7**, O_{11} is a 16-bits bus that contains the responses of the convolutional filters, while O_{12} is a 12-bits bus with the RGB color (4 bits per channel) of each pixel of the monitor. Thus, O_{11} feeds block (2) and the AXI block that connects the PS with the PL (feature maps are transmitted to MATLAB through UART), while O_{12} directly feeds block (6). The color encodes the information related to the percentage of the responses for each pixel of the feature map.

Once the neural network has extracted the feature maps, the FPGA performs the equalization step. The process reads the 16-bits bus coming from block (1), one bit for every response of the convolution between the input and the filters. Firstly, the VHDL code scans the part of the bus related to the responses coming from the CFs. If none of these filters has given a '1', the logic reads the second part of the input bus. Thus, following

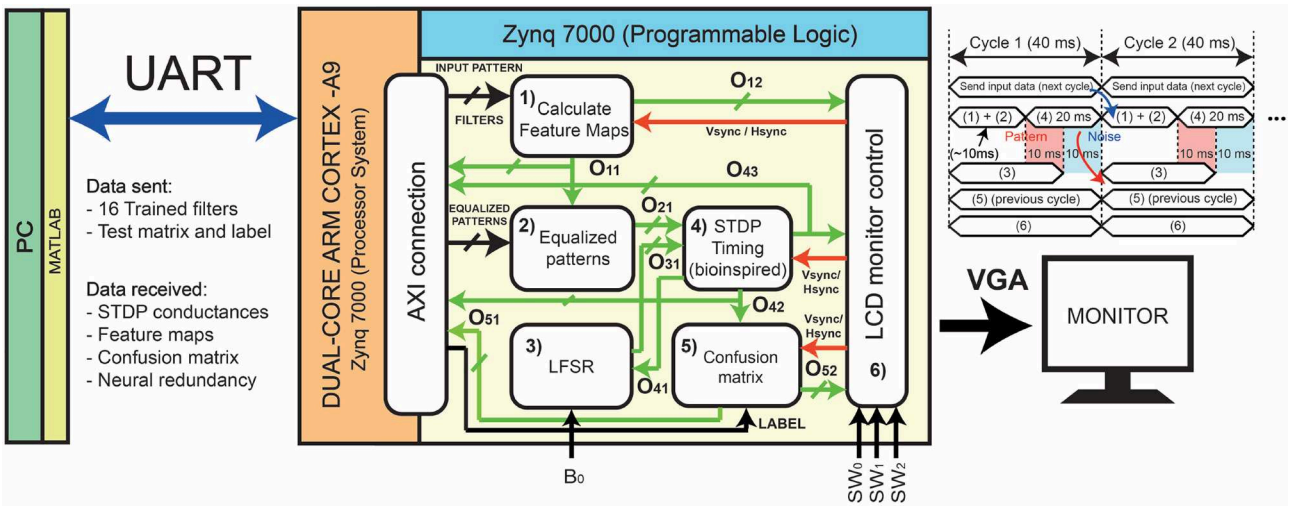


FIGURE 7 | Detailed schematic implementation of the system in the Xilinx Zedboard Dual-Core ARM Cortex (SoC). The filters are transmitted using the UART protocol at a baud rate of 230400 bps, while the operations for classification have been determined in the Programmable Logic part. The external user can decide which classes are trained and which not, and can select the channels of the monitor for observing the intermediate or the final results. Note that the management of the data transfer is pipelined with the convolution plus STDP computation.

the hierarchy set by CFs and FFs, this block assigns to the input feature map an equalized pattern from those stored in the internet FPGA register, which works as a Look-Up-Table (LUT). The equalization block has one main output, O_{21}, a bus of 16-bits containing the equalized 4×4 pattern that feeds block (4).

4.4. Linear Feedback Shift Register (LFSR)

In order to implement STDP algorithm, it is necessary to introduce stochasticity inside the SoC (Maass, 2014). Stochasticity in our STDP protocol is given by "noise" spikes, i.e., an uncorrelated spiking activity that is alternated with the presentation of pattern spikes at the input of the unsupervised layer (Pedretti et al., 2017).

Noise is an essential element for on-line learning in STDP, as it induces depression of background synapses. It also allows to remove or "forget" a previously learnt pattern when a new one is submitted, thus introducing plasticity in the network (Maass, 2014; Pedretti et al., 2018). It is important to set a correct noise density (the number of stochastic spikes) for improving the learning dynamics: a high noise density makes faster the background depression, but learning becomes unstable, as pattern and noise compete for synaptic potentiation.

We have set the noise density equal to 5%, thus 1 pixel turned on. Noise spikes are submitted at a noise rate probability (R_N) equal to the input rate probability (R_I) = 50%.

For generating pseudo-random numbers in the FPGA, we have developed a 4-bits linear feedback shift register (LFSR). In order to improve the uncertainty, the seed of the shift register is variable and it is generated as a function of the interaction time of the user with the board. Each time the user presses the bottom B_0, a counter with a pre-defined prescaler resets the seed of the LFSR. The output of this block (O_{31}) feeds block (4) (**Figure 7**).

4.5. STDP Timing

Block (4) of **Figure 7** performs the STDP calculations by managing the information of pattern and noise, i.e., buses O_{31} and O_{21}. Once the noise appears at the input, a signal is sent back to block (3) in order to update the noise bus for the next operation. In this way, the LFSR is accordingly incremented and generates different noise patterns.

The excitatory synapses are implemented with counters. These counters increase or decrease their values according to the STDP dynamics, 2.1. The duration of the signals of the PRE-neurons and the POST-neurons follows the bio-inspired evolution time of 10 ms. A more detailed representation of the evolution of the signals is shown in **Figure 8B**. The fire signal is sent back to block (5) and to the AXI block as a bus (O_{42}) of N_N bits, where N_N refers to the total number of neurons.

Additional VHDL processes perform the integration, the comparison and the fire operations for each of the output neurons (**Figure 8C**). The neuronal integration is implemented by adders (as many as the number of output neurons). If the value reached by one counter is higher than a certain threshold value (Pedretti et al., 2017), the associated neuron fires and the value of the synapses is accordingly modified. The firing activity not only causes the inhibition of the integrators (they are reset to zero), but it also causes the gradual increment of the neuronal threshold. The neuronal threshold is implemented with binary counters too, one for each output neuron. This operating methodology helps the STDP learning mechanism, as each neuron specializes in a particular pattern. Indeed, a fire event of a target neuron occurs only when a specific pattern arrives, thus avoiding learning errors (Pedretti et al., 2017; Muñoz-Martin et al., in press).

In order to have a visual representation of the evolution of the synapses, a further 12-bits bus that encodes the synaptic values as

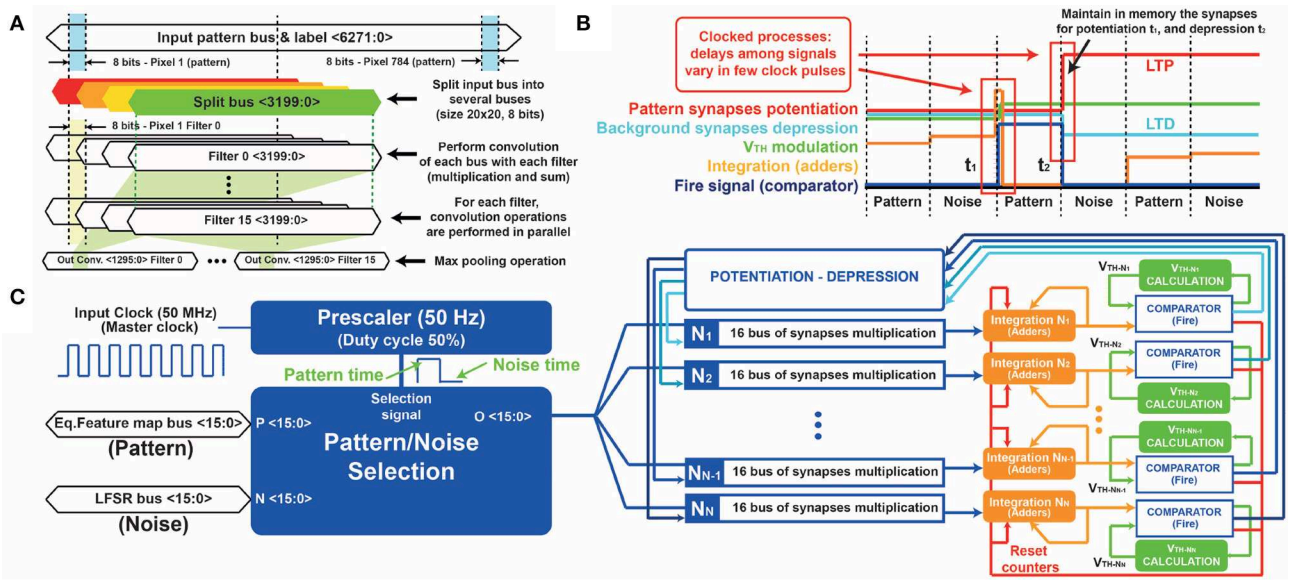


FIGURE 8 | (A) Detailed representation of the digital convolution operation. Input patterns are codified in buses of 6,272 bits (grayscale images of 28 × 28 pixels converted into 8 bits-integer binary data). Note that the input pattern is split into several subsets of buses, all with the same size of the convolutional filters. The elements of the split buses are multiplied by the elements of the convolutional filters one by one. The results are then summed up and stored into additional registers (16 buses of 81 elements each one). **(B)** Signal evolution during the STDP operation. Each time the pattern appears at the input, the adders increase their values, till the integration reaches the threshold. Then, the digital comparator maintains high its output for 10 ms, causing the potentiation/depression of the synapses at the falling edge. Since all the FPGA processes are synchronous, the delay between signals is much smaller (μs) than the time scale used during the STDP operation (tens of ms). **(C)** Detailed STDP-WTA scheme. Initially, the pattern/noise block alternates pattern (equalized feature maps) and noise (LFSR bus) at 50% rate of appearance, for a duration of 10 ms each. Each output neuron {1 ... N } performs the integration and the comparison respect to the internal threshold. If the comparator fires, the integrating adders are reset (inhibition) and the potentiation-depression block is activated to drive the excitatory synapses. In addition, the corresponding neuronal threshold is increased in order to improve the specialization capability.

RGB information (signal O_{43}) is sent to block (6) for displaying the information on the LCD monitor.

4.6. Confusion Matrix

The real time computation of the confusion matrix is carried out by comparing the input label of the image with the spiking output neuron (signals O_{42} and "label" in **Figure 7**). Labels are sent as 4 bits integers.

Note that a specific VHDL process calculates the accuracy of classification. This process performs statistical operations related to the number of times a fire event occurred compared to the label of the input class. As an example, get into consideration the situation in which there are two non-trained classes, "1" and "2," and one output neuron. If the neuron fires for the first time, and the label of the input class is "1," the accuracy for that neuron is 100% for class "1" and 0% for class "2." Now, the neuron fires again, but this time the label is "2," and the accuracy of that neuron is 50% for class "1" and 50% for class "2." If the neuron fires for a third time, with the input label equal to "2," now the accuracy changes again, being 33% for class "1" and 66% for class "2." After this test phase, which lasts 100 fire activities, we link a neuron to a non-trained class (the one that has given the maximum accuracy).

A second process of block (5) manages two output buses: the first bus contains the accuracies of the results and it feeds the AXI block for data transmission to the PC (O_{51}), for debugging purposes. The other bus is related to the RGB color of each pixel of the screen (O_{52}), and it is connected to block (6).

Figure 9 shows that the management of the information enables the track of the neuronal redundancy for investigating the classification accuracy of the non-trained classes (in this case, 3, 5 and 6). Note that the neuronal redundancy is essential to achieve higher classification accuracy. However, some confusion between the further output neurons avoids a completely correct clustering of the non-trained information.

4.7. Control of the LCD Monitor

The LCD monitor block of **Figure 7** manages the VGA using the synchronizing signals (*Hsync* and *Vsync*), and the RGB 12-bits bus for each pixel of the screen. The programmable logic selects the correct output RGB bus (O_{12}, O_{43}, or O_{52}) after reading the position of the switches (SW_0, SW_1, and SW_2) for showing the output screen requested by the user. As an example, if the user decides to check the evolution of the digital synapses of the non-trained classes, this block selects the RGB bus coming from block (1).

5. RESULTS

The fully digital approach of hybrid supervised-unsupervised network has been tested for continual learning of up to 50% non-trained classes. We have then compared the digital approach

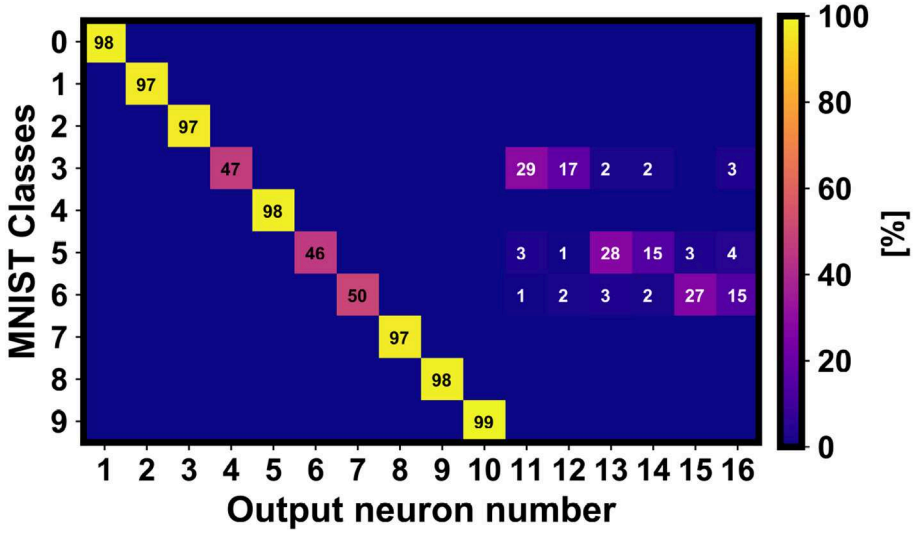


FIGURE 9 | Confusion matrix related to the classification results of the non-trained classes 3, 5, and 6 including the neural redundancy of 3 output neurons per non-trained class. As observed, the additional output neurons improve significantly the global accuracy.

with a memristive-based network with PCM synapses, in terms of area, energy consumption and testing efficiency (Munoz-Martin et al., 2019).

5.1. Continual Learning Results

Figure 10 shows the classification results of the continual learning accuracy for every combination of two non-trained classes of the MNIST (a) and the Fashion-MNIST (c) datasets. Concerning the MNIST, the classification accuracy of the non-trained classes varies from 69 to 95%. Note that this value is dependent on the similarities between the two non-trained digits. For instance, non-trained class 4 has a classification accuracy higher than 90% when it appears as a non-trained digit together with any other digit, except when the other non-trained class is number 9. In fact, numbers 4 and 9 have, on average, common shapes that could respond to the same feature filter FF (**Figure 10B**).

Note that, in **Figure 10C**, classes from 0 to 9 accordingly refer to clothes: t-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Due to the more complexity of the Fashion-MNIST dataset in terms of number and type of shapes, the accuracy of every combination of two non-trained classes is lower respect to the MNIST case.

Other average statistical results are shown in **Figure 11**, both for the MNIST (a) and the Fashion-MNIST (b) datasets. Note that the accuracy of the non-trained classes is strictly dependent on the number and type of the non-trained digits. For instance, it is interesting to observe that number 9, when it is not trained together with other four non-trained classes, shows a very low classification accuracy (from 21% to 50%) while, in the same conditions, number 6 can be classified with an accuracy from 48% to 73%. The degradation of the accuracy is mainly dependent on the confusion among the non-trained classes and the lack of efficient features-extraction from the reduced trained part of the dataset. The global classification results for the non-trained classes are summarized in **Figure 12** for both the MNIST (a) and the Fashion-MNIST (b). Note that the spread of the distribution strongly increases when more than 40% of non-trained classes are taken into consideration.

5.2. Discussion and Comparison With Memristive-Based Approaches

The results about continual learning of section 5.1 demonstrate that the network is able to re-use previously learnt information to develop further knowledge during inference. However, the FPGA-based fully digital approach is not the only feasible way to perform continual learning. In particular, other works have described the possibility of implementing a hybrid supervised-unsupervised neural network using a PCM-based approach (Bianchi et al., 2019; Munoz-Martin et al., 2019). PCM devices are among the best candidates for building efficient synaptic elements, especially for their 3D stacking integration and multilevel programming capability (Kuzum et al., 2013). **Figure 13A** shows a comparison between the fully digital approach and the memristive-based design of the network for the MNIST dataset. Note that the FPGA-based approach is more accurate with respect to the memristive one, in terms of accuracy of both trained and non-trained classes. This is mainly due to the fact the multilevel capability of the devices is not as good as the digital values implemented in the FPGA, that can codify the synaptic weights with a big number of bits for better precision. On the other hand, the area and power requirements of the digital design are worse with respect to the PCM-based approach, as evident from the table reported in **Figure 13B**. This is strongly related to the efficiency of PCM devices that can be operated in a parallel matrix-vector-multiplication architecture during convolution, for improved timing and energy efficiency. The power estimation of the FPGA has been extrapolated by

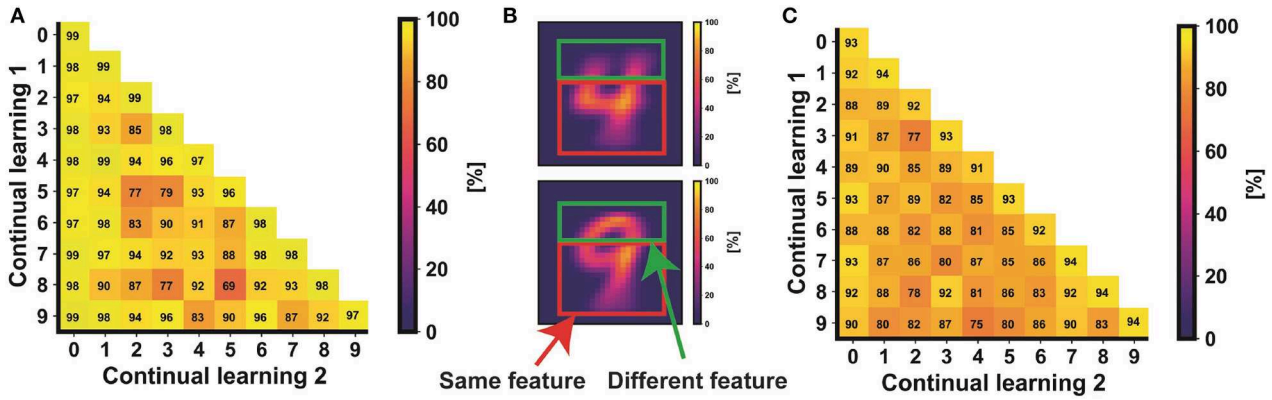


FIGURE 10 | Classification accuracy of continual learning for all the combinations of two non-trained classes of the MNIST **(A)** and the Fashion-MNIST **(C)** datasets. Note that the diagonal represents the case of only one non-trained class. Panel **(B)** shows the average shapes of classes 4 and 9 of the MNIST dataset. Since the shapes have common features, it is important to provide the system with a generic set of filters able to differentiate the objects joining the two classes.

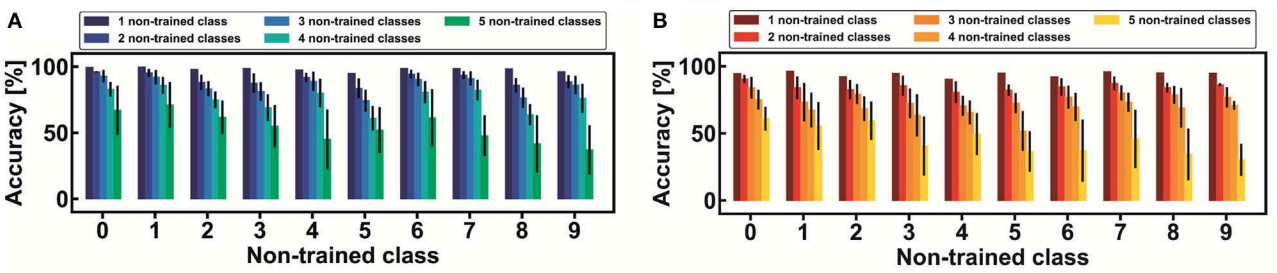


FIGURE 11 | Bar charts of continual learning classification accuracy for each one of the classes of the MNIST **(A)** and the Fashion-MNIST **(B)** datasets. Note that, on average, the accuracy of a reference non-trained class varies depending on the number of the further non-trained classes. In particular, as the number of non-trained classes increases, the continual learning accuracy degrades: this is mainly due to the confusion between new classes and to the worse efficiency in achieving good transfer learning from the reduced training sub-dataset. Note that every bar is provided with the 3 σ standard deviation.

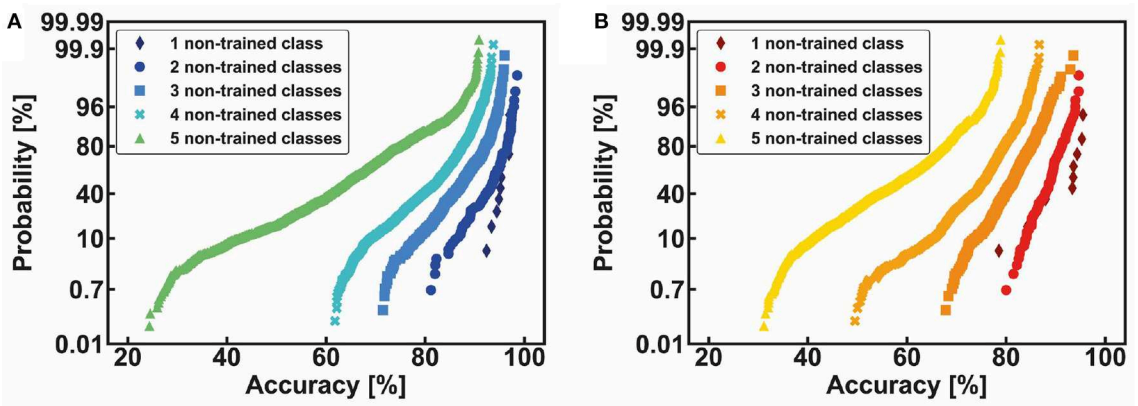
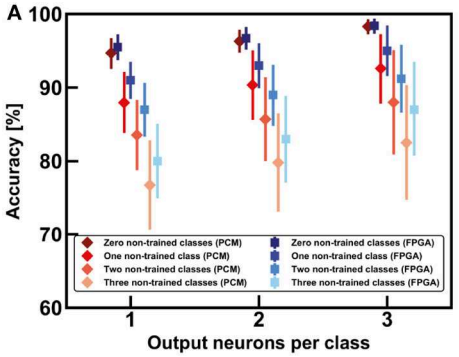


FIGURE 12 | Cumulative distributions of classification accuracy as a function of the increasing number of non-trained classes, from 1 to 5, for the MNIST **(A)** and the Fashion-MNIST datasets **(B)**.

using the internal software of Xilinx, Vivado. On the other hand, the power analysis of a hardware realization based on PCM synapses has been studied referring to the PCM devices described in Bianchi et al. (2019) and to a peripheral circuitry designed using a 90 nm node technology. The power required by convolution has been estimated in simulation taken into consideration the power for reading the PCM devices, the number of total steps required for performing the convolution of



B

	Memristive devices (PCM)	Digital implementation (FPGA)
Area estimation	5 mm^2	90 mm^2
Power consumption	170 mW	1.85W
MNIST global accuracy	98 %	99 %
Additional notes	Trial and error to adjust multilevel capability	FPGA programmable logic

FIGURE 13 | (A) Global accuracies for 0, 1, 2, 3 non-trained classes of MNIST dataset for both an FPGA-based design and a PCM-based approach. **(B)** Table of comparison for the two approaches in terms of area estimation, power consumption and global accuracy after training all the dataset.

each filter and a peripheral circuitry for the management of the results (operational amplifiers and decoders).

Note that the simulations claim that the possibility of parallel matrix-vector-multiplication with memristive devices accelerates the overall computation of the neural network and ease the peripheral circuitry for data management. However, if a higher accuracy is required, an increased number of levels of the weights in the convolutional filters is necessary. If this necessity is easily obtained in the FPGA by increasing the number of bits, in a fully analog approach a more precise multilevel capability of the PCM synapse depends on both the structure of the device and on the programming precision.

5.3. Extension to Other Datasets

In order to provide a good behavior for larger datasets (e.g., CIFAR-10), it is necessary to increment the number of convolutional filters (i.e., increasing the training complexity), and provide more output neurons per non-trained class (e.g., a neural redundancy of 5 output neurons instead of 3). The main problem associated to the scalability of our network is the exponential growing of resources required by the FPGA, both in terms of area and power consumption. In particular we simulated in software that, in order to obtain a full testing capability for CIFAR10 at 91.5%, the required computational power would double respect to what needed for MNIST. The area consumption would increase accordingly (we simulated an increment of 60%). In order to reduce these losses, it would be possible to optimize the training procedure, by means, for instance, the use of a validation set. Furthermore, it would be possible to seek for a sub-selection of filters which could enable an acceptable classification accuracy. However, this would require a much more complex training procedure and would not assure high classification standards.

6. CONCLUSIONS

In this paper, we proposed a new kind of hybrid supervised-unsupervised neural network capable of continually learn new concepts without forgetting the previous information. To prove the capability of the network for lifelong learning we used two datasets, (i) the MNIST and (ii) the Fashion-MNIST. The network mimics the functionality of the human brain. In particular, a section of the network stabilizes the learnt information, as it happens in the neocortex, while another part provides plasticity for accepting new information, as the hippocampus. The first section of the network is constituted by a set of convolutional filters which are specialized on the recognition of a particular trained class or on the extraction of generic features from the training dataset. Then, during inference, the responses of the convolutional filters form a pattern of responses, that is on-line learnt exploiting the benefits of unsupervised spike-trimming-dependent plasticity, STDP. We showed that the learnt pattern is original for both trained classes and new classes, i.e., classes that were not used for training the convolutional filters. We demonstrated the continual learning capability of the network by building a fully digital system on a System-on-Chip, SoC. A user-friendly interface was implemented in order to challenge the network by choosing the number and type of non-trained classes of the datasets. The classification accuracy significantly improves when other bio-inspired techniques are introduced in the digital framework of the demonstration. In particular, the spike-frequency adaptation, achieved by controlling the firing threshold of every neuron, and the neuronal redundancy, boost the learning activity of the non-trained classes. We showed that the network can classify up to 30% of new classes with an accuracy around 80%. Furthermore, we provided a comparison between a fully digital approach and an analog one using non-volatile synapses such as Phase-Change-Memories. This work highlights the possibility of achieving continual learning in neural networks using bio-inspired algorithms capable of merging the need of both stability and plasticity of an intelligent system. Thus, it paves the way for the creation of autonomous machines able to infer concepts and continually learn without catastrophically forgetting previously stored information.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

SB and IM-M have contributed equally in the planning and digital implementation of the system, the extraction and the interpretation of the results, the figures realization and the text writing. DI has supervised the conceptual planning and the design of this project.

FUNDING

This article has received fundings from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 648635) and from the Italian Minister for University and Research (grant agreement No. R164TYLBZP).

REFERENCES

Abbott, L., and Nelson, S. (2000). Synaptic plasticity: taming the beast. *Nat. Neurosci.* 3(Suppl.):1178–1183. doi: 10.1038/81453

Abbott, L. F., Varela, J. A., Sen, K., and Nelson, S. B. (1997). Synaptic depression and cortical gain control. *Science* 275, 221–224. doi: 10.1126/science.275.5297.221

Ambrogio, S., Balatti, S., Milo, V., Carboni, R., Wang, Z., Calderoni, A., et al. (2016). "Novel RRAM-enabled 1T1R synapse capable of low power STDP via burst-mode communication and real-time unsupervised machine learning," in *2016 IEEE Symposium on VLSI Technology* (Honolulu, HI), 1–2. doi: 10.1109/VLSIT.2016.7573432

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R., Boybat, I., Nolfo, C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558, 60–67. doi: 10.1038/s41586-018-0180-5

Awoyemi, J. O., Adetunmbi, A. O., and Oluwadare, S. A. (2017). "Credit card fraud detection using machine learning techniques: a comparative analysis," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, Lagos, Nigeria, 1–9. doi: 10.1109/ICCNI.2017.8123782

Barnett, S., and Ceci, S. (2002). When and where do we apply what we learn? A taxonomy for far transfer. *Psychol. Bull.* 128, 612–637. doi: 10.1037/0033-2909.128.4.612

Bianchi, S., Muñoz-Martin, I., Hashemkhani, S., Pedretti, G., and Ielmini, D. (2020). "A bio-inspired recurrent neural network with self-adaptive neurons and PCM synapses for solving reinforcement learning tasks" in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)* (Seville).

Bianchi, S., Munoz-Martin, I., Pedretti, G., Melnic, O., Ambrogio, S., and Ielmini, D. (2019). "Energy-efficient continual learning in hybrid supervised-unsupervised neural networks with PCM synapses," in *2019 Symposium on VLSI Technology* (Kyoto), T172–T173. doi: 10.23919/VLSIT.2019.8776559

Binas, J., Rutishauser, U., Indiveri, G., and Pfeiffer, M. (2014). Learning and stabilization of winner-take-all dynamics through interacting excitatory and inhibitory plasticity. *Front. Comput. Neurosci.* 8:68. doi: 10.3389/fncom.2014.00068

Burr, G. W., Shelby, R. M., Sidler, S., di Nolfo, C., Jang, J., Boybat, I., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electron Dev.* 62, 3498–3507. doi: 10.1109/TED.2015.2439635

Camuñas-Mesa, L., Zamarreno-Ramos, C., Linares-Barranco, A., Acosta-Jimenez, A. J., Serrano-Gotarredona, T., and Linares-Barranco, B. (2012). An event-driven multi-kernel convolution processor module for event-driven vision sensors. *IEEE J. Solid State Circ.* 47, 504–517. doi: 10.1109/JSSC.2011.2167409

Chicca, E., Stefanini, F., Bartolozzi, C., and Indiveri, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* 102, 1367–1388. doi: 10.1109/JPROC.2014.2313954

Cichon, J., and Gan, W.-B. (2015). Branch-specific dendritic ca2+ spikes cause persisten synaptic plasticity. *Nature* 520, 180–185. doi: 10.1038/nature14251

Connors, B. W., Gutnick, M. J., and Prince, D. A. (1982). Electrophysiological properties of neocortical neurons *in vitro*. *J. Neurophysiol.* 48, 1302–1320. doi: 10.1152/jn.1982.48.6.1302

Diehl, P., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. (2015). Learning in nonstationary environments: a survey. *IEEE Comput. Intell. Mag.* 10, 12–25. doi: 10.1109/MCI.2015.2471196

Doumas, L., Hummel, J., and Sandhofer, C. (2008). A theory of the discovery and predication of relational concepts. *Psychol. Rev.* 115, 1–43. doi: 10.1037/0033-295X.115.1.1

Ferré, P., Mamalet, F., and Thorpe, S. J. (2018). Unsupervised feature learning with winner-takes-all based STDP. *Front. Comput. Neurosci.* 12:24. doi: 10.3389/fncom.2018.00024

Friedemann Zenke, W. G., and Ganguli, S. (2017). The temporal paradox of Hebbian learning and homeostatic plasticity. *Curr. Opin. Neurobiol.* 43, 166–176. doi: 10.1016/j.conb.2017.03.015

Gokhale, V., Jin, J., Dundar, A., Martini, B., and Culurciello, E. (2014). "A 240 G-OPS/s mobile coprocessor for deep neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (Columbus, OH), 696–701. doi: 10.1109/CVPRW.2014.106

Indiveri, G., and Liu, S. (2015). Memory and information processing in neuromorphic systems. *Proc. IEEE* 103, 1379–1397. doi: 10.1109/JPROC.2015.2444094

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U.S.A.* 114, 3521–3526. doi: 10.1073/pnas.1611835114

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Neural Inform. Process. Syst.* 25. doi: 10.1145/3065386

Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends Cogn. Sci.* 20, 512–534. doi: 10.1016/j.tics.2016.05.004

Kuzum, D., Yu, S., and Wong, H.-S. P. (2013). Synaptic electronics: materials, devices and applications. *Nanotechnology* 24:382001. doi: 10.1088/0957-4484/24/38/382001

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA), 3431–3440. doi: 10.1109/CVPR.2015.7298965

Maass, W. (2014). Noise as a resource for computation and learning in networks of spiking neurons. *Proc. IEEE* 102, 860–880. doi: 10.1109/JPROC.2014.2310593

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APS and EPSPS. *Science* 275, 213–215. doi: 10.1126/science.275.5297.213

Martial Mermillod, A. B., and Bonin, P. (2013). The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Front. Psychol.* 4:504. doi: 10.3389/fpsyg.2013.00504

Merrikh-Bayat, F., Prezioso, M., Chakrabarti, B., Kataeva, I., and Strukov, D. (2017). Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. Commun.* 9:2331. doi: 10.1038/s41467-018-04482-4

Muñoz-Martin, I., Bianchi, S., Hashemkhani, S., Pedretti, G., and Ielmini, D. (in press). "Hardware implementation of PCM-based neurons with self-regulating threshold for homeostatic scaling in unsupervised learning," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)* (Seville).

Munoz-Martin, I., Bianchi, S., Pedretti, G., Melnic, O., Ambrogio, S., and Ielmini, D. (2019). Unsupervised learning to overcome catastrophic forgetting in neural networks. *IEEE J. Explorat. Solid State Comput. Dev. Circ.* 5, 58–66. doi: 10.1109/JXCDC.2019.2911135

Oster, M., Douglas, R., and Liu, S.-C. (2009). Computation with spikes in a winner-take-all network. *Neural Comput.* 21, 2437–2465. doi: 10.1162/neco.2009.07-08-829

Ou, C.-M., Li, H.-Y., and Hwang, W.-J. (2012). Efficient k-winner-take-all competitive learning hardware architecture for on-chip learning. *Sensors* 12, 11661–83. doi: 10.3390/s120911661

Palatucci, M., Pomerleau, D. A., Hinton, G. E., and Mitchell, T. (2009). "Zero-shot learning with semantic output codes," in *NIPS'09: Proceedings of the 22nd International Conference on Neural Information Processing Systems,* 1410–1418.

Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowledge Data Eng.* 22, 1345–1359. doi: 10.1109/TKDE.2009.191

Parisi, G., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Netw.* 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Pedretti, G., Bianchi, S., Milo, V., Calderoni, A., Ramaswamy, N., and Ielmini, D. (2017). "Modeling-based design of brain-inspired spiking neural networks with RRAM learning synapses," in *2017 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 28.1.1–28.1.4. doi: 10.1109/IEDM.2017.8268467

Pedretti, G., Milo, V., Ambrogio, S., Carboni, R., Bianchi, S., Calderoni, A., Ramaswamy, N., Spinelli, A. S., and Ielmini, D. (2018). Stochastic learning in neuromorphic hardware via spike timing dependent plasticity with rRAM synapses. *IEEE J. Emerg. Select. Top. Circ. Syst.* 8, 77–85. doi: 10.1109/JETCAS.2017.2773124

Power, J. D., and Schlaggar, B. L. (2017). Neural plasticity across the lifespan. Wiley Interdiscipl. *Rev. Dev. Biol.* 6:e216. doi: 10.1002/wdev.216

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. (2017). iCaRL: Incremental classifier and representation learning. *arXiv:1611.07725*. doi: 10.1109/CVPR.2017.587

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., et al. (2016). Progressive neural networks. *arXiv:1606.04671*.

Stefan, M., and Ernst, N. (2009). A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural Comput.* 21, 704–718. doi: 10.1162/neco.2008.12-07-680

Takiyama, K., and Okada, M. (2012). Maximization of learning speed in the motor cortex due to neuronal redundancy. *PLoS Comput. Biol.* 8:e1002348. doi: 10.1371/journal.pcbi.1002348

Vinyals, O., Blundell, C., Lillicrap, T., and Wierstra, D. (2016). "Matching networks for one shot learning," in *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems,* 3637–3645.

Zucker, R. S., and Regehr, W. G. (2002). Short-term synaptic plasticity. *Annu. Rev. Physiol.* 64, 355–405. doi: 10.1146/annurev.physiol.64.092501.114547

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Bianchi, Muñoz-Martin and Ielmini. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 30 April 2020
doi: 10.3389/fnins.2020.00309

Double MgO-Based Perpendicular Magnetic Tunnel Junction for Artificial Neuron

*Dong Won Kim[1], Woo Seok Yi[2], Jin Young Choi[3], Kei Ashiba[4], Jong Ung Baek[1], Han Sol Jun[1], Jae Joon Kim[2] and Jea Gun Park[1,4]**

[1] Department of Nanoscale Semiconductor Engineering, Hanyang University, Seoul, South Korea, [2] Department of Creative IT Engineering, Pohang University of Science and Technology, Pohang, South Korea, [3] MRAM Center, Department of Electronics and Computer Engineering, Hanyang University, Seoul, South Korea, [4] Wafer Engineering Department, SUMCO Corporation, Imari, Japan

A perpendicular spin transfer torque (p-STT)-based neuron was developed for a spiking neural network (SNN). It demonstrated the integration behavior of a typical neuron in an SNN; in particular, the integration behavior corresponding to magnetic resistance change gradually increased with the input spike number. This behavior occurred when the spin electron directions between double $Co_2Fe_6B_2$ free and pinned layers in the p-STT-based neuron were switched from parallel to antiparallel states. In addition, a neuron circuit for integrate-and-fire operation was proposed. Finally, pattern-recognition simulation was performed for a single-layer SNN.

Keywords: neuromorphic, MRAM, spiking neuron, spiking neural network, artificial neuron

OPEN ACCESS

Edited by:
Erika Covi,
Politecnico di Milano, Italy

Reviewed by:
Halid Mulaosmanovic,
NaMLab gGmbH, Germany
Farshad Moradi,
Aarhus University, Denmark

****Correspondence:***
Jea Gun Park
parkjgl@hanyang.ac.kr

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *03 December 2019*
Accepted: *16 March 2020*
Published: *30 April 2020*

Citation:
Kim DW, Yi WS, Choi JY, Ashiba K, Baek JU, Jun HS, Kim JJ and Park JG (2020) Double MgO-Based Perpendicular Magnetic Tunnel Junction for Artificial Neuron. Front. Neurosci. 14:309. doi: 10.3389/fnins.2020.00309

INTRODUCTION

Artificial neural network (ANN)-based artificial intelligence (AI) has been one of the most successful technologies in recent years. Today, it is applied in numerous fields, such as education, security, finance, science, and entertainment. In particular, the performance of the AI has already exceeded the ability of human beings (Szegedy et al., 2015; He et al., 2016; Silver et al., 2016; Hu J. et al., 2018) in fields such as image recognition and the Go game. However, there is a limitation to conventional ANNs working on the von-Neumann architecture. The low bandwidth between processor and memory in the von-Neumann architecture hinders efficient neural networks processing (Merolla et al., 2014; Monroe, 2014). Neuromorphic computing systems that mimic the human brain has been designed to overcome this limitation using complementary metal oxide semiconductor (CMOS)-based artificial neuron devices. However, it is a major challenge to implement high neuronal density by means of conventional CMOS technology because emulating the integration function of the neuron relies on the capacitor where the area of capacitor would be prohibitively large ($\sim$1,000 F^2) to obtain the desired capacitance ($\sim$10 fF/μm^2) (Gentet et al., 2000; Indiveri et al., 2013). Therefore, an artificial neuron device without a capacitor is necessary to implement high-density neuromorphic chip. Recently, emerging artificial neuron devices have been reported as an alternative to CMOS-based neuron devices such as partially depleted silicon-on-insulator n-MOSFET (PD-SOI n-MOSFET) (Dutta et al., 2017), phase change random-access memory (PCRAM) (Tuma et al., 2016), and magnetic

random-access memory (MRAM) (Grollier et al., 2016; Sengupta et al., 2016; Shim et al., 2017; Srinivasan et al., 2017; Torrejon et al., 2017; Mizrahi et al., 2018; Kurenkov et al., 2019). Among them, MRAM has been proposed as a promising candidate for artificial neuron device due to its high-area efficiency, fast operating speed, and low power consumption (Zhang et al., 2016; Liyanagedera et al., 2017; Hu G. et al., 2018). However, past researches have mainly focused on stochastic behavior of MRAM, and its integration behavior has not yet been reported. In this work, we first demonstrated the integration behavior of perpendicular spin transfer torque magnetic tunneling junction (p-STT MTJ) spin valve when switching from parallel to antiparallel states between $Co_2Fe_6B_2$ free and pinned layers. In addition, its integration behavior was discussed with grain boundary in MgO tunneling barrier. Finally, we conducted a pattern recognition simulation of a spiking neural network (SNN) using our p-STT-based neuron.

MATERIALS AND METHODS

Device Fabrication

p-STT MTJ

A p-STT MTJ spin valve structure was fabricated using a 12-in SiO_2 wafer multichamber cluster magnetron sputtering system under a high vacuum of $<1 \times 10^{-8}$ Torr. In particular, it was vertically stacked with a W/TiN bottom electrode, Ta buffer layer, Pt seed layer, [Co (0.47 nm)/Pt (0.23 nm)]$_6$/Co (0.51 nm) lower SyAF layer, Ru spacer layer (0.85 nm), Co (0.51 nm)/Pt (0.23 nm)/[Co (0.47 nm)/Pt (0.23 nm)]$_3$ upper SyAF layer, Co buffer layer (0.4 nm), W bridge layer (0.2 nm), $Co_2Fe_6B_2$ pinned layer (0.95 nm), MgO tunneling barrier (1.0 nm), Fe insertion layer (0.3 nm), $Co_2Fe_6B_2$ lower free layer (0.8 nm), W spacer layer (0.4 nm), $Co_2Fe_6B_2$ upper free layer (0.8 nm), MgO capping layer (0.8 nm)/Fe diffusion barrier (0.19 nm), W capping layer (4.0 nm), and Ta/Ru top electrode. An amorphous Ta buffer layer was used to prevent the texturing of the polycrystallinity of the W/TiN bottom electrode. A Pt seed layer thickness was optimized for the face-centered cubic (f.c.c) texturing of the [Co/Pt] SyAF multilayers. The [Co/Pt]$_6$ lower SyAF layer and [CoPt]$_3$ upper SyAF layer were perfectly antiferromagnetic coupled by inserting an optimized Ru spacer layer by Ruderman–Kittel–Kasuya–Yosida (RKKY) coupling. Then, the $Co_2Fe_6B_2$ pinned layer was ferrocoupled to the [CoPt]$_3$ upper SyAF layer by a W bridge layer. Then, the p-STT MTJ spin valve was *ex situ* annealed at 350°C for 30 min under a vacuum below 10^{-6} Torr and a perpendicular magnetic field of 3 T. The p-STT MTJ spin valve was cut into 1×1 cm^2 pieces and was patterned into p-STT MTJ with a device size of 1.6×1.6 μm^2 using ion milling and E-beam lithography. Then, p-STT MTJ was passivated, and their contact pads were wire bonded to a sample holder to estimate the electrical characteristics. The magnetic resistance versus applied magnetic field (*R*–*H*) curve and integration characteristic of the p-STT MTJ were measured with a homemade electrical probing system with a ∼1-T electromagnet using a Keithley 236 source measure unit and an Agilent B2902A semiconductor parameter analyzer.

IGZO-Based ReRAM

Five-nanometer-thick indium gallium zinc oxide (IGZO) film was deposited on a 113-nm diameter plug-type TiN-bottom-electrode-patterned wafer by radio frequency (RF) magnetron sputtering at 40 W RF power, 40 sccm Ar flowrate, and 1 sccm O_2 flowrate for an IGZO target, followed by 400°C annealing for 30 min in N_2 ambient. For a top electrode patterning, 850 μl photoresist (AZ5214E) was dropped on the IGZO thin film layer followed by spin coating with 5,000 rpm for 30 s and 120°C hard baking for 1 min and 40 s. Then, a photomask with 60×60 μm^2 pattern size was aligned on the substrate followed by exposure to UV light with a beam intensity of 20 mW/cm^2 for 12 s. The exposed photoresist was developed for 50 s using a developer (AZ300MIF) followed by deionized water rinse for 4 min. Afterward, the top Al electrode was deposited by direct current (DC) magnetron sputtering at 30 W DC power and 30 sccm Ar flowrate for an Al target. Finally, lift-off process was performed to make the top electrode pattern by acetone for 4 min followed by methanol rinse for 4 min and deionized water rinse for 4 min. Thus, the synapse devices have a sandwich device structure of a bottom TiN electrode, an IGZO layer, and a top Al electrode. Electrical characteristic was measured using a Keithley 4200A semiconductor parameter analyzer.

Pattern Recognition Simulation

Neuron

An empirical model was used to simulate the integration characteristic of the p-STT MTJ. The logistic function was used to fit a measured data (**Supplementary Figure 1A**). Thus, resistance of the p-STT MTJ is given as follows:

$$r(n) = \frac{r_{\min} - r_{\max}}{1 + \left(\frac{n}{n_v + n_\sigma}\right)^p} + r_{\max} + r_\sigma \quad (1)$$

where n, r_{min}, r_{max}, p, and n_v were the number of applied pulse, minimum and maximum resistance of the p-STT MTJ, fitting constant (=0.3142), and curve fitting parameter depending on the voltage, respectively. The integration characteristic of the p-STT MTJ is determined by n_v, which depends on the applied pulse amplitude (**Supplementary Figure 1B**). In this empirical model, n_σ and r_σ were added to account for device variation where $n_\sigma \sim N(\mu_n, {\sigma_n}^2)$ ($\mu_n = 0$ and $\sigma_n = 0.5$) and $r_\sigma \sim N(\mu_r, {\sigma_r}^2)$ ($\mu_r = 0$ and $\sigma_r = 0.2$) are Gaussian random variables (**Supplementary Figures 1C,D**).

Synapse

In this simulation, IGZO-based ReRAM is used as the artificial synapse, as shown in **Supplementary Figure 2A**. The IGZO-based ReRAM shows typical bistable current versus voltage (*I*–*V*) curve of interface-type ReRAM, as shown in **Supplementary Figure 2B**. To emulate synaptic property, we used a synapse model similar to Ziegler et al. (2015) and Hansen et al. (2017). In this model, change in synaptic weight is given by

$$\triangle w_{p,d}(t) = \beta_{p,d}(w)\, w(t) \left[1 - \frac{1}{w_{\max}} w(t)\right] \quad (2)$$

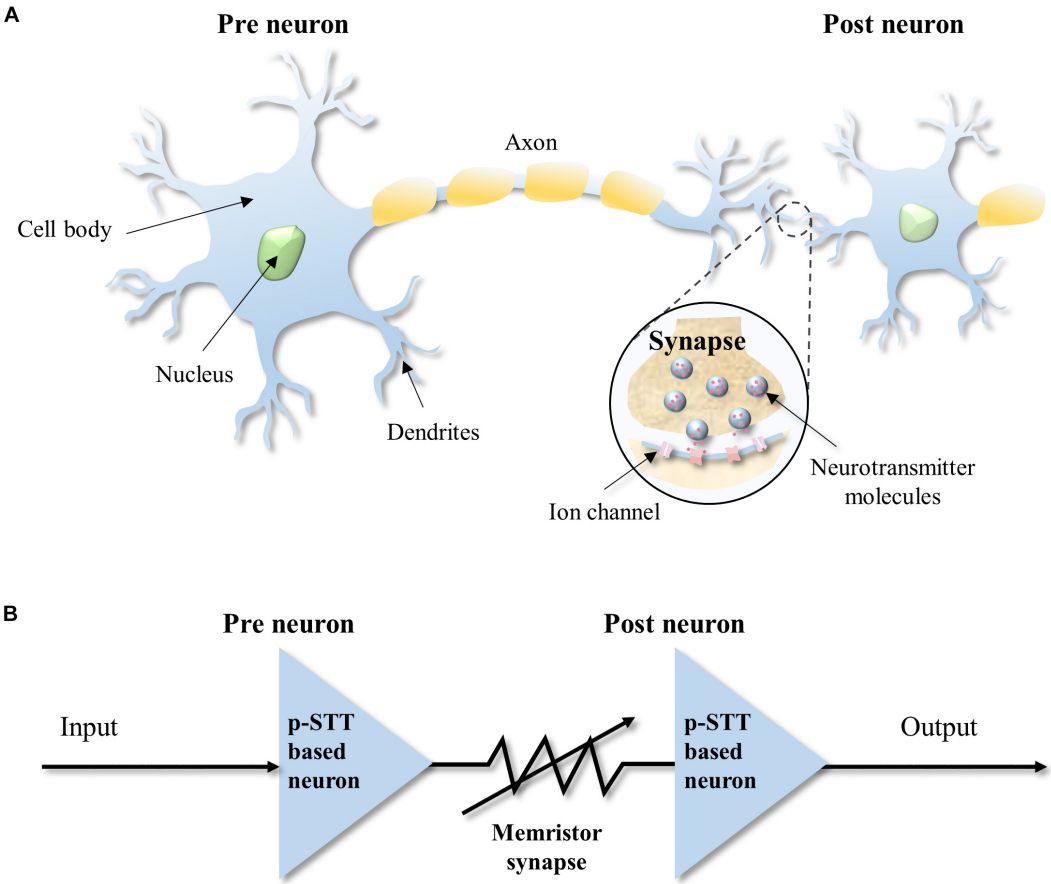


FIGURE 1 | Schematic of neural network. **(A)** Biological neural network. **(B)** Artificial neural network using the perpendicular spin transfer torque (p-STT)-based neurons and memristor synapse.

where w, β, and w_{max} represent the synaptic weight, the weight-dependent learning rate, and maximum synaptic weight, respectively. β determines the potentiation and depression curves depending on the switching mechanism of the ReRAM (Ziegler et al., 2015; Hansen et al., 2017). In order to obtain synaptic weight change, β should be determined. Here, we use a learning rate model given by

$$\beta_{p,d}(\triangle V, \triangle t, w) = \begin{cases} c_p(\triangle V, \triangle t) \times (1 - \gamma w), & potentiation : (V > 0) \\ -c_d(\triangle V, \triangle t) \times w, & depression : (V < 0) \end{cases} \quad (3)$$

where γ is a positive constant, and c_p and c_d are $\triangle V$ and $\triangle t$ dependent function. In our model, c_p (=0.275) and c_d (=0.063) are constant since $\triangle V$ and $\triangle t$ were fixed for the potentiation and depression. The simulation is well correlated with potentiation/depression of the experimental data, as shown in **Supplementary Figure 2C**.

Synaptic Weight Update

We used simplified spike timing-dependent plasticity (STDP) learning rule for training SNN. Synaptic weight was updated with the following equation:

$$w(t_{n+1}) = \begin{cases} w(t_n) + \triangle w_p(t_n), & 0 \le t_{post} - t_{pre} < 10T \\ w(t_n) + \triangle w_d(t_n), & otherwise \end{cases} \quad (4)$$

where $\triangle w_p$ and $\triangle w_{nd}$ are the synaptic weight change for the potentiation and depression, respectively. T is the time of a one cycle of integration–read–reset. Since we assumed a synchronous system, T is constant. Additional circuits are required for STDP operation. However, it is beyond the scope of this paper to deal with synaptic learning circuit in detail. When the spiking time difference between a preneuron (t_{pre}) and a postneuron (t_{post}) was <10 cycles (1 cycle = integration–read–reset), the synapses connected with the pre- and postneurons were potentiated, and the remaining synapses were depressed.

RESULTS

Artificial Neural Network Based on p-STT-Based Neuron

In biological neural networks, neurons are connected to other neighboring neurons via synapses, as shown in **Figure 1A**.

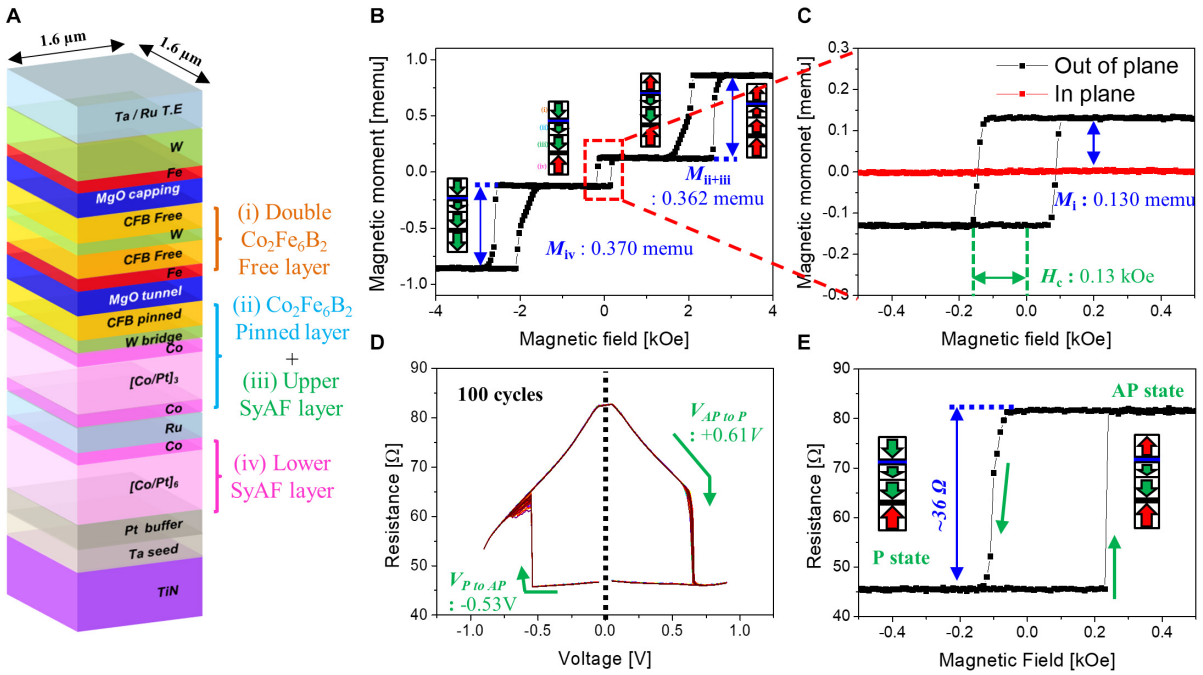


FIGURE 2 | Magnetic and electrical properties of the perpendicular spin transfer torque (p-STT)-based neuron (1.6× 1.6 μm^2). **(A)** Schematic structure. **(B)** *M–H* curve in a wide scanning range of the applied perpendicular magnetic field (i.e., −4 ∼ + 4 KOe). **(C)** *M–H* curve in a narrow scanning range of the applied perpendicular magnetic field (i.e., −0.5 ∼ + 0.5 KOe). **(D)** *R–V* curve. **(E)** *R–H* curve of the p-STT-based neuron.

Neurons integrate input spike signals from adjacent neurons via synapses, i.e., integrate. In addition, neurons generate output spike signals when membrane potentials reach a threshold value, i.e., fire. This neuronal behavior is called "integrate-and-fire," which is the key operation of neuron (Hodgkin and Huxley, 1952; Izhikevich, 2003). Similarly, artificial neurons could be connected with other artificial neurons via artificial synapses, where p-STT-based neurons are connected with memristor-type synapse, as shown in **Figure 1B**. The p-STT-based neurons receive spike signals through synapses connected with preneurons, integrate the signals, and then sends out output spike signals when the resistance of the p-STT-based neurons reaches a certain threshold value. In the following sections, we will describe in detail how p-STT-based neuron works.

Magnetic Properties of p-MTJ

Figure 2A shows schematic structure of p-STT MTJ. Its magnetic moment versus applied perpendicular magnetic field (*M–H*) loop was investigated to determine the static magnetic behavior of the p-STT MTJ, as shown in **Figures 2B,C**. It includes four perpendicular magnetic anisotropy (PMA) layers: a double $Co_2Fe_6B_2$ free layer (i in **Figure 2A**), $Co_2Fe_6B_2$ pinned layer (ii in **Figure 2A**), upper $[Co/Pt]_3$ SyAF layer (iii in **Figure 2A**), and lower $[Co/Pt]_6$ SyAF layer (iv in **Figure 2A**). Here, the $Co_2Fe_6B_2$ pinned layer was ferrocoupled with the upper SyAF layer, whereas the upper $[Co/Pt]_3$ SyAF layer was antiferro coupled with the lower $[Co/Pt]_6$ SyAF layer. The magnetic moments of the double $Co_2Fe_6B_2$ free layer, $Co_2Fe_6B_2$ pinned layer ferrocoupled with the upper $[Co/Pt]_3$ SyAF layer, and lower $[Co/Pt]_6$ SyAF layer were 0.130 (M_i in the inset of **Figure 2C**), 0.362 ($M_{ii\ +\ iii}$ in **Figure 2B**), and 0.370 (M_{iv} in **Figure 2B**) memu, respectively. In addition, the double $Co_2Fe_6B_2$ free layer showed an excellent interface PMA characteristic with a good squareness and fair coercivity (H_c, ∼0.13 kOe), as shown in **Figure 2C**. This result indicates that the MgO tunneling barrier had good face-centered cubic crystallinity that enhanced the coherent tunneling of the spin electrons (Lee et al., 2016a,c,d). The magnetic resistance versus voltage (*R–V*) behavior at room temperature (295 K) was measured to investigate the spin transfer torque switching behavior of the p-MTJ, as shown in **Figure 2D**. The switching voltage from parallel to antiparallel states was −0.53 *V* (V_{PtoAP}), while the switching voltage from antiparallel to parallel states was + 0.61 *V* (V_{APtoP}). The magnetic resistance versus magnetic field (*R–H*) loop of the p-STT MTJ is shown in **Figure 2E**. When the applied perpendicular magnetic field was scanned from + 0.5 to −0.5 kOe, the electron spin direction of the double $Co_2Fe_6B_2$ free layer was rotated from upward to downward so that the electron spin directions between the double $Co_2Fe_6B_2$ free and pinned layers were switched from antiparallel to parallel states. As a result, the resistance of the p-STT MTJ decreased from 82 to 46 Ω. The squareness and coercivity of the p-STT MTJ measured with an *R–H* loop was almost the same as that measured with an *M–H* loop, indicating that this device could maintain a stable magnetic state in a zero magnetic field so that the integration behavior would be characterized during the switch from parallel to antiparallel between the double $Co_2Fe_6B_2$ free and pinned layers.

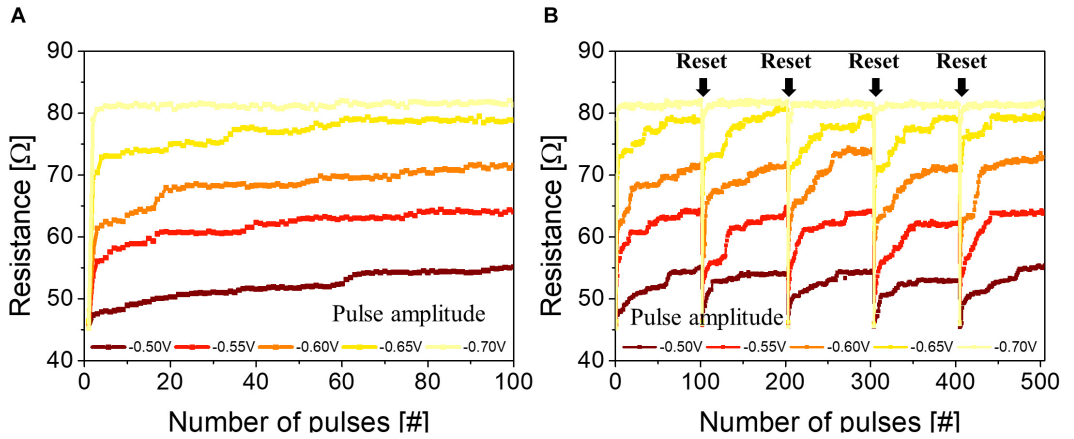


FIGURE 3 | Integration characteristics of the perpendicular spin transfer torque (p-STT) magnetic tunneling junction (MTJ)-based neuron. **(A)** Dependence of the integration behavior on the input spike number and amplitude. **(B)** Repeated integration characteristic of the p-STT MTJ (five sets of 100 input pulse spikes).

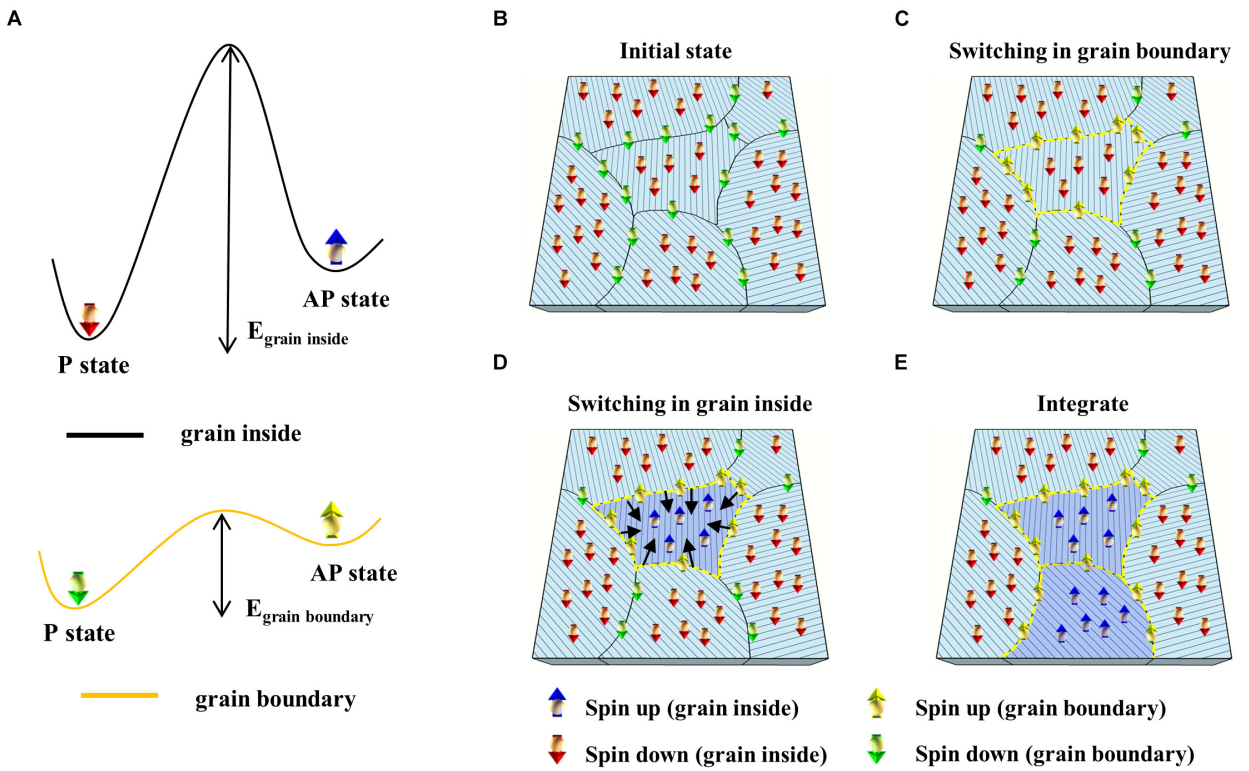


FIGURE 4 | Integration mechanism of the perpendicular spin transfer torque (p-STT) magnetic tunneling junction (MTJ). **(A)** Schematic of switching energy diagram at grain inside (black) and grain boundary (yellow). Schematic illustration of integration mechanism: **(B)** initial state, **(C)** switching at grain boundary, **(D)** switching at grain inside, and **(E)** integration.

Integration Property of p-MTJ Spin Valves

Interestingly, the p-STT MTJ showed integration property when consecutive voltage pulses (spike) were applied, as shown in **Figure 3A**. The spike width was 50 μs, and the spike amplitude was varied from −0.50 to −0.70 V. At all spike amplitudes, i.e., −0.50, −0.55, −0.60, −0.65, and −0.70 V, the p-STT MTJ performed the integration at input spikes of ∼100 pulses. In addition, the resistance difference increased when the input spike amplitude increased from −0.50 to −0.70 V at input spikes of ∼100 pulses, as shown in **Figure 3A**. Over an input spike amplitude of −0.7 V, no integration behavior was found. In addition, the p-STT MTJ showed a good repeatability for five sets of ∼100 input spike pulses, where the resistance

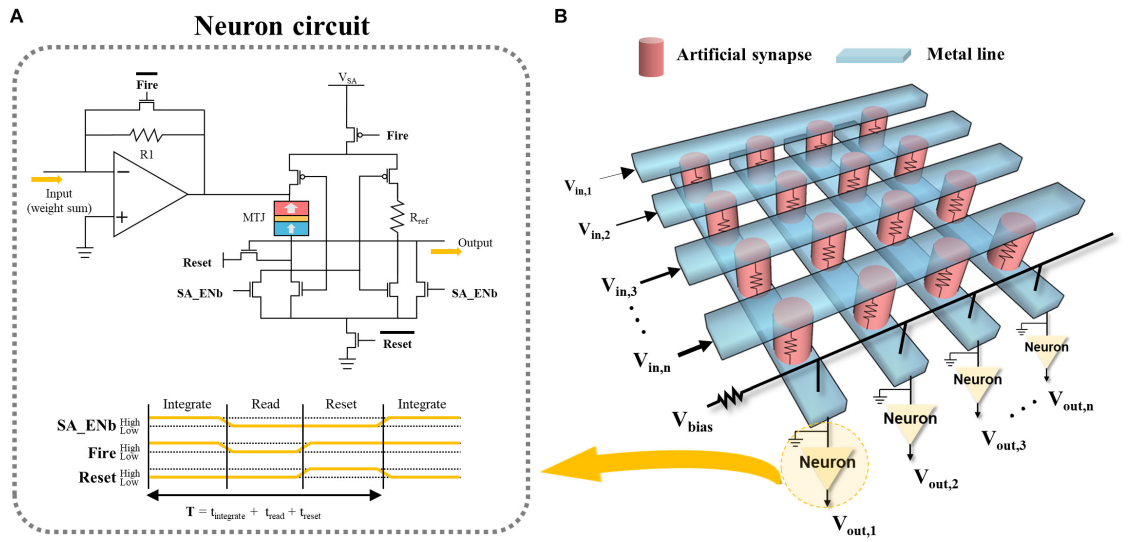


FIGURE 5 | Schematic of artificial neural network. **(A)** Crossbar array of artificial synapses and **(B)** neuron circuit for integrate-and-fire.

increment by the 100 input spike pulses increased with the input spike, as shown in **Figure 3B**. Our proposed p-STT MTJ in **Figure 3** showed a unique neuron characteristic (i.e., integration characteristic) compared to MTJ-based neurons (stochastic characteristic with a two-terminal device or leaky-integrate-and-fire characteristic with a three-terminal device), as shown in **Supplementary Table 1**. The mechanism of this behavior could be explained by understanding the grain-size distribution of the polycrystalline MgO tunneling barrier. The distribution of the sputtered polycrystalline MgO tunneling barrier was 0.6 to ∼1.8 nm, where the average grain size was ∼0.94 nm, as shown in **Supplementary Figure 3**. This indicates that even for a p-STT MTJ with a cell size of 35× 35 nm^2, multiple grains would exist within the p-STT MTJ cell, as shown in **Supplementary Figure 4**. As a result, we can expect that the p-STT MTJ with a cell size of 35× 35 nm^2 would show an integrate characteristic similar to **Figure 3** since it has a large number of grain within the p-STT MTJ cell. The interfacial PMA of both the double $Co_2Fe_6B_2$ free and pinned layers originated from the hybridization between O atoms and X (Fe or Co) atoms at the MgO tunneling barrier and $Co_2Fe_6B_2$ layer interface. Thus, the polygrain size distribution of the polycrystalline MgO tunneling barrier directly and strongly affects the ferromagnetic properties of both the double $Co_2Fe_6B_2$ free and pinned layers, i.e., resistance difference between parallel and antiparallel states of the p-MTJ. In addition, the hybridized Fe–O and Co–O bonds within the grains would be well oriented with the crystallinity of the MgO tunneling barrier, so the electron spins would require a high activation energy to switch from parallel to antiparallel. Otherwise, the spin electrons at the grain boundaries would have a relatively low energy barrier to switch from parallel to antiparallel, compared with the spin electrons within the grains (MacLaren and Willoughby, 2001; Victora et al., 2003; Kondo et al., 2018), as shown in **Figure 4A**. Thus, the spin electrons at the grain boundaries (**Figure 4B**) would first be switched from parallel to antiparallel states (**Figure 4C**), and the spin electrons inside the grain would then rotate due to the ferrocoupling between the spin electrons at the grain boundary and inside the grain (**Figure 4D**). As a result, the spin electrons in the grains would be switched from parallel to antiparallel, which would be a similar switching behavior to a previous report (Suzuki et al., 2016). This switching process would induce the integration behavior when the spikes are sequentially applied to p-STT MTJ (**Figure 4E**). The integration behavior of a p-STT MTJ was influenced by the crystallinity of the MgO tunneling barrier in **Figure 2A**, i.e., a better crystallinity of the MgO tunneling barrier led to a better integration characteristic, as shown in **Supplementary Figure 5**. This integration behavior of the p-STT MTJ would suggest that the p-STT MTJ could be applied with the complementary metal–oxide–semiconductor field-effect transistor (C-MOSFET) technology to produce artificial neuron. In general, the perpendicular spin torque switching time of a p-STT MTJ has been reported as ∼10 ns, which is the fastest switching time among other semiconductor devices (Hu G. et al., 2018). In addition, the operation of the integration by a p-STT MTJ in **Figure 3** was performed prior to a full the perpendicular spin torque switching. Thus, the width of a spike pulse in **Figure 3** could be less than ∼10 ns if the size of a neuron using a p-STT MTJ can be scaled down–up to 35 × 35 nm^2, suggesting a lowest power consumption per a spike in neuron (i.e., 1.6 × 1.6 μm^2), as shown in **Supplementary Table 2**.

p-STT MTJ-Based Integrate-and-Fire Neuron

Although the p-STT MTJ exhibited integration behavior depending on the input spike amplitude, it requires an additional circuit to perform the fire operation. Thus, the p-STT MTJ-based neuron circuit was designed using one p-STT MTJ, seven n-MOS FETs, three p-MOS-FETs, and one reference resistance to conduct

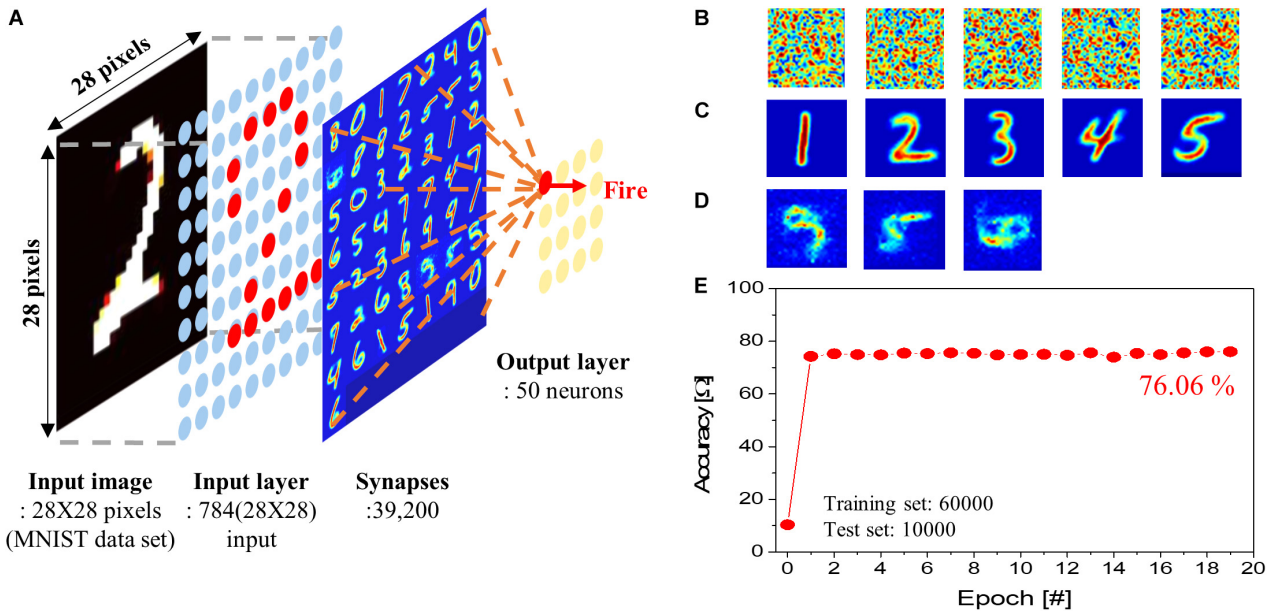


FIGURE 6 | Pattern recognition simulation. **(A)** Schematic of a single-layer spiking neural network (SNN). **(B)** Normalized synaptic weight before learning. **(C)** Normalized synaptic weight connected with active neurons after learning. **(D)** Normalized synaptic weight connected with silent neurons after learning. **(E)** Pattern recognition accuracy.

the integrate-and-fire operation as shown in **Figure 5A**. Note that we calculated the area of the p-STT MTJ-based integrate-and-fire neuron using 1.6×1.6 μm^2 p-STT MTJ (i.e., ∼8.2 μm^2), which was approximately one-fourth smaller than the previous report (Sourikopoulos et al., 2017), as shown in **Supplementary Figure 6** and **Supplementary Table 3**. In this circuit, "fire" occurs when the resistance of the p-STT-based neuron exceeds the reference resistance (R_{ref}). The neuron receives control signals from a controller and performs integration, read, and reset operations in each clock cycle, as shown in **Figure 5A**. One controller can control multiple neurons simultaneously. In order to implement neural network, cross-point array can be used to realize analog matrix-vector multiplication. **Figure 5B** shows the schematic illustration of typical cross-point neural network implementation, which was fabricated by a cross-point synapse array being connected with our proposed p-STT MTJ neuron. Synapse would be IGZO-based memristor (in our experiment shown in **Supplementary Figure 2**). Where the bias voltage (V_{bias}) serves to ensure that the p-STT-based neuron is within its proper operating range.

Pattern Recognition

To investigate the performance of the SNN, a single-layer SNN consisting of input and output layers (50 p-STT-based neuron) was designed, as shown in **Figure 6A**. In this simulation, IGZO-based ReRAM was used in artificial synapse. A performance test of the SNN was carried out using the MNIST handwritten image set. MNIST images (6×10^4) were used for training, and 1×10^4 images not included in the training were used for testing. The probability of the input spike occurrence was set to be proportional to the pixel value of an input image, and the amplitude of an input spike was set to −1 V. The neurons integrate the input spike signals and fire when the resistance of the p-STT MTJ exceed R_{th} (=70 Ω). When the neurons fire, they generated an output spike. The winner takes all (WTA) was applied to the output neuron nodes. WTA improved the accuracy of a single-layer SNN since the WTA guarantees non-linear mapping in a single-layer SNN (Du et al., 2015; Hansen et al., 2017). Finally, only the synaptic weights associated with the fired output neurons were updated. In the initial synaptic weight map, the conductance of the synapses was randomly distributed. After training, the distribution of synaptic weights was changed. The weights for active and silent neurons are shown in **Figures 6C,D**, respectively. Even if there were more than 10 epochs, there were some silent neurons, as shown in **Figure 6D**. These silent neurons exhibited almost no firing during training. The reason for this is that the WTA updates only synaptic weights associated with neurons that have fired; consequently, synaptic weights connected with neurons that rarely fire are slower to learn. As a result, these less learned synapses reduce the firing rate of the connected silent neurons compared to other neurons. In the end, learning is rarely achieved for the silent neurons. In biological neural networks, there is a mechanism called "homeostasis" to overcome these problems. With this mechanism, a neuron that frequently fires increases the threshold required to fire, and a neuron that rarely fires decreases it (Lee et al., 2016b,d; Johnson et al., 2018). This mechanism lowers the fire threshold of neurons where learning has not been achieved; thus, it causes neurons to be more likely to fire during subsequent learning. However, it is difficult to change the reference resistance R_{th} once it is set in the circuit. This remains a problem to be solved in the future.

We use simplified STDP learning rule for synaptic learning. The synaptic weights before training are shown in **Figure 6B**. First, we simulated the dependence of pattern recognition accuracy on read error using our proposed the cross-point synapse array (i.e., **Figure 5B**) being connected with our proposed p-STT MTJ neuron (i.e., **Figure 5A**), as shown in **Supplementary Figure 7**. The pattern recognition accuracy sustained at ∼76% up to read error of 5% and then rapidly decreased with read error larger than 5%. In addition, we tested the dependence of pattern recognition accuracy on the reference resistance by simulation, as shown below **Supplementary Figure 8**. We determined the reference resistance that showed the highest accuracy of pattern recognition simulation. Using the simulated reference resistance, the pattern recognition accuracy rapidly increased to ∼76% in two epochs, as shown in **Figure 6E**. Since the single-layer SNN used in training is learned through STDP unsupervised learning, so only clustering was performed for each output stage. Therefore, the most frequent output values of each node were compared with the determined input value to measure the pattern recognition accuracy. The single-layer SNN, composed of p-STT-based neurons, showed a maximum recognition accuracy of ∼76%, which was somewhat lower than that of other reported neural networks (Burr et al., 2014). In the single-layer SNN, pattern recognition accuracy increases with the number of output neurons (Querlioz et al., 2015; Zahari et al., 2015; Hansen et al., 2017). However, even if the number of output neurons is increased to 100, it is difficult to obtain more than 90% accuracy. The major reason for the low accuracy is the lack of proper learning algorithms to train SNN. The spike signals are not differentiable, so global learning rule such as backpropagation cannot be used for training SNN. Therefore, local learning rule such as STDP is mainly used for training SNN. This limits the structure of neural network to a single layer. Therefore, in order to increase the accuracy of the SNN, further study of the learning algorithm is necessary.

DISCUSSION

p-STT MTJ could perform integration when the spin electron directions at double $Co_2Fe_6B_2$ free and pinned layers were switched from parallel to antiparallel states. However, for the integrate-and-fire operation, a neuron circuit performing the fire behavior was essentially designed. Pattern recognition accuracy of ∼76% was achieved using a ReRAM-based synapse model and the STDP learning rule. In summary, the p-STT-based neuron could perform like a typical neuron showing integrate-and-fire behavior and would be a suitable for SNN. In addition, a cross-point synapse array is essentially necessary, where a selector is vertically stacked on a synapse to eliminate a sneak current between synapses. Thus, further studies are necessary on processes for fabricating cross-point synapse arrays connected with p-STT-based neurons. In addition, since the two-terminal p-STT-based neuron can perform only the integration behavior, a circuit performing the fire behavior should also be designed. Therefore, further study is also necessary on a three-terminal p-STT-based neuron that uses a magnetic domain moving mechanism. Finally, since a strong merit of the p-STT-based neuron would be its power consumption; further study is necessary for a neuron circuit design with low power consumption.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

JP conceived and designed the study. JC, HJ, KA, and JB fabricated the samples and carried out measurements. JK and WY designed the neuron circuit. DK performed the simulations with the help of JP. All authors contributed to discussions regarding the research. DK, JC, and JP wrote the manuscript.

FUNDING

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2016M3A7B4910249).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.00309/full#supplementary-material

REFERENCES

Burr, G. W. W., Shelby, R. M. M., di Nolfo, C., Jang, J. W. W., Shenoy, R. S. S., Narayanan, P., et al. (2014). "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," in *Proceedings of the 2014 IEEE Int. Electron Devices Meet*, San Francisco, CA, 29.5.1–29.5.4. doi: 10.1109/IEDM.2014.7047135

Du, Z., Ben-Dayan Rubin, D. D., Chen, Y., He, L., Chen, T., Zhang, L., et al. (2015). "Neuromorphic accelerators: a comparison between neuroscience and machine-learning approaches," in *Proc. 48th Int. Symp. Microarchitecture - MICRO-48*, Waikiki, HI, 494–507. doi: 10.1145/2830772.2830789

Dutta, S., Kumar, V., Shukla, A., Mohapatra, N. R., and Ganguly, U. (2017). Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET. *Sci. Rep.* 7:8257. doi: 10.1038/s41598-017-07418-y

Gentet, L. J., Stuart, G. J., and Clements, J. D. (2000). Direct measurement of specific membrane capacitance in neurons. *Biophys. J.* 79, 314–320. doi: 10.1016/S0006-3495(00)76293-X

Grollier, J., Querlioz, D., and Stiles, M. D. (2016). Spintronic nanodevices for bioinspired computing. *Proc. IEEE* 104, 2024–2039. doi: 10.1109/JPROC.2016.2597152

Hansen, M., Zahari, F., Ziegler, M., and Kohlstedt, H. (2017). Double-barrier memristive devices for unsupervised learning and pattern recognition. *Front. Neurosci.* 11:91. doi: 10.3389/fnins.2017.00091

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.* 2016, 770–778. doi: 10.1109/CVPR.2016.90

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1007/BF02459568

Hu, G., Gottwald, M. G., He, Q., Park, J. H., Lauer, G., Nowak, J. J., et al. (2018). Key parameters affecting STT-MRAM switching efficiency and improved device performance of 400°C-compatible p-MTJs. *Tech. Dig. Int. Electron Devices Meet. IEDM* 5, 38.3.1–38.3.4. doi: 10.1109/IEDM.2017.8268515

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-Excitation Networks. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.* 2018, 7132–7141. doi: 10.1109/CVPR.2018.00745

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24:384010. doi: 10.1088/0957-4484/24/38/384010

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Johnson, A. P., Liu, J., Millard, A. G., Karim, S., Tyrrell, A. M., Member, S., et al. (2018). Homeostatic fault tolerance in spiking neural networks?: a dynamic hardware perspective. *IEEE Trans. Circuits Syst. I Regul. Pap.* 65, 687–699. doi: 10.1109/TCSI.2017.2726763

Kondo, K., Choi, J., Baek, J., and Jun, H. (2018). A two-terminal perpendicular spin-transfer torque based artificial neuron. *J. Phys. D Appl. Phys.* 51:504002. doi: 10.1088/1361-6463/aad592

Kurenkov, A., DuttaGupta, S., Zhang, C., Fukami, S., Horio, Y., and Ohno, H. (2019). Artificial neuron and synapse realized in an antiferromagnet/ferromagnet heterostructure using dynamics of spin–orbit torque switching. *Adv. Mater.* 31, 1–7. doi: 10.1002/adma.201900636

Lee, D. Y., Hong, S. H., Lee, S. E., and Park, J. G. (2016a). Dependency of tunneling-magnetoresistance ratio on nanoscale spacer thickness and material for double mgo based perpendicular-magnetic-tunneling-junction. *Sci. Rep.* 6, 1–9. doi: 10.1038/srep38125

Lee, D. Y., Lee, S. E., Shim, T. H., and Park, J. G. (2016b). Tunneling-magnetoresistance ratio comparison of mgo-based perpendicular-magnetic-tunneling-junction spin valve between top and bottom Co2Fe6B2Free layer structure. *Nanoscale Res. Lett.* 11, 1–7. doi: 10.1186/s11671-016-1637-9

Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016c). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508

Lee, S. E., Shim, T. H., and Park, J. G. (2016d). Perpendicular magnetic tunnel junction (p-MTJ) spin-valves designed with a top Co2Fe6B2free layer and a nanoscale-thick tungsten bridging and capping layer. *NPG Asia Mater.* 8, e324–e325. doi: 10.1038/am.2016.162

Liyanagedera, C. M., Sengupta, A., Jaiswal, A., and Roy, K. (2017). Stochastic spiking neural networks enabled by magnetic tunnel junctions: from nontelegraphic to telegraphic switching regimes. *Phys. Rev. Appl.* 8, 1–13. doi: 10.1103/PhysRevApplied.8.064017

MacLaren, J. M., and Willoughby, S. D. (2001). First principles determination of the effects of a grain boundary on the anisotropy and exchange energies in Co1Pt5 superlattices. *J. Appl. Phys.* 89, 6895–6897. doi: 10.1063/1.1357839

Merolla, P. A., Arthur, J. V., Alvarez-icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Sciencemag.Org* 345, 668–673. doi: 10.1126/science.1254642

Mizrahi, A., Hirtzlin, T., Fukushima, A., Kubota, H., Yuasa, S., Grollier, J., et al. (2018). Neural-like computing with populations of superparamagnetic basis functions. *Nat. Commun.* 9, 1–11. doi: 10.1038/s41467-018-03963-w

Monroe, D. (2014). Neuromorphic computing gets ready for the (really) big time. *Commun. ACM* 57, 13–15. doi: 10.1145/2601069

Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616

Sengupta, A., Panda, P., Wijesinghe, P., Kim, Y., and Roy, K. (2016). Magnetic tunnel junction mimics stochastic cortical spiking neurons. *Sci. Rep.* 6, 1–9. doi: 10.1038/srep30039

Shim, Y., Chen, S., Sengupta, A., and Roy, K. (2017). Stochastic spin-orbit torque devices as elements for bayesian inference. *Sci. Rep.* 7, 1–9. doi: 10.1038/s41598-017-14240-z

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961

Sourikopoulos, I., Hedayat, S., Loyez, C., Danneville, F., Hoel, V., Mercier, E., et al. (2017). A 4-fJ/spike artificial neuron in 65 nm CMOS technology. *Front. Neurosci.* 11:123. doi: 10.3389/fnins.2017.00123

Srinivasan, G., Sengupta, A., and Roy, K. (2017). Magnetic tunnel junction enabled all-spin stochastic spiking neural network. *Proc. 2017 Des. Autom. Test Eur. DATE* 2017, 530–535. doi: 10.23919/DATE.2017.7927045

Suzuki, M., Yasui, A., Kotani, Y., Tsuji, N., Nakamura, T., and Hirosawa, S. (2016). Magnetic domain evolution in Nd-Fe-B:Cu sintered magnet visualized by scanning hard X-ray microprobe. *Acta Mater.* 106, 155–161. doi: 10.1016/j.actamat.2016.01.011

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, Boston, MA, 1–9.

Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., et al. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature* 547, 428–431. doi: 10.1038/nature23011

Tuma, T., Pantazi, A., Le Gallo, M., Sebastian, A., and Eleftheriou, E. (2016). Stochastic phase-change neurons. *Nat. Nanotechnol.* 11, 693–699. doi: 10.1038/nnano.2016.70

Victora, R. H., Willoughby, S. D., MacLaren, J. M., and Xue, J. (2003). Effects of grain boundaries on magnetic properties of recording media. *IEEE Trans. Magn.* 39, 710–715. doi: 10.1109/TMAG.2003.808998

Zahari, F., Hansen, M., Mussenbrock, T., Ziegler, M., and Kohlstedt, H. (2015). Pattern recognition with TiOx-based memristive devices. *AIMS Mater. Sci.* 2, 203–216. doi: 10.3934/matersci.2015.3.203

Zhang, D., Zeng, L., Zhang, Y., Zhao, W., and Klein, J. O. (2016). Stochastic spintronic device based synapses and spiking neurons for neuromorphic computation. *Proc. 2016 IEEE/ACM Int. Symp. Nanoscale Archit. NANOARCH* 2016, 173–178. doi: 10.1145/2950067.2950105

Ziegler, M., Riggert, C., Hansen, M., Bartsch, T., and Kohlstedt, H. (2015). Memristive hebbian plasticity model: device requirements for the emulation of hebbian plasticity based on memristive devices. *IEEE Trans. Biomed. Circuits Syst.* 9, 197–206. doi: 10.1109/TBCAS.2015.2410811

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Kim, Yi, Choi, Ashiba, Baek, Jun, Kim and Park. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 07 July 2020
doi: 10.3389/fnins.2020.00423

On-Chip Training Spiking Neural Networks Using Approximated Backpropagation With Analog Synaptic Devices

*Dongseok Kwon, Suhwan Lim, Jong-Ho Bae, Sung-Tae Lee, Hyeongsu Kim, Young-Tak Seo, Seongbin Oh, Jangsaeng Kim, Kyuho Yeom, Byung-Gook Park and Jong-Ho Lee**

Department of Electrical and Computer Engineering, Inter-University Semiconductor Research Center, Seoul National University, Seoul, South Korea

Hardware-based spiking neural networks (SNNs) inspired by a biological nervous system are regarded as an innovative computing system with very low power consumption and massively parallel operation. To train SNNs with supervision, we propose an efficient on-chip training scheme approximating backpropagation algorithm suitable for hardware implementation. We show that the accuracy of the proposed scheme for SNNs is close to that of conventional artificial neural networks (ANNs) by using the stochastic characteristics of neurons. In a hardware configuration, gated Schottky diodes (GSDs) are used as synaptic devices, which have a saturated current with respect to the input voltage. We design the SNN system by using the proposed on-chip training scheme with the GSDs, which can update their conductance in parallel to speed up the overall system. The performance of the on-chip training SNN system is validated through MNIST data set classification based on network size and total time step. The SNN systems achieve accuracy of 97.83% with 1 hidden layer and 98.44% with 4 hidden layers in fully connected neural networks. We then evaluate the effect of non-linearity and asymmetry of conductance response for long-term potentiation (LTP) and long-term depression (LTD) on the performance of the on-chip training SNN system. In addition, the impact of device variations on the performance of the on-chip training SNN system is evaluated.

Keywords: neuromorphic, spiking neural networks, deep neural networks, on-chip training, supervised learning, hardware-based neural networks, synaptic devices

OPEN ACCESS

Edited by:
Elisa Donati,
ETH Zürich, Switzerland

Reviewed by:
Michael Pfeiffer,
Bosch Center for Artificial Intelligence, Germany
Shuangming Yang,
Tianjin University, China

***Correspondence:**
Jong-Ho Lee
jhl@snu.ac.kr

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *02 January 2020*
Accepted: *07 April 2020*
Published: *07 July 2020*

Citation:
Kwon D, Lim S, Bae J-H, Lee S-T, Kim H, Seo Y-T, Oh S, Kim J, Yeom K, Park B-G and Lee J-H (2020) On-Chip Training Spiking Neural Networks Using Approximated Backpropagation With Analog Synaptic Devices. Front. Neurosci. 14:423. doi: 10.3389/fnins.2020.00423

INTRODUCTION

Recently, artificial neural networks (ANNs) have shown superior performance in several fields, such as pattern recognition or object detection (Gokmen and Vlasov, 2016; Ambrogio et al., 2018; Kim C.-H. et al., 2018; Kim J. et al., 2018; Kim et al., 2019). The structure of ANNs was inspired by models of cortical hierarchies in neuroscience and neuroengineering (Fukushima, 1988; Riesenhuber and Poggio, 1999; Pfeiffer and Pfeil, 2018). In particular, convolutional neural networks (CNNs) inspired by the biological vision model have significantly improved the accuracy of deep neural networks (Krizhevsky et al., 2012).

However, it is difficult to say that the ANNs with a Von Neumann architecture perfectly imitate a human's brain, which is a very high-speed and massively parallel operating system with ultra-low power consumption (O'Connor et al., 2013; Shrestha et al., 2018; Kang et al., 2019). In light of this, hardware-based spiking neural networks (SNNs) capable of massively parallel operation by using analog synaptic devices have been regarded as an innovative type of computing system that can partially replace ANNs (Hwang et al., 2018).

Spiking neural networks can imitate biological behavior with various neuron and synapse models (Jo et al., 2010; Yang et al., 2016). Neurons in SNNs generate spikes to communicate between adjacent neurons. The input intensity of the neuron is represented as the number of spikes generated from the neurons (Oh et al., 2019). The spikes transmit through synapses and are integrated into the membrane capacitor of neurons in the next layer. When the membrane potential exceeds the threshold voltage, the neuron generates a spike to the deeper layer. This biological behavior of the neuron in SNNs can be matched to the behavior of the rectified linear unit (ReLU) activation function in ANNs (Diehl et al., 2015; Rueckauer et al., 2017). Since their behavior can be matched with each other, weights trained in ANNs with ReLU can be exactly converted to the weights in SNNs with very slight accuracy degradation. Using the ANN-to-SNN conversion method, SNNs have achieved state-of-the-art accuracy in MNIST, CIFAR-10, and Imagenet classification (Pfeiffer and Pfeil, 2018). However, the weights in SNNs should be trained from ANNs in serial operation, and the conversion is performed once. Therefore, SNNs adopting the ANN-to-SNN conversion cannot update themselves depending on various system situations and only perform the inference process for a given task. For this reason, the performance of SNNs that adopt conversion is sensitive to unexpected variations of hardware and cannot save the power consumption required for training a weight (Kim H. et al., 2018; Yu, 2018). In contrast, SNNs using on-chip training schemes that can update weights on the chip can have immunity against device variation or noise (Querlioz et al., 2013; Kwon et al., 2019). In addition, the on-chip training SNN systems train a weight by applying an update pulse to a synaptic device representing a weight, which leads to low power consumption for training a weight (Hasan et al., 2017).

There are two types of training weight methods for SNNs on the chip. One imitates the unsupervised training behavior in the human brain, for example, spike-timing-dependent plasticity (STDP) algorithms (Bi and Poo, 1998; Milo et al., 2016; Kheradpisheh et al., 2018). The other type is the supervised training method, which updates weights by approximating the backpropagation algorithm to match the behavior of the SNNs (Lee et al., 2016; Tavanaei and Maida, 2019). SNNs using unsupervised STDP have been reported to be implemented with synaptic devices, such as RRAM or Flash devices (Pedretti et al., 2017; Kim C.-H. et al., 2018; Prezioso et al., 2018). However, compared to conventional ANNs, the performance of SNNs using STDP is limited in terms of accuracy. In contrast to STDP, the performance of SNNs using approximated backpropagation is close to that of conventional ANNs. However, even in this case, signals representing an error value should be propagated backward while calculating and storing the values for updating weights, which is the main reason why it is difficult to implement hardware-based SNNs using on-chip training schemes.

Here, we propose a new supervised on-chip training scheme that efficiently approximates the backpropagation algorithm suitable for SNNs. The proposed on-chip training scheme dramatically reduces the memory usage required for the weight update by using 1 bit of memory per neuron to determine whether the neuron generates a spike at the last time step, and 1 bit of memory per neuron to store the derivative of the neuron's activation function. By using the stochastic characteristic of neurons in SNNs, the performance of SNNs using the proposed training scheme achieves the performance of ANNs. For the hardware configuration of on-chip training SNN systems, a gated Schottky diode (GSD), which has a saturated current, is fabricated as a synaptic device (Bae et al., 2017; Lim et al., 2019b). This characteristic greatly improves the reliability of the SNN system by allowing the GSDs to represent accurate weights even if an unexpected voltage drop occurs in the system (Lim et al., 2019a). In addition, a parallel conductance update scheme that speeds up the SNN system is validated for GSDs. We then design and simulate an on-chip training SNN system based on the results measured from GSDs and verify the performance of the system based on its ability to classify MNIST data sets. Lastly, the system is evaluated for non-ideal characteristics of synaptic devices, such as non-linearity, asymmetry, and device variation.

MATERIALS AND METHODS

Gated Schottky Diode

A three-terminal gated Schottky diode (GSD) that cuts off the Schottky forward current was previously fabricated to act as a synaptic device (Lim et al., 2019b). However, the GSD in the previous paper was damaged by the sputtering process for the deposition of metal electrodes. By reducing the sputtering power, the current level of the GSD is improved. **Figure 1A** shows a bird's eye view of the GSD. The bottom gate (BG) and ohmic contact (O) are made of n^+-poly silicon. A $SiO_2/Si_3N_4/SiO_2$ (ONO) stack is then deposited, and the Si_3N_4 layer acts as a charge storage layer. As an active layer, undoped Si is deposited on the ONO stack and O electrode. Contact holes are opened on the active layer after the layer of SiO_2 has formed. A Ti/TiN/Al/TiN stack is deposited on the exposed active layer by sputtering and forms the Schottky contact (S). **Figure 1B** shows a circuit diagram of an n-type GSD when the voltage applied to the BG is positive. If V_{BG} is positive, the Schottky junction is formed at the S contact, and NMOS is formed intrinsically within the structure of the GSD. Measured I_O-V_{BG} curves of GSDs for different V_O values are shown in **Figure 1C**. The effective Schottky barrier height for electrons decreases as V_{BG} increases, and the operating current of GSD is the reverse Schottky diode current. Therefore, the reverse Schottky diode current also increases as V_{BG} increases and can be used as a weight for SNNs. In addition, since the magnitude of the reverse Schottky diode current is low, an SNN system using GSDs operates with low power consumption. **Figure 1D** shows the measured I_O-V_O curves of GSDs with

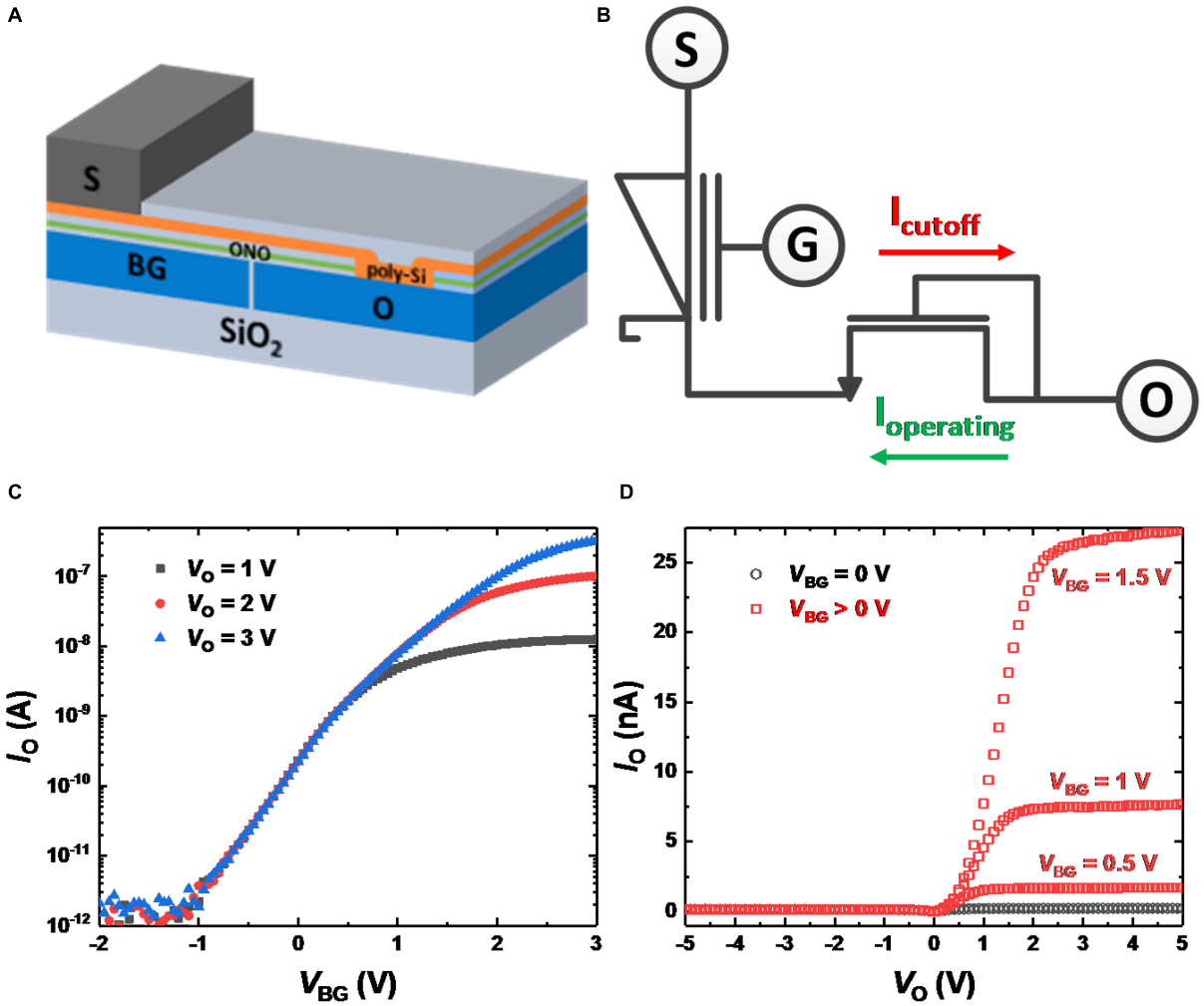


FIGURE 1 | (A) Bird's eye view of a GSD. **(B)** Circuit diagram of a GSD. When V_{BG} is positive, operating current flows with positive V_O, and the current is cut off with negative V_O. **(C)** Measured I_O-V_{BG} curves of GSDs with different V_O values. **(D)** I_O-V_O curves of GSDs with different V_{BG} values. When V_O is positive, saturated current is shown as V_O increases. Since a negative V_O depletes electrons in the poly-Si active layer, the I_O current is cut off.

different V_{BG} values. Since V_O above a certain value (e.g., 1.5 V at V_{BG} = 1 V) is dropped between O and S in **Figure 1B**, the reverse Schottky diode current is saturated with respect to the input voltage of V_O. With the help of the saturation behavior, the current of a GSD does not change despite voltage drops along metal wires in a crossbar array, and voltage drops by electronic switches do not affect the voltage across the device (Lim et al., 2019a). In addition, negative V_O depletes electrons in the Si active layer when V_{BG} is positive, and Schottky forward current is blocked.

Figures 2A,B show the conductance response (I_O at V_O = 3 V, V_{BG} = 0 V, V_S = 0 V) with respect to the time the erase pulse (V_{BG} = −7 V, V_O = 0 V, V_S = 0 V) and program pulse (V_{BG} = 5.5 V, V_O = 0 V, V_S = 0 V) are applied, respectively. Long-term potentiation (LTP) and long-term depression (LTD) curves are shown by applying the erase and program pulses, respectively. After GSDs are initialized, each pulse with a different pulse width is applied to the GSDs 10 times. Since the amount of charge stored in the Si_3N_4 layer is determined by the total time the FN tunneling current flows (Kim et al., 2010), the conductance can be changed continuously with the time of the program or erase pulses applied to the devices. The normalized conductance response of the GSD is fitted by the model of conductance with respect to the total time a pulse is applied to a synaptic device (Querlioz et al., 2011; Ernoult et al., 2019; Kwon et al., 2019), as follows:

$$G_{LTP}(t) = a_{LTP} + \frac{1}{\beta_{LTP}} \ln(t + c_{LTP}), \; for\ LTP \tag{1}$$

$$G_{LTD}(t) = a_{LTD} - \frac{1}{\beta_{LTD}} \ln(t + c_{LTD}), \; for\ LTD \tag{2}$$

where G is the conductance of the synaptic device, t is the total time the pulse is applied, a and c are the fitting parameters, and β is a non-linearity factor. As shown in **Figure 2**, the

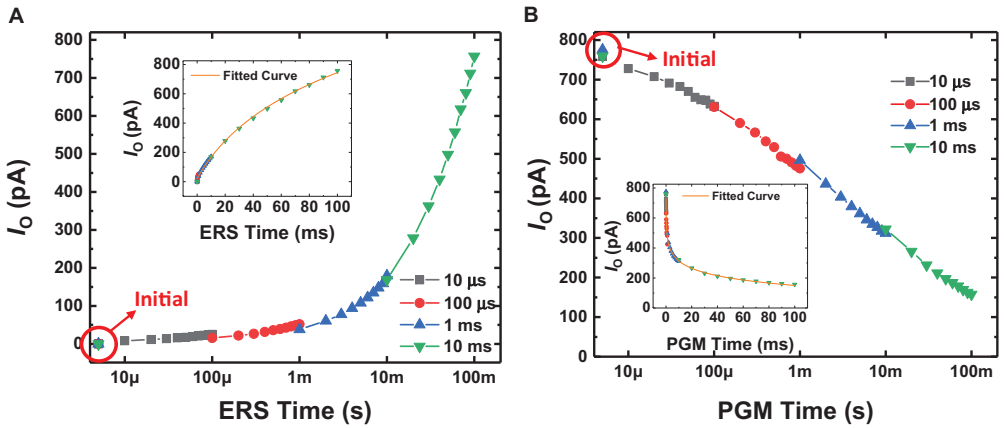


FIGURE 2 | I_O behavior with respect to **(A)** erase time and **(B)** program time when V_O is 3 V and V_{BG} is 0 V. After the GSD is initialized, erase (–7 V) or program (5.5 V) pulses are applied to the BG electrode, with 0 V applied to the S and O electrodes. Each pulse with a different pulse width is applied to the GSD 10 times. (Inset) I_O behavior with respect to the erase or program time on a linear scale.

GSDs have a near-linear LTP curve (β_{LTP} of ∼1.60) and a non-linear LTD curve (β_{LTD} of ∼8.03). The normalized conductance responses as a parameter of the non-linearity factor are described in **Supplementary Figure S1**.

On-Chip Training Algorithm

The behavior of an integrate-and-fire (I&F) neuron in an SNN can approximate the conventional ReLU activation function in ANNs (Tavanaei and Maida, 2019). A ReLU activation function, $f(y)$, is defined as follows:

$$f(y) = \max(0,\ y), \tag{3}$$

$$\frac{df}{dy} = \begin{cases} 1, & y > 0 \\ 0, & y \le 0 \end{cases} \tag{4}$$

where y is the input signal of the activation function. When the input signal of ReLU exceeds 0, the activated value is propagated to the next layer, and the derivative of ReLU is set to 1. This behavior of ReLU is similar to the behavior of I&F neurons, which also generate and propagate a spike when the membrane potential exceeds the threshold voltage. In this regard, I&F neurons are used in the forward-propagation phase (FP), the phase for the inference process. In addition, we approximate the derivative of the activation function of I&F neurons in the form of a derivative of ReLU.

In SNNs, a weight is represented by the conductance difference between two synaptic devices representing positive and negative values. In the case of a network having L layers, a weight connecting neuron i in layer l to neuron j in layer $l+1$ is represented by $W_{ij}^{l} = G_{+,\ ij}^{l} - G_{-,\ ij}^{l}$, where $l \in \{1,\ \ldots,\ L-1\}$ (Burr et al., 2015). The input of the first layer is converted to a Poisson-distributed spike train, and the input intensity is encoded as a spike rate. The input spikes are fed into the GSD arrays, which represent the weight matrix. An I&F neuron integrates charge resulting from the weighted sum into its membrane capacitor:

$$V_j^l(t_{FP}) = V_j^l(t_{FP}-1) + \sum_i^{N^{l-1}} \frac{S_i^{l-1}(t_{FP})}{C_{mem}} (G_{+,\ ij}^{l-1} - G_{-,\ ij}^{l-1}), \tag{5}$$

where $V^l{}_j(t_{FP})$ is the membrane potential of I&F neuron j in layer l at time step t_{FP}, N^{l-1} is the total number of neurons in layer l-1, $S^{l-1}{}_i(t_{FP})$ is a spike in the form of a voltage pulse generated from neuron i in layer l-1 at time t_{FP}, and C_{mem} is the membrane capacitance of an I&F neuron. The voltage pulses propagate along the O lines in the GSD array, and the currents along the O lines are added to the S lines in the array. The current output from the GSD array charges or discharges the membrane capacitor of an I&F neuron. The I&F neuron generates a spike when its membrane potential exceeds the firing threshold voltage of the I&F neuron (V_{th}). V_{th} is then subtracted from the membrane potential of the neuron:

$$if \quad V_j^l(t_{FP}) > V_{th}: \begin{cases} V_j^l(t_{FP}) = V_j^l(t_{FP}) - V_{th} \\ S_j^l(t_{FP}) = 1 \\ g_j^l = 1 \end{cases} \tag{6}$$

$$else: \quad S_j^l(t_{FP}) = 0, \tag{7}$$

where g is an approximated derivative of the neuron's activation function. When FP starts for a given input signal, the approximated derivative g of each neuron's activation function is initialized to 0. Then, if the neuron generates a spike during FP, g is set to 1. If the neuron does not generate a spike during FP, g remains 0. Although the behavior of an I&F neuron cannot be differentiable, neural networks have been reported to show comparable performance when storing a derivative with only 1 bit (Narayanan et al., 2017; Tavanaei and Maida, 2019). In the last layer ($l = L$), spikes generated from the neurons and target spikes that supervise the correct answer are

accumulated to obtain the "delta" value in the last layer (δ^L):

$$\delta_j^L = k \sum_{t_{FP}=1}^{T} \left(Target_j\left(t_{FP}\right) - S_j^L\left(t_{FP}\right) \right), \tag{8}$$

where T is the total time step for the FP, and k is a constant that converts the number of spikes into the voltage amplitude. For the correct label, a target spike train has a value of 1, and its firing frequency is set to the maximum. In other words, a target spike is generated every time step with the value of 1 for the correct label, and no target spike is generated for the wrong label. The constant k is set to the value with the maximum δ_j^L of 1 V. The whole process performed in the FP is simply described with a 1-layer network in **Figure 3**.

In the backward-propagation phase (BP), the delta values reversely propagate to the previous layer through the synaptic devices and are integrated to obtain the delta sum (Burr et al., 2015; Hasan et al., 2017; Narayanan et al., 2017; Ambrogio et al., 2018):

$$\delta_i^l = \sum_j^{N^{l+1}} \frac{\lambda_{BP}\delta_j^{l+1}}{C_{BP}} \left(G_{+,\ ij}^l - G_{-,\ ij}^l\right) g_i^l, \tag{9}$$

where g is the derivative of the neuron's activation function determined in the FP. λ is a constant representing the ratio of voltage pulse width to voltage amplitude, and C_{BP} is the capacitance to store δ. The δ is obtained in the form of voltage amplitude and is converted to a voltage pulse ($\lambda\delta$) with a width proportional to the voltage amplitude using the pulse-width modulation circuit (Hasan et al., 2017; Lim et al., 2019a). Although the current direction of GSDs in the BP should be kept the same as in the FP to maintain their conductance value, the delta sum can be performed along the O line of GSD arrays while maintaining the current flow direction (Lim et al., 2018). Then, δ_i^l is obtained when the corresponding derivative (g_i^l) determined in the FP is 1.

In the update phase (UP), the conductance of synaptic devices is updated depending on δ. In the conventional backpropagation algorithm, the weight (W_{ij}^l) update is calculated as $\Delta W_{ij}^l \propto x_i^l \delta_j^{l+1}$, where x_i^l is the activated value. When this update rule is applied to the SNNs, x_i^l is matched to the number of spikes generated from the neuron during the FP. However, significant power consumption and memory usage are required for counting and storing the number of spikes for every neuron, which can become a bottleneck for the entire SNN system (Yu, 2018). In this work, we use a 1-bit spike value (0 or 1) per neuron depending on whether the neuron generated a spike at the last time step:

$$\Delta t_{+,\ ij}^l = \Delta t_{-,\ ij}^l = \left| S_i^l\left(T\right) \times \lambda_{UP}\delta_j^{l+1} \right|, \tag{10}$$

where λ is a constant representing the ratio of voltage pulse width to voltage amplitude and Δt is the width of the voltage pulse applied to the corresponding synaptic device. In the UP, since the amount of conductance update is modulated by λ, λ represents the learning rate of conventional ANNs. Whether it is a program pulse or an erase pulse is determined by the sign of the delta value. When the weight increases, the conductance

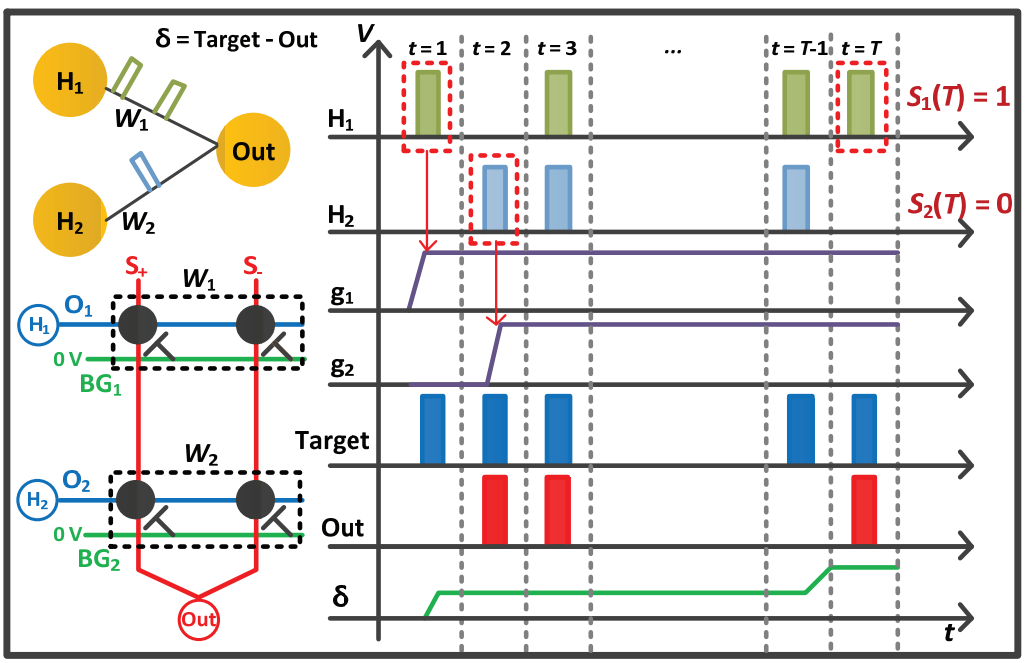


FIGURE 3 | Conceptual diagram with a 1-layer network for the forward phase (FP) of the proposed on-chip training scheme. The spikes from previous layers propagate along the O line of the G^+ and G^- array, and the current sum of the array is integrated into the membrane capacitor of the I&F. When the neuron H_1 and H_2 generate a spike, the derivative (g_1 and g_2) of the neurons is set to a value of 1.

of the synaptic device representing the positive weight increases by the erase pulse and the conductance of the synaptic device representing the negative weight decreases by the program pulse. On the contrary, when the weight decreases, the program pulse is applied to the device representing the positive weight and the erase pulse is applied to the device representing the negative weight. The whole training process of the proposed scheme is represented in **Algorithm 1**.

Updating Method

After all delta values (δ) except in the first input layer have been obtained, the conductance of the GSDs is updated by δ and $S(T)$. To update the conductance of GSDs in parallel, we apply DC bias to the BG and O lines of the array and a program or erase pulse to the S lines of the array. **Figure 4** shows the 2-by-2 layout of GSD arrays and the bias conditions of program and erase in the UP. The red dotted square represents the condition along the BG and O lines for $S(T)$ of 1, and the green dotted square stands for the condition along the S line if δ is not equal to 0. The width of the program and erase pulses is proportional to δ, which can be implemented by the pulse-width modulation circuit (Lim et al., 2019a). In this case, only cell 1 in **Figure 4** should be updated by a program or erase pulse, and the others should be inhibited in this condition. When a program pulse with an amplitude of −3.5 V is applied to the S line, the voltage of 2 V is applied to the BG line of cell 1. The voltage difference between the BG and S in cell 1 is then 5.5 V, which is the condition for programming a GSD. On the contrary, the voltage difference between BG and S of the other cells does not exceed 5.5 V, so the other cells are inhibited in this program scheme. In case of applying an erase pulse to the S line, the erase pulse has the same width as the program pulse width, but it has an amplitude of 5 V. The conductance change of each cell condition is shown in **Figure 5**. The width of the program pulse is 10 ms, and the width of the erase pulse is 100 ms. In both the cases of program and of erase, only the conductance of cell 1 is updated, and the others are inhibited successfully. By using this scheme, the GSDs in the array can be updated in parallel, which can improve the update speed of the entire SNN system. Note that the on-chip training SNN system updates weights as frequently as the training iterations, so a parallel conductance update of the device array is required to boost the training speed of the system.

Algorithm 1 | On-chip training scheme in SNNs with synaptic devices.

Input: Poisson-distributed input spike train (S^1), Total time step (T), threshold voltage of I&F neuron (V_{th}), ratio of the voltage pulse width to the voltage amplitude (λ), capacitance (C)

Initialize: Membrane voltage (V), derivative of the neuron's activation function (g)

for $t_{FP} = 1$ **to** T **do** // Forward-propagation phase

  for $l = 2$ **to** L **do**

    $V^l \leftarrow V^l + S^{l-1}\,(G_+^{l-1} - G_-^{l-1})/C_{mem}$

    if $V^l > V_{th}$ **then** // If membrane voltage exceeds V_{th}, the neuron generates a spike

      $V^l \leftarrow V^l - V_{th}$

      $S^l \leftarrow 1$

      $g^l \leftarrow 1$ (if $l \neq L$) // The corresponding derivative of the neuron's activation function is set to 1

    else

      $S^l \leftarrow 0$

    end if

  end for

  $\delta^L \leftarrow \delta^L + k\,(TargetSpike - S^L)$

end for

for $l = L$-1 **to** 2 **do** // Backward-propagation phase

  $\delta^l \leftarrow \lambda_{BP}\delta^{l+1}\,g^l\,(G_+^l - G_-^l)/C_{BP}$

  // δ^l is obtained in the form of voltage amplitude and converted to a voltage pulse with a width proportional to the amplitude ($\lambda\delta^l$)

end for

for $l = 1$ **to** L-1 **do**//Update phase

  $\Delta t \leftarrow \left|S^l\,(T)\;\lambda_{UP}\delta^{l+1}\right|$ // Applying a program or erase pulse to the corresponding synaptic device

  if $\delta^{l+1} > 0$ **then** // Case of increasing a weight

    $G_+^l \leftarrow G_{+,\,LTP}^l(t + \Delta t)$ // Increasing conductance of G_+^l by applying an erase pulse

    $G_-^l \leftarrow G_{-,\,LTD}^l(t + \Delta t)$ // Decreasing conductance of G_-^l by applying a program pulse

  end if

  if $\delta^{l+1} < 0$ **then** // Case of decreasing a weight

    $G_+^l \leftarrow G_{+,\,LTD}^l(t + \Delta t)$ // Decreasing conductance of G_+^l by applying a program pulse

    $G_-^l \leftarrow G_{-,\,LTP}^l(t + \Delta t)$ // Increasing conductance of G_-^l by applying an erase pulse

  end if

end for

RESULTS

Evaluation of On-Chip Training Scheme

We design and simulate fully connected (FC) neural networks for MNIST classification to verify the proposed on-chip training scheme for SNNs. The batch size of training is 1 to reduce memory usage and the area footprint required for the memory. The accuracy of SNNs is evaluated with the membrane voltage of the neuron at the last layer. The parameters in the training scheme for MNIST classification are described in **Table 1**. **Figure 6A** shows the MNIST test set accuracy of SNNs using the proposed on-chip training scheme according to the total time step (T). Here we assume that synaptic devices have a linear conductance response and no variation, and the baseline accuracy in **Figure 6A** is evaluated in ANNs that have the same network size. If T is 20, the maximum number of input, hidden, and output spikes are 20. The increased T precisely represents the activation value of each neuron and δ, resulting in improved accuracy for SNNs. When T is equal or more than 20, the SNNs show saturated accuracy but achieve accuracy near the baseline accuracy of ANNs. **Figures 6B,C** show whether the proposed on-chip training scheme can be applied to wider and deeper networks. The on-chip training SNNs achieve higher accuracy as the layer width increases, but the accuracy decreases as the depth of the network increases with the same T. In this case, since increased T represents more accurate neuron activation values and δ, the accuracy in deeper networks is expected to be improved. As a result of increasing T to 50, the accuracy of

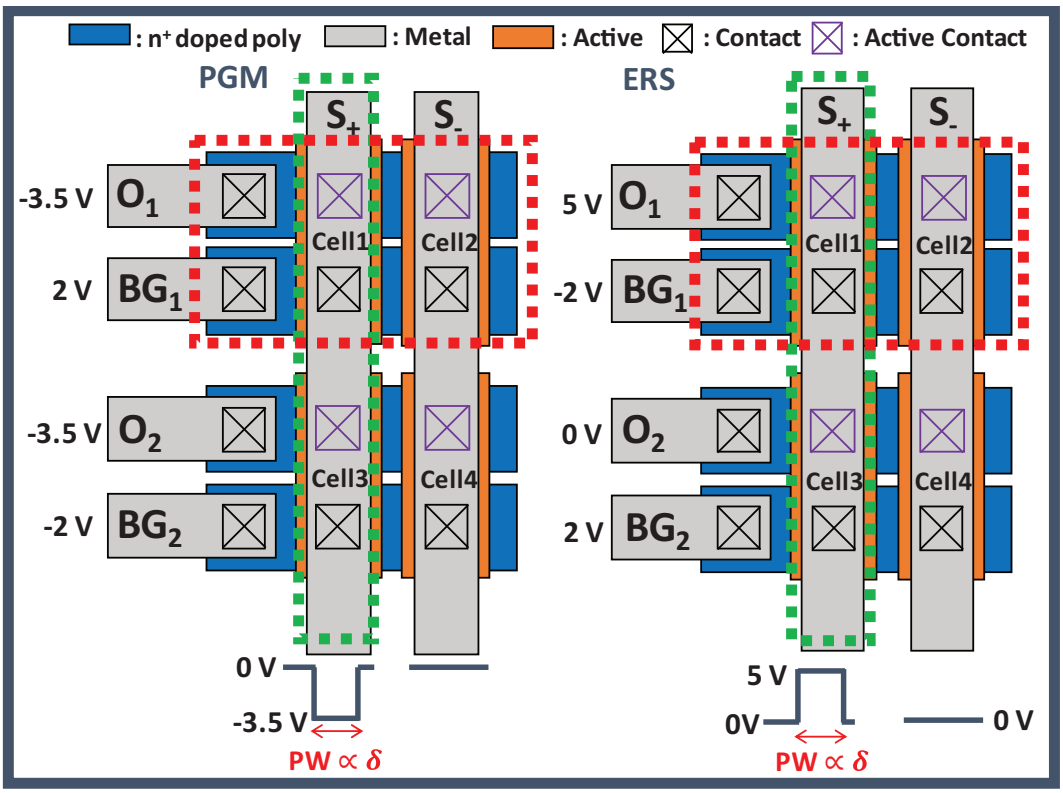


FIGURE 4 | Bias condition in the update phase with the GSD array. The red dotted square is the condition along the BG and O line for $S(T)$ of 1, and the green dotted square along the S line is the condition when δ is not equal to 0.

SNNs with 4 hidden layers increases, as shown in **Figure 6C**. Nevertheless, the training curve for the network with 4 hidden layers oscillates over epochs due to the large λ_{UP}. Since λ_{UP} is multiplied by δ, a large λ_{UP} increases the amount of weight update and causes the oscillating training curve. Thus, we scale λ_{UP} to train deeper networks. After reducing λ_{UP} to $0.2\lambda_{UP}$ at epoch 11 in **Figure 6D**, a stable training curve is obtained, and the accuracy increases to 98.25%.

Table 2 compares this work with conventional on-chip training schemes using analog synaptic devices for MNIST classification. The proposed on-chip training scheme achieves an accuracy near that of conventional ANNs even when the batch size of training is 1 with a single hidden layer. In addition, we increase the batch size to 100 to improve the proposed scheme for SNNs with 4 hidden layers. Although increasing batch size for training directly increases memory usage, it improves the accuracy of deep networks. As a result, the network achieves an accuracy of 98.44% (0.1% lower than the accuracy of an ANN using the Adam optimizer), and shows excellent performance compared to other on-chip training schemes.

When ANNs are converted to SNNs, I&F neurons generate spikes at each time step with a probability proportional to the activated value in the ANN. Then, the weights connected to the neuron that generates a large number of spikes are updated with a high probability in one training iteration. This weight update scheme using a 1-bit spike event of a neuron is less accurate than that using the total number of spikes of a neuron. However, the average of total weight updates using a 1-bit spike approximates the average of total weight updates using the number of spikes of the neuron. To compare the weight update schemes, we trace the sum of total weight updates in each layer with respect to the training iterations. Case 1 is the sum of total weight updates using 1-bit spike events (this work), and Case 2 is the sum using the total number of spike events. In Case 1, $S(T)$ of the equation (10) is 0 or 1, determined by the spike event at the last time step. In Case 2, $S(T)$ in the equation (10) is converted to the number of spikes in the FP divided by T. For example, if the neuron generates spikes 14 times in the FP with a T of 20, the $S(T)$ in the equation (10) is converted to 0.7 for Case 2. The actual weight update is performed with the 1-bit spike of a neuron, but the amount of the weight update is calculated by both ways at each iteration to compare them. **Figure 7** shows the difference between the sum of total weight updates for Case 1 and Case 2. As shown in **Figure 7**, the sums of total weight updates in both cases are not exactly the same, but the values in Case 1 fluctuate around the values in Case 2. In addition, we trace the sum of weight updates of the random position in each layer: a synapse connecting the 358th neuron as the input layer and the 124th neuron as the hidden layer, and a synapse connecting the 97th neuron as the hidden layer and the 5th neuron as the output layer. As shown in **Figure 8**, the sum of weight updates in case 1 follows the curve for case 2, although the curves are not exactly the same.

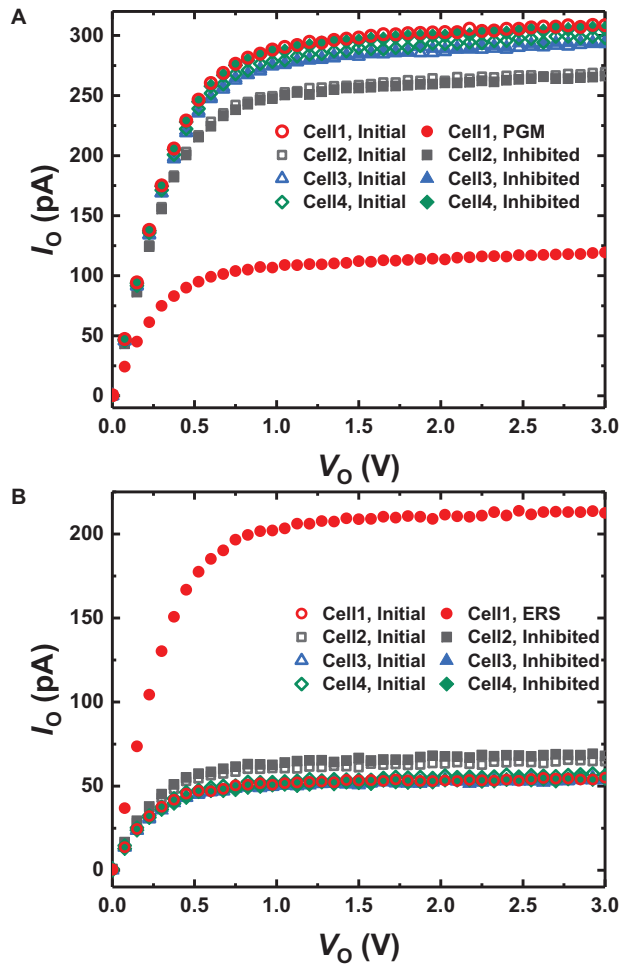


FIGURE 5 | I_O-V_O curves of the GSDs in the array **(A)** programming for 10 ms and **(B)** erasing for 100 ms depending on each condition in **Figure 4**. Only the current of cell 1 is changed, while others are inhibited successfully.

This indicates that the proposed on-chip training scheme for SNNs can achieve performance similar to that of ANNs by using the stochastic characteristics of SNNs. In other words, a spike from a neuron is generated at every time step with a probability proportional to the value of the neuron's activation function, so the 1-bit spike event approximates the behavior of the neuron's activation function during training.

Non-ideal Device Characteristics

The accuracy of on-chip training SNNs versus the non-linearity of conductance response is shown in **Figure 9A**. Although the delta value (δ) can be applied to the synaptic devices in the form of the program or erase pulse, the conductance response is non-linear with respect to the updating pulse. As a result, the expected weight updates cannot be reflected in the conductance updates, which causes accuracy degradation of SNNs. Nevertheless, an accuracy of higher than 93% is obtained when the non-linearity factor (β) is 8 for both LTP and LTD, which is an extremely non-linear conductance response of

TABLE 1 | The parameters when the GSDs are used as synaptic devices in the SNNs.

Parameters	Description	Value
a_{LTP}, a_{LTD} c_{LTP}, c_{LTD}	Parameters of the fitted curve for the normalized conductance response of GSDs	2.270, 1.422 0.0278, 18.25
β_{LTP},β_{LTD}	Non-linearity factor of GSDs	1.60, 8.03
S	Spike in the form of voltage pulse	Pulse amplitude: 3 V Pulse duration: 10 μs
C_{mem}	Membrane capacitance of I&F neuron	$4T-8T$ fF (l = 1) 40–80 fF (l > 1)
V_{th}	Threshold voltage of I&F neuron	0.1 V
λ	Ratio of voltage pulse width to voltage amplitude	BP 50 μs/V UP 50 μs/V ($l = L$) 500 μs/V ($l \neq L$)
C_{BP}	Capacitance for BP	40 fF

synaptic devices. Since the conductance of synaptic devices is updated continuously with the program or erase time, the on-chip training SNN system can achieve high accuracy even with highly non-linear devices. The accuracy of SNNs depending on the non-linearity for LTD is shown in **Figure 9A** to investigate the effect of asymmetry between the LTP and LTD curves on the accuracy. The non-linearity factor of the LTP curve has fixed values of 1 and 3. The accuracy of SNNs decreases as the non-linearity factor for LTD increases, represented as the red and black lines in **Figure 9A**. However, the degree of accuracy reduction resulting from the asymmetry is less than when β values for both LTP and LTD increase. In the case of a GSD as a synaptic device, the on-chip training SNN achieves an accuracy of 96.5%. The near-linear conductance change in the LTP curve can mitigate the effect of non-linear conductance change in LTD.

Inherent device variation is inevitable in neurons and synaptic devices. We categorize the device variation into three types: pulse-to-pulse variation (Chen et al., 2015), device-to-device variation (Gong et al., 2018; Sun and Yu, 2019), and stuck-at-off variation (Li et al., 2018). The performance of the proposed on-chip training scheme is evaluated with the degree of each variation and is compared with the performance of the off-chip training scheme (Kwon et al., 2019). When the off-chip training scheme is adopted to SNNs, the weights trained in ANNs using ReLU are exactly converted to the weights in SNNs by modulating the width of pulses applied to the synaptic devices. The synaptic device used in the SNNs is the GSD device, which has β values of ∼1.60 and ∼8.03 for LTP and LTD. In the off-chip training scheme, the trained weights are transferred to conductance along the LTP curve. When the ANN-to-SNN conversion is adopted, the accuracy of off-chip training SNNs with a T of 20 is 98.04% for MNIST data classification as a baseline. All accuracy datapoints in **Figures 9B–D** were evaluated five times and then averaged. The error bars show 1 standard deviation over five simulations.

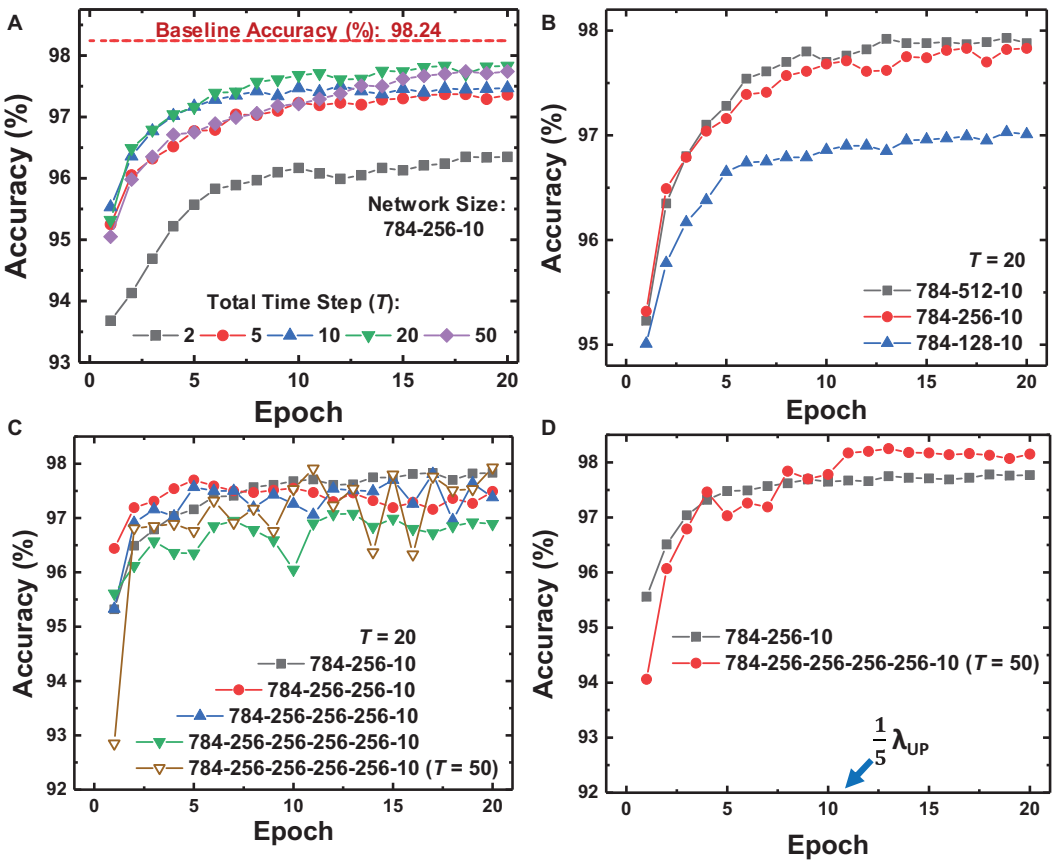


FIGURE 6 | Training curves of the proposed on-chip training scheme depending on **(A)** the total time step (T) in FP, **(B)** the width of the hidden layer, and **(C)** the number of hidden layers. **(D)** Training curves when modulating λ at epoch 11. Deep networks show variance over epochs with large λ, but modulating λ stabilizes the training curve and improves the accuracy of the network.

TABLE 2 | Comparison of the proposed with conventional on-chip training schemes for hardware-based neural networks using analog synaptic devices.

Architecture	Training method	Batch size	Network size	Accuracy
FC (Zhang et al., 2018)	Sign BP	–	784-300-10	94.50%
FC (Chang et al., 2018)	BP-based	10–50	784-300-10	97.93%
FC (Fu et al., 2019)	BP-based	–	400-100-10	95.55%
FC (Ambrogio et al., 2018)	Adam	1	528-250-125-10	97.94%
FC (Lim et al., 2018)	Manhattan learning rule	1	784-200-10	95.36%
FC (ANN*)	Adam	100	784-256-256-256-256-10	98.54%
FC (This work)	BP-based	1	784-256-10	97.83%
FC (This work)	BP-based	100	784-256-256-256-256-10	98.44%

*The accuracy of all reported data was evaluated through the classification of the MNIST test set and obtained through hardware-based simulation. The proposed on-chip training scheme shows excellent accuracy compared to other on-chip training schemes. *Software-based neural network simulated in the Pytorch framework.*

Figure 9B shows a comparison between the accuracy of SNNs using the on-chip and off-chip training scheme by taking pulse-to-pulse variation into account. When an update pulse is applied to a synaptic device, a Gaussian distribution function is used to indicate fluctuations in weight updates. The variation is applied to the on-chip training SNN system whenever an update pulse is applied. On the other hand, the variation affects the off-chip training system only once when transferring the trained weights to the conductance of synaptic devices in SNNs. As shown in **Figure 9B**, the accuracy significantly decreases when a large conductance variation is applied to the synapses in the off-chip training SNN system. However, even if σ/μ increases to 2, the accuracy of the on-chip training SNN system is maintained (accuracy loss of 0.2% at $\sigma/\mu = 2$).

We also evaluate the effects of device-to-device variation on the SNNs. Synaptic devices in the array can have various characteristics for one non-linearity factor. We assume that the non-linearity factor of synaptic devices in the array follows a

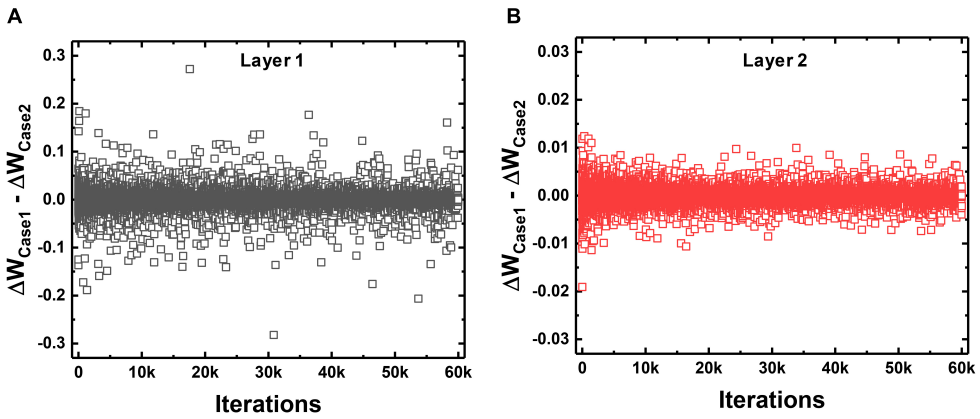


FIGURE 7 | Comparison between the updating method that uses only a 1-bit spike event at the last time step per neuron (Case 1) and the total number of generated spikes in the neuron divided by the total time step (Case 2). The difference in the sum of total weight updates for Case 1 and Case 2 with respect to the training iterations in **(A)** the first layer and **(B)** the second layer.

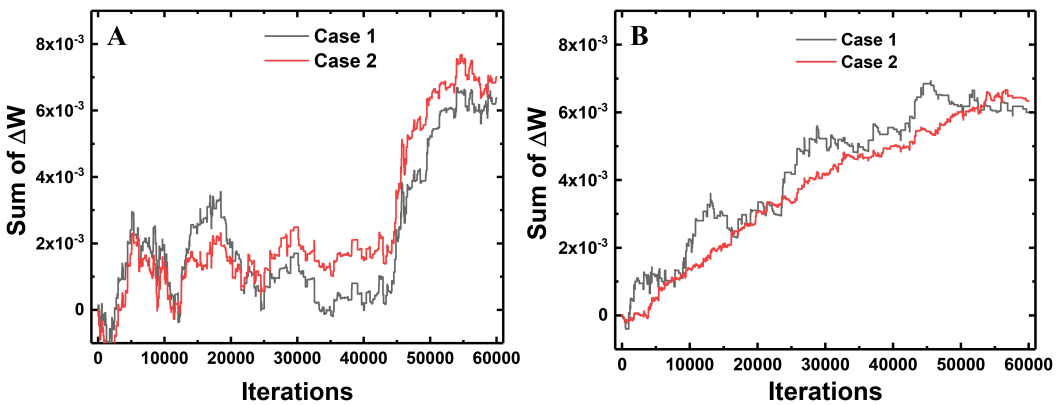


FIGURE 8 | Sum of weight updates in a synapse connecting two adjacent neurons in the 784-256-10 network. **(A)** Sum of weight updates in the synapse between the 358th neuron as the input layer and the 124th neuron as the hidden layer. **(B)** Sum of weight updates in the synapse connecting the 97th neuron as the hidden layer and the 5th neuron as the output layer.

Gaussian distribution, and the accuracy of SNNs is evaluated with respect to the degree of variation. As a result of applying the device-to-device variation, the synaptic device array has various conductance responses with different non-linearity factors. However, the on-chip training SNN systems also maintain their accuracy, but the accuracy of off-chip training SNN systems decreases as the degree of device-to-device variation increases, as shown in **Figure 9C**.

Lastly, we investigate the effect of the stuck-at-off ratio on the accuracy of SNNs. The stuck-at-off ratio is defined as the ratio of the number of stuck-at-off devices to the total number of devices in the array. Note that the number of devices with a conductance of 0 increases as the stuck-at-off ratio increases, and the stuck devices cannot be updated. As shown in **Figure 9D**, the accuracy of on-chip training SNNs decreases as the stuck-at-off ratio increases. A device pair represents a weight in SNNs, and both devices in the pair are updated when the corresponding weight is updated. Therefore, the weight updates are always performed using both near-linear LTP and LTD curves, which can mitigate the abrupt conductance change in the highly non-linear LTD curve of the GSDs. However, if one device in the pair is stuck-at-off with respect to all training iterations, the abrupt changes of stuck devices cannot be mitigated and degrade the performance of SNNs, even if the on-chip training scheme is adopted. When the SNNs adopt the off-chip training scheme, the accuracy of SNNs also degrades as the ratio increases, and the degree of accuracy loss is more severe than in the case of adopting the on-chip training scheme.

DISCUSSION

In this work, we proposed an on-chip training scheme suitable for hardware-based SNNs using analog synaptic devices. This scheme requires 2 bits of memory per neuron to update a weight: 1 bit for storing the spike event of the neuron at the last time step and the other for storing the derivative of the neuron's activation function. Since the input of the first layer is

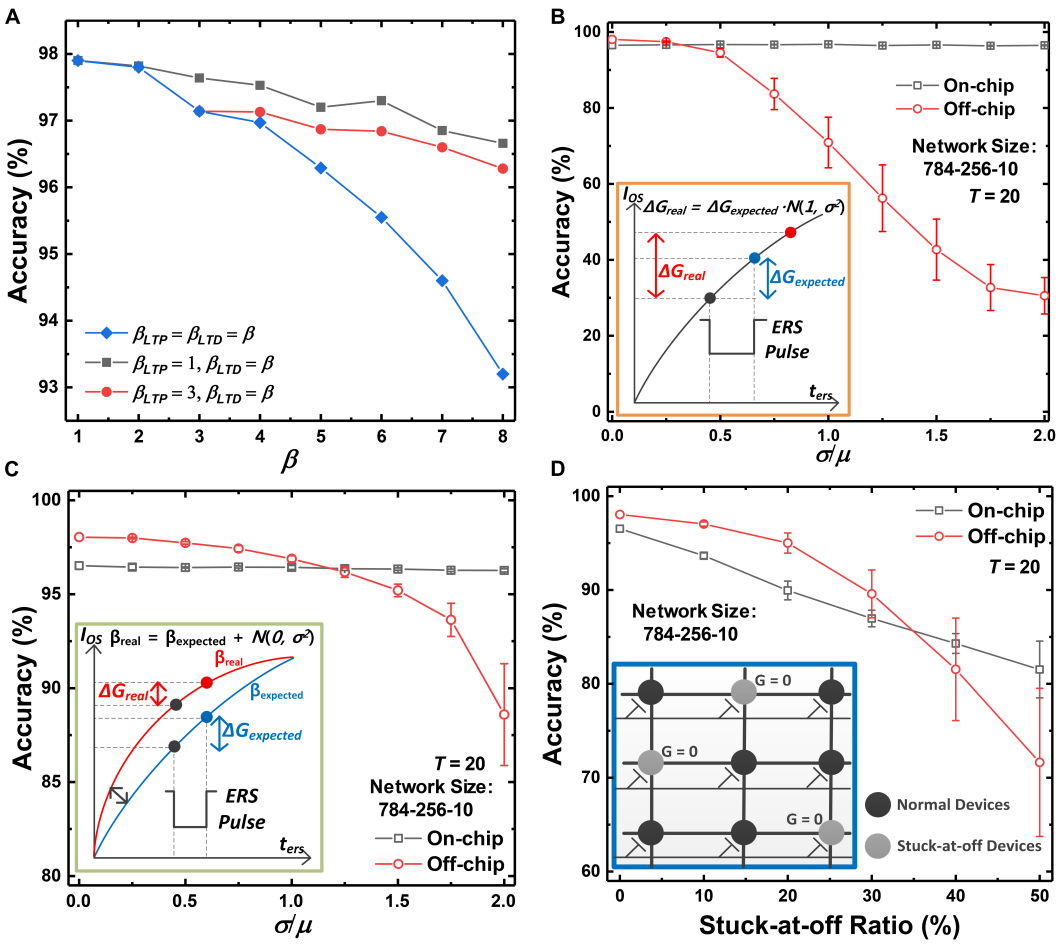


FIGURE 9 | Accuracy of the on-chip training SNN systems versus **(A)** the non-linearity factor (β) of conductance response, **(B)** pulse-to-pulse variation, **(C)** device-to-device variation, and **(D)** stuck-at-off ratio. Although extremely non-linear and asymmetric devices are used as synaptic devices, high accuracy is obtained. Degradation due to pulse-to-pulse variation and device-to-device variation is negligible, but degradation due to the stuck-at-off ratio is significant.

converted to a Poisson-distributed spike train, the probability of generating a spike at each time step is determined by the activated value of the neuron. The stochastic 1-bit spike event of an I&F neuron helps the system achieve high accuracy while using the minimum memory. In addition, we evaluated the performance of the proposed training scheme in classifying N-MNIST data that cannot be represented as Poisson-distributed spike trains. As shown in **Supplementary Table S1**, the on-chip training SNN system achieved 97.64% accuracy with real spike data from event-based sensors (N-MNIST data) and still has the advantages of low power consumption and hardware efficiency.

As a synaptic device, we fabricated a gated Schottky diode (GSD), which has saturated current with respect to the input voltage. Even if a noisy input voltage is applied to the GSD, the weight represented by the GSD is stable because almost constant saturation current is maintained. When the on-chip training SNN system uses GSDs as synaptic devices, the array of GSDs can be updated and inhibited in parallel operation, which greatly boosts the training speed of the SNN system. In addition, the energy consumption per spike in a GSD is about 30 fJ (∼1 nA current at 3 V amplitude and 10 μs pulse width), so the on-chip training SNN system is estimated to operate at very low power consumption.

The on-chip training SNN system was verified with fully connected neural networks for MNIST data classification. The accuracy of SNNs (784-256-10) using the on-chip training scheme achieved 97.83% with T of 20, compared to an accuracy of 98.04% when ANN-to-SNN conversion was used with the same network. Since we did not use regularization methods such as dropout (Srivastava et al., 2014) or L2 regularization, training curves with a large λ_{UP} in deep networks can show variance, and the accuracy of deep networks can decrease. In this case, increasing T is a way to recover accuracy, because the activated and delta values of the neuron are more precisely represented by increased T. However, increasing T can be a burden on the overall system because the forward-propagation process is repeated T times in on-chip training SNNs. Increasing the batch size of the training process is also a way to enhance the accuracy of deep networks by averaging stochastic spike events of neurons within a single batch training. We confirmed that the accuracy of deep

networks with increased batch size (98.44%) is very close to that of conventional ANNs (98.54%). In addition, the accuracy of deep networks can be improved by controlling the λ_{UP}, which is used as the learning rate of conventional ANNs. Since the proposed on-chip training scheme uses a 1-bit spike event at the last time step, the weight updates are calculated less precisely compared to the conventional backpropagation algorithm. Therefore, setting a small λ_{UP} allows deep networks to achieve high accuracy.

We investigated the effect of the non-ideal characteristics of synaptic devices on the performance of on-chip training SNNs. Digital SNN systems seem to alleviate the influence of the non-ideal characteristics of synaptic devices (Pani et al., 2017; Yang et al., 2018; Yang et al., 2020), but analog SNN systems can be affected by such synaptic characteristics. Therefore, their influence needs to be considered when evaluating the performance of analog SNN systems. In this work, the non-linearity and asymmetry of devices affected the performance of SNNs, but high accuracy was still achieved even in the extreme case. Since the width of pulses to be applied to synaptic devices is obtained in proportion to the delta value, degradation due to non-linear weight update is mitigated in this training scheme. Compared with conventional on-chip training algorithms that use the number of pulses to be applied to update the weights, this scheme has the advantage of continuously and accurately updating the conductance of synaptic devices. As a result, this training system allows the conductance of analog synaptic devices with continuous characteristics to be reflected in the training process, thereby improving the accuracy of SNNs with non-linear synaptic devices.

Furthermore, the effects of three types of device variations on the performance of SNNs were evaluated with respect to the degree of the variation when the GSDs are used as synaptic devices: pulse-to-pulse variation, device-to-device variation, and the stuck-at-off device ratio. Since on-chip training SNNs can mitigate the impact of variation on the system performance, the accuracies of on-chip training SNN systems with GSDs are slightly affected by the pulse-to-pulse variation and device-to-device variation. In contrast, if one of the pairs of devices is stuck-at-off, non-linear weight updates by the LTD curve of one GSD device have a significant impact on the training process and degrade the performance of on-chip training SNNs. However, since GSDs are fabricated with reliable CMOS processes, the stuck-at-off ratio in the GSD array is expected to be negligibly small.

The main challenge of the proposed on-chip training scheme for SNNs is realizing the performance of convolutional neural networks (CNNs) or recurrent neural networks (RNNs). To achieve this, weight sharing in the CNN structure should be implemented in SNN systems with low power consumption (Bartunov et al., 2018). Although the max-pooling layer and softmax layer in CNNs can be implemented in SNNs (Rueckauer et al., 2017), the batch normalization layer, which significantly improves the performance of CNNs, should be implemented in hardware-based SNNs while updating parameters during training iterations. In addition, the long short-term memory (LSTM) layer in RNNs should be implemented in the form of SNNs without much memory usage. If the conditions mentioned above are met, the proposed on-chip training scheme is expected to achieve state-of-the-art performance for hardware-based SNNs with low power consumption and high-speed parallel operation.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

DK, SL, and J-HB conceived and designed the experiments. DK, SL, and HK performed the simulation for MNIST classification. DK, SL, J-HB, S-TL, HK, Y-TS, JK, and J-HL performed the theoretical analyses. DK and Y-TS measured the device characteristics. DK, SO, KY, B-GP, and J-HL wrote the manuscript. All of the authors discussed the results and commented on the manuscript.

FUNDING

This work was supported by the National Research Foundation of Korea (NRF-2016M3A7B4909604) and the Brain Korea 21 Plus Project in 2020.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.00423/full#supplementary-material

REFERENCES

Ambrogio, S., Narayana, P., Tsai, H., Shelby, R. M., Boybat, I., Nolfo, C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558, 60–67. doi: 10.1038/s41586-018-0180-5

Bae, J.-H., Lim, S., Park, B.-G., and Lee, J.-H. (2017). High-density and near-linear synaptic device based on a reconfigurable gated schottky diode. *IEEE Electron Device Lett.* 38, 1153–1156. doi: 10.1109/led.2017.2713460

Bartunov, A., Santoro, A., Richards, B. A., Hinton, G. E., and Lillicrap, T. P. (2018). Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Adv. Neural Inform. Proc. Syst.*

Bi, G.-Q., and Poo, M.-M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci* 18, 10464–10472. doi: 10.1523/JNEUROSCI.18-24-10464.1998

Burr, G. W., Shelby, R. M., Sidler, S., Norfo, C., di Jang, J., Irem, B., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network

(165000 Synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electron Devices* 62, 3498–3507. doi: 10.1109/ted.2015.2439635

Chang, C.-C., Chen, P.-C., Chou, T., Wang, I.-T., Hudec, B., Chang, C.-C., et al. (2018). Mitigating asymmetric nonlinear weight update effects in hardware neural network based on analog resistive synapse. *IEEE J. Emerg. Select. Top. Circ. Syst.* 8, 116–124. doi: 10.1109/JETCAS.2017.2771529

Chen, P.-Y., Seo, J.-S., Cao, Y., and Yu, S. (2015). "Mitigating effects of non-ideal synaptic device characteristics for on-chip learning," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, New York, NY: ACM, 194–199.

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Piscataway, NJ: IEEE.

Ernoult, M., Grollier, J., and Querlioz, D. (2019). Using memristors for robust local learning of hardware restricted boltzmann machines. *Sci. Rep.* 9, 1851.

Fu, J., Liao, Z., Gong, N., and Wang, J. (2019). Mitigating nonlinear effect of memristive synaptic device for neuromorphic computing. *IEEE J. Emerg. Select. Top. Circ. and Sys.* 9, 377–387. doi: 10.1109/JETCAS.2019.2910749

Fukushima, K. (1988). Neocognitron: a hierarchical neural network capable of visual pattern recognition. *Neural. Netw.* 1, 119–130. doi: 10.1016/0893-6080(88)90014-7

Gokmen, T., and Vlasov, Y. (2016). Acceleration of deep neural network training with resistive cross-point devices: design considerations. *Front. Neurosci.* 10:333. doi: 10.3389/fnins.2016.00333

Gong, N., Ide, T., Kim, S., Boybat, I., Sebastian, A., Narayanan, V., et al. (2018). Signal and noise extraction from analog memory elements for neuromorphic computing. *Nat, Commun.* 9:2102. doi: 10.1038/s41467-018-04485-1

Hasan, R., Taha, T. M., and Yakopcic, C. (2017). On-chip training of memristor crossbar based multi-layer neural networks. *Microelectron. J.* 66, 31–40. doi: 10.1016/j.mejo.2017.05.005

Hwang, S., Kim, H., Park, J., Kwon, M.-W., Baek, M.-H., Lee, J.-J., et al. (2018). System-level simulation of hardware spiking neural network based on synaptic transistors and i&f neuron circuits. *IEEE Electron Device Lett.* 39, 1441–1444. doi: 10.1109/LED.2018.2853635

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Kang, W.-M., Kim, C.-H., Lee, S., Woo, S.-Y., Bae, J.-H., Park, B.-G., et al. (2019). "A Spiking neural network with a global self-controller for unsupervised learning based on spike-timing-dependent plasticity using flash memory synaptic devices," in *International Joint Conference on Neural Networks (IJCNN)*, Piscataway, NJ: IEEE, 1–7.

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., and Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* 99, 56–67. doi: 10.1016/j.neunet.2017.12.005

Kim, C.-H., Lee, S., Woo, S.-Y., Kang, W.-M., Lim, S., Bae, J.-H., et al. (2018). Demonstration of unsupervised learning with spike-timing-dependent plasticity using a TFT-Type NOR flash memory array. *IEEE Trans. Electron Devices* 65, 1774–1780. doi: 10.1109/ted.2018.2817266

Kim, C.-H., Lim, S., Woo, S. Y., Kang, W.-M., Seo, Y.-T., Lee, S.-T., et al. (2019). Emerging memory technologies for neuromorphic computing. *Nanotechnology* 30:03200. doi: 10.1088/1361-6528/aae975

Kim, D.-H., Cho, S., Li, D. H., Yun, J.-G., Lee, J.-H., Lee, G. S., et al. (2010). Program/erase model of nitride-based nand-type charge trap flash memories. *Jap. J. Appl. Phys.* 49:084301. doi: 10.1143/jjap.49.084301

Kim, H., Hwang, S., Park, J., Yun, S., Lee, J.-H., and Park, B.-G. (2018). Spiking neural network using synaptic transistors and neuron circuits for pattern recognition with noisy images. *IEEE Electron Device Lett.* 39, 630–633. doi: 10.1109/LED.2018.2809661

Kim, J., Koo, J., Kim, T., and Kim, J. J. (2018). Efficient synapse memory structure for reconfigurable digital neuromorphic hardware. *Front. Neurosci.* 12:829. doi: 10.3389/fnins.2018.00829

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, New York, NY: AMC.

Kwon, D., Lim, S., Bae, J.-H., Lee, S.-T., Kim, H., Kim, C.-H., et al. (2019). Adaptive weight quantization method for nonlinear synaptic devices. *IEEE Trans. Electron Devices* 66, 395–401. doi: 10.1109/TED.2018.2879821

Lee, J.-H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508

Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., et al. (2018). Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* 9:2385. doi: 10.1038/s41467-018-04484-2

Lim, S., Bae, J.-H., Eum, J.-H., Lee, S., Kim, C.-H., Kwon, D., et al. (2018). Adaptive learning rule for hardware-based deep neural networks using electronic synapse devices. *Neural Comput. Appl.* 31, 8101–8116. doi: 10.1007/s00521-018-3659-y

Lim, S., Kwon, D., Eum, J.-H., Lee, S.-T., Bae, J.-H., Kim, H., et al. (2019a). Highly reliable inference system of neural networks using gated schottky diodes. *IEEE J. Electron Devices Soc.* 7, 522–528. doi: 10.1109/JEDS.2019.2913146

Lim, S., Kwon, D., Lee, S.-T., Kim, H., Bae, J.-H., and Lee, J.-H. (2019b). "Investigation of neural networks using synapse arrays based on gated schottky diodes," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest. doi: 10.1109/jeds.2019.2913146

Milo, V., Pedretti, G., Carboni, R., Calderoni, A., Ramaswamy, N., Ambrogio, S., et al. (2016). "). Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity," in *2016 IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA: IEEE.

Narayanan, P., Fumarloa, A., Sanches, L. L., Hosokawa, K., Lewis, S. C., Shelby, R. M., et al. (2017). Toward on-chip acceleration of the backpropagation algorithm using nonvolatile memory. *IBM J. Res. Dev.* 61, 11.1–11.11. doi: 10.1147/jrd.2017.2716579

O'Connor, P., Neil, D., Liu, S. C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* 7:178. doi: 10.3389/fnins.2013.00178

Oh, S., Kim, C.-H., Lee, S., Kim, J. S., and Lee, J.-H. (2019). Unsupervised online learning of temporal information in spiking neural network using this-film transistor-type NOR flash memory devices. *Nanotechnology* 30:435206. doi: 10.1088/1361-6528/ab34da

Pani, D., Meloni, P., Tuveri, G., Palumbo, F., Massobrio, P., and Raffo, L. (2017). An FPGA platform for real-time simulation of spiking neuronal networks. *Front. Neurosci.* 11:90. doi: 10.3389/fnins.2017.00090

Pedretti, G., Milo, V., Ambrogio, S., Carboni, R., Bianchi, S., Calderoni, A., et al. (2017). Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. *Sci. Rep.* 7:5288. doi: 10.1038/s41598-017-05480-0

Pfeiffer, M., and Pfeil, T. (2018). Deep learning with spiking neurons: opportunities and challenges. *Front. Neurosci.* 12:774. doi: 10.3389/fnins.2018.00774

Prezioso, M., Mahmoodi, M. R., Bayat, F. M., Nili, H., Kim, H., Vincent, A., et al. (2018). Spike-timing-dependent plasticity learning of coincidence detection with passively integrated memristive circuits. *Nat. Commun.* 9:5311. doi: 10.1038/s41467-018-07757-y

Querlioz, D., Bichler, O., Dollfus, P., and Gamrat, C. (2013). Immunity to Device variations, in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* 12, 288–295. doi: 10.1109/tnano.2013.2250995

Querlioz, D., Dollfus, P., Bichler, O., and Gamrat, C. (2011). "Learning with memristive devices: how should we model their behavior?," in *Proceedings of the 2011 IEEE/ACM International Symposium on Nanoscale Architectures*, San Diego, CA: IEEE.

Riesenhuber, M., and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat, Neurosci.* 2:1019. doi: 10.1038/14819

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., and Liu, S.-C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* 11:682. doi: 10.3389/fnins.2017.00682

Shrestha, A., Ahmed, K., Wang, Y., Widemann, D. P., Moody, A. T., Essen, B. C. V., et al. (2018). Modular spiking neural circuits for mapping long short-term memory on a neurosynaptic processor. *IEEE J. Emerg. Select. Top. Circ. Syst.* 4, 782–795. doi: 10.1109/JETCAS.2018.2856117

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

Sun, X., and Yu, S. (2019). Impact of Non-ideal characteristics of resistive synaptic devices on implementing convolutional neural networks. *IEEE*

J. Emerg. Select. Top. Circ. Syst. 9, 570–579. doi: 10.1109/jetcas.2019.2933148

Tavanaei, A., and Maida, A. (2019). BP-STDP: approximating backpropagation using spike timing dependent plasticity. *Neurocomputing* 330, 39–47. doi: 10.1016/j.neucom.2018.11.014

Yang, S., Deng, B., Wang, J., Li, H., Lu, M., Che, Y., et al. (2020). Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. *IEEE Trans. Neural Learn. Syst.* 31, 148–162. doi: 10.1109/TNNLS.2019.2899936

Yang, S., Wang, J., Deng, B., Liu, C., Li, H., Fietkiewicz, C., et al. (2018). Real-time neuromorphic system for large-scale conductance-based spiking neural networks. *IEEE Trans. on Cyber.* 49, 2490–2502. doi: 10.1109/TCYB.2018.2823730

Yang, S., Wang, J., Li, S., Li, H., Wei, X., Yu, H., et al. (2016). Digital implementations of thalamocortical neuron models and its application in thalamocortical control using FPGA for Parkinson's disease. *Neurocomputing* 177, 274–289. doi: 10.1016/j.neucom.2015.11.026

Yu, S. (2018). Neuro-inspired computing with emerging nonvolatile memory. *Proc. IEEE* 106, 260–285. doi: 10.1109/JPROC.2018.2790840

Zhang, Q., Wu, H., Yao, P., Zhang, W., Gao, B., Deng, N., et al. (2018). Sign backpropagation: an on-chip learning algorithm for analog RRAM neuromorphic computing systems. *Neural Netw.* 108, 217–223. doi: 10.1016/j.neunet.2018.08.012

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Kwon, Lim, Bae, Lee, Kim, Seo, Oh, Kim, Yeom, Park and Lee. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 12 November 2020
doi: 10.3389/fnins.2020.586424

Blood Pressure Modulation With Low-Intensity Focused Ultrasound Stimulation to the Vagus Nerve: A Pilot Animal Study

*Ning Ji[1,2†], Wan-Hua Lin[1,3†], Fei Chen[3], Lisheng Xu[2], Jianping Huang[1] and Guanglin Li[1]**

[1] CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems and Research Center for Neural Engineering, Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS), and the SIAT Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China, [2] College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China, [3] Department of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen, China

OPEN ACCESS

Edited by:
Elisa Donati,
ETH Zurich, Switzerland

Reviewed by:
Soumyajit Mandal,
University of Florida, United States
Shinsuke Yasukawa,
Kyushu Institute of Technology, Japan

***Correspondence:**
Guanglin Li
gl.li@siat.ac.cn

[†] These authors have contributed equally to this work

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: 23 July 2020
Accepted: 06 October 2020
Published: 12 November 2020

Citation:
Ji N, Lin W-H, Chen F, Xu L, Huang J and Li G (2020) Blood Pressure Modulation With Low-Intensity Focused Ultrasound Stimulation to the Vagus Nerve: A Pilot Animal Study. Front. Neurosci. 14:586424. doi: 10.3389/fnins.2020.586424

Objective: For hypertensive individuals, their blood pressure (BP) is often managed by taking medications. However, antihypertensive drugs might cause adverse effects such as congestive heart failure and are ineffective in significant numbers of the hypertensive population. As an alternative method for hypertension management, non-drug devices-based neuromodulation approaches such as functional electrical stimulation (FES) have been proposed. The FES approach requires the implantation of a stimulator into the body. One recently emerging technique, called low-intensity focused ultrasound stimulation (FUS), has been proposed to non-invasively modulate neural activities. In this pilot study, the feasibility of adopting low-intensity FUS neuromodulation for BP regulation was investigated using animal models.

Methods: A FUS system was developed for BP modulation in rabbits. For each rabbit, the low-intensity FUS with different acoustic intensities was used to stimulate its exposed left vagus nerve, and the BP waveform was synchronously recorded in its right common carotid artery. The effects of the different FUS intensities on systolic blood pressure (SBP), diastolic blood pressure (DBP), mean blood pressure (MAP), and heart rate (HR) were extensively examined from the BP recordings.

Results: The results demonstrated that the proposed FUS method could successfully induce changes in SBP, DBP, MAP, and HR values. When increasing acoustic intensities, the values of SBP, DBP, and MAP would tend to decrease more substantially.

Conclusion: The findings of this study suggested that BP could be modulated through the FUS, which might provide a new way for non-invasive and non-drug management of hypertension.

Keywords: blood pressure management, low-intensity focused ultrasound stimulation, vagus nerve, neuromodulation, hypertension animal study

INTRODUCTION

High blood pressure (BP) is one of the leading causes of morbidity and mortality worldwide. Clinically, a common way for BP management is to regularly take antihypertensive medications for the hypertensive population. While a variety of antihypertensive drugs could effectively regulate BP with a primary goal to prevent the occurrence of cardiovascular and cerebrovascular complications such as stroke, a long-term medication of antihypertensive administration may cause some potential side effects such as congestive heart failure, depression, immunological disease, orthostatic symptoms, palpitations, precipitate angina, sexual dysfunction, and syncope (Husserl and Franz, 1981; Marc et al., 2019). Furthermore, some of the existing antihypertensive drugs have been reported to be associated with an increased risk of myocardial infarction (Psaty et al., 1995) as well as ischemic stroke (Klungel et al., 2001). On the other hand, for a substantial portion of patients with hypertension, their BP is uncontrolled by currently available antihypertensive drugs, which are designated as having resistant hypertension (Bisognano et al., 2011).

Owing to those potential side effects of taking antihypertensive medications regularly for a long time (even during the rest of life) and the issue of resistant hypertension, device-based non-drug neuromodulation approaches have been proposed and developed for the administration of resistant hypertension. This is because resistant hypertension is mainly a neurogenic disease characterized by enhanced sympathetic nerve activity. Thus, novel neuromodulation approaches targeting sympathetic nerve inhibition might be potential for the treatment of resistant hypertension. Tremendous evidences have proved that neuromodulation techniques such as functional electrical stimulation (FES) of the carotid baroreceptor (Scheffers et al., 2010; Bisognano et al., 2011; Lohmeier and Iliescu, 2011; Bakris et al., 2012; Hoppe et al., 2012)/vagus nerve (Plachta et al., 2014; Gierthmuehlen et al., 2016; Annoni et al., 2019) and renal sympathetic denervation (RSD) by different devices and techniques [including surgical sympathectomy (Smithwick, 1948), laparoscopic sympathectomy (Gao et al., 2019), catheter-based radiofrequency ablation (Krum et al., 2009), endovascular ultrasound (Fengler et al., 2019), injection of neurotoxic agents (Lohmeier and Hall, 2019), external stereotactic radiofrequency (Cai et al., 2019), external high-intensity focused ultrasound (Wang et al., 2013), etc.,] might reduce BP through sympathetic nerve activity inhibition. However, these procedures of current neuromodulation methods are either invasive or associated with complete nerve damage. The invasive surgery for the implant of FES stimulator would lead to some difficulties and/or adverse effect such as complicated surgical implantation and perioperative and post-surgery risks. In addition, dealing with the damaged implanted electrodes wrapped in the scar tissue remains unclear and difficult (Plachta et al., 2014). Besides that, despite that the catheter-based RSD procedures are minimally invasive and the procedures of performing external stereotactic radiofrequency or external high-intensity focused ultrasound for RSD are non-invasive, BP is reduced by completely destroying the renal sympathetic nerve by utilizing the high intensity of the radiofrequency/ultrasound energy.

By contrast with the electrical approaches (such as FES) and denervation methods (such as RSD), one emerging technology that is called low-intensity focused ultrasound stimulation (FUS) has been shown in a number of literatures to be promising in non-invasive neuromodulation without damaging the nerve (Baek et al., 2017; Landhuis, 2017); thus, it should be a potential approach for BP modulation. Specifically, the penetrability of ultrasound allows it to penetrate non-invasively from the body surface into a deep targeted nerve or tissue without a need of surgical implantation. The focused characteristics of ultrasound could ensure the precise stimulation of a targeted nerve or tissue. Furthermore, by setting different acoustic parameters such as intensity and frequency, neural activity could be selectively activated or inhibited without nerve damage. Owing to those advantages, FUS has opened a new era for non-invasive neuromodulation and has been recently applied in a number of studies in the field of neurosciences (Hakimova et al., 2015; Baek et al., 2017; Landhuis, 2017). By far, the FUS neuromodulation technique has been widely applied for brain stimulation (Hakimova et al., 2015; Baek et al., 2017; Landhuis, 2017). Furthermore, the FUS has also been used to target different peripheral nerves for neuromodulations. For example, targeting FUS at the retina could activate a visual-evoked potential equal to strong visual responses (Menz et al., 2013), at ear labyrinth it could cause auditory sensation corresponding to an audio-modulating signal (Tsirulnikov et al., 1988; Gavrilov and Tsirulnikov, 2012), at the peripheral sensory neuroreceptors or nerve fibers it could excite tactile, thermal, and pain sensations (Bystritsky et al., 2011; Gavrilov and Tsirulnikov, 2012; Legon et al., 2012), and at the sciatic nerve it could modulate motor neuron activity (Kim et al., 2020).

The successful and the promising applications of low-intensity FUS as described above for the neuromodulation of both the central and the peripheral nerves inspired us to consider the feasibility of utilizing FUS technique to stimulate the peripheral nerves (such as vagus nerve) for BP regulation. One previous animal study suggested that using focused pulsed ultrasound for vagus nerve modulation could induce the change of its compound action potential (CAP) (Juan et al., 2014). As it is well known, the BP value is regulated by the vagus nerve (Plachta et al., 2014; Annoni et al., 2019). Hence, we hypothesized that it might be feasible to control the BP by stimulating the vagus nerve *via* low-intensity FUS. In this pilot study, by using animals, we investigated whether BP could be effectively controlled through low-intensity FUS neuromodulation and explored how the different acoustic intensities would influence on the BP modulation as well as heart rate (HR). This study would be worth looking forward to provide an effective way for non-invasive and non-drug management of hypertension.

MATERIALS AND METHODS

In this study, the experiments of BP control through FUS modulation were conducted on eight white rabbits (six

New Zealand white rabbits and two Japanese white rabbits, all male, body weight 3.5–4.5 kg). For each rabbit, the BP modulation experiments included three sections: (1) animal preparation, (2) ultrasonic stimulation, and (3) BP data acquisition (as shown in **Figure 1**). All the animal experimental procedures were approved by the Institutional Animal Care and Use Committee (IACUC) of Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (SIAT-IACUC-190801-YGS-LWH-A0454-01). The details of the animal experiment are described in the following subsections.

Animal Preparation

For each rabbit, its left vagus nerve was chosen as the targeted nerve of BP modulation, and its right common carotid artery was selected as BP detecting site. Although the FUS can be applied non-invasively, a surgery was performed to expose the left vagus nerve to ensure that the focus of the ultrasound transducer precisely stimulates the targeted vagus nerve, and the right common carotid artery was also exposed by surgery to measure the BP values. Before the surgery commenced, a face mask was initially put on the face of the rabbits. The rabbits were induced to anesthetize using 5% isoflurane delivered with oxygen at a rate of 0.8 L/min, and then the anesthesia level was reduced to 2.5% isoflurane for maintenance. After that, the rabbits were placed on a platform in supine position, their neck hairs were shaved off with a razor, and then the surgical area was sterilized with alcohol. The underlying sternohyoid muscle was exposed through a ventral neck incision, and then the left vagus nerve and the right common carotid artery were exposed and separated from the neurovascular bundles, respectively. The exposed left vagus nerve was targeted with a low-intensity FUS probe for neuromodulation, and the exposed right common carotid artery was catheterized for continuous BP wave recording. During the experiment, 0.3% heparin sodium, an anticoagulant, was used to prevent blood coagulation when necessary to ensure that the experiment goes on smoothly. After successfully conducting the experiment, the rabbits were sacrificed with an overdose of isoflurane.

Ultrasonic Stimulation

Sonication Setup and Acoustic Measurement

An ultrasonic stimulation system was built using a function generator, a power amplifier, and a focused ultrasound transducer (shown in **Figure 1**). The driving signal from the functional generator (SDG 1032X, SIGLENT, Shenzhen, China) was amplified by a power amplifier (A075, E&I, Ltd., Rochester, NY, United States) and then sent to a focused ultrasound transducer. The focused ultrasound transducer with a fundamental frequency (FF) of 3.7 MHz, a diameter of 19.5 mm, and a focal length of 17 mm was connected to an acoustic collimator. The collimator was designed based on the characteristics of the ultrasound transducer and was fabricated with a three-dimensional (3D) printer, which was used to precisely focus the ultrasound on the stimulation target. During the experiment, the collimator was filled with ultrasound gel for better acoustic coupling.

Using the FUS with different acoustic intensities to stimulate a nerve may cause different biological responses. In this study, two kinds of acoustic intensities, spatial-peak pulse-average intensity (I_{sppa}) and spatial-peak time-average intensity (I_{spta}), were examined to explore their influences on BP modulation. The I_{sppa} and the I_{spta} represent the degree of acoustic pressure given by the driving voltage and the energy deposition rate in

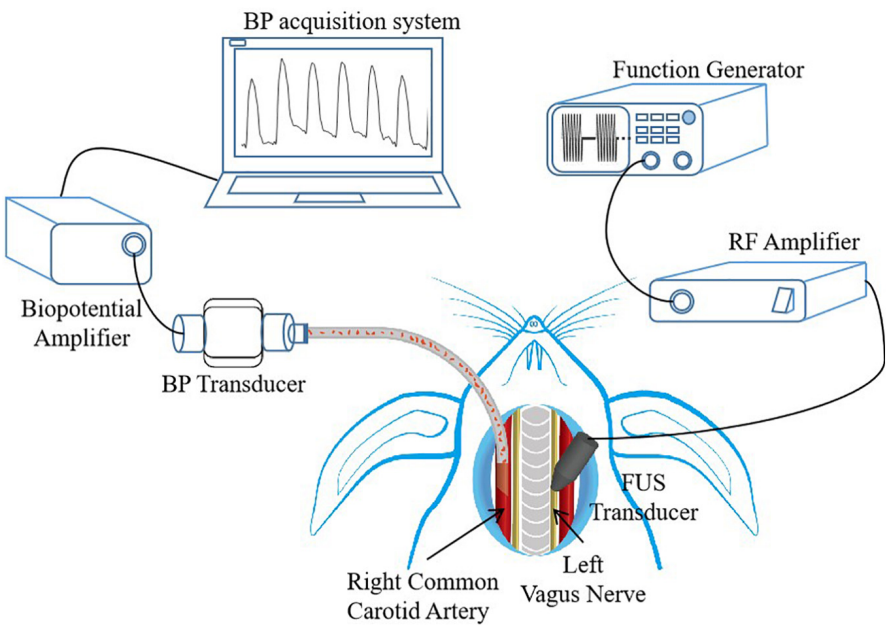


FIGURE 1 | Experimental setup, with description of animal preparation, focused ultrasound stimulation (FUS) control, and data acquisition system. The placements of FUS transducer and blood pressure (BP) transducer are shown. RF, radio frequency.

TABLE 1 | Sonication parameters used in the stimulation trials with fixed fundamental frequency (3.7 MHz), sonication duration (5 s), and inter-stimulus interval (1 s).

AI: I_{sppa} (W/cm^2)	Duty cycle (%)	Tone-burst duration (ms)	AI: I_{spta} (W/cm^2)
18.0	30	0.3	5.4
48.5	30	0.3	14.6
87.3	30	0.3	26.2
18.0	50	0.5	9.0
48.5	50	0.5	24.3
87.3	50	0.5	43.6
18.0	70	0.7	12.6
48.5	70	0.7	34.0
87.3	70	0.7	61.1
18.0	100	–	18.0
48.5	100	–	48.5
87.3	100	–	87.3

the tissue, respectively. The I_{sppa} can be theoretically calculated according to the American Institute of Ultrasound Medicine standards (NEMA, 2004) using the following equation (1):

$$I_{\text{sppa}} = \frac{P_0^2}{2\rho c} \quad (1)$$

where P_0 is the acoustic peak pressure, ρ is the density of the medium (1,000 kg/m^3), and c is the sound speed in the medium (1,480 m/s). Before the experiments, three sets of I_{sppa} parameters (shown in **Table 1**) were determined, and the acoustic pressure fields in the focal region generated by the ultrasonic stimulation system were practically measured using a 3D acoustic scanning system (UMS3, Precision Acoustics, Dorchester, United Kingdom) equipped with a calibrated needle-type hydrophone (HNP-0400, Onda, Sunnyvale, CA, United States). **Figure 2A** shows a typical example of acoustic pressure distributions in the axial plane (X–Z section) with a 0.5-mm step at the focus position (Y-axis).

The I_{spta} can be calculated using the equation (2):

$$I_{\text{spta}} = \text{DC} \times I_{\text{sppa}} \quad (2)$$

where DC represents the ultrasound duty cycle, a percentage ratio of sonication active time to a total period. Thus, the I_{spta} would be determined by both acoustic peak pressure and time-related parameters (DC). During the experiment, the I_{spta} values ranged from 5.40 to 87.3 W/cm^2 by setting the DC values to be 30, 50, 70, and 100%, with different I_{sppa} values (18.0, 48.5, and 87.3 W/cm^2), as shown in **Table 1**.

Ultrasound Stimulation Trials

The FUS trials were conducted by setting parameters of the function generator to examine the effects of different acoustic intensities on the BP regulation. One channel of the function generator was used to control the ultrasound FF and tone-burst duration (TBD) and was triggered by another channel of the function generator which was used to generate the bursts of sinusoidal pulse waves and control the pulse repetition frequency (PRF), sonication duration (SD), and inter-stimulus interval (ISI), as shown in **Figure 2B**. The duty cycle (DC) equals to TBD divided by 1/PRF. Note that the FUS with a DC of 100% represents continuous stimulation and that with a DC less than 100% represents the pulsed stimulation.

In this study, the FF, PRF, SD, and ISI were fixed and set to 3.7 MHz, 1 kHz, 5 s, and 1 s, respectively, and the TBD was changed with different values to obtain different DC and I_{spta}, as listed in **Table 1**. For each trial with a set of predefined sonication parameters, the FUS duration lasted for about 20 s to clearly observe the BP changes, and then we waited for around 60 s after cessation of the stimulation until the BP returned to the baseline level. The sequence of sonication trials was pseudo-randomized and balanced across the animals. Repeated ultrasound stimulation trials on the same animal were conducted to ensure the effectiveness of the different sonication parameter sets.

Data Acquisition

During the FUS experiments, the BP waveform was continuously recorded in the exposed right common carotid artery with a commercially available data acquisition system (ADInstruments Pty Ltd., Bella Vista, NSW, Australia). The sampling rate was set

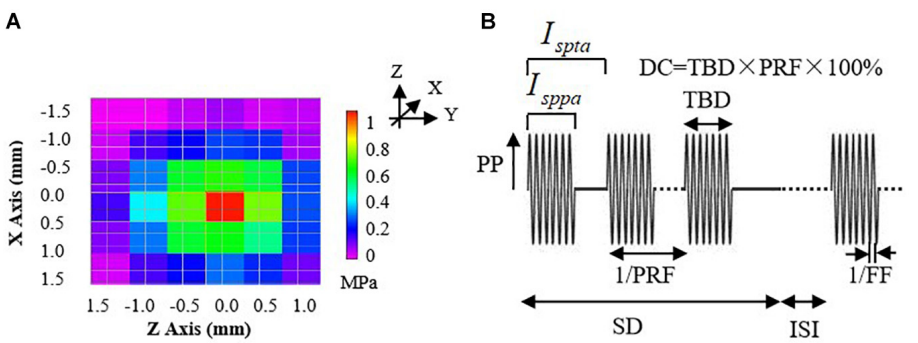


FIGURE 2 | (A) The acoustic pressure distributions of the 3.7-MHz FUS transducer in the axial plane (X–Z section) with 0.5-mm steps at the focus position (Y-axis). **(B)** An illustration of parameters for a typical pulsed sonication: PP, peak pressure; I_{sppa}, spatial-peak pulse-average intensity; I_{spta}, spatial-peak time-average intensity; DC, duty cycle; TBD, tone-burst duration; PRF, pulse repetition frequency; SD, sonication duration; ISI, inter-stimulus interval; FF, fundamental frequency.

as 1,000 Hz. Four important cardiovascular parameters, beat-to-beat systolic BP (SBP), diastolic BP (DBP), mean arterial pressure (MAP), as well as heart rate (HR), were calculated from the BP recordings by using a commercial software (LabChart toolbox, ADInstruments). SBP and DBP are defined as the amplitude of the peak and the trough of BP waveform, respectively. MAP represents an average blood pressure within a single cardiac cycle, which could be calculated using equation (3). HR is defined as the number of heartbeats in a minute and herein is calculated by equation (4).

$$\mathrm{MAP} = \frac{1}{3}\mathrm{SBP} + \frac{2}{3}\mathrm{DBP} \tag{3}$$

$$\mathrm{HR} = 60/\mathrm{IBI} \tag{4}$$

where IBI indicates inter-beat interval.

Statistical Analysis

A paired *t*-test was conducted to assess the significant difference of the changes in arterial BP and HR when the FUS was turned off and on. A correlation analysis measured with Pearson coefficient was performed to estimate the relationship between the BP control and each of the ultrasonic stimulation parameters, as well as the relevance between the BP changes and the HR changes in response to the FUS. The analysis results were expressed as mean ± standard deviation (SD). A value of $p < 0.05$ was considered to be statistically significant. All statistical tests were conducted using the SPSS software package for data analysis.

RESULTS

BP Modulations With a Low-Intensity FUS to the Vagus Nerve

Figure 3A shows a representative segment of the continuous BP waveform recordings during the FUS to the left vagus nerve of a rabbit with an incremental acoustic intensity (18.0, 48.5, and 87.3 W/cm^2 I_{sppa}), in which the FUS durations were indicated by red bars. It was clearly observed from **Figure 3A** that the arterial BP waveform gradually decreased from the baseline level when the FUS was turned on, and the BP waveform slowly returned back to the baseline level when the FUS was turned off. Similar changes in the BP waveform characteristics were observed when the FUS was conducted repeatedly. With the increase of acoustic intensity (from 18.0 to 87.3 W/cm^2 I_{sppa}), the BP value decreased more substantially. To further illustrate this phenomenon in a clearer manner, a zoom-in of the BP waveform under the FUS at 87.3 W/cm^2 I_{sppa} is shown in **Figure 3B**.

Figure 4 shows the overall changes of SBP, DBP, MAP, and HR of eight rabbits when applying FUS on the left vagus nerve at 3.7 MHz FF, 1 kHz PRF, 5 s SD, and 18.0–87.3 W/cm^2 I_{sppa}. The white boxes indicated the values before stimulation, and the gray boxes demonstrated the values recorded during the stimulation. As shown in the boxplots, the mean values of SBP decreased by 1.91, 9.71, and 10.09 mmHg when applying FUS at 18.0, 48.5, and 87.3 W/cm^2 I_{sppa}, respectively. Meanwhile, the mean values of DBP decreased by 3.01, 13.75, and 14.55 mmHg, MAP decreased by 2.64, 12.41, and 13.06 mmHg, and HR decreased by 7.02, 19.21, and 24.90 bpm when applying FUS at 18.0, 48.5, and 87.3 W/cm^2

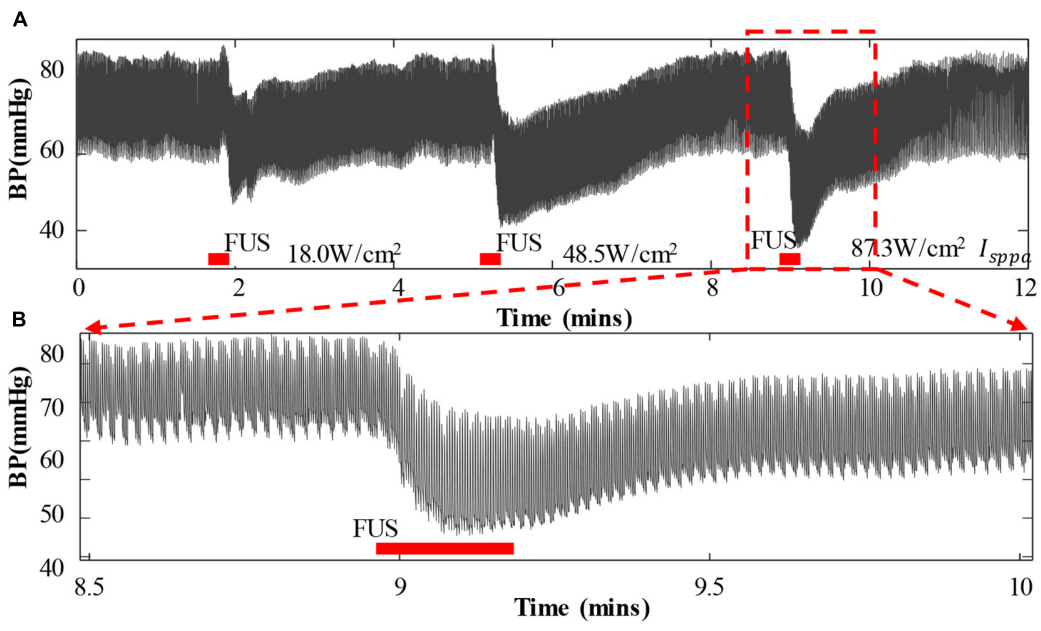


FIGURE 3 | Arterial blood pressure (ABP) and heart rate change when applying focused ultrasound stimulation (FUS) to the vagus nerve. **(A)** A typical recording of the original ABP waveform of a white rabbit responding to the FUS with increasing acoustic intensities (18.0, 48.5, and 87.3 W/cm^2 I_{sppa}). ABP decreased when the FUS was turned on. **(B)** A zoom-in view of ABP waveform under the FUS at 87.3 W/cm^2 I_{sppa}. The red bar indicates the period of stimulation.

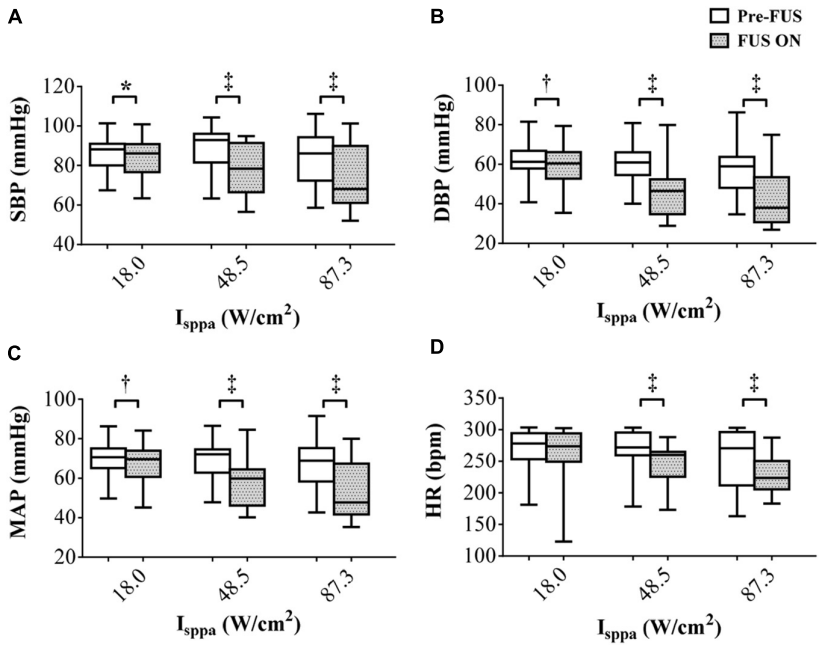


FIGURE 4 | The overall changes (n = 8) in **(A)** systolic blood pressure, **(B)** diastolic blood pressure, **(C)** mean arterial pressure, and **(D)** heart rate responding to focused ultrasound stimulation. The white boxes indicate the values before stimulation, and the gray boxes indicate the values during the stimulation. *$p < 0.05$, †$p < 0.01$, ‡$p < 0.001$.

I_{sppa}, respectively. A paired t test showed that the decrease in SBP, DBP, and MAP was significant when applying FUS at 18.0, 48.5, and 87.3 W/cm^2 I_{sppa} (**Figures 4A–C**). HR was significantly decreased at 48.5 and 87.3 W/cm^2 I_{sppa} (**Figure 4D**). Thus, FUS on the left vagus nerve at 3.7 MHz FF, 1 kHz PRF, 5 s SD, 1 s ISI, and 18.0–87.3 W/cm^2 I_{sppa} significantly decreased SBP, DBP, and MAP. The effectiveness and the repeatability of the SBP, DBP, MAP, and HR reductions in response to the FUS on the left vagus nerve were validated by multiple stimulation trials under different stimulation parameters.

Effect of Acoustic Intensity on BP Modulations

I_{spta}

Figure 5 shows the reduction percentage of SBP, DBP, and MAP as I_{spta} increases when applying FUS on the left vagus nerve at 3.7 MHz FF, 1 kHz PRF, and 5 s SD. On average, decreases of 0.92–19.33% in SBP, 2.04–36.49% in DBP, and 1.61–29.36% in MAP were recorded when I_{spta} increased from 5.40 to 87.30 W/cm^2. Furthermore, the decrease in SBP, DBP, and MAP was significantly correlated with I_{spta} ($r = 0.55$, $p < 0.01$ for SBP reduction; $r = 0.62$, $p < 0.01$ for DBP reduction; $r = 0.61$, $p < 0.01$ for MAP reduction). Hence, the results suggested that the BP modulations had a significant correlation with acoustic intensity (I_{spta}) when applying FUS on the left vagus nerve. The SBP, DBP, and MAP reduction tends to increase more substantially with a higher I_{spta}. However, it is also worthy to note that the SBP, DBP, and MAP reductions are not completely monotonically increasing as I_{spta} increases. Note that I_{spta} is composed of stimuli with different DC and I_{sppa} combinations, and the DC and I_{sppa} might contribute to the BP reduction differently, which might explain the non-monotonicity between I_{spta} increase and BP reduction. Therefore, the respective effects of I_{sppa} and DC on BP modulations are presented in the following discussion.

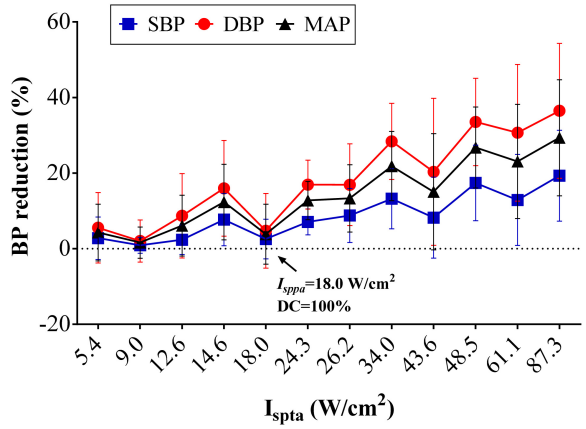


FIGURE 5 | Effects of acoustic intensity (AI) on blood pressure (BP). The overall BP changes (n = 8) in percentage relative to the baseline level in the different AIs (I_{spta}) with the set 3.7-MHz fundamental frequency, 1-kHz pulse repetition frequency, and 5-s sonication duration. Reductions in systolic blood pressure, diastolic blood pressure, and mean arterial pressure increased with AI.

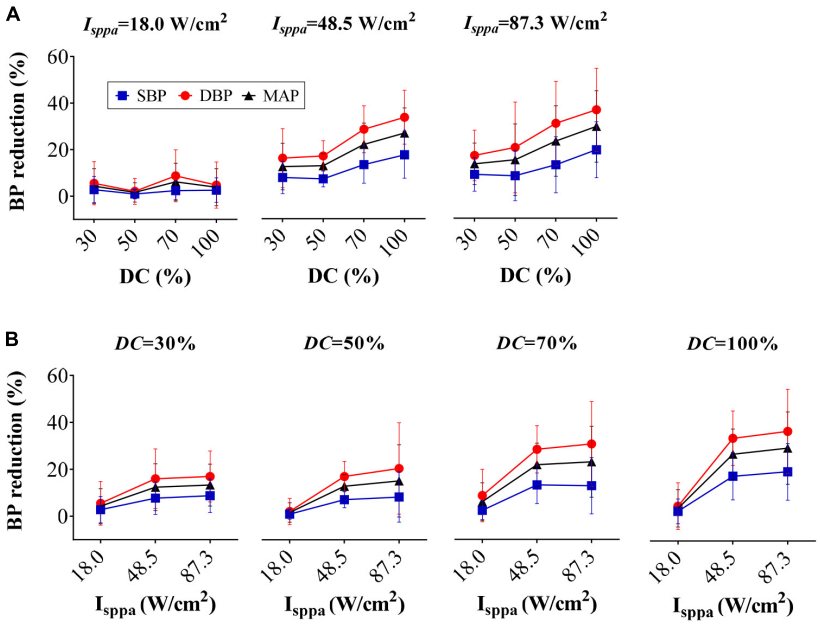


FIGURE 6 | Reduction percentage of systolic blood pressure, diastolic blood pressure, and mean arterial pressure as **(A)** duty cycle increases and **(B)** I_{sppa} increases.

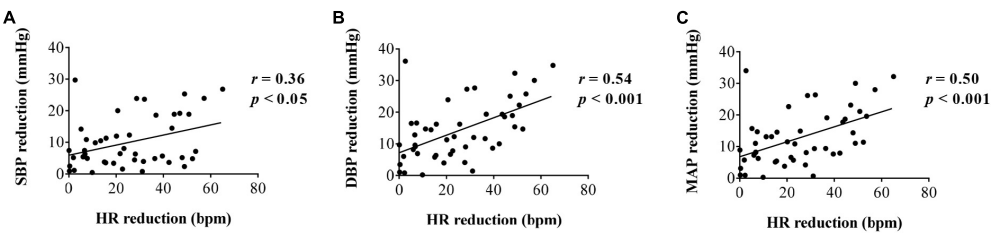


FIGURE 7 | Correlation analysis between heart rate reduction and **(A)** systolic blood pressure reduction, **(B)** diastolic blood pressure reduction, and **(C)** mean arterial pressure reduction, respectively.

I_{sppa} and DC

Figure 6A shows the reduction percentage of SBP, DBP, and MAP as DC increases when applying FUS with I_{sppa} fixed at 18.0, 48.5, and 87.3 W/cm^2, respectively. As shown in the figure, BP reduction increased monotonically with the rise of DC when the I_{sppa} was fixed. A correlation analysis further confirmed that the magnitude of SBP, DBP, and MAP reduction was also significantly correlated with DC (r = 0.29, $p < 0.05$ for SBP reduction; r = 0.31, $p < 0.01$ for DBP reduction; r = 0.31, $p < 0.01$ for MAP reduction). **Figure 6B** shows the reduction percentage of SBP, DBP, and MAP as I_{sppa} increases when applying FUS with DC fixed at 30, 50, 70, and 100%, respectively. In general, when the DC was fixed, the SBP, DBP, and MAP reductions increase monotonically with the rise of I_{sppa}. Furthermore, the magnitude of SBP, DBP, and MAP reduction was also significantly correlated with I_{sppa} (r = 0.42, $p < 0.01$ for SBP reduction; r = 0.49, $p < 0.01$ for DBP reduction; r = 0.48, $p < 0.01$ for MAP reduction). Thus, both I_{sppa} and DC could affect the magnitude of BP reduction, which might explain the non-monotonic relationship between I_{spta} and BP reduction (as shown in **Figure 5**). As labeled in **Figure 5**, FUS with I_{spta} of 18.0 W/cm^2 produced a relatively lower BP induction than the adjacent points; this might be because of the small I_{sppa} (18.0 W/cm^2), leading to a smaller BP decline.

Effect of HR Changes on BP Modulations

In order to investigate the effect of HR changes on BP modulation, a correlation analysis between the changes of HR and changes of BP was conducted, and the results are shown in **Figure 7**. As can be seen from the figure, the decrease in SBP, DBP, and MAP was significantly correlated with the decrease in HR (r = 0.36, $p < 0.05$ for SBP reduction; r = 0.54, $p < 0.001$ for DBP reduction; r = 0.50, $p < 0.001$ for MAP reduction). It demonstrates that the HR changes have an important effect on BP regulation. However, the correlation coefficients are moderate (0.36–0.54, the maximum scope is [0 1]), which demonstrates that the changes of BP elicited by FUS are not completely caused by HR changes. There are other factors that might influence the changes in BP.

DISCUSSION AND CONCLUSION

Clinically, the first-line treatment for patients with hypertension would be pharmacological antihypertensive therapy. Currently, while there are a plethora of available antihypertensive drugs that have been used to effectively treat hypertension, the drug safety issue has been sustainedly concerned by the physicians and the patients (Husserl and Franz, 1981; Psaty et al., 1995; Klungel et al., 2001; Marc et al., 2019). In addition, the need of administrating BP is unmet in the resistant hypertensive population. Therefore, developing devices-based neuromodulation approaches would be one alternative way to treat hypertension and its comorbidities (Scheffers et al., 2010; Bisognano et al., 2011; Lohmeier and Iliescu, 2011; Bakris et al., 2012; Hoppe et al., 2012). Recently, a new advanced neuromodulation technology that uses focused ultrasound stimulation to suppress or boost neurons' activity has been approved to be promising to treat some neuropathies such as movement disorders, depression, and anxiety (Bystritsky et al., 2011; Legon et al., 2012; Lohmeier and Hall, 2019). Whether the newly approved neuromodulation technique would be feasible and effective for BP modulation was investigated in this animal study.

In this pilot study, the acute response of BP under low-intensity FUS to the vagus nerve of rabbits was investigated. The experimental results indicated that BP could be effectively modulated through low-intensity FUS technique when appropriate sonication parameters were set, to the best of our knowledge, which should be the first time to demonstrate the feasibility of using FUS on peripheral nerve for BP neuromodulation. When FUS was targeted at the vagus nerve with the sonication parameters of 3.7 MHz FF and 18.0–87.3 W/cm^2 I_{sppa}, the values of the SBP, MAP, and DBP were observed to be significantly reduced. Meanwhile, the acoustic intensities of the FUS had a significant effect on the degree of BP variation, such that the higher the acoustic intensity, the more substantial reduction in the values of the SBP, DBP, and MAP. In addition, the HR also decreased during the FUS period, which indicates that the FUS at the vagus nerve may modulate BP through regulating cardiac function and peripheral vascular function. The correlation analysis between BP and HR produced intermediate values (r = 0.36 for ΔSBP, r = 0.54 for ΔDBP, r = 0.50 for ΔMAP), demonstrating that the reduction of BP was not only induced by the decline of HR. The decrease of HR when the FUS was utilized also suggested that low-frequency FUS of vagus nerve plays a role on the relief of tachycardia but should avoid bradycardia.

It is worth noting that, compared to the FES-based neuromodulation approach that has been proposed for hypertension managements in clinical trials (Hoppe et al., 2012), the acute response of BP to the proposed FUS approach was found to be similar as that of the FES reported in a previous study (Plachta et al., 2014). It was also found that the values of the SBP, DBP, and MAP reduced substantially when the focused ultrasound stimulation was turned on and then returned back to the baseline level when the stimulation was turned off. BP response exhibited similar waveform characteristics when a repeated stimulation was administered during the experiment (as illustrated in **Figure 3**). This suggested that the FUS approach could be promising as an alternative non-drug treatment method for hypertension, similar to the FES. In comparison with applying the FES for BP regulation, which requires the implantation of the stimulator into the targeted nerve by invasive surgery (Scheffers et al., 2010; Bisognano et al., 2011; Lohmeier and Iliescu, 2011; Bakris et al., 2012; Hoppe et al., 2012; Plachta et al., 2014; Gierthmuehlen et al., 2016; Annoni et al., 2019), ultrasound energy could penetrate into the deep tissue in a non-invasive way, which has been proven in a lot of previous studies (Bystritsky et al., 2011; Gavrilov and Tsirulnikov, 2012; Baek et al., 2017), that may make the proposed FUS approach outperform the existing device-based methods for BP regulation. In addition, unlike the high-intensity focused ultrasound stimulation used to ablate the renal sympathetic

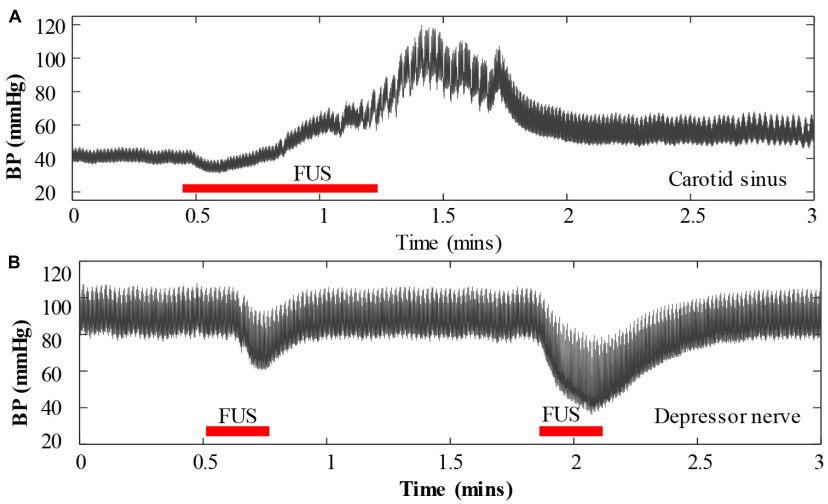


FIGURE 8 | An example of the response of blood pressure when applying focused ultrasound stimulation to a carotid sinus nerve **(A)** and to a depressor nerve **(B)**.

nerve for drug-resistant hypertension treatment (Wang et al., 2013), FUS could induce an antihypertensive effect without damaging the nerve or tissues surrounding it. Therefore, FUS might provide a better way for non-invasive and non-drug management of hypertension. Although the FUS can be applied non-invasively, which has been proven to be feasible in lots of previous studies (Baek et al., 2017; Landhuis, 2017), in this pilot study, it is noteworthy that, in order to identify the exact targeted nerve that could induce an obvious antihypertensive effect through FUS, an invasive animal surgery model was used to expose the target nerve, and the ultrasound energy was directly focused on the target nerve using a collimator fully filled with a coupling agent. After identifying the target nerves specifically, future works would test the effect of BP control by non-invasive stimulation from the body surface. Whereas delivering the FUS with exact alignment to the target nerve is a challenging issue, image-guided technique may provide a promising option to address the alignment issue, which has been proven to be feasible in some previous studies (Kim et al., 2020).

The vagus nerve was initially selected in this pilot study because BP is more responsive to changes in vagus nerve activity than those in other nerves, and the vagus nerve is relatively easy to locate. However, other autonomic nerves such as sympathetic nerve and stretch-sensitive nerves such as carotid sinus nerve and depressor nerve may also be used as a targeted nerve for BP modulation with FUS method. In this study, we also preliminarily investigated the response of BP when applying FUS to the carotid sinus nerve and to the depressor nerve, respectively. However, the experimental results showed that the sonication parameters used in this study increased BP rather than decreased it when FUS was targeted at the carotid sinus nerve, and when FUS was targeted at the depressor nerve, it could reduce BP similar to that of targeting the vagus nerve, but the BP decrease induced by the depressor nerve stimulation was not as significant as that induced by the vagus nerve stimulation. This phenomenon was also similar as electrical stimulation (Douglas and Ritchie, 1956). An example of the response of BP when applying FUS to a carotid sinus and to a depressor nerve is shown in **Figures 8A,B**, respectively. The intensity (I_{spta}) of the FUS used in **Figure 8A** was 34 W/cm^2 and in **Figure 8B** were 14.6 and 26.2 W/cm^2, respectively. Future works will explore the appropriate sonication parameters that could induce an antihypertensive effect when FUS was targeted to the carotid sinus nerve, which may further broaden the window of understanding on the potential applications of FUS for BP regulations in clinical practice.

Noting that FF of ultrasound is an important parameter for FUS, according to previous literatures (Bystritsky et al., 2011; Gavrilov and Tsirulnikov, 2012; Baek et al., 2017), various FF parameters have been adopted for FUS neuromodulation. Generally, low-frequency ultrasound with FFs of less than 1 MHz was mostly utilized, while some studies adopted relatively high FFs such as 1.68, 1.9, 2.5, 2.7, 3.2, 3.5, 4.6, 5, 2–7, and 8 MHz for FUS neuromodulation. In order to learn the response of BP induced by FUS at different frequencies, three ultrasonic transducers with FF values of 548 kHz, 1.05 MHz, and 3.7 MHz were tested in this preliminary experiment, respectively. Our results showed that the 548-kHz ultrasonic transducer induced little BP response, the 1.05-MHz ultrasonic transducer induced a slight BP response, and the 3.7-MHz ultrasonic transducer induced a significant BP response. Thus, a 3.7-MHz ultrasonic transducer was used in this pilot study, and its effect on BP modulation was systematically investigated. Other ultrasonic parameters, such as PRF, SD, and ISI, were also chosen based on previous literatures (Bystritsky et al., 2011; Gavrilov and Tsirulnikov, 2012; Baek et al., 2017) and preliminary experimental results.

In summary, for the first time, to the best of our knowledge, this pilot animal study provided the evidence of acute response of BP by low-intensity FUS. The experimental data indicated that BP could be effectively modulated through low-intensity FUS of the vagus nerve when the sonication parameters were appropriately determined. The acute response of BP to low-intensity FUS was similar as that of electrical stimulation, which indicates that this new approach may provide an alternative way for non-invasive and non-drug management of hypertension and other diseases associated with vagal activity modulation. However, the long-term antihypertensive effect still needs to be verified by chronic stimulation. In the future, the influence of additional sonication parameters such as FF, PRF, SD, and ISI on short-term and long-term BP attenuation/regulation as well as its corresponding thermal effect would be further investigated. In addition, the mechanism by which FUS induces BP response remains to be further investigated.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

ETHICS STATEMENT

The animal study was reviewed and approved by the Institutional Animal Care and Use Committee (IACUC) of Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences.

AUTHOR CONTRIBUTIONS

GL initiated and supported this project and checked through all the writing. FC and LX participated in the design of the study and provided technical supports. W-HL and NJ carried out the clinical examination, participated in the work of analysis, and drafted the manuscript. JH carried out the clinical examination and provided clinical support. All the authors read and approved the final manuscript.

FUNDING

This work was partially supported by the National Key R&D Program of China (2019YFC1710400 and 2019YFC1710402), the National Natural Science Foundation of China (under grants

#U1613222 and #61901461), the China Postdoctoral Science Foundation (#2020M672701), the Science and Technology Planning Project of Guangdong Province (under grant #2019B090915002), and the Fundamental Research Funds for the Central Universities (#N181906001).

ACKNOWLEDGMENTS

The authors would like to thank Ms. Na Pang for configuring the ultrasound stimulation system and Dr. Oluwarotimi Williams Samuel for editing and proofreading the manuscript.

REFERENCES

Annoni, E. M., Van Helden, D., Guo, Y., Levac, B., Libbus, I., KenKnight, B. H., et al. (2019). Chronic low-level vagus nerve stimulation improves long-term survival in salt-sensitive hypertensive rats. *Front. Physiol.* 10:25. doi: 10.3389/fphys.2019.00025

Baek, H., Pahk, K. J., and Kim, H. (2017). A review of low-intensity focused ultrasound for neuromodulation. *Biomed. Eng. Lett.* 7, 135–142. doi: 10.1007/s13534-016-0007-y

Bakris, G. L., Nadim, M. K., Haller, H., Lovett, E. G., Schafer, J. E., and Bisognano, J. D. (2012). Baroreflex activation therapy provides durable benefit in patients with resistant hypertension: results of long-term follow-up in the rheos pivotal trial. *J. Am. Soc. Hypertens.* 6, 152–158. doi: 10.1016/j.jash.2012.01.003

Bisognano, J. D., Bakris, G., Nadim, M. K., Sanchez, L., Kroon, A. A., Schafer, J., et al. (2011). Baroreflex activation therapy lowers blood pressure in patients with resistant hypertension: results from the double-blind, randomized, placebo-controlled rheos pivotal trial. *J. Am. Coll. Cardiol.* 58, 765–773.

Bystritsky, A., Korb, A. S., Douglas, P. K., Cohen, M. S., Melega, W. P., Mulgaonkar, A. P., et al. (2011). A review of low-intensity focused ultrasound pulsation. *Brain Stimul.* 4, 125–136. doi: 10.1016/j.brs.2011.03.007

Cai, X., Yang, Y., Shen, Y., Wang, W., and Li, Y. G. (2019). Noninvasive stereotactic radiotherapy for renal denervation in a swine model. *J. Am. Coll. Cardiol.* 74, 1697–1709. doi: 10.1016/j.jacc.2019.07.053

Douglas, W. W., and Ritchie, J. M. (1956). Cardiovascular reflexes produced by electrical excitation of non-medullated afferents in the vagus, carotid sinus and aortic nerves. *J. Physiol. Lond.* 134, 167–178. doi: 10.1113/jphysiol.1956.sp005632

Fengler, K., Rommel, K. P., Blazek, S., Besler, C., Hartung, P., Roeder, M. V., et al. (2019). A three-arm randomized trial of different renal denervation devices and techniques in patients with resistant hypertension. *Circulation* 139, 590–600. doi: 10.1161/circulationaha.118.037654

Gao, C., Zhao, L., Zhu, L., Li, M., Ding, D., Liu, Z., et al. (2019). Laparoscopic-based perivascular unilateral renal sympathetic nerve denervation for treating resistant hypertension a case report. *Hypertension Res.* 42, 1162–1165. doi: 10.1038/s41440-019-0237-3

Gavrilov, L. R., and Tsirulnikov, E. M. (2012). Focused ultrasound as a tool to input sensory information to humans (review). *Acoust. Phys.* 58, 1–21. doi: 10.1134/s1063771012010083

Gierthmuehlen, M., Stieglitz, T., Zentner, J., and Plachta, D. T. (2016). Haemodynamic responses to selective vagus nerve stimulation under enalapril medication in rats. *PLoS One* 11:e0147045. doi: 10.1371/journal.pone.0147045

Hakimova, H., Kim, S., Chu, K., Lee, S. K., Jeong, B., and Jeon, D. (2015). Ultrasound stimulation inhibits recurrent seizures and improves behavioral outcome in an experimental model of mesial temporal lobe epilepsy. *Epilepsy Behav.* 49, 26–32. doi: 10.1016/j.yebeh.2015.04.008

Hoppe, U. C., Brandt, M. C., Wachter, R., Beige, J., Rump, L. C., Kroon, A. A., et al. (2012). Minimally invasive system for baroreflex activation therapy chronically lowers blood pressure with pacemaker-like safety profile: results from the barostim neo trial. *J. Am. Soc. Hypertens.* 6, 270–276. doi: 10.1016/j.jash.2012.04.004

Husserl, F. E., and Franz, H. M. (1981). Adverse effects of antihypertensive drugs. *Drugs* 22, 188–210. doi: 10.2165/00003495-198122030-00002

Juan, E. J., Gonzalez, R., Albors, G., Ward, M. P., and Irazoqui, P. (2014). Vagus nerve modulation using focused pulsed ultrasound: potential applications and preliminary observations in a rat. *Int. J. Imaging Syst. Technol.* 24, 67–71. doi: 10.1002/ima.22080

Kim, M. G., Kamimura, H. A. S., Lee, S. A., Aurup, C., Kwon, N., and Konofagou, E. E. (2020). Image-guided focused ultrasound modulates electrically evoked motor neuronal activity in the mouse peripheral nervous system in vivo. *J. Neural Eng.* 17:026026. doi: 10.1088/1741-2552/ab6be6

Klungel, O. H., Heckbert, S. R., and Longstreth, W. T. (2001). Antihypertensive drug therapies and the risk of ischemic stroke. *Arch. Intern. Med.* 161, 37–43. doi: 10.1001/archinte.161.1.37

Krum, H., Schlaich, M., Whitbourn, R., Aobotka, P. A., Sadowski, J., Bartus, K., et al. (2009). Catheter-based renal sympathetic denervation for resistant hypertension: a multicentre safety and proof-of-principle cohort study. *Lancet* 373, 1275–1281.

Landhuis, E. (2017). Ultrasound for the brain. *Nature* 551, 257–259.

Legon, W., Rowlands, A., Opitz, A., Sato, T. F., and Tyler, W. J. (2012). Pulsed ultrasound differentially stimulates somatosensory circuits in humans as indicated by EEG and fMRI. *PLoS One* 7:e51177. doi: 10.1371/journal.pone.0051177

Lohmeier, T. E., and Hall, J. E. (2019). Device-based neuromodulation for resistant hypertension therapy. *Circ. Res.* 124, 1071–1093. doi: 10.1161/circresaha.118.313221

Lohmeier, T. E., and Iliescu, R. (2011). Chronic lowering of blood pressure by carotid baroreflex activation: mechanisms and potential for hypertension therapy. *Hypertension* 57, 880–886. doi: 10.1161/hypertensionaha.108.119859

Marc, A. S., Martijn, J. S., Harlan, M. K., Seng, C. Y., Chen, R. J., Nicole, P., et al. (2019). Comprehensive comparative effectiveness and safety of first-line antihypertensive drug classes: a systematic, multinational, large-scale analysis. *Lancet* 39, 32317–32317.

Menz, M. D., Oralkan, O., Khuri-Yakub, P. T., and Baccus, S. A. (2013). Precise neural stimulation in the retina using focused ultrasound. *J. Neurosci.* 33, 4550–4560. doi: 10.1523/jneurosci.3521-12.2013

NEMA (2004). *Acoustic Output Measurement Standard for Diagnostic Ultrasound Equipment*. Washinton, DC: National Electrical Manufactures Association.

Plachta, D. T., Gierthmuehlen, M., Cota, O., Espinosa, N., Boeser, F., Herrera, T. C., et al. (2014). Blood pressure control with selective vagus nerve stimulation and minimal side effects. *J. Neural Eng.* 11:036011. doi: 10.1088/1741-2560/11/3/036011

Psaty, B. M., Heckbert, S. R., Koepsell, T. D., Siscovick, D. S., Raghunathan, T. E., Weiss, N. S., et al. (1995). The risk of myocardial infarction associated with antihypertensive drug therapies. *J. Am. Med. Assoc.* 274, 620–625.

Scheffers, I. J., Kroon, A. A., Schmidli, J., Jordan, J., Tordoir, J. J., Mohaupt, M. G., et al. (2010). Novel baroreflex activation therapy in resistant hypertension: results of a european multi-center feasibility study. *J. Am. Coll. Cardiol.* 56, 1254–1258.

Smithwick, R. H. (1948). Surgical treatment of hypertension. *Am. J. Med.* 4, 744–759.

Tsirulnikov, E. M., Vartanyan, I. A., Gersuni, G. V., Rosenblyum, A. S., Pudov, V. I., and Gavrilov, L. R. (1988). Use of amplitude-modulated focused ultrasound for diagnosis of hearing disorders. *Ultrasound Med. Biol.* 14, 277–285. doi: 10.1016/0301-5629(88)90093-2

Wang, Q., Guo, R., Rong, S., Yang, G., Zhu, Q., Jiang, Y., et al. (2013). Noninvasive renal sympathetic denervation by extracorporeal high-intensity focused ultrasound in a pre-clinical canine model. *J. Am. Coll. Cardiol.* 61, 2185–2192. doi: 10.1016/j.jacc.2013.02.050

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Ji, Lin, Chen, Xu, Huang and Li. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 17 November 2020
doi: 10.3389/fnins.2020.534619

Toward New Modalities in VEP-Based BCI Applications Using Dynamical Stimuli: Introducing Quasi-Periodic and Chaotic VEP-Based BCI

Zahra Shirzhiyan[1,2,3], Ahmadreza Keihani[2,3], Morteza Farahi[2,3], Elham Shamsi[2], Mina GolMohammadi[3], Amin Mahnam[4], Mohsen Reza Haidari[5] and Amir Homayoun Jafari[2,3]**

[1] Computational Neuroscience, Institute of Medical Technology, Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany, [2] Department of Medical Physics & Biomedical Engineering, School of Medicine, Tehran University of Medical Sciences, Tehran, Iran, [3] Research Center for Biomedical Technologies and Robotics (RCBTR), Tehran University of Medical Sciences, Tehran, Iran, [4] Department of Biomedical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran, [5] Section of Neuroscience, Department of Neurology, Faculty of Medicine, Baqiyatallah University of Medical Sciences, Tehran, Iran

OPEN ACCESS

Edited by:
Elisa Donati,
ETH Zürich, Switzerland

Reviewed by:
Omid Kavehei,
The University of Sydney, Australia
Eugenie Roudaia,
McMaster University, Canada

***Correspondence:**
Mohsen Reza Haidari
drmohsinraza2012@yahoo.com
Amir Homayoun Jafari
h_jafari@tums.ac.ir

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: 13 February 2020
Accepted: 15 September 2020
Published: 17 November 2020

Citation:
Shirzhiyan Z, Keihani A, Farahi M, Shamsi E, GolMohammadi M, Mahnam A, Haidari MR and Jafari AH (2020) Toward New Modalities in VEP-Based BCI Applications Using Dynamical Stimuli: Introducing Quasi-Periodic and Chaotic VEP-Based BCI. Front. Neurosci. 14:534619. doi: 10.3389/fnins.2020.534619

Visual evoked potentials (VEPs) to periodic stimuli are commonly used in brain computer interfaces for their favorable properties such as high target identification accuracy, less training time, and low surrounding target interference. Conventional periodic stimuli can lead to subjective visual fatigue due to continuous and high contrast stimulation. In this study, we compared quasi-periodic and chaotic complex stimuli to common periodic stimuli for use with VEP-based brain computer interfaces (BCIs). Canonical correlation analysis (CCA) and coherence methods were used to evaluate the performance of the three stimulus groups. Subjective fatigue caused by the presented stimuli was evaluated by the Visual Analogue Scale (VAS). Using CCA with the M2 template approach, target identification accuracy was highest for the chaotic stimuli (M = 86.8, SE = 1.8) compared to the quasi-periodic (M = 78.1, SE = 2.6, p = 0.008) and periodic (M = 64.3, SE = 1.9, p = 0.0001) stimulus groups. The evaluation of fatigue rates revealed that the chaotic stimuli caused less fatigue compared to the quasi-periodic (p = 0.001) and periodic (p = 0.0001) stimulus groups. In addition, the quasi-periodic stimuli led to lower fatigue rates compared to the periodic stimuli (p = 0.011). We conclude that the target identification results were better for the chaotic group compared to the other two stimulus groups with CCA. In addition, the chaotic stimuli led to a less subjective visual fatigue compared to the periodic and quasi-periodic stimuli and can be suitable for designing new comfortable VEP-based BCIs.

Keywords: VEP-based BCI, chaotic stimuli, quasi-periodic stimuli, CCA, coherence

INTRODUCTION

Electroencephalogram (EEG) is commonly used for EEG-based brain computer interfaces (BCIs) as a non-invasive and low-cost method for measuring the brain neural activities (Wolpaw et al., 2002; Lebedev and Nicolelis, 2006). BCI applications employing EEG use visual evoked potentials (VEPs) (Middendorf et al., 2000; Müller-Putz et al., 2005; Lee et al., 2006, 2008;

Wang et al., 2006,Wang Y. et al., 2017; Martinez et al., 2007; Allison et al., 2008; Guo et al., 2008; Chen et al., 2015; Xie et al., 2017, 2018), which are responses of the visual system to visual stimuli. Various types of visual stimuli, such as flickering LED, can be decoded from the EEG activity of the visual cortex and used for diverse BCI applications (Takano et al., 2009; Lee et al., 2011; Gao et al., 2014; Kapgate and Kalbande, 2015).

In VEP-based BCIs, the target stimuli are identified by decoding all the gazed stimuli. The stimuli are required to be orthogonal or almost orthogonal in time-, frequency-, or code-based BCIs (Bin et al., 2009a). The oddball paradigm is an example of time-based BCIs, whereby the target stimuli are presented at different times and evoke event-related potentials (ERPs) like the P300 (Jin et al., 2005; Lee et al., 2008; McCane et al., 2015). Frequency-based BCIs may use visual stimuli that are modulated in time according to a sine wave with different temporal frequencies, which also evoke EEG responses at the same frequencies and their harmonics (Middendorf et al., 2000; Müller-Putz et al., 2005; Wang et al., 2006). Orthogonal visual stimuli in code-based BCIs are generated by random codes such as *m*-sequences, whereby different shifts of a modulating code have been used to evoke code-modulated VEPs (c-VEPs) (Nakanishi et al., 2014; Riechmann et al., 2016; Wei et al., 2016, 2018; Spüler, 2017; Liu et al., 2018; Shirzhiyan et al., 2019).

Due to their high decoding accuracy, external stimuli such as periodic flickers are commonly used in VEP-based BCIs evoking steady-state visual evoked potentials (SSVEPs) (Vialatte et al., 2010; Keihani et al., 2018). In SSVEP-based BCIs, the stimulus comprises a constant frequency that varies from low to high (1–100 Hz), which then leads to specific EEG responses that correlate with the stimulus frequency (Vialatte et al., 2010). Therefore, the gazed target stimuli could be identified from their EEG responses. However, among different frequency sets, the lower (1–12 Hz) and medium (13–16 Hz) ones lead to high subjective discomfort, fatigue, and possible epileptic seizures (Volosyak et al., 2011). Various dynamical approaches have also been used for improving SSVEP-based BCIs, such as dynamic stopping methods and the detection of SSVEP responses for higher information transfer rate (ITR) SSVEP-based BCIs (Yin et al., 2014; Jiang et al., 2018).

Visual stimuli have diverse dynamical patterns such as periodic, quasi-periodic, and chaotic. Biological systems also exhibit these dynamical behaviors (Attinger et al., 1966; Petrov et al., 1997; Suzuki et al., 2016) including non-oscillatory chaotic behavior, which is more complex than quasi-periodic oscillation (Camazine et al., 2003; Saha and Galic, 2018; Strogatz, 2018). Neuronal systems exhibit both complex oscillatory behavior (Llinás, 1988; Zhanabaev and Kozhagulov, 2013; Zhanabaev et al., 2016; Feng et al., 2017) as well as the non-oscillatory chaotic behavior that is seen in neurons (Aihara et al., 1984; Hong, 2011; Lv et al., 2016; Ma and Tang, 2017) and networks (Aihara, 1989; Freeman, 1992; Potapov and Ali, 2000; Rössert et al., 2015; Nobukawa and Nishimura, 2016) due to various underlying mechanisms (Hoebeek et al., 2010; Ishikawa et al., 2015). Stimuli with dynamical patterns such as chaotic behaviors are thus expected to be more in harmony with the visual system.

Natural visual stimuli rarely flicker at a constant rate, but rather exhibit more complex dynamics with quasi-periodic temporal characteristics (Kayser et al., 2003; Butts et al., 2007; Mazzoni et al., 2011). Natural visual stimuli are efficiently encoded by the visual system which is capable of processing and detecting information from complex natural environments (Blake and Lee, 2005; Mazzoni et al., 2011). These visual stimuli have similar spatial and temporal patterns resembling the $1/f$ amplitude spectrum, features that are encoded more efficiently by the retina and other components of the visual system (Atick and Redlich, 1992; Yoshimoto et al., 2017). In addition, chaotic patterns also follow the $1/f$ spectrum observed in natural scenes and phenomena (Relano et al., 2002; Molina et al., 2010). Quasi-periodic visual stimuli can generate phase-locked responses (Keitel et al., 2017; Obleser et al., 2017; Haegens and Golumbic, 2018) and also evoke responses with independent dynamics that correlate with their corresponding stimuli (Keitel et al., 2017). Temporal dynamics of the presented visual stimuli leads to adjustment of the visual system based on its inherent characteristics (Lasley and Cohn, 1981; Correa and Nobre, 2008). For these reasons, it is possible to assume that complex stimuli with dynamical temporal patterns such as quasi-periodic and chaotic may generate correlated responses which may lead to greater visual comfort for the viewer compared to the periodic stimuli.

One of the important issues in VEP-based BCI applications is the subjective visual fatigue caused by the flickering stimuli (Volosyak et al., 2011; Chang et al., 2014; Won et al., 2015). Periodic stimuli generating SSVEPs, due to their high contrast flashes, are not comfortable and can lead to subjective visual fatigue (Kardan et al., 2015; Xie et al., 2016). These periodic patterns may also lead to migraine headache (DeTommaso et al., 1999) or even epileptic seizures (Fisher et al., 2005). Studies have used various methods including the use of high-frequency stimuli rather than lower frequencies (Allison et al., 2010; Sakurada et al., 2015; Ajami et al., 2018), polychromatic stimuli (Chien et al., 2017), motion Newton's rings and motion checkerboards (Xie et al., 2012, 2017; Yan et al., 2017; Han et al., 2018), and rhythmic pattern stimuli (Keihani et al., 2018) to minimize the subjective visual fatigue, which still remains an important problem in VEP-based BCI applications.

Utilization of visual stimuli with quasi-periodic and chaotic patterns that are closer to natural scenes in BCI applications requires further research. In our previous study, we used chaotic and pseudo-random *m*-sequence binary codes and found that chaotic codes lead to comparatively less fatigue (Shirzhiyan et al., 2019). In this study, we introduce a new kind of visual stimuli with quasi-periodic and chaotic characteristics to evoke distinct visual potentials in normal subjects for their possible application in VEP-based BCIs. For comparison, we used periodic stimuli commonly employed in SSVEP-based BCI applications and also compared subjective visual fatigue caused by these three groups of stimuli.

MATERIALS AND METHODS

In this study, first of all, the stimulus groups were designed and proper setup for the experiment was prepared. The data recording step started with EEG and behavioral data (fatigue data) recording from normal subjects. After preprocessing of

the EEG data, the data analysis was done to decode the presented stimuli from their corresponding data using canonical correlation analysis (CCA) and coherence analysis methods. These two methods calculate the similarity of templates with EEG signals where the stimuli could be considered as templates (template generation approach M1) or obtained from a training dataset (template generation approach M2). Finally, the target identification results using the above-mentioned methods and the fatigue data were analyzed separately. The flowchart of this study is presented in **Figure 1**.

Study Participants

Our study was announced in the faculties of medicine and biomedical engineering *via* notice boards as well as in students' social media groups. Forty-eight volunteers were initially enrolled based on the inclusion criteria (normal or corrected vision with no history of head trauma and without current use of drugs). All subjects were informed about the study aims and procedure of signal recording and were allowed to leave the experiment at any point if they wished. Thirty-eight subjects participated in all the sessions (18 females), aged 20–33 years old (23.01 ± 4.32). The remaining 10 subjects did not participate in all the sessions or left the session due to urgent work, and so their data were excluded from the study. Written informed consent was signed by the participants before joining the study in accordance with the Declaration of Helsinki. The study was approved by the Office of Research Review Board and the Research Ethics Committee of the Tehran University of Medical Sciences, with LREC protocol number IR.TUMS.REC.1394.2110.

Stimuli

We used visual stimuli consisting of modulating the brightness of a red color LED measuring 4 cm × 4 cm according to three different temporal patterns: periodic, quasi-periodic, or chaotic. Each of these three categories had four different target stimuli that had their orthogonal characteristics. All the stimuli were generated using MATLAB software (Release 2016b, The MathWorks, 193 Inc., Natick, MA, United States).

Periodic Stimuli

For generating the periodic stimuli, we used four sine waves at the target frequencies of $f_1 - f_4$ (20, 25, 35, and 40 Hz), as shown in Equation 1 and schematically illustrated in **Figure 2A**. It can be seen that the simple periodic stimulus group (P_1 – P_4) had constant frequencies and that their spectrum was sparse in the frequency domain representation.

$$P_i = \sin\left(2\pi f_i t\right). \quad t = 0 : 6\,\text{sec}. \quad f = [20\,\text{Hz}.\ 25\,\text{Hz}.\ 35\,\text{Hz}.\ 40\,\text{Hz}] \tag{1}$$

Quasi-Periodic Stimuli

A sine-circle map was used to generate four quasi-periodic stimuli (Essl, 2006). Equation 2 models the sinusoidal oscillators that were perturbed by non-linear function.

$$\theta_{n+1} = \theta_n + \Omega - K/2\pi \ \sin(2\pi\theta_n) \tag{2}$$

where Ω is the frequency ratio and K is the coupling length of non-linear perturbation.

If the frequency ratio Ω is a rational number (p/q) with p and $q \in N$ (natural numbers), the map shows periodic behavior. For irrational numbers of Ω and appropriate parameters of K, the behavior of the sine-circle map is called quasi-periodic oscillation (Essl, 2006).

Quasi-periodic stimuli were generated using a sine-circle map by considering the parameter $K = 0.5$ and then Ω were selected as irrational numbers $\left(\sqrt{5}-1\right)/2$, $\sqrt{3}-1$, $\sqrt{3}/2$, and $\sqrt{2}/9$, where the sine-circle map showed quasi-periodic behaviors. These parameters were used to generate quasi-periodic stimuli Q_1, Q_2, Q_3, and Q_4. After that, the generated sequences from the sine-circle maps were considered as a time series with a sampling frequency of 90 Hz. Thus, each sample of the sequences was applied for a duration of 1/90 ms. The waveform and spectrum of the quasi-periodic stimuli are shown in **Figure 2B**. The waveform and spectrum of the quasi-periodic stimuli were more complex compared to the periodic stimulus group.

Chaotic Stimulus Group

For generating chaotic stimuli, we used a logistic map which is a one-dimensional map capable of generating chaotic signals with low cross-interferences. This map is seen in most of the natural phenomena and population growth of biological species (Costantino et al., 1997), as defined in Equation 3.

$$x\,(i+1) = A\ x\,(i)\,(1 - x\,(i)) \tag{3}$$

where x is in the interval of [0 1] and indicates the ratio of an existing population to the maximum possible population, $x(0)$ as the initial value of x, and A is the rate of reproduction and starvation that is in the interval of [0 4]. This simple map could generate chaotic dynamics in some values of parameter A generally between 3.5 and 4 (May, 1976). Parameter A was chosen in a way that the logistic map exhibited chaotic behaviors for generating four chaotic sequences and were then presented at the rate of 90 Hz as A = 3.982, 3.885, 3.987, and 4, respectively. In this way, four different chaotic stimuli, Ch_1, Ch_2, Ch_3, and Ch_4, were generated by the logistic map.

Figure 2C shows the waveforms of four chaotic stimuli with their amplitude spectra.

The dotted black curved line in the amplitude spectra plots shows the $1/f$ line, where f is the frequency vector (horizontal axis). It can be seen that the amplitude spectra of the stimuli in the chaotic group are closer to the $1/f$ spectrum line compared to the quasi-periodic and periodic stimulus groups. Please note that for better illustration, only 1 s of the total 6 s duration of every stimulus is shown in the plot.

Auto- and Cross-Correlation Function of Target Stimuli

The auto- and cross-correlation functions of the periodic, quasi-periodic, and chaotic stimulus groups were calculated to investigate their individual orthogonal characteristics. This was done to verify overlapping characteristics in order to avoid interference between the target stimuli. The auto- and cross-correlation functions of the periodic, quasi-periodic, and chaotic

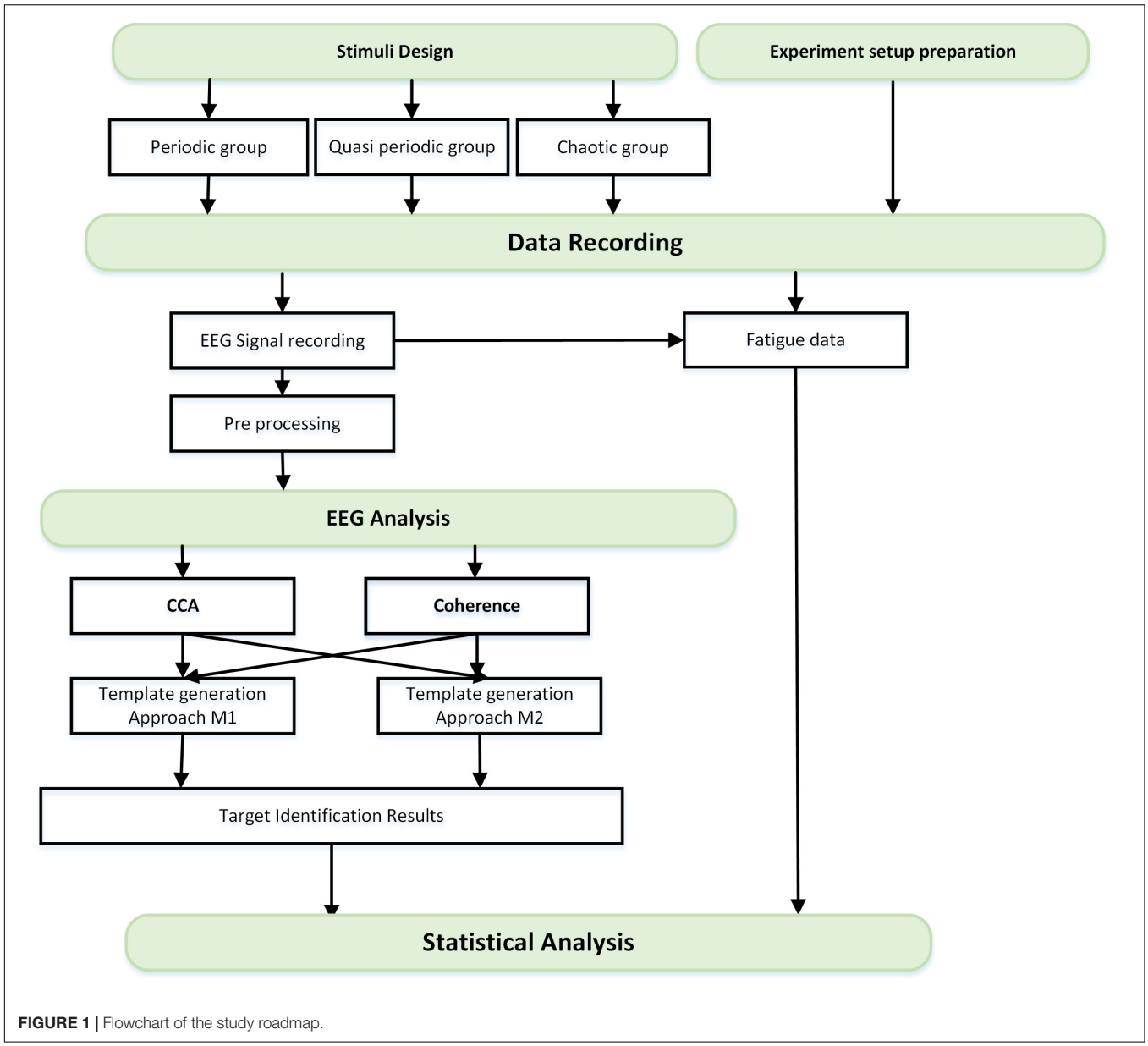


FIGURE 1 | Flowchart of the study roadmap.

groups are presented in **Figures 3A–C**, respectively. It is obvious that the auto-correlation function of each target stimulus group is high, while within-group cross-correlations of stimuli with others is comparatively low.

Stimulus Presentation Paradigm

All the subjects were presented with periodic, quasi-periodic, and chaotic stimuli in a single lab visit. They were presented with each of the periodic (P_1 – P_4), quasi-periodic (Q_1 – Q_4), and chaotic (Ch_1–Ch_4) stimuli as 12 different sessions. The total duration of each session (for each stimulus) was 90 s consisting of 10 trials. In each trial (6 s duration), the same stimulus was presented to the subject with a 2 s rest time in between the trials. An initial rest of 10 s was included in each session. After each session (90 s), the subjective fatigue was evaluated (see below). The maximum duration of a whole stimulus presentation paradigm including rest time was approximately 30 min. The stimulus presentation paradigm is shown in **Figure 4**.

The subjects were informed before the experiment that they will be asked to evaluate their own visual fatigue by considering the amount of tiredness and discomfort caused by gazing at the stimuli. They were asked to grade their visual fatigue level by choosing a number between 0 for no fatigue and 10 for the highest fatigue level. After each session (90 s), the subjects were asked to self-report the level of visual fatigue caused by stimulation. Their fatigue rate was recorded and they were asked for permission to start the next session. The order of presentation of stimuli in all groups was randomly distributed for all subjects to avoid possible bias in subjective visual fatigue caused by the order of presentation.

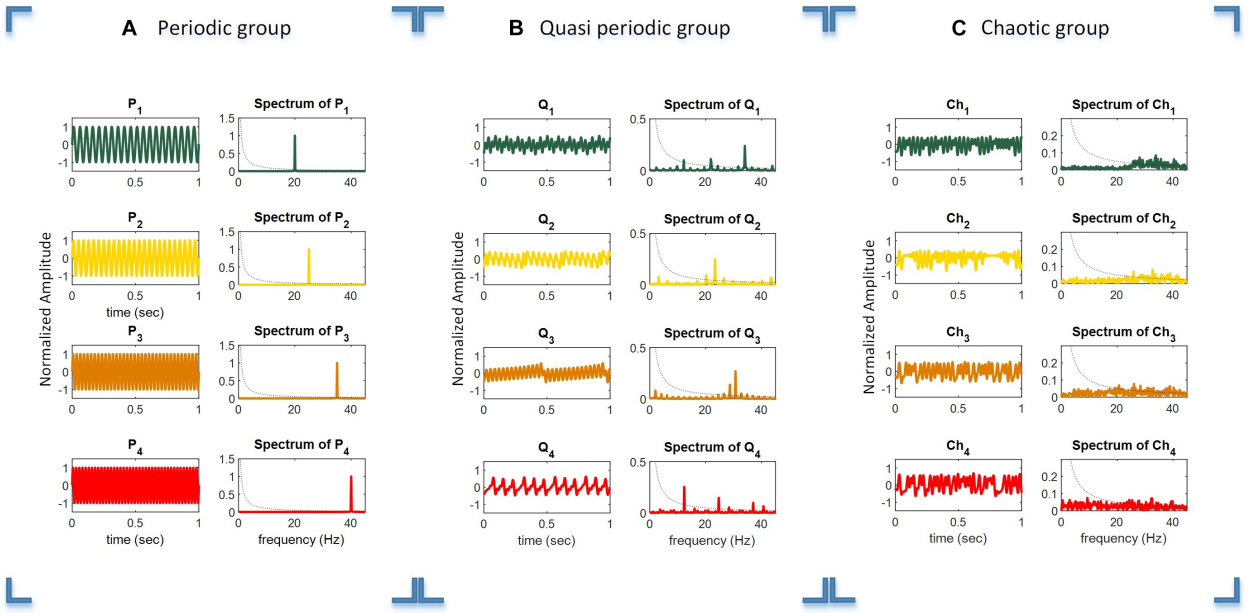


FIGURE 2 | Time duration, amplitude, and frequency spectrum of the periodic **(A)**, quasi-periodic **(B)**, and chaotic **(C)** stimulus groups. *Columns 1* and *2* in each plot show the relevant waveforms and spectra of the stimuli of each group. For better illustration of the waveforms, they are shown in a 1 s timescale, while the total duration of stimuli was 6 s.

Fatigue Evaluation Process

The level of subjective visual fatigue was measured using graded values of the Visual Analogue Scale (VAS), which is suitable for grading continuous phenomena (Aitken, 1969). The VAS is a subjective estimation method for quantifying a feeling and attitude which is hard to estimate directly (Gift, 1989; Grant et al., 1999; Crichton, 2001; Tseng et al., 2010; Klimek et al., 2017). This scale is mostly used in clinical research for measuring the intensity of various symptoms (Paul-Dauphin et al., 1999) such as pain (Bijur et al., 2001). It is commonly used in BCI applications for the evaluation of a patient's motivation and mood (Holz et al., 2013), the level of subjective fatigue (Guger et al., 2013; Käthner et al., 2014), pain (Choi, 2017), discomfort (Verwulgen et al., 2018), and control ability (Chumerin et al., 2012).

Signal Recording Setup

The EEG signals were recorded using g.USB Amp with a sampling rate of 1,200 Hz. Four active g.Ladybird electrodes were placed at Oz, O1, O2, and Pz positions on the scalp of the subjects where the visual evoked potentials have maximum amplitude (Bin et al., 2011; Aminaka et al., 2015). Fpz and right earlobe were used as the ground and reference electrodes, respectively. An online bandpass filter with cutoff frequencies of 0.05 and 120 Hz was applied.

The generated stimuli were applied to a custom-made digital-to-analog converter (DAC) board as a stimulus presenter box (shown in **Supplementary Figure 1**) for driving an LED. The LED was placed at a distance of 70 cm from the subject. The trigger output of g.USB Amp (start time of EEG recording) and the output of a Texas Instruments optical sensor (visual stimuli) were sent to National Instruments (NI) DAQ. Details of the signal recording setup are reported in our previous studies (Keihani et al., 2018; Shirzhiyan et al., 2019).

Data Processing

The signal analysis procedure was carried out for the recorded responses for each stimulus group (periodic, quasi-periodic, and chaotic) (**Supplementary Figure 2**) separately to compare the results of the three different groups. Tenfold cross-validation was used as our validation method. In this method, nine-tenths of the trials were used as the training data and one-tenth was used as the testing data. The training data was used for template generation and the testing data was used for target identification.

Preprocessing

The recorded trigger from g.USB Amp and sensor output and the presented stimuli in NI DAQ were used for the detection of the beginning of each trial and then each trial EEG data was extracted. A zero-phase eighth-order band pass filter with cutoff frequencies of 1 and 50 Hz was applied for all the trials.

Processing (CCA and Coherence Analysis)

To analyze the EEG data, we used two methods that are commonly employed in BCI studies: CCA and coherence analysis (Zhang et al., 2013, 2014; Vaid et al., 2015). These methods measure the amount of correlation in the time and frequency domains, respectively.

Canonical correlation analysis is a multivariable data analysis method that measures the underlying time domain correlation between two multidimensional signals and attempts to reveal a linear time domain correlation by maximizing the correlation

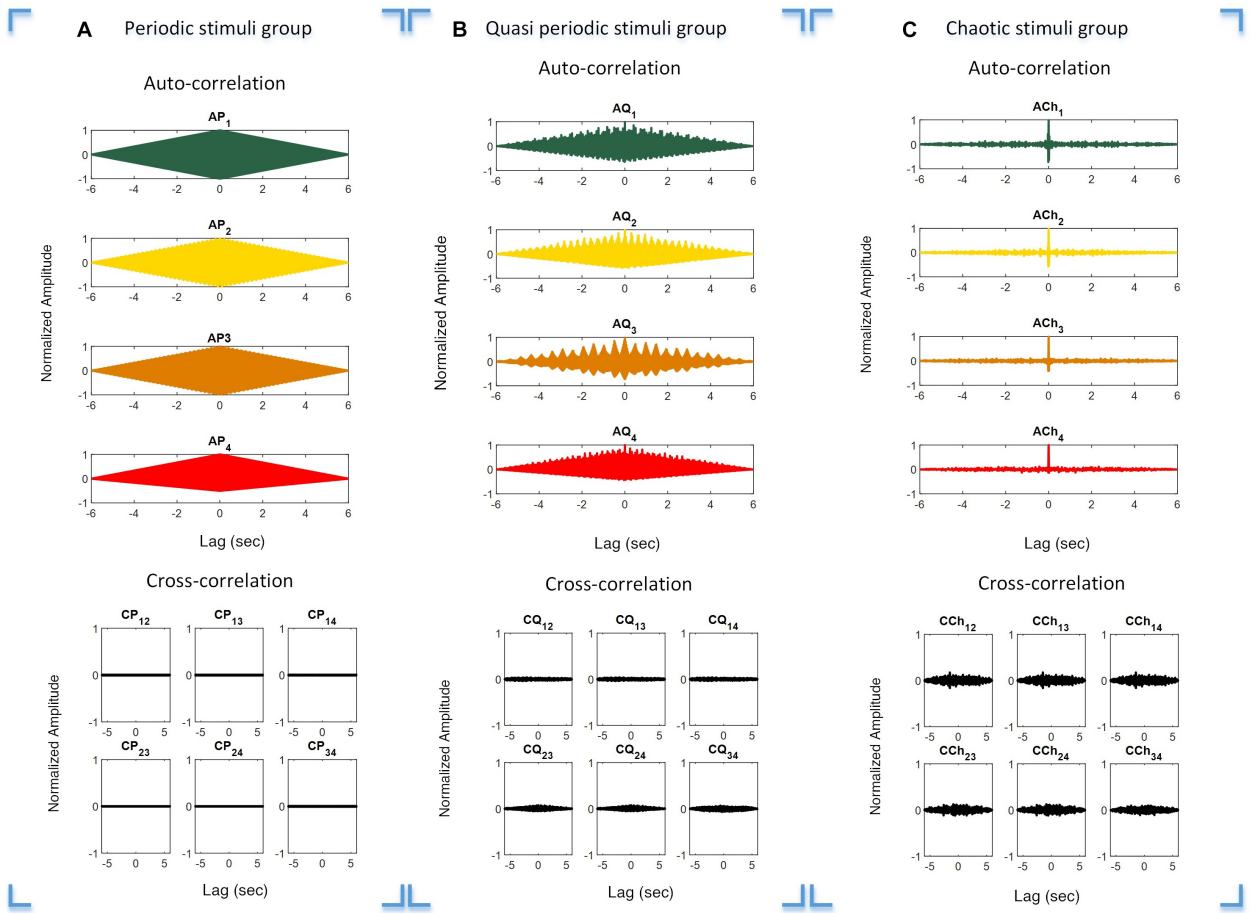


FIGURE 3 | Auto- and cross-correlation functions of the three stimulus groups. Plot **(A–C)** Correlation functions over different time lags of the periodic **(A)**, quasi-periodic **(B)**, and chaotic **(C)** groups. *Rows 1* and *2* in each plot show the auto- and cross-correlations of the target stimuli in each category, respectively.

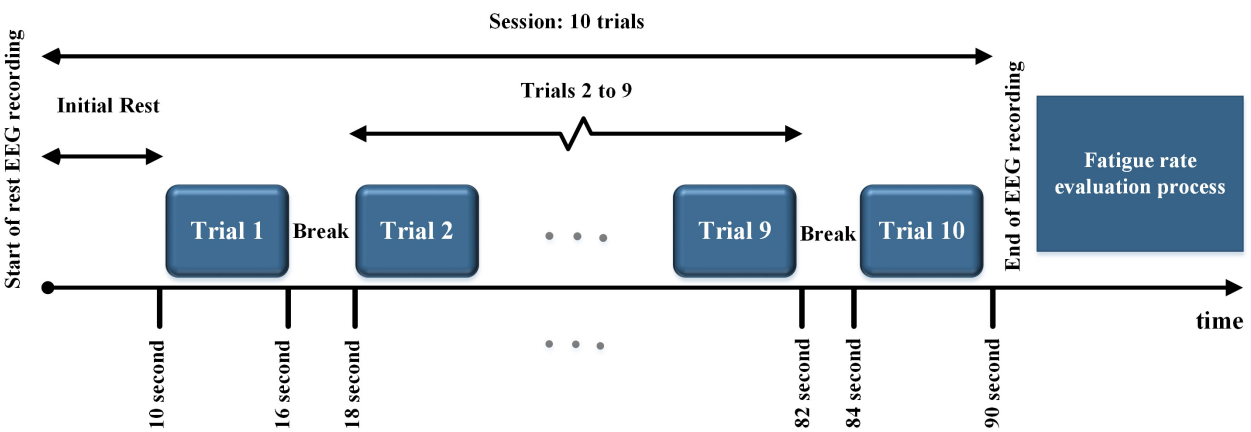


FIGURE 4 | Stimulus presentation paradigm. All the stimuli in each group were presented in 10 trials. After each trial, a 2 s rest time was considered. At the end of each session, the fatigue rate was evaluated.

of the two signals (Lin et al., 2006). CCA has been successfully used in target identification and in the analysis of visual evoked potentials (Lin et al., 2006; Bin et al., 2009b, 2011). Equation 4 defines the CCA coefficient of variables x and y, where $E(x^t y)$ stand for the covariance of x and y and $E(x^t x)$ and $E(y^t y)$ represent the variance of x and y, respectively. This

method finds the canonical correlation vectors W_x and W_y for two multidimensional variables, x and y, by maximizing their canonical covariants.

$$\underset{w_x.\ w_y}{\text{Max}}\rho\left(x.y\right) = \frac{E\left(x^t y\right)}{\sqrt{E\left(x^t x\right)E(y^t y)}} \tag{4}$$

Coherence analysis has been used to investigate the synchronization process of brain regions (Nunez et al., 1997; Lachaux et al., 1999; La Rocca et al., 2014). In addition, it has been used as a feature extraction and target identification method in BCI applications (Gysels and Celka, 2004; Krusienski et al., 2012; Liew et al., 2015). The coherence of two signals is sometimes called magnitude-squared coherence, as shown by Equation 5 which defines the amount of coherence of two signals at a specific frequency.

$$C_{xy}\left(f\right) = \frac{\left|S_{xy}\left(f\right)\right|^2}{S_{xx}\left(f\right)\ S_{yy}(f)} \tag{5}$$

where S_{xx} and S_{yy} are the power spectral densities of variables x and y, respectively, and $S_{xy}\left(f\right)$ is the cross power spectral density between x and y (Sanei and Chambers, 2007).

In this study, variable x is the template and y is the EEG signal. In BCI studies, it is common to use either the presented stimuli or the average EEG signal of the training session trials to generate the template. Therefore, in this study, we used two approaches for generating templates. In the first approach (M1), the stimuli were used as templates, while in the second approach (M2), we attempted to extract templates from the training dataset.

Template Generation

Templates were generated by using the two aforementioned approaches as described below.

Approach 1 (M1): using the presented stimuli as templates

The target stimuli were resampled to the sampling frequency of the EEG responses (1,200 Hz) and zero-padded the resampled target stimulus i by lag time D_i. The lag time D_i represents the systematic lag for the presented stimuli and was calculated by cross-correlating the target stimuli with the grand averaged EEG responses and determining the time lag that yielded the maximum cross-correlation values.

This step was not important for coherence analysis because the magnitude-squared coherence was not sensitive to the time lag between templates (stimuli) and responses, while the CCA coefficients were maximum where the lag was considered.

Approach 2 (M2): generating templates using the training dataset

In VEP-based BCI studies, it is also common to create templates using the EEG signals from a subset of the data (i.e., a training dataset) instead of the stimulus waveform itself, as this approach allows capturing information related to the non-linear processing of the system. Given that we had access to the training dataset, we used this approach to generate templates by EEG data from the training dataset.

This approach included extracting the EEG responses from r trials in the training set, $X_{\text{Train}_i}^{r\times m\times n}$, and averaging over r trials to generate the template for stimulus i, $T_i^{m\times n}$, where m is the number of channels and n is the number of samples per trial.

Target Identification

After generating templates separately for the targets in each group, the CCA coefficient and coherence were calculated by Equations 4 and 5, where T was considered as variable x and the EEG response was considered as variable y. For template generation using the M1 approach, all the trials were separately considered as testing trials, while 10-fold cross-validation was considered for template generation using the M2 approach. Here, ninefold of the dataset was considered as the training dataset and the one remaining fold was considered as the testing trial. Therefore, all the trials were tested once. The details of both the approaches of target identification are given below.

For the CCA method:

(1) Extraction of testing trials $X_{\text{Test}}^{m\times n}$, where m and n are the channel numbers and samples in a trial, respectively.
(2) Calculation of the CCA coefficient of templates T_i and $X_{\text{Test}}^{m\times n}$ as vector $\text{P}_i{}^{1\times\ m}$.
(3) Calculation of the mean value of $\text{P}_i^{1\times m}$ to create the feature vector.
(4) Selection of the maximum value of feature vector.

For the coherence method:

(1) Extraction of testing trials $X_{\text{Test}}^{m\times n}$, where m and n are the channel numbers and samples in a trial, respectively.
(2) Calculation of the coherence function of templates T_i and $X_{\text{Test}}^{m\times n}$ for obtaining the vector $C_i\left(f\right)$.
(3) Extraction of the coherence coefficient from vector of $C_i(f)$ in the target frequencies or a specific frequency band.
(a) Periodic group: the target frequencies in the periodic group were the target frequencies of the presented stimuli as shown in **Figure 2A**.
(b) Quasi-periodic group: the dominant frequencies of the presented stimuli are shown in the spectra of stimuli in **Figure 2B**.
(c) Chaotic group: the total frequency band of the chaotic stimuli as shown in **Figure 2C**.
(4) Calculation of the mean value of $C_i^{m\times f}$ to create the feature vector.
(5) Selection of the maximum value of the feature vector.

Figure 5 schematically shows the template generation and target identification processes.

Statistical Analysis

Statistical analysis was done with SPSS software (version 16.0, SPSS Inc., IBM Corp., Chicago, released 2011) for comparing the analysis methods and also for evaluating the subjective visual fatigue rate between the three groups of stimuli (periodic, quasi-periodic, and chaotic).

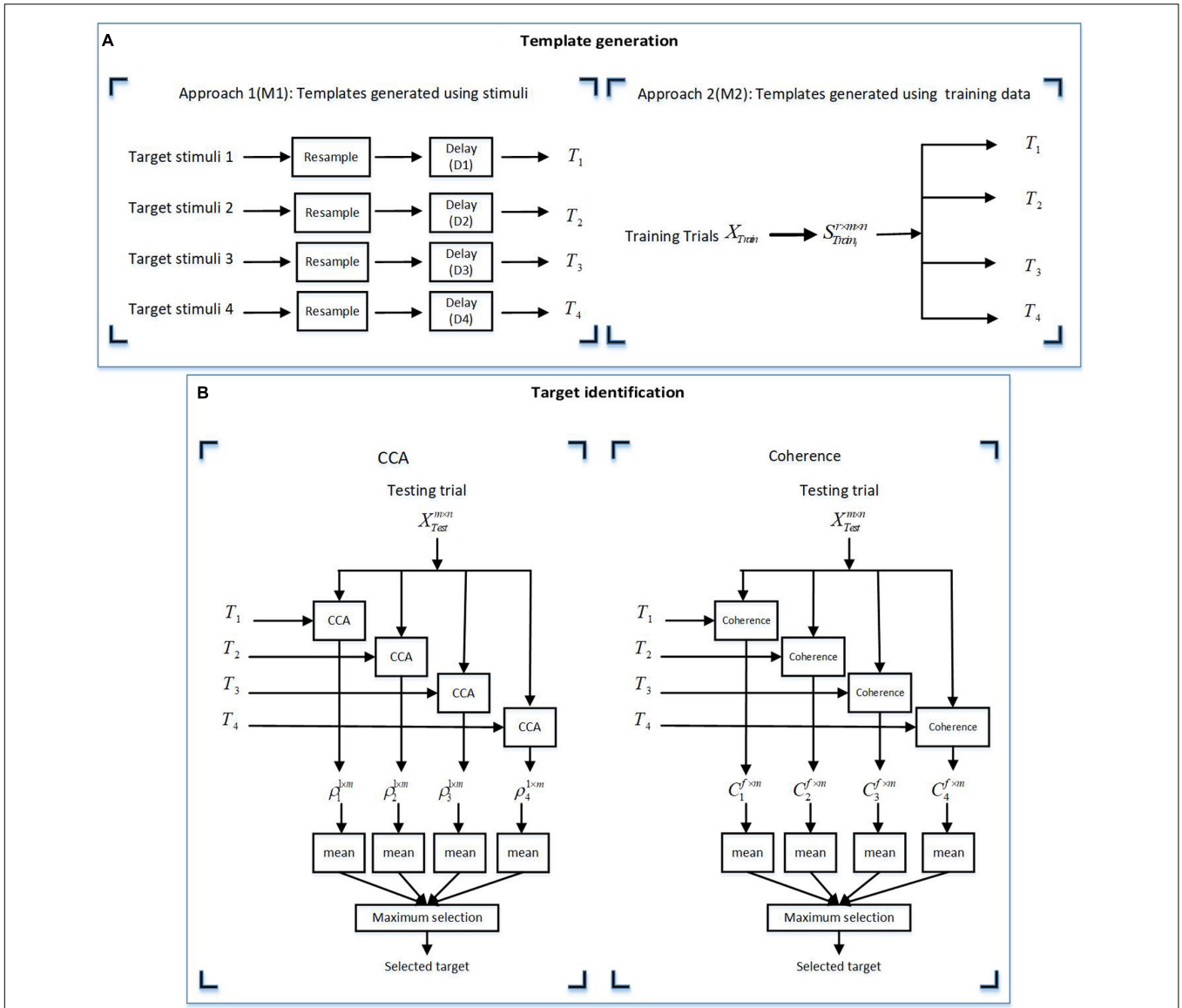


FIGURE 5 | Illustration of the two methods for **(A)** template generation using approaches M1 and M2. **(B)** Target identification using the canonical correlation analysis (CCA) and coherence methods. **(A)** The templates for the CCA and coherence methods were derived by two approaches: M1 (using target stimuli) and M2 (using training data). **(B)** For the target identification, the derived templates from the template generation approaches **(A)** were used for analysis by the CCA and coherence methods. In this figure *m*, *n*, and *r* are the channel numbers, samples in a trial, and the trial numbers in a training dataset, respectively. *f* represents the frequency vector in a specific frequency band in each stimulus group. X_{Train} are all training trials and S_{train_i} represents trial response to the *i*th stimulus in each stimulus group.

Statistical Analysis of Accuracies

Three-way repeated measures ANOVA was used to test the effects of three factors—methods (CCA and coherence), approaches (M1 and M2), and stimulus groups (periodic, quasi-periodic, and chaotic)—with assumed sphericity (significance level $\alpha = 0.05$). Confidence intervals were adjusted by Bonferroni correction for pairwise comparisons.

Statistical Analysis of Fatigue Rates

Within-group analysis of fatigue rates

To compare the VAS scores across stimuli within each stimulus type (periodic, quasi-periodic, and chaotic), the Friedman test (significance level $\alpha = 0.05$) was used. Then, the scores for each pair of stimuli were compared using the Wilcoxon signed-rank test, with a Bonferroni-corrected alpha set to 0.008.

Between-group analysis of subjective visual fatigue rates

For comparison of the subjective visual fatigue caused by the periodic, quasi-periodic, and chaotic groups, the VAS scores of each group were averaged for the four sessions for each stimulus type for each subject and were then compared using the Friedman test (significance level $\alpha = 0.05$). The Wilcoxon signed-rank test with Bonferroni correction was

used for the comparison of each pair while α was set at 0.0168.

RESULTS

Accuracy Analysis Results

The descriptive statistics of all accuracies of the three stimulus groups (periodic, quasi-periodic, and chaotic) obtained from two different methods (CCA and coherence) and template generation approaches (M1 and M2) are reported in **Table 1**, and **Figure 6** shows the estimated marginal means plots of accuracies.

The 2 (analysis method) × 2 (template approach) × 3 (stimulus group) ANOVA on target identification accuracy revealed significant main effects of analysis method [$F(1,37) = 60.253$, $p = 0.0001$, $\eta_p^2 = 0.620$] and template approach [$F(1,37) = 28.56$, $p = 0.0001$, $\eta_p^2 = 0.435$], but not stimulus group [$F(1,37) = 1.143$, $p = 0.324$, $\eta_p^2 = 0.030$], with overall higher accuracy for the CCA method relative to coherence and higher accuracy for the M1 template approach relative to M2.

However, these effects were qualified by significant two-way interactions between analysis method and stimulus group [$F(2,74) = 47.0009$, $p = 0.0001$, $\eta_p^2 = 0.56$] and template approach and stimulus group [$F(2,74) = 8.776$, $p = 0.0001$, $\eta_p^2 = 0.192$]. The interaction between analysis method and template approach was not significant [$F(1,37) = 3.695$, $p = 0.062$, $\eta_p^2 = 0.091$]. Finally, the three-way interaction was significant [$F(2,74) = 35.74$, $p = 0.0001$, $\eta_p^2 = 0.491$]. To decompose the three-way interaction, we

TABLE 1 | Descriptive statistics of all accuracies obtained by two methods (CCA and coherence analysis) with two template generation approaches (M1 and M2) in three different stimulus groups.

Method	Approach	Stimulus group	Mean (%)	Standard error (%)	95% Confidence interval	
					Lower bound (%)	Upper bound (%)
CCA	M1	P	79.5	2.4	74.8	84.3
		Q	73.2	2.3	68.5	77.9
		Ch	85.1	2.2	80.6	89.7
	M2	P	64.3	1.9	60.5	68.2
		Q	78.1	2.6	72.8	83.4
		Ch	86.8	1.8	83.1	90.4
Coherence	M1	P	75.0	2.2	70.5	79.5
		Q	70.5	2.7	65.1	76.0
		Ch	70.2	2.7	64.7	75.7
	M2	P	74.5	2.5	69.5	79.6
		Q	65.7	2.7	60.1	71.2
		Ch	58.8	3.0	52.7	64.8

CCA, canonical correlation analysis; M1, using target stimuli; M2, using training data; P, periodic; Q, quasi-periodic; Ch, chaotic.

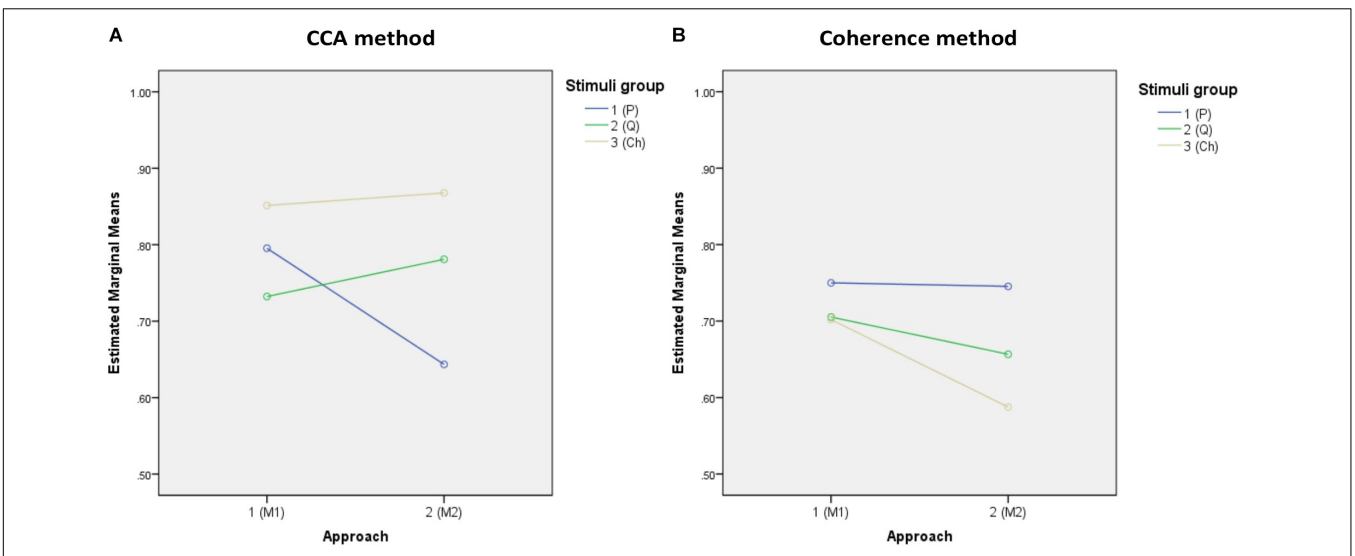


FIGURE 6 | Marginal estimated means of all accuracies. **(A,B)** Marginal means of the accuracies of both approaches for the CCA and coherence methods, respectively. **(C,D)** Marginal means of the target identification accuracies of the stimulus groups for approaches M1 and M2, respectively.

examined the effects of stimulus group for each analysis method and template separately and conducted pairwise comparisons between the stimuli groups.

For CCA and template approach M1, accuracy was higher for the chaotic stimuli (M = 85.1, SE = 2.2) than for periodic (M = 79.5, SE = 2.4, p = 0.0001) and quasi-periodic (M = 73.2, SE = 2.3, p = 0.008), with quasi-periodic also being lower than periodic (p = 0.0001). With template approach M2, CCA accuracy was again higher for the chaotic group (M = 86.8, SE = 1.8) compared to the other two groups, the accuracy now being better for the quasi-periodic (M = 78.1, SE = 2.6) than for the periodic stimuli (p = 0.0001) (**Figure 6A**).

For coherence analysis, using M1 template approach, multiple comparison with $\alpha = 0.0168$ did not show significant differences between the periodic (M = 75.0, SE = 2.2), chaotic (M = 70.2, SE = 2.7), and quasi-periodic (M = 70.5, SE = 2, $p > 0.0168$) stimulus groups. Using M2 template approach, the periodic group (M = 74.5, SE = 2.5) showed higher accuracy than did the chaotic group (M = 58.8, SE = 3, p = 0.001), and the quasi-periodic group (M = 65.5, SE = 2.7) did not significantly differ from the other groups ($p > 0.02$) (**Figure 6B**).

VAS Scores Analysis Results

The subjective fatigue VAS scores for the four stimuli within each stimulus group are shown in **Figure 7** and the averaged scores for each stimulus group are shown in **Figure 8**.

Results of Within-Group Analysis of the Periodic Group

The Friedman test showed significant differences in the VAS scores across stimuli in the periodic group [$\chi^2(3) = 37.857$, p = 0.0001]. Participants reported on average highest subjective fatigue scores for P_1 (20 Hz) (M = 4.92, SE = 0.38) and P_2 (25 Hz) (M = 4.5, SE = 0.30), which did not differ from each other (Z = 1.99, p = 0.046). The scores for P_3 (30 Hz) were significantly lower (M = 3.2, SE = 0.22) than for P_2 ($p < 0.001$), and P_4 had the lowest scores (M = 2.71, SE = 0.28).

Results of Within-Group Analysis of the Quasi-Periodic Group

The Friedman test showed significant differences among the VAS scores of stimuli in the quasi-periodic group [$\chi^2(3) = 14.848$, p = 0.002]. Q4 (M = 3.65, SE = 0.32) caused relatively higher fatigue scores compared to Q1 (M = 3.08, SE = 0.30, Z = 2.85, p = 0.004) in the pairwise comparison of within-group quasi-periodic stimuli. Q2 (M = 3.13, SE = 0.30) and Q3 (M = 3.05, SE = 0.326) did not differ significantly from the others.

Results of Within-Group Analysis of the Chaotic Group

The Friedman test showed significant differences in the VAS scores for the stimuli in the chaotic group [$\chi^2(3) = 20.125$, p = 0.0001]. Ch_1 (M = 2.02, SE = 0.28) had lower fatigue scores compared to Ch_3 (M = 2.78, SE = 0.31, Z = 3.53, p = 0.0001), and Ch_4 (M = 2.84, SE = 0.30, Z = 2.79, p = 0.005). Ch_2 (M = 1.78, SE = 0.28) did not differ from Ch_1 and Ch_4.

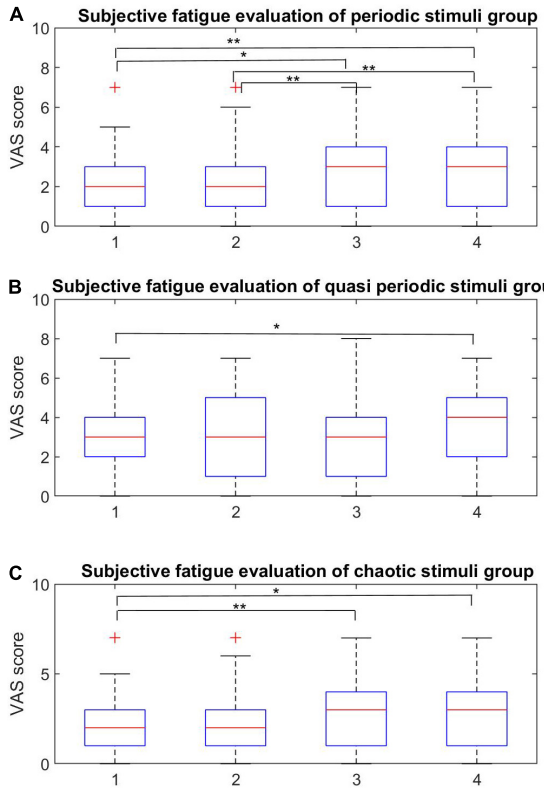


FIGURE 7 | Within-group subjective fatigue evaluation. **(A–C)** Within-group analysis of the fatigue rates for the periodic, quasi-periodic, and chaotic groups, respectively. **(A)** In the periodic group, the pairs of (P_1, P_3), (P_1, P_4), (P_2, P_4), and (P_3, P_4) showed significant differences. Higher frequencies led to lower fatigue rates. **(B)** There was only one significant difference within Q_1 and Q_4 in the quasi-periodic group. **(C)** The chaotic stimulus group showed significant difference between the Ch_1–Ch_3 and Ch_1–Ch_4 pairs (*$p < 0.001$, **$p < 0.0001$).

Details of the p-values for subjective fatigue comparisons using the Friedman test with Bonferroni correction are presented in **Supplementary Table 1**.

Between-Group Analysis of Subjective Visual Fatigue Rate

The Friedman test showed that the subjective visual fatigue scores differed across the three stimulus groups [$\chi^2(2) = 28.69$, p = 0.0001]. Participants reported higher subjective fatigue for the periodic stimuli compared to the quasi-periodic (Z = 2.931, p = 0.003) and chaotic stimuli (Z = 4.429, p = 0.0001). In addition, they also felt higher fatigue for the quasi-periodic stimuli compared to the chaotic group (Z = 3.466, p = 0.001).

DISCUSSION

In this study, for the first time, we used quasi-periodic and chaotic stimuli with different orthogonal characteristics and compared them with periodic stimuli commonly employed in SSVEP-based BCIs that use EEG data. We also compared the level of subjective

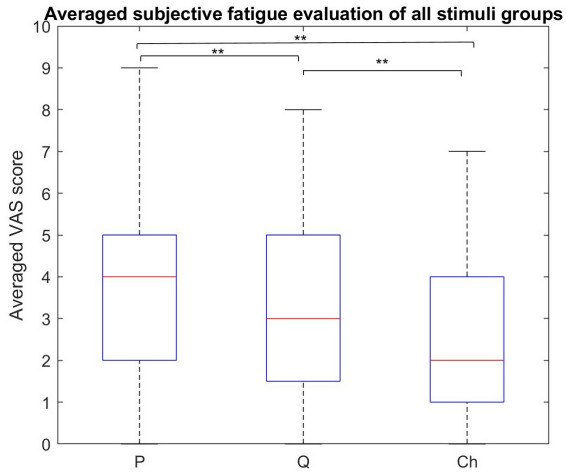


FIGURE 8 | Subjective fatigue rates comparison between all the stimulus groups. The periodic group had the highest fatigue rate compared to the other two groups. The chaotic group had the least fatigue rate compared to the periodic and quasi-periodic groups (**$p < 0.0001$, *$p < 0.016$).

visual fatigue caused by these three stimuli on young adult participants. For this purpose, three groups of visual stimuli with different temporal dynamics (periodic, quasi-periodic, and chaotic) from simple sinusoidal frequencies to complex stimuli were generated from sine-circle and logistic maps and used for evoking visual potentials.

Periodic stimuli have been used for years in VEP generation for eliciting SSVEP responses that are known for their relatively high ITR, less training time (Parini et al., 2009), and practical BCI applications (Lalor et al., 2005; Muller-Putz and Pfurtscheller, 2008; Bin et al., 2009b; Yin et al., 2015; Lin et al., 2016; Wang Y.T. et al., 2017; Wang et al., 2018). Our results showed that the introduced dynamical visual stimuli (quasi-periodic and chaotic stimulus groups) could also evoke discriminative responses and can have even better target identification accuracies than the periodic visual stimulus group using the CCA method. In addition, compared to the other stimulus groups (periodic and quasi-periodic), the obtained accuracy values of target identification for the chaotic group by employing the CCA method for template generation approaches M1 (stimuli waveforms considered as a template) and M2 (templates generated from training EEG dataset) were the highest, with values of $M = 86.78\%$, $SE = 1.8\%$ and $M = 85.1\%$, $SE = 2.2\%$, respectively. The results of the M1 approach for the periodic, quasi-periodic, and chaotic stimuli indicate that their corresponding EEG responses correlated with their stimuli waveforms. It has been reported that the temporal structures of quasi-rhythmic stimuli are reflected in the brain responses in the visual cortex (Keitel et al., 2017).

Auto- and Cross-Correlation Function of Stimulus Groups

The stimuli in the chaotic and quasi-periodic groups as well as in the periodic group had orthogonal characteristics (**Figure 3**). This was demonstrated by the fact that the pairwise cross-correlation values between the stimuli were less than the auto-correlation values for the target stimuli (**Figure 3**). The cross-correlation functions showed that the stimuli within each stimulus group were not correlated because their cross-correlation values were close to zero. This meant that these stimuli were nearly orthogonal and the interference between the stimuli would be reduced in possible BCI applications. It is worth noting that the chaotic group's auto-correlation exhibited a Dirac-like function, meaning that while these stimuli were orthogonal, they did not correlate with themselves. This feature was absent in the periodic and quasi-periodic groups, which required a higher level of synchronization between the visual stimuli and their EEG responses. The concept of Dirac-like auto-correlation function has been used in code-modulated BCIs for generating uncorrelated target stimuli from one code by the process of shifting (Bin et al., 2011).

Cross-Correlation Function of Stimuli and Responses

The cross-correlation function of the presented stimuli and their corresponding responses suggest that the visual pathway system serves as an input and the evoked potentials as the system output. The lag of maximum of the cross-correlation function is considered as a system delay, which was used in our analysis especially in the CCA method for the M1 approach to generate templates from the presented stimuli. Due to the periodic and semi-periodic nature of the cross-correlation function (as seen in **Figures 9A,B**), compensating for the delay in templates was not necessary. However, compensating for the delay was vital for the chaotic group because the chaotic stimuli correlated with their corresponding responses in a specific time delay (**Figure 9C**). For the CCA method using template generation from the training dataset (M2 approach), the time delay compensation was not needed as the inherent time delay was embedded in the templates extracted from the training dataset.

Target Identification Results Comparison

The highest accuracy for the chaotic group was obtained by CCA using the M2 approach, while the lowest accuracy was obtained by the coherence method using the M2 approach. The much lower accuracy seen with the chaotic stimuli using coherence analysis may be explained by the fact that the spectra of these stimuli are highly similar, making them less discriminative compared to the other stimulus groups (periodic and quasi-periodic) (column 2 in **Figure 2C**). As the coherence analysis quantified the synchronization of the spectral information of two variables, for the periodic stimuli (single frequency) and even the quasi-periodic stimuli (containing multiple dominant frequencies), measuring the amount of synchronization between the narrow frequency bands (Vaid et al., 2015) was relatively less likely to be impacted by noise. However, as the chaotic stimulus spectra are similar and less discriminant (column 2 in **Figure 2C**) compared to the other stimulus groups, coherence analysis using the M2 approach is not recommended for chaotic stimuli. We suggest

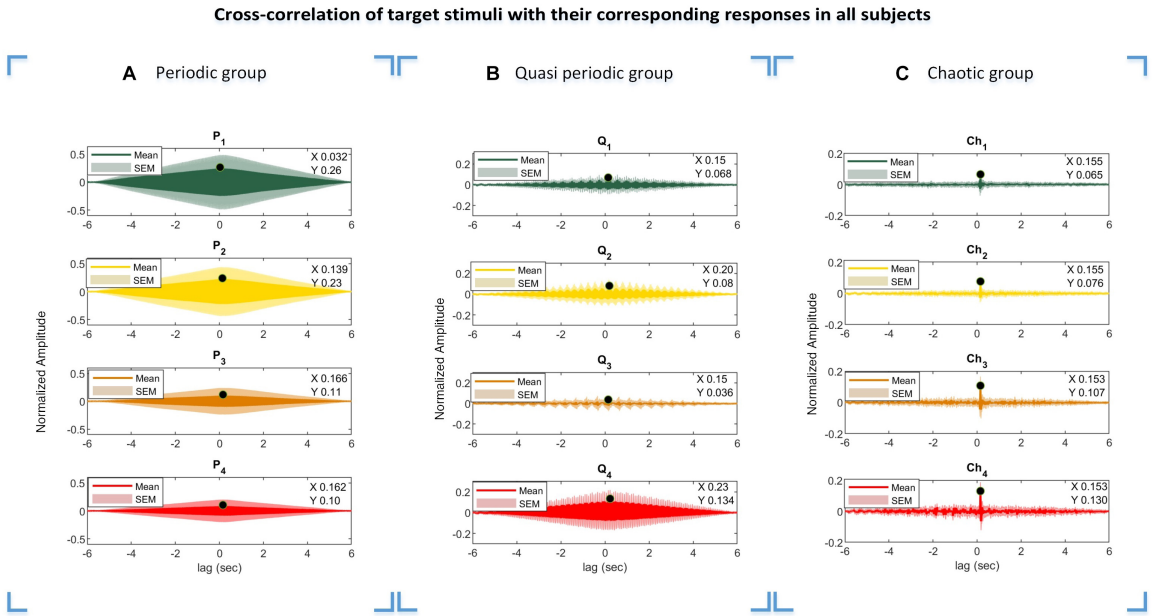


FIGURE 9 | Cross-correlations of the periodic, quasi-periodic, and chaotic stimuli with their corresponding EEG responses. Results of the periodic **(A)**, quasi-periodic **(B)**, and chaotic **(C)** groups in all subjects. The lag of maximum correlation values in the plots were considered as delay time (D_l, l = [1 2 . . . 4]) in template generation using the M1 approach. *X*- and *Y*-values represent the lag *black dot* values where the cross-correlation function of the stimuli and their corresponding responses are maximum, respectively.

the coherence method with the M1 approach for the analysis of chaotic stimuli.

While using the M1 templates with the coherence method led to a lower accuracy than with the CCA method, there is some benefit to using the coherence method with chaotic stimuli. Specifically, unlike CCA, coherence analysis is not sensitive to the time lag between variables (Guevara and Corsi-Cabrera, 1996). Therefore, using the coherence method could reduce the training time because it removes the need to obtain training data in order to extract the time lag needed for the time domain correlation.

The accuracy values of the M2 approach were found to be relatively higher compared to M1 in the CCA analysis for the quasi-periodic and chaotic stimuli (**Figure 6** and **Table 1**). This means that the EEG response to the chaotic and quasi-periodic stimuli may contain not only the stimulus-locked components but also more complex dynamics that did not correlate with the visual stimuli while being discriminative.

Within-Group Subjective Visual Fatigue Rate Evaluation

From **Figure 7**, it can be seen that the mean VAS scores of the periodic stimulus group (P_1 – P_4, corresponding to frequencies of 20, 25, 35, and 40 Hz) decreased as the frequency of the target stimuli increased. These results confirm the fact that higher frequencies cause a less subjective visual fatigue level compared to lower ones (Allison et al., 2010; Volosyak et al., 2011; Yoshimoto et al., 2017). The statistical results show significant differences between all the pairs, except for P_1 – P_2 and P_3 – P_4 which were close to each other compared to the other pairs.

Q_4 stimulus had higher VAS scores compared to Q_1. This may be due to the fact that Q_4 stimulus had dominant components in lower frequencies (column 2 in **Figure 2B**) compared to Q_1, possibly leading to a more subjective visual fatigue. Ch_1 stimuli caused lower subjective visual fatigue compared to Ch_3 and Ch_4. This could be due to the differences in the spectrum of Ch_1 compared to those of Ch_3 and Ch_4 (column 2 in **Figure 2C**) which tend to be in the higher frequencies.

In summary, the periodic stimulus group was less favorable considering the higher subjective visual fatigue level compared to the quasi-periodic and chaotic stimulus groups. For designing homogenous BCI, it is recommended to optimize the quasi-periodic and chaotic groups' orthogonal stimuli by evaluating their auto- and cross-correlation functions while at the same time choosing appropriate frequency bandwidths to minimize variations in the subjective visual fatigue.

Between-Group Subjective Fatigue Rate Evaluation

The comparison of the subjective visual fatigue rates of the periodic, quasi-periodic, and chaotic stimulus groups showed that the chaotic group caused less visual fatigue compared to the other two stimulus groups. The quasi-periodic group caused lower levels of visual fatigue compared to the periodic one (**Figure 8**). These results indicate the superiority of using the chaotic group for designing new comfortable and ergonomic VEP-based BCIs. Our recent study also showed that visual stimuli with chaotic characteristics lead to significantly less visual fatigue (Shirzhiyan et al., 2019).

The visual fatigue reduction seen in the chaotic and even the quasi-periodic stimuli group could be due to their dynamical and complex nature which is more compatible with the visual system compared to synthesized single frequencies. It has been shown that the periodic stimuli that exist in nature are not very pure in tone and have more than a single frequency as they contain quasi-rhythmic components and have a complex dynamical structure (Kayser et al., 2003; Butts et al., 2007; Mazzoni et al., 2011).

A simple deterministic dynamical system is also able to generate extremely unpredictable, divergent, and fractal behaviors (Boeing, 2016). These behaviors contain infinitely self-similar patterns avoiding exact repetition (periodicity). It is shown that fractal images and natural patterns are more appealing compared to the synthesized ones (Redies, 2007; Chapeau-Blondeau et al., 2009; Hagerhall et al., 2015; Kardan et al., 2015), and also the chaotic patterns with high fractal dimension and Lyapunov exponent are more aesthetically pleasing (Aks and Sprott, 1996). The application of natural visual stimuli that have sparse encoding can induce a resonant state leading to the adaptation of the visual system to natural patterns (Sekuler and Bennett, 2001; Redies, 2007) and possibly lower visual discomfort. The results of the current study show that visual fatigue scores are lower for quasi-periodic and chaotic stimuli compared to periodic stimuli having one frequency. It is speculated that this could be potentially because of the adaptation of the visual system to the presented complex dynamical stimuli.

Additionally, the chaotic, quasi-periodic, and periodic stimulus groups were closer to the $1/f$ amplitude spectrum, with the chaotic group being the nearest (dotted black line in the amplitude spectra plots of **Figure 2**). This pattern matches our results of the VAS scores as the chaotic group had the least visual fatigue in the same order with the other two groups. The relevance of a comparatively less visual fatigue and nearness of the chaotic stimuli to the $1/f$ amplitude spectrum is in agreement with previous studies reporting that the visual system encodes stimuli with $1/f$ amplitude spectral information (Tan and Yao, 2009; Ellemberg et al., 2012; Isherwood et al., 2017; Yoshimoto et al., 2017).

Limitations and Plans for Future Studies

Our study has several limitations. Within-group analysis of the subjective visual fatigue rates shows significant differences in all the three groups. In practice, it is not favorable that different target stimuli have different discomfort levels. To avoid possible within-group differences, the parameters of the stimulus-generated maps for each stimulus group could be selected in order to have similar stimulus spectra while at the same time preferring a higher frequency range instead of a lower one.

In this research, we did not study the optimization process for selecting the appropriate parameters of the quasi-periodic and chaotic stimuli. For future studies, optimization of the parameters with the aim of having a sharper and greater auto-correlation function of the target stimuli and a lower cross-correlation function with other stimuli should be considered. It is possible that such optimization can lead to better accuracy results.

The calculated lag times from the cross-correlation functions in three different stimulus groups represent interesting patterns. As can be seen from **Figure 9**, the obtained delay for each target stimulus group differed from each other mainly in the periodic stimulus group, while this delay was almost constant among the target stimuli in the quasi-periodic and chaotic stimulus groups. The diversity in the delay lags could potentially be due to the non-linear behavior of the visual system to the presented input. We plan to investigate this effect in a future project and study the reasons leading to differential system delays for different stimulus characteristics.

In our previous study, we have shown that using binary chaotic codes *versus* *m*-sequences could decrease the subjective visual fatigue, and this could be used as a modulating code in c-VEP-based BCIs. Additionally, in this study, we found that the chaotic stimulus group provided very high discrimination between its individual stimuli, Ch_1–Ch_4, and could reduce the fatigue rate better when compared to the traditional stimuli for VEP generation (periodic stimuli). For further studies, it would be probably feasible to attempt using chaotic stimuli generated from other chaotic maps (such as Hanon map) and as short as codes commonly used in c-VEP studies. This could lead to the design of more comfortable and ergonomic c-VEP-based BCIs.

CONCLUSION

In this study, we introduced for the first time quasi-periodic and chaotic visual stimuli for evoking VEPs in order to use them in VEP-based BCIs and compared them with traditional periodic stimuli used in SSVEP-based studies. The presented complex stimuli (quasi-periodic and chaotic stimuli) satisfy the necessities for use as visual stimuli in VEP-based BCIs. The best target identification accuracy was obtained for the chaotic stimuli. Potentially due to the more natural-like characteristics of the chaotic and quasi-periodic stimuli, they led to less subjective visual fatigue compared to the periodic stimulus group.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/**Supplementary Material**.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the office of research review board and the research ethics committee of the Tehran University of Medical Sciences with the LREC protocol number IR.TUMS.REC.1394.2110. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

ZS: data acquisition, analysis and interpretation, and first draft of manuscript. AK: data acquisition and analysis and contribution to manuscript writing. MF: hardware design and data acquisition. ES and MG: data acquisition and analysis. AM: study design,

hardware, data interpretation, and supervision. MH: data interpretation, design, overall supervision, and manuscript review. AJ: study design, funding, project supervision, and principal investigator. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by Tehran University of Medical Sciences (grant no. 30946).

ACKNOWLEDGMENTS

The authors would like to thank all subjects of this study for their cooperation.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2020.534619/full#supplementary-material

REFERENCES

Aihara, K. (1989). Chaotic neural networks (Bifurcation phenomena in nonlinear systems and theory of dynamical systems). *Res. Instit. Math. Analys.* 710, 145–163.

Aihara, K., Matsumoto, G., and Ikegaya, Y. (1984). Periodic and non-periodic responses of a periodically forced Hodgkin-Huxley oscillator. *J. Theoret. Biol.* 109, 249–269. doi: 10.1016/s0022-5193(84)80005-3

Aitken, R. C. (1969). A growing edge of measurement of feelings [Abridged] measurement of feelings using visual analogue scales. *J. R. Soc. Med.* 62, 989–993.

Ajami, S., Mahnam, A., and Abootalebi, V. (2018). Development of a practical high frequency brain-computer interface based on steady-state visual evoked potentials using a single channel of EEG. *Biocybernet. Biomed. Eng.* 38, 106–114. doi: 10.1016/j.bbe.2017.10.004

Aks, D. J., and Sprott, J. C. (1996). Quantifying aesthetic preference for chaotic patterns. *Empir. Stud. Arts* 14, 1–16. doi: 10.2190/6v31-7m9r-t9l5-cdg9

Allison, B., Luth, T., Valbuena, D., Teymourian, A., Volosyak, I., and Graser, A. (2010). BCI demographics: how many (and what kinds of) people can use an SSVEP BCI? *IEEE Trans. Neural Syst. Rehabil. Eng.* 18, 107–116. doi: 10.1109/tnsre.2009.2039495

Allison, B. Z., McFarland, D. J., Schalk, G., Zheng, S. D., Jackson, M. M., and Wolpaw, J. R. (2008). Towards an independent brain-computer interface using steady state visual evoked potentials. *Clin. Neurophysiol.* 119, 399–408. doi: 10.1016/j.clinph.2007.09.121

Aminaka, D., Makino, S., and Rutkowski, T. M. (2015). "Eeg filtering optimization for code-modulated chromatic visual evoked potential-based brain-computer interface," in *Proceedings of the International Workshop on Symbiotic Interaction*, Berlin.

Atick, J. J., and Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Comput.* 4, 196–210. doi: 10.1162/neco.1992.4.2.196

Attinger, E., Anne, A., and McDonald, D. (1966). Use of fourier series for the analysis of biological systems. *Biophys. J.* 6:291. doi: 10.1016/s0006-3495(66)86657-2

Bijur, P. E., Silver, W., and Gallagher, E. J. (2001). Reliability of the visual analog scale for measurement of acute pain. *Acad. Emerg. Med.* 8, 1153–1157. doi: 10.1111/j.1553-2712.2001.tb01132.x

Bin, G., Gao, X., Wang, Y., Hong, B., and Gao, S. (2009a). VEP-based brain-computer interfaces: time, frequency, and code modulations [Research Frontier]. *IEEE Comput. Intellig. Magaz.* 4, 22–26. doi: 10.1109/mci.2009.934562

Bin, G., Gao, X., Yan, Z., Hong, B., and Gao, S. (2009b). An online multi-channel SSVEP-based brain-computer interface using a canonical correlation analysis method. *J. Neural Eng.* 6:046002. doi: 10.1088/1741-2560/6/4/046002

Bin, G., Gao, X., Wang, Y., Li, Y., Hong, B., and Gao, S. (2011). A high-speed BCI based on code modulation VEP. *J. Neural Eng.* 8:025015. doi: 10.1088/1741-2560/8/2/025015

Blake, R., and Lee, S.-H. (2005). The role of temporal structure in human vision. *Behav. Cogn. Neurosci. Rev.* 4, 21–42. doi: 10.1177/1534582305276839

Boeing, G. (2016). Visual analysis of nonlinear dynamical systems: chaos, fractals, self-similarity and the limits of prediction. *Systems* 4:37. doi: 10.3390/systems4040037

Butts, D. A., Weng, C., Jin, J., Yeh, C.-I., Lesica, N. A., Alonso, J.-M., et al. (2007). Temporal precision in the neural code and the timescales of natural vision. *Nature* 449:92. doi: 10.1038/nature06105

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Bonabeau, E., and Theraula, G. (2003). *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press.

Chang, M. H., Baek, H. J., Lee, S. M., and Park, K. S. (2014). An amplitude-modulated visual stimulation for reducing eye fatigue in SSVEP-based brain-computer interfaces. *Clin. Neurophysiol.* 125, 1380–1391. doi: 10.1016/j.clinph.2013.11.016

Chapeau-Blondeau, F., Chauveau, J., Rousseau, D., and Richard, P. (2009). Fractal structure in the color distribution of natural images. *Chaos Solitons Fract.* 42, 472–482. doi: 10.1016/j.chaos.2009.01.008

Chen, X., Wang, Y., Nakanishi, M., Gao, X., Jung, T.-P., and Gao, S. (2015). High-speed spelling with a noninvasive brain-computer interface. *Proc. Natl. Acad. Sci. U.S.A.* 112, E6058–E6067.

Chien, Y.-Y., Lin, F.-C., Zao, J. K., Chou, C.-C., Huang, Y.-P., Kuo, H.-Y., et al. (2017). Polychromatic SSVEP stimuli with subtle flickering adapted to brain-display interactions. *J. Neural Eng.* 14:016018. doi: 10.1088/1741-2552/aa550d

Choi, I. (2017). *A Sensorimotor Rhythm (SMR)-Based Brain-Computer Interface (BCI) Controlled Functional Electrical Stimulation (FES) for Restoration of Hand Grasping and Extension Functions*. Raleigh, CA: NC State University Libraries.

Chumerin, N., Manyakov, N. V., van Vliet, M., Robben, A., Combaz, A., and Van Hulle, M. M. (2012). Steady-state visual evoked potential-based computer gaming on a consumer-grade EEG device. *IEEE Trans. Comput. Intellig. Games* 5, 100–110. doi: 10.1109/tciaig.2012.2225623

Correa, A., and Nobre, A. C. (2008). Neural modulation by regularity and passage of time. *J. Neurophysiol.* 100, 1649–1655. doi: 10.1152/jn.90656.2008

Costantino, R., Desharnais, R., Cushing, J., and Dennis, B. (1997). Chaotic dynamics in an insect population. *Science* 275, 389–391. doi: 10.1126/science.275.5298.389

Crichton, N. (2001). Visual analogue scale (VAS). *J. Clin. Nurs.* 10:706.

DeTommaso, M., Sciruicchio, V., Guido, M., Sasanelli, G., and Puca, F. (1999). Steady-state visual-evoked potentials in headache: diagnostic value in migraine and tension-type headache patients. *Cephalalgia* 19, 23–26. doi: 10.1046/j.1468-2982.1999.1901023.x

Ellemberg, D., Hansen, B. C., and Johnson, A. (2012). The developing visual system is not optimally sensitive to the spatial statistics of natural images. *Vis. Res.* 67, 1–7.

Essl, G. (2006). *Circle Maps as Simple Oscillators for Complex Behavior: I. Basics*. Available online at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.5143&rep=rep1&type=pdf doi: 10.1016/j.visres.2012.06.018 (accessed January 2, 2020).

Feng, P., Wu, Y., and Zhang, J. (2017). A Route to chaotic behavior of single neuron exposed to external electromagnetic radiation. *Front. Comput. Neurosci.* 11:94. doi: 10.3389/fncom.2017.00094

Fisher, R. S., Harding, G., Erba, G., Barkley, G. L., and Wilkins, A. (2005). Photic- and pattern-induced seizures: a review for the epilepsy foundation of america working group. *Epilepsia* 46, 1426–1441. doi: 10.1111/j.1528-1167.2005.31405.x

Freeman, W. J. (1992). Tutorial on neurobiology: from single neurons to brain chaos. *Intern. J. Bifurc. Chaos* 2, 451–482. doi: 10.1142/s0218127492000653

Gao, S., Wang, Y., Gao, X., and Hong, B. (2014). Visual and auditory brain-computer interfaces. *IEEE Trans. Biomed. Eng.* 61, 1436–1447. doi: 10.1109/tbme.2014.2300164

Gift, A. G. (1989). Visual analogue scales: measurement of subjective phenomena. *Nurs. Res.* 38, 286–288.

Grant, S., Aitchison, T., Henderson, E., Christie, J., Zare, S., Mc Murray, J., et al. (1999). A comparison of the reproducibility and the sensitivity to change of visual analogue scales, Borg scales, and Likert scales in normal subjects during submaximal exercise. *Chest* 116, 1208–1217. doi: 10.1378/chest.116.5.1208

Guevara, M. A., and Corsi-Cabrera, M. (1996). EEG coherence or EEG correlation? *Intern. J. Psychophysiol.* 23, 145–153. doi: 10.1016/s0167-8760(96)00038-4

Guger, C., Allison, B., and Edlinger, G. (2013). "State of the art in bci research: BCI award 2011," in *Brain-Computer Interface Research*, ed. R. Fazel-Rezai (London: IntechOpen), 1–5. doi: 10.1007/978-3-642-36083-1_1

Guo, F., Hong, B., Gao, X., and Gao, S. (2008). A brain-computer interface using motion-onset visual evoked potential. *J. Neural Eng.* 5:477. doi: 10.1088/1741-2560/5/4/011

Gysels, E., and Celka, P. (2004). Phase synchronization for the recognition of mental tasks in a brain-computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* 12, 406–415. doi: 10.1109/tnsre.2004.838443

Haegens, S., and Golumbic, E. Z. (2018). Rhythmic facilitation of sensory processing: a critical review. *Neurosci. Biobehav. Rev.* 86, 150–165. doi: 10.1016/j.neubiorev.2017.12.002

Hagerhall, C. M., Laike, T., Kuller, M., Marcheschi, E., Boydston, C., and Taylor, R. (2015). Human physiological benefits of viewing nature: EEG responses to exact and statistical fractal patterns. *Nonlinear Dyn. Psychol. Life Sci.* 19, 1–12.

Han, C., Xu, G., Xie, J., Chen, C., and Zhang, S. (2018). Highly interactive brain-computer interface based on flicker-free steady-state motion visual evoked potential. *Sci. Rep.* 8:5835.

Hoebeek, F. E., Witter, L., Ruigrok, T. J., and De Zeeuw, C. I. (2010). Differential olivo-cerebellar cortical control of rebound activity in the *Cerebellar nuclei*. *Proc. Natl. Acad. Sci. U.S.A.* 107, 8410–8415. doi: 10.1073/pnas.0907118107

Holz, E. M., Höhne, J., Staiger-Sälzer, P., Tangermann, M., and Kübler, A. (2013). Brain-computer interface controlled gaming: evaluation of usability by severely motor restricted end-users. *Artif. Intellig. Med.* 59, 111–120. doi: 10.1016/j.artmed.2013.08.001

Hong, K.-S. (2011). Synchronization of coupled chaotic FitzHugh-Nagumo neurons via lyapunov functions. *Math. Comput. Simul.* 82, 590–603. doi: 10.1016/j.matcom.2011.10.005

Isherwood, Z. J., Schira, M. M., and Spehar, B. (2017). The tuning of human visual cortex to variations in the 1/fα amplitude spectra and fractal properties of synthetic noise images. *Neuroimage* 146, 642–657. doi: 10.1016/j.neuroimage.2016.10.013

Ishikawa, T., Shimuta, M., and Häusser, M. (2015). Multimodal sensory integration in single cerebellar granule cells in vivo. *eLife* 4:e12916.

Jiang, J., Yin, E., Wang, C., Xu, M., and Ming, D. (2018). Incorporation of dynamic stopping strategy into the high-speed SSVEP-based BCIs. *J. Neural Eng.* 15:046025. doi: 10.1088/1741-2552/aac605

Jin, J., Sellers, E. W., Zhou, S., Zhang, Y., Wang, X., and Cichocki, A. (2005). A P300 brain-computer interface based on a modification of the mismatch negativity paradigm. *Intern. J. Neural Syst.* 25:1550011. doi: 10.1142/s0129065715500112

Kapgate, D., and Kalbande, D. (2015). "A review on visual brain computer interface," in *Advancements of Medical Electronics. Lecture Notes in Bioengineering*, eds S. Gupta, S. Bag, K. Ganguly, I. Sarkar, and P. Biswas (New Delhi: Springer), 193–206.

Kardan, O., Demiralp, E., Hout, M. C., Hunter, M. R., Karimi, H., Hanayik, T., et al. (2015). Is the preference of natural versus man-made scenes driven by bottom-up processing of the visual features of nature? *Front. Psychol.* 6:471. doi: 10.3389/fncom.2017.00471

Käthner, I., Wriessnegger, S. C., Müller-Putz, G. R., Kübler, A., and Halder, S. (2014). Effects of mental workload and fatigue on the P300, alpha and theta band power during operation of an ERP (P300) brain-computer interface. *Biol. Psychol.* 102, 118–129. doi: 10.1016/j.biopsycho.2014.07.014

Kayser, C., Salazar, R. F., and Konig, P. (2003). Responses to natural scenes in cat V1. *J. Neurophysiol.* 90, 1910–1920. doi: 10.1152/jn.00195.2003

Keihani, A., Shirzhiyan, Z., Farahi, M., Shamsi, E., Mahnam, A., Makkiabadi, B., et al. (2018). Use of sine shaped high-frequency rhythmic visual stimuli patterns for SSVEP response analysis and fatigue rate evaluation in normal subjects. *Front. Hum. Neurosci.* 12:201. doi: 10.3389/fnhum.2018.00201

Keitel, C., Thut, G., and Gross, J. (2017). Visual cortex responses reflect temporal structure of continuous quasi-rhythmic sensory stimulation. *Neuroimage* 146, 58–70. doi: 10.1016/j.neuroimage.2016.11.043

Klimek, L., Bergmann, K.-C., Biedermann, T., Bousquet, J., Hellings, P., Jung, K., et al. (2017). Visual analogue scales (VAS): measuring instruments for the documentation of symptoms and therapy monitoring in cases of allergic rhinitis in everyday health care. *Allergo J. Intern.* 26, 16–24. doi: 10.1007/s40629-016-0006-7

Krusienski, D. J., McFarland, D. J., and Wolpaw, J. R. (2012). Value of amplitude, phase, and coherence features for a sensorimotor rhythm-based brain-computer interface. *Brain Res. Bull.* 87, 130–134. doi: 10.1016/j.brainresbull.2011.09.019

La Rocca, D., Campisi, P., Vegso, B., Cserti, P., Kozmann, G., Babiloni, F., et al. (2014). Human brain distinctiveness based on EEG spectral coherence connectivity. *IEEE Trans. Biomed. Eng.* 61, 2406–2412. doi: 10.1109/tbme.2014.2317881

Lachaux, J. P., Rodriguez, E., Martinerie, J., and Varela, F. J. (1999). Measuring phase synchrony in brain signals. *Hum. Brain Mapp.* 8, 194–208. doi: 10.1002/(sici)1097-0193(1999)8:4<194::aid-hbm4>3.0.co;2-c

Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., et al. (2005). Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment. *EURASIP J. Adv. Signal. Process.* 2005:706906.

Lasley, D. J., and Cohn, T. (1981). Detection of a luminance increment: effect of temporal uncertainty. *JOSA* 71, 845–850. doi: 10.1364/josa.71.000845

Lebedev, M. A., and Nicolelis, M. A. (2006). Brain-machine interfaces: past, present and future. *Trends Neurosci.* 29, 536–546. doi: 10.1016/j.tins.2006.07.004

Lee, P.-L., Hsieh, J.-C., Wu, C.-H., Shyu, K.-K., Chen, S.-S., Yeh, T.-C., et al. (2006). The brain computer interface using flash visual evoked potential and independent component analysis. *Ann. Biomed. Eng.* 34, 1641–1654. doi: 10.1007/s10439-006-9175-8

Lee, P.-L., Hsieh, J.-C., Wu, C.-H., Shyu, K.-K., and Wu, Y.-T. (2008). Brain computer interface using flash onset and offset visual evoked potentials. *Clin. Neurophysiol.* 119, 605–616. doi: 10.1016/j.clinph.2007.11.013

Lee, P.-L., Yeh, C.-L., Cheng, J. Y.-S., Yang, C.-Y., and Lan, G.-Y. (2011). An SSVEP-based BCI using high duty-cycle visual flicker. *IEEE Trans. Biomed. Eng.* 58, 3350–3359. doi: 10.1109/tbme.2011.2162586

Liew, S.-H., Choo, Y.-H., Low, Y. F., and Yusoh, Z. I. M. (2015). Identifying visual evoked potential (VEP) electrodes setting for person authentication. *Int. J. Adv. Soft Comput. Appl.* 7, 85–99.

Lin, K., Cinetto, A., Wang, Y., Chen, X., Gao, S., and Gao, X. (2016). An online hybrid BCI system based on SSVEP and EMG. *J. Neural Eng.* 13:026020. doi: 10.1088/1741-2560/13/2/026020

Lin, Z., Zhang, C., Wu, W., and Gao, X. (2006). Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *IEEE Trans. Biomed. Eng.* 53, 2610–2614. doi: 10.1109/tbme.2006.886577

Liu, Y., Wei, Q., and Lu, Z. (2018). A multi-target brain-computer interface based on code modulated visual evoked potentials. *PLoS One* 13:e0202478. doi: 10.1371/journal.pone.202478

Llinás, R. R. (1988). The intrinsic electrophysiological properties of mammalian neurons: insights into central nervous system function. *Science* 242, 1654–1664.

Lv, M., Wang, C., Ren, G., Ma, J., and Song, X. (2016). Model of electrical activity in a neuron under magnetic flow effect. *Nonlinear Dyn.* 85, 1479–1490. doi: 10.1007/s11071-016-2773-6

Ma, J., and Tang, J. (2017). A review for dynamics in neuron and neuronal network. *Nonlinear Dyn.* 89, 1569–1578. doi: 10.1007/s11071-017-3565-3

Martinez, P., Bakardjian, H., and Cichocki, A. (2007). Fully online multicommand brain-computer interface with visual neurofeedback using SSVEP paradigm. *Comput. Intellig. Neurosci.* 94561. doi: 10.1155/2007/94561

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature* 261:459. doi: 10.1038/261459a0

Mazzoni, A., Brunel, N., Cavallari, S., Logothetis, N. K., and Panzeri, S. (2011). Cortical dynamics during naturalistic sensory stimulations: experiments and models. *J. Physiol. Paris* 105, 2–15. doi: 10.1016/j.jphysparis.2011.07.014

McCane, L. M., Heckman, S. M., McFarland, D. J., Townsend, G., Mak, J. N., Sellers, E. W., et al. (2015). P300-based brain-computer interface (BCI) event-related

potentials (ERPs): people with amyotrophic lateral sclerosis (ALS) vs. age-matched controls. *Clin. Neurophysiol.* 126, 2124–2131. doi: 10.1016/j.clinph.2015.01.013

Middendorf, M., McMillan, G., Calhoun, G., and Jones, K. S. (2000). Brain-computer interfaces based on the steady-state visual-evoked response. *IEEE Trans. Rehabil. Eng.* 8, 211–214. doi: 10.1109/86.847819

Molina, R., Relaño, A., Retamosa, J., Muñoz, L., Faleiro, E., and Gómez, J. (2010). "Perspectives on 1/f noise in quantum chaos," in *Proceedings of the Journal of Physics: Conference Series, Volume 239, XXXIII Symposium on Nuclear Physics 5–8 January 2010*, Morelos.

Muller-Putz, G. R., and Pfurtscheller, G. (2008). Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Trans. Biomed. Eng.* 55, 361–364. doi: 10.1109/tbme.2007.897815

Müller-Putz, G. R., Scherer, R., Brauneis, C., and Pfurtscheller, G. (2005). Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *J. Neural Eng.* 2:123.

Nakanishi, M., Wang, Y., Wang, Y. T., Mitsukura, Y., and Jung, T. P. (2014). A high-speed brain speller using steady-state visual evoked potentials. *Intern. J. Neural Syst.* 24:1450019. doi: 10.1142/s0129065714500191

Nobukawa, S., and Nishimura, H. (2016). Chaotic resonance in coupled inferior olive neurons with the Llinás approach neuron model. *Neural Comput.* 28, 2505–2532. doi: 10.1162/neco_a_00894

Nunez, P. L., Srinivasan, R., Westdorp, A. F., Wijesinghe, R. S., Tucker, D. M., Silberstein, R. B., et al. (1997). EEG coherency: I: statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalogr. Clin. Neurophysiol.* 103, 499–515.

Obleser, J., Henry, M. J., and Lakatos, P. (2017). What do we talk about when we talk about rhythm? *PLoS Biol.* 15:e2002794. doi: 10.1371/journal.pbio.2002794

Parini, S., Maggi, L., Turconi, A. C., and Andreoni, G. (2009). A robust and self-paced BCI system based on a four class SSVEP paradigm: algorithms and protocols for a high-transfer-rate direct brain communication. *Comput. Intellig. Neurosci.* 2009:864564.

Paul-Dauphin, A., Guillemin, F., Virion, J.-M., and Briançon, S. (1999). Bias and precision in visual analogue scales: a randomized controlled trial. *Am. J. Epidemiol.* 150, 1117–1127. doi: 10.1093/oxfordjournals.aje.a009937

Petrov, V., Ouyang, Q., and Swinney, H. L. (1997). Resonant pattern formation in achemical system. *Nature* 388:655. doi: 10.1038/41732

Potapov, A., and Ali, M. (2000). Robust chaos in neural networks. *Phys. Lett. A* 277, 310–322.

Redies, C. (2007). A universal model of esthetic perception based on the sensory coding of natural stimuli. *Spat. Vis.* 21, 97–117. doi: 10.1163/156856808782713780

Relano, A., Gómez, J., Molina, R., Retamosa, J., and Faleiro, E. (2002). Quantum chaos and 1/f noise. *Phys. Rev. Lett.* 89:244102.

Riechmann, H., Finke, A., and Ritter, H. (2016). Using a cVEP-based brain-computer interface to control a virtual agent. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 692–699. doi: 10.1109/tnsre.2015.2490621

Rössert, C., Dean, P., and Porrill, J. (2015). At the edge of chaos: how cerebellar granular layer network dynamics can provide the basis for temporal filters. *PLoS Comput. Biol.* 11:e1004515. doi: 10.1371/journal.pone.1004515

Saha, T., and Galic, M. (2018). Self-organization across scales: from molecules to organisms. *Philos. Trans. R. Soc. B* 373:20170113. doi: 10.1098/rstb.2017.0113

Sakurada, T., Kawase, T., Komatsu, T., and Kansaku, K. (2015). Use of high-frequency visual stimuli above the critical flicker frequency in a SSVEP-based BMI. *Clin. Neurophysiol.* 126, 1972–1978. doi: 10.1016/j.clinph.2014.12.010

Sanei, S., and Chambers, J. A. (2007). *EEG Signal Processing*. New York, NY: Wiley.

Sekuler, A. B., and Bennett, P. J. (2001). Visual neuroscience: resonating to natural images. *Curr. Biol.* 11, R733–R736.

Shirzhiyan, Z., Keihani, A., Farahi, M., Shamsi, E., GolMohammadi, M., Mahnam, A., et al. (2019). Introducing chaotic codes for the modulation of code modulated visual evoked potentials (c-VEP) in normal adults for visual fatigue reduction. *PLoS One* 14:e0213197. doi: 10.1371/journal.pone.0213197

Spüler, M. (2017). A high-speed brain-computer interface (BCI) using dry EEG electrodes. *PLoS One* 12:e0172400. doi: 10.1371/journal.pone.0172400

Strogatz, S. H. (2018). *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering*. Boca Raton, FL: CRC Press.

Suzuki, Y., Lu, M., Ben-Jacob, E., and Onuchic, J. N. (2016). Periodic, quasi-periodic and chaotic dynamics in simple gene elements with time delays. *Sci. Rep.* 6:21037.

Takano, K., Komatsu, T., Hata, N., Nakajima, Y., and Kansaku, K. (2009). Visual stimuli for the P300 brain-computer interface: a comparison of white/gray and green/blue flicker matrices. *Clin. Neurophysiol.* 120, 1562–1566. doi: 10.1016/j.clinph.2009.06.002

Tan, Z., and Yao, H. (2009). The spatiotemporal frequency tuning of LGN receptive field facilitates neural discrimination of natural stimuli. *J. Neurosci.* 29, 11409–11416. doi: 10.1523/jneurosci.1268-09.2009

Tseng, B. Y., Gajewski, B. J., and Kluding, P. M. (2010). Reliability, responsiveness, and validity of the visual analog fatigue scale to measure exertion fatigue in people with chronic stroke: a preliminary study. *Stroke Res. Treat.* 2010: 412964.

Vaid, S., Singh, P., and Kaur, C. (2015). "EEG signal analysis for BCI interface: a review," in *Proceedings of the 2015 Fifth International Conference on Advanced Computing & Communication Technologies*, Haryana.

Verwulgen, S., Lacko, D., Justine, H., Kustermans, S., Moons, S., Thys, F., et al. (2018). "Determining comfortable pressure ranges for wearable EEG headsets," in *Proceedings of the International Conference on Applied Human Factors and Ergonomics*, Cham.

Vialatte, F.-B., Maurice, M., Dauwels, J., and Cichocki, A. (2010). Steady-state visually evoked potentials: focus on essential paradigms and future perspectives. *Prog. Neurobiol.* 90, 418–438. doi: 10.1016/j.pneurobio.2009.11.005

Volosyak, I., Valbuena, D., Luth, T., Malechka, T., and Graser, A. (2011). BCI demographics II: how many (and what kinds of) people can use a high-frequency SSVEP BCI? *IEEE Trans. Neural Syst. Rehabil. Eng.* 19, 232–239. doi: 10.1109/tnsre.2011.2121919

Wang, Y., Wang, R., Gao, X., Hong, B., and Gao, S. (2006). A practical VEP-based brain-computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 234–240. doi: 10.1109/tnsre.2006.875576

Wang, Y.-T., Nakanishi, M., Wang, Y., Wei, C.-S., Cheng, C.-K., and Jung, T.-P. (2017). An online brain-computer interface based on SSVEPs measured from non-hair-bearing areas. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 14–21. doi: 10.1109/tnsre.2016.2573819

Wang, Y., Chen, X., Gao, X., and Gao, S. (2017). A benchmark dataset for SSVEP-based brain-computer interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 1746–1752. doi: 10.1109/tnsre.2016.2627556

Wang, Z., Yu, Y., Xu, M., Liu, Y., Yin, E., and Zhou, Z. (2018). Towards a Hybrid BCI gaming paradigm based on motor imagery and SSVEP. *Intern. J. Hum. Comput. Interact.* 35, 197–205. doi: 10.1080/10447318.2018.1445068

Wei, Q., Feng, S., and Lu, Z. (2016). Stimulus specificity of brain-computer interfaces based on code modulation visual evoked potentials. *PLoS One* 11:e0156416. doi: 10.1371/journal.pone.0156416

Wei, Q., Liu, Y., Gao, X., Wang, Y., Yang, C., Lu, Z., et al. (2018). A novel c-VEP BCI paradigm for increasing the number of stimulus targets based on grouping modulation with different codes. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 1178–1187. doi: 10.1109/tnsre.2018.2837501

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* 113, 767–791.

Won, D.-O., Hwang, H.-J., Dähne, S., Müller, K.-R., and Lee, S.-W. (2015). Effect of higher frequency on the classification of steady-state visual evoked potentials. *J. Neural Eng.* 13:016014. doi: 10.1088/1741-2560/13/1/016014

Xie, J., Xu, G., Luo, A., Li, M., Zhang, S., Han, C., et al. (2017). The role of visual noise in influencing mental load and fatigue in a steady-state motion visual evoked potential-based brain-computer interface. *Sensors* 17:1873. doi: 10.3390/s17081873

Xie, J., Xu, G., Wang, J., Li, M., Han, C., and Jia, Y. (2016). Effects of mental load and fatigue on steady-state evoked potential based brain computer interface tasks: a comparison of periodic flickering and motion-reversal based visual attention. *PLoS One* 11:e0163426. doi: 10.1371/journal.pone.00163426

Xie, J., Xu, G., Wang, J., Zhang, F., and Zhang, Y. (2012). Steady-state motion visual evoked potentials produced by oscillating newton's rings: implications for brain-computer interfaces. *PLoS One* 7:e39707. doi: 10.1371/journal.pone.039707

Xie, J., Xu, G., Zhao, X., Li, M., Wang, J., Han, C., et al. (2018). Enhanced Plasticity of Human Evoked Potentials by Visual Noise During the Intervention of Steady-State Stimulation Based Brain-Computer Interface. *Front. Neurorobot.* 12:82. doi: 10.3389/fnbot.2018.00082

Yan, W., Xu, G., Li, M., Xie, J., Han, C., Zhang, S., et al. (2017). Steady-state motion visual evoked potential (SSMVEP) based on equal luminance colored enhancement. *PLoS One* 12:e0169642. doi: 10.1371/journal.pone.0016 9642

Yin, E., Zhou, Z., Jiang, J., Yu, Y., and Hu, D. (2014). A dynamically optimized SSVEP brain-computer interface (BCI) speller. *IEEE Trans. Biomed. Eng.* 62, 1447–1456. doi: 10.1109/tbme.2014.2320948

Yin, E., Zhou, Z., Jiang, J., Yu, Y., and Hu, D. (2015). A dynamically optimized SSVEP brain-computer interface (BCI) speller. *IEEE Trans. Biomed. Eng.* 62, 1447–1456.

Yoshimoto, S., Garcia, J., Jiang, F., Wilkins, A. J., Takeuchi, T., and Webster, M. A. (2017). Visual discomfort and flicker. *Vis. Res.* 138, 18–28. doi: 10.1016/j.visres. 2017.05.015

Zhanabaev, Z. Z., Grevtseva, T. Y., and Kozhagulov, Y. (2016). Nonlinear characteristics of neural signals. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1610.01147#:~:text=We%20have%20shown%20that% 20time,%2Dsimilarity%20and%20self%2Daffinity (accessed January 13, 2020).

Zhanabaev, Z. Z., and Kozhagulov, Y. T. (2013). A generic model for scale-invariant neural networks. *J. Neurosci. Neuroeng.* 2, 267–271. doi: 10.1166/jnsne.2013. 1057

Zhang, Y., Zhou, G., Jin, J., Wang, M., Wang, X., and Cichocki, A. (2013). L1-regularized multiway canonical correlation analysis for SSVEP-based BCI. *IEEE Trans. Neural Syst. Rehabil. Eng.* 21, 887–896. doi: 10.1109/tnsre.2013.2279680

Zhang, Y., Zhou, G., Jin, J., Wang, X., and Cichocki, A. (2014). Frequency recognition in SSVEP-based BCI using multiset canonical correlation analysis. *Intern. J. Neural Syst.* 24:1450013. doi: 10.1142/s0129065714500130

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Shirzhiyan, Keihani, Farahi, Shamsi, GolMohammadi, Mahnam, Haidari and Jafari. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 01 February 2021
doi: 10.3389/fnins.2021.580909

Non-linear Memristive Synaptic Dynamics for Efficient Unsupervised Learning in Spiking Neural Networks

Stefano Brivio[1]*, *Denys R. B. Ly*[2], *Elisa Vianello*[2] and *Sabina Spiga*[1]

[1] CNR - IMM, Unit of Agrate Brianza, Agrate Brianza, Italy, [2] Université Grenoble Alpes, CEA, Leti, Grenoble, France

Spiking neural networks (SNNs) are a computational tool in which the information is coded into spikes, as in some parts of the brain, differently from conventional neural networks (NNs) that compute over real-numbers. Therefore, SNNs can implement intelligent information extraction in real-time at the edge of data acquisition and correspond to a complementary solution to conventional NNs working for cloud-computing. Both NN classes face hardware constraints due to limited computing parallelism and separation of logic and memory. Emerging memory devices, like resistive switching memories, phase change memories, or memristive devices in general are strong candidates to remove these hurdles for NN applications. The well-established training procedures of conventional NNs helped in defining the desiderata for memristive device dynamics implementing synaptic units. The generally agreed requirements are a linear evolution of memristive conductance upon stimulation with train of identical pulses and a symmetric conductance change for conductance increase and decrease. Conversely, little work has been done to understand the main properties of memristive devices supporting efficient SNN operation. The reason lies in the lack of a background theory for their training. As a consequence, requirements for NNs have been taken as a reference to develop memristive devices for SNNs. In the present work, we show that, for efficient CMOS/memristive SNNs, the requirements for synaptic memristive dynamics are very different from the needs of a conventional NN. System-level simulations of a SNN trained to classify hand-written digit images through a spike timing dependent plasticity protocol are performed considering various linear and non-linear plausible synaptic memristive dynamics. We consider memristive dynamics bounded by artificial hard conductance values and limited by the natural dynamics evolution toward asymptotic values (soft-boundaries). We quantitatively analyze the impact of resolution and non-linearity properties of the synapses on the network training and classification performance. Finally, we demonstrate that the non-linear synapses with hard boundary values enable higher classification performance and realize the best trade-off between classification accuracy and required training time. With reference to the obtained results, we discuss how memristive devices with non-linear dynamics constitute a technologically convenient solution for the development of on-line SNN training.

Keywords: spiking neural network, MNIST, neuromorphic, analog memory, STDP, memristive synapse, memristor, memristive devices

OPEN ACCESS

Edited by:
Michael Pfeiffer, Bosch Center for Artificial Intelligence, Germany

Reviewed by:
Hesham Mostafa, Intel, United States
Gopalakrishnan Srinivasan, MediaTek, Taiwan

***Correspondence:**
Stefano Brivio
stefano.brivio@mdm.imm.cnr.it

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *07 July 2020*
Accepted: *06 January 2021*
Published: *01 February 2021*

Citation:
Brivio S, Ly DRB, Vianello E and Spiga S (2021) Non-linear Memristive Synaptic Dynamics for Efficient Unsupervised Learning in Spiking Neural Networks. Front. Neurosci. 15:580909. doi: 10.3389/fnins.2021.580909

1. INTRODUCTION

Spiking Neural Networks (SNNs) received a renewed wave of interest from a computational point of view as a tool to move the huge overload in data analysis from the cloud to the edge. Indeed, they couple the neural network computing power with spike coding of information, which is considered a valid approach to reduce power requirement for the real-time analysis of unstructured data. This enables the process of *in-situ* decision making of autonomous systems (Indiveri et al., 2013). SNNs are a complementary solution to conventional Neural Networks (NNs), which compute with real-valued numbers and are currently used to remotely analyze the data uploaded to the cloud or at the edge only for inference, without online training (Yu, 2018). Both NNs and SNNs require specific hardware to boost their performance and computing speed. On one side, hardware accelerators of NNs, like graphical processing units and tensor processing units, are now widespread in the market. On the contrary, hardware supporting SNNs are mainly based on research platforms. In both cases, though, the lack of parallelism and separation between storage and computing units is still an issue, for which solutions are under investigation. To this aim, emerging memory devices, compatible with back-end of the production line of CMOS technology, and in particular resistive switching random access memories (RRAM), also named memristive devices, are considered among the best candidates for hardware solutions supporting NNs and SNNs. In particular, the so-called *neuromorphic* systems intend to use memristive devices to update, during training, and store, for inference, the synaptic weights of a network.

Since well-established robust and reliable training algorithms, like the back-propagation of the gradient, is available for NNs, the requirements for memristive devices for NN accelerators have already been determined (Chen et al., 2015; Gokmen and Vlasov, 2016; Sidler et al., 2016; Ambrogio et al., 2018; Fumarola et al., 2018; Moon et al., 2018). It has been shown that the memristive dynamics of the synapses, i.e., the evolution of the memristor conductance driven by train of identical pulses, determines the performance of the network (Chen et al., 2015; Gokmen and Vlasov, 2016; Sidler et al., 2016; Fumarola et al., 2018; La Barbera et al., 2018; Brivio et al., 2019a). In particular, NN accelerators trained through back-propagation require a memristive conductance evolving through many evenly-spaced levels (linear dynamics) (Chen et al., 2015; Gokmen and Vlasov, 2016; Sidler et al., 2016; Fumarola et al., 2018). The same agreement on the required memristive synaptic dynamics in SNN can hardly be reached because various training protocols have been investigated with different results (Brivio et al., 2019a). Currently, SNN training attempts include *on-line* spike-based procedures (Payvand et al., 2018; Brivio et al., 2019a; Donati et al., 2019) and *off-line* conventional training of a non-spiking NN that must be afterwards converted into an equivalent SNN (Diehl et al., 2015, 2016; Sengupta et al., 2019). The former allows exploiting the full potential of memristive devices tuneability to achieve a real-time on-line adaptive operation. Among the spike-based training procedures, supervised learning rules inspired by the back-propagation exist (Urbanczik and Senn, 2014; Müller et al., 2017; Donati et al., 2019), which are seldom investigated for systems including realistic simulations of memristive devices (Nair et al., 2017; Payvand et al., 2018). On the contrary, the literature is extremely rich of reports dealing with networks trained by supervised (Brivio et al., 2019a) and unsupervised versions of the so-called Spike Timing Dependent Plasticity (STDP) (Diehl and Cook, 2015; Garbin et al., 2015; Querlioz et al., 2015; Ambrogio et al., 2016; La Barbera et al., 2018). Few reports indicate that non-linear memristive dynamics may be beneficial for STDP-based SNNs (La Barbera et al., 2018; Brivio et al., 2019a). A comprehensive review about neural networks and spiking neural networks including also memristive devices can be found in Bouvieret al. (2019). In addition, the deployment of all the various emerging technologies for brain-inspired computing is extensively described in Spiga et al. (2020).

In this paper, we aim at moving the first steps toward the optimization of the training of a SNN through system-level simulations as a function of various experimentally-inspired memristive dynamics. Neuron model, training protocol, and architecture are also compatible with a hardware implementation in CMOS technology, as in the silicon chip described in Valentian et al. (2019) and Regev et al. (2020). The investigated memristive dynamics include linear and non-linear evolution bounded within extreme maximum and minimum values, as well as a non-linear evolution asymptotically approaching the boundary values (details are reported below). The response of the network is monitored throughout its training against the classification of hand-written digits from the MNIST dataset (Lecun et al., 1998). We choose this particular task in order to allow a direct comparison with other results reported in the literature for NNs (Chen et al., 2015; Garbin et al., 2015; Ambrogio et al., 2018) and SNNs. (La Barbera et al., 2018; Brivio et al., 2019a) Furthermore, the comparison among the various memristive dynamics is performed in a quantitative manner through the definition of figures of merit that apply to any mathematical formulation for synaptic dynamics. We found that non-linear dynamics bounded within extreme values is the most versatile dynamics, which guarantees the best classification performance and the best compromise between training duration and classification accuracy. This result marks a clear difference with respect to the recent finding related to conventional neural network accelerators trained through the back-propagation algorithm, which, according to a general agreement, require linear synapses (Chen et al., 2015; Gokmen and Vlasov, 2016; Ambrogio et al., 2018; Fumarola et al., 2018).

2. METHODS

2.1. Network Architecture and Training

Figure 1A presents the two-layers fully-connected feed-forward SNN simulated for the classification of hand-written digits from the MNIST dataset (Lecun et al., 1998). Simulations are performed with the event-based N2D2 simulator tool (Bichler et al., 2013). The full MNIST dataset is presented only once for training (60,000 training digits), then testing (10,000 testing digits). Each digit is composed of 28×28 pixels. The input layer

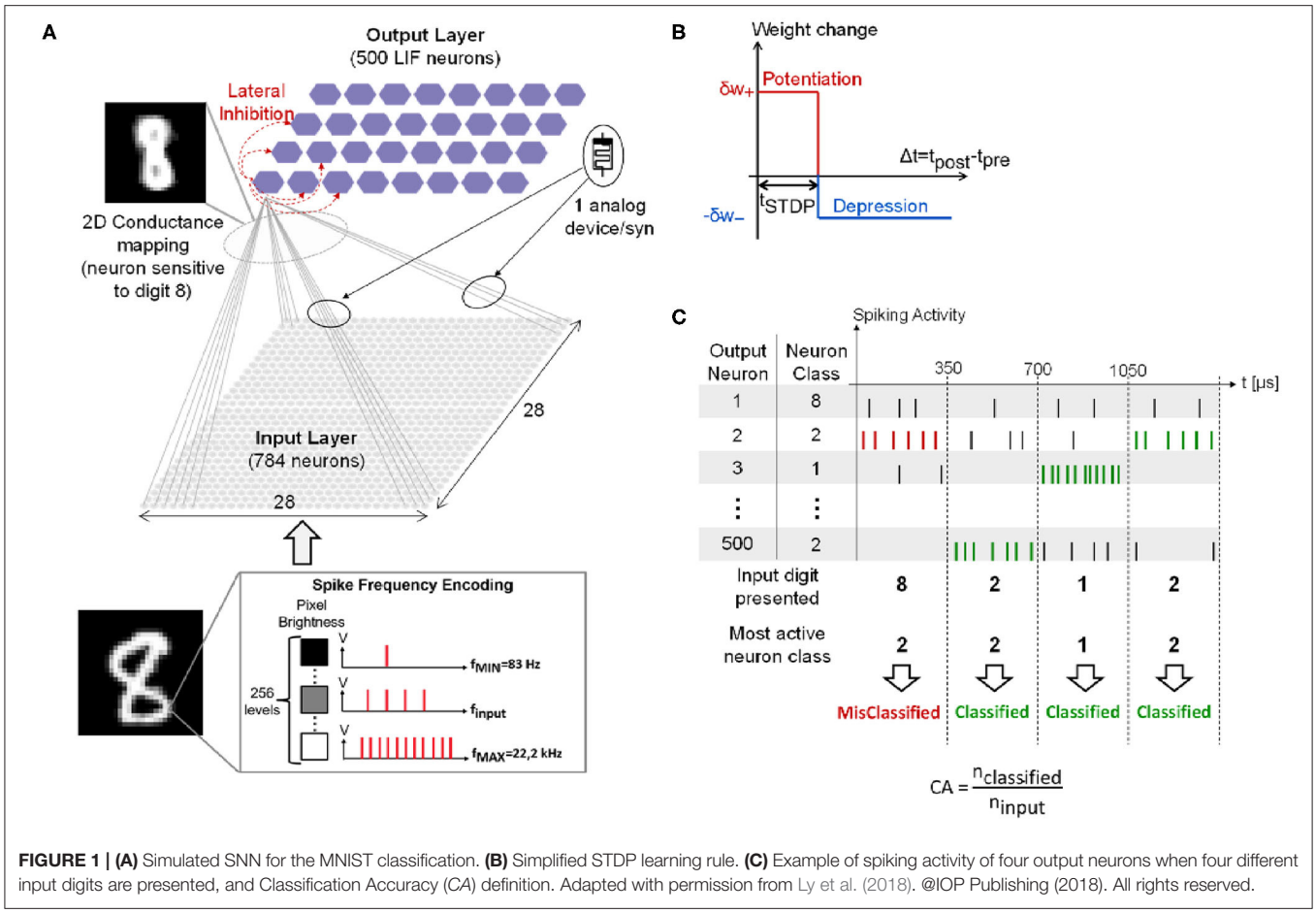


FIGURE 1 | (A) Simulated SNN for the MNIST classification. **(B)** Simplified STDP learning rule. **(C)** Example of spiking activity of four output neurons when four different input digits are presented, and Classification Accuracy (*CA*) definition. Adapted with permission from Ly et al. (2018). @IOP Publishing (2018). All rights reserved.

converts the input digit with a spike frequency encoding: each input neuron generates a spike train with a spiking rate f_{input} proportional to the gray level of the corresponding input pixel. f_{input} ranges from $f_{MIN} = 83$ Hz to $f_{MAX} = 22.2$ kHz with a total of 256 different gray levels. Spike trains are generated according to a Normal distribution. Each input digit is presented to the network for 350 μs during the training phase. The input layer is composed of 28×28 input neurons fully-connected by weighted synapses to the output layer composed of 500 Leaky Integrate-and-Fire (LIF) output neurons with a leak time constant $\tau_{leak} = 120.0$ μs. Note that after an output neuron fires a spike, it cannot integrate any incoming spikes for a refractory period $t_{refrac} = 1$ ns. It also prevents all the other neurons of the layer from integrating incoming spikes for a period $t_{inhibit} = 10$ μs, referred to as *lateral inhibition*. This allows implementing a Winner-Take-All (WTA) network between all the neurons (Bichler et al., 2013). In addition, a slight delay in the firing time of output neurons has been introduced: when an output neuron reaches its threshold value, it fires a spike after a delay t_{emit}. The parameter t_{emit} for each output neuron has been randomly drawn from a normal distribution with a mean value $\mu = 0.1$ ns and a standard deviation $\sigma = 1$ ps. This facilitates the implementation of the WTA process. These parameters have been optimized by a genetic algorithm. The network is trained with an unsupervised simplified Spike-Timing-Dependent Plasticity (STDP) rule (**Figure 1B**) (Suri et al., 2011; Querlioz et al., 2015): if the post-synaptic neuron spikes after the pre-synaptic neuron within a STDP time window $t_{STDP} = 60.0$ μs, the synapse increases its synaptic weight by a quantity δw_+ (synaptic *potentiation* event). Otherwise, its synaptic weight is decreased by a quantity δw_- (synaptic *depression* event). Quantities δw_+ and δw_- follow different dynamics models described in the following section. The weights are bounded between [0, 1] and are initialized to the value of 0.8 before training. From a hardware point of view, the initialization of devices to a predefined value is more straightforward than a random initialization. In particular, the weight value of 0.8 (i.e., high memristive conductance) is coherent with an initialization performed in hardware with only an electroforming step, which is required for a large class of memristive devices (Brivio and Menzel, 2020). Furthermore, the initialization does not influence the obtained classification performances as demonstrated in Querlioz et al. (2013).

During the training phase, each output neuron becomes sensitive to a specific class of digit as illustrated in the 2D conductance mapping in the top left of **Figure 1A** (class of digit "8"). After training, each output neuron is associated with the digit it is the most sensitive to. This represents the class of the neuron. To assess network performance during the testing

phase, the Classification Accuracy (*CA*) is computed as defined in **Figure 1C**. Each input digit is presented to the network for 350 μs and the output neuron that spikes the most within this time window—the most active neuron—corresponds to the network response. If the most active neuron class coincides with the input digit, the digit is successfully classified (green spikes). Otherwise, the digit is misclassified (red spikes). The *CA* is calculated as the ratio between the number of successfully classified digits, $n_{classified}$, and the number of input digits, n_{input} (bottom of **Figure 1C**). As there are multiple ways to hand-write the same digit, increasing the number of output neurons allows for an improvement of network performance as demonstrated in Querlioz et al. (2015). Indeed, this enables the network to have at its disposal several neurons specialized to different hand-writings of the same digit. As shown in Querlioz et al. (2015), the increase of *CA* with the number of output neurons saturates after 500 output neurons.

It is worth pointing out that the network architecture and operation are implemented according to the real hardware possibilities of the current CMOS and memristive technologies. In particular, contrary to Querlioz et al. (2015) who implemented the same network as the present one, homeostasis, which, e.g., adjusts each individual output neuron threshold on the basis on its instantaneous firing rate, is not included. As a matter of fact, Querlioz et al. (2015) shows that homeostasis can improve the classification accuracy by about 10%. On the other hand, the hardware implementation of homeostasis would require memory banks to store each individual neuron threshold values and one additional capacitor per neuron to probe each neuron firing rate, which will have a prohibitive impact on the required silicon real estate (Dalgaty et al., 2019). Some pioneering works are trying to address this issue with the help of memristive technology (Dalgaty et al., 2019), but a hardware-compatible homeostatic process over a large number of neurons has not been elaborated yet.

Furthermore, the classification scheme can also be improved with a voting procedure that takes into account the average firing rate of each neuron pool as in Diehl and Cook (2015), instead of considering only the individual neuron that fires the most as in the present implementation. However, the voting procedure based on the individual neuron firing rate eases the circuit complexity and is only marginally influencing the network performances. Indeed, Querlioz et al. (2015) obtained a classification accuracy (94.5%) very close to that of Diehl et al. (95%).

2.2. Models for Memristive Dynamics

The synaptic dynamics corresponds to the evolution of the weight of an artificial synapse (proportional to the memristive device conductance) when subjected to a train of identical pulses. Considering bipolar memristive synapses, trains of pulses of a given voltage polarity can lead to weight potentiation and trains of pulses with the opposite polarity lead to weight depression. As evident from the recent literature (Fumarola et al., 2018) and pointed out by part of the present authors in Frascaroli et al. (2018), the more general memristive conductance dynamics usually follows a non-linear evolution with a slow approach to the maximum and minimum values. Such dynamics can be described by a *non-linear soft-bound* (NL-SB) model, which has a particular importance in the field of computational neuroscience. Indeed, Fusi and Abbott (2007) demonstrated that NL-SB synapses generally endow a SNN with a larger memory capacity (capacity of storage of memories) compared to synapses whose weight evolve linearly between two boundary values. This latter synaptic model will be referred to as *linear hard-bound* (L-HB) synapses in the following. Fusi and Abbott (2007) showed that L-HB synapses perform better than NL-SB ones only in the particular case of a balanced network, i.e., a network in which the rate of potentiation is the same as the rate of depression events. From an experimental point of view, a memristive dynamics is usually approximated with a L-HB dynamics by interrupting a NL-SB one after a certain number of pulses at the cost of reduced conductance window, $^{G_{max}}/_{G_{min}}$. Examples of experimental reports can be found in Jang et al. (2015), Wang et al. (2016), and Bousoulas et al. (2017). A third generic case, which we will call *non-linear hard-bound* (NL-HB), consists in a non-linear dynamics interrupted at arbitrary boundary values. The boundary values are strictly reached after a certain finite number of consecutive weight increase or decrease events. As already mentioned, the NL-SB case is different because the weight boundaries are reached as asymptotic values after an infinite number of pulses (from a experimental point of view, tests up to few thousand pulses have been performed; Brivio et al., 2019a). The investigated L-HB and NL-SB dynamics in potentiation (conductance increase) and depression (conductance decrease) are shown in **Figures 2A,B** as solid and dotted lines, respectively. **Figures 2C,D** report various investigated NL-HB cases, for potentiation and depression operations as solid and dotted lines, respectively. The examples reported in **Figure 2** correspond to specific mathematical expressions and parameterizations of the dynamics models as described in the following.

Formally, the weight dynamics can be expressed in a differential form in the continuous domain as a variation of the weight, *dw*, per pulse, *dn*. The weights are always positive because they are represented by the conductance value of a physical device. Furthermore, it must be pointed out that hard-bound cases are experimentally obtained by interrupting a generic soft-bound dynamics, which therefore reduces the conductance window of hard-bound cases. Despite this fact, all the dynamics cases are simulated with the same conductance window considering the weight as the normalized version of the conductance between $[0, 1]$, as plotted in **Figure 2**. Therefore, for all the following equations one should consider $w \in [0, 1]$ and $\frac{dw_\pm}{dn} = 0$ outside the interval $[0, 1]$. In particular, the L-HB dynamics is given by

$$\frac{dw_\pm}{dn} = \pm\alpha_\pm, \tag{1}$$

with $\alpha_\pm \in (0, 1]$ and where the $(\cdot)_+$ and the $(\cdot)_-$ stand for potentiation and depression, respectively. Following Fusi and Abbott (2007), Frascaroli et al. (2018), and Brivio et al. (2019a),

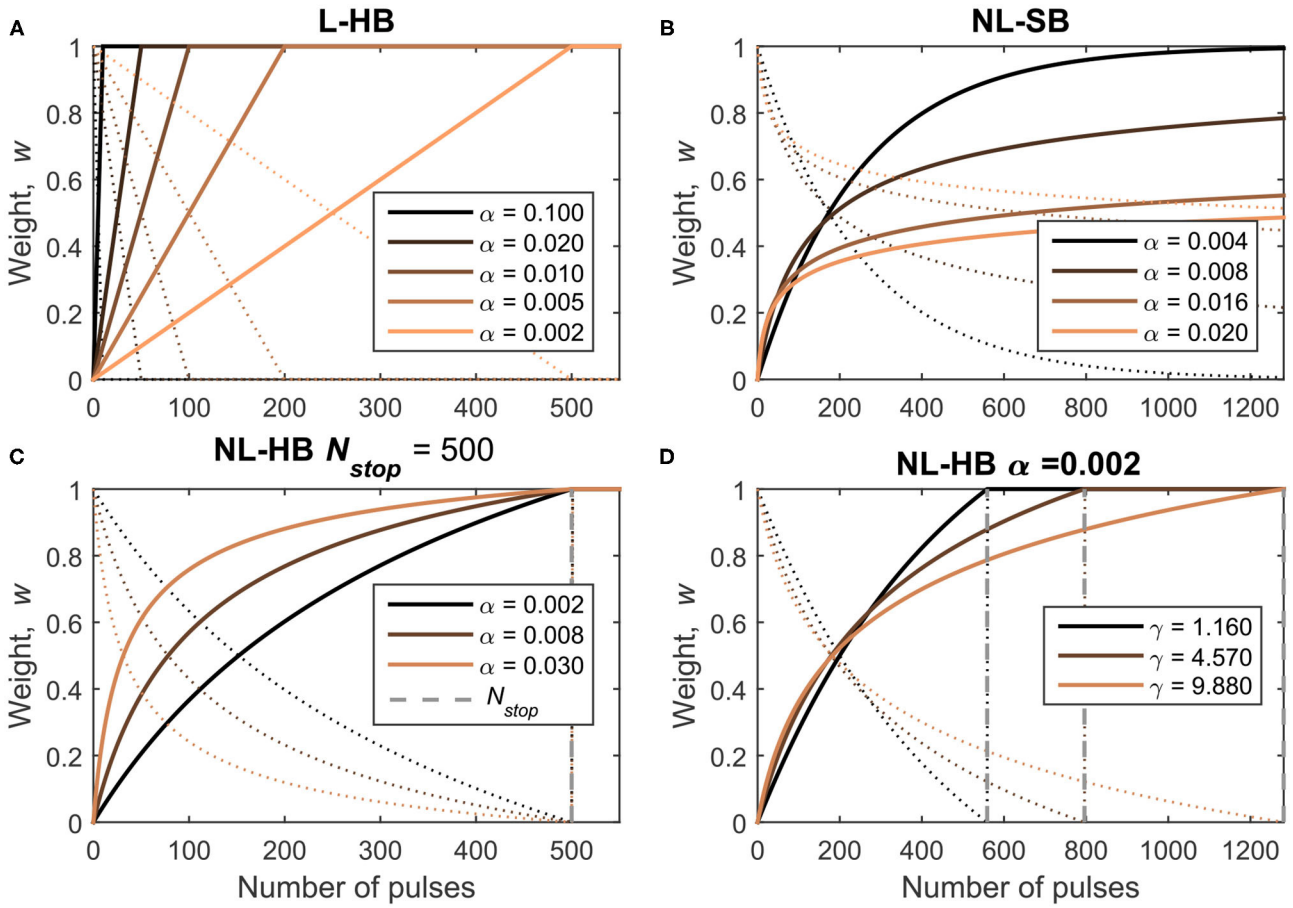


FIGURE 2 | Investigated dynamics: **(A)** Linear Hard-Bound (L-HB); **(B)** Non-Linear Soft-Bound (NL-SB) and **(C,D)** Non-Linear Hard-Bound (NL-HB). Dynamics are plotted as a function of the number of pulses for potentiation and depression operations (straight and dotted lines, respectively). The various dynamics cases are defined in **Table 1**.

the NL-SB equation is given by

$$\begin{cases} \frac{dw_+}{dn} = \alpha_+(1-w)^{\gamma_+} \\ \frac{dw_-}{dn} = -\alpha_- w^{\gamma_-} \end{cases}, \tag{2}$$

with $\alpha_\pm \in (0,1]$ and $\gamma_\pm \geq 1$. It is evident from Equation (2) that the weight variation tends to nullify as w approaches the boundary values. The NL-HB dynamics is the truncated version of the NL-SB properly re-scaled between 0 and 1, as follows

$$\begin{cases} \frac{dw_+}{dn} = \frac{\alpha_+}{w_{stop,+}}(1 - w \cdot w_{stop,+})^{\gamma_+} \\ \frac{dw_-}{dn} = -\frac{\alpha_-}{w_{stop,-}}(w \cdot w_{stop,-} + 1 - w_{stop,-})^{\gamma_-} \end{cases}, \tag{3}$$

where $\alpha_\pm \in (0,1]$, $\gamma_\pm \in [1, +\infty)$. $N_{stop,\pm}$ are the values of n at which the corresponding NL-SB dynamics is truncated to get a NL-HB one. $w_{stop,\pm}$ are the normalization terms that depend on the value of $N_{stop,\pm}$, as shown in the **Supplementary Material**.

It is worth making some additional clarifications. Each dynamics case is described by one or more free parameters which are chosen as described in the following. It is clear from Equations (1) and (2) that $\alpha_\pm$ is the step height when departing from the boundary value for the L-HB and the NL-SB dynamics. Indeed, the weight moves away from the lower boundary value, $w = 0$ for potentiation (resp. higher boundary value, $w = 1$ for depression) with a weight change equal to α_+ (resp. $-\alpha_-$). For the NL-HB case, the first step height is $\alpha_\pm / w_{stop,\pm}$. In addition, the weight change step is constant throughout the entire weight range for the L-HB case; it decreases from α to 0 for the NL-SB case; and it decreases from $\alpha_\pm / w_{stop,\pm}$ to a finite value greater than 0 for the NL-HB case. The parameter $\gamma_\pm$ introduces an additional non-linearity factor, whose effect can be appreciated from **Figure 2B**. For each dynamics case, potentiation and depression evolution are considered identical, i.e., with the same values of the free parameters, $\alpha_+ = \alpha_-$, $\gamma_+ = \gamma_-$, and $N_{stop,+} = N_{stop,-}$. As a consequence, the pace of approaching and departing to and from a given weight value is the same only for linear synapses. On the contrary, non-linear synapses are characterized by a certain asymmetry between potentiation and depression. For instance, a NL-SB synapse can be potentiated with a significant rate away from a weight value close to 0 ($w \approx 0$). In turn, at the same value, the

depression rate is close to 0 because $^{dw_-}/_{dn} = -\alpha_- w^{\gamma_-} \approx 0$. As a matter of fact, the asymmetry between potentiation and depression dynamics is usually present in real devices (Lee et al., 2015; Frascaroli et al., 2018). The impact of asymmetry between potentiation and depression dynamics on the performances of a neuromorphic system has been investigated in some detail for networks trained through back-propagation of the error (Chen et al., 2015; Agarwal et al., 2016; Fumarola et al., 2018) and only partially in spiking networks (La Barbera et al., 2018). However, a procedure to decouple the effect of asymmetry from that of non-linearity has not been proposed yet.

The memristive evolution in the network is determined by the STDP rule described in the previous section and the Equations (1)–(3). In fact, when the pre- and post-spikes are emitted according to the potentiation (depression) window in **Figure 1B**, a potentiation (depression) pulse is delivered to the memristive synapse driving a weight change equal to δw_+ (δw_-). The quantity $\delta w_\pm$ is determined by the current synaptic weight w and by the dynamics parameters in **Table 1** according to Equations (1)–(3). The programming of memristive device with CMOS neuron circuit, in STDP-based schemes, has been investigated in a number of works, which highlighted the need to include compact interface electronics. Pedretti et al. (2017) demonstrated STDP protocols on real 1 transistor-1 memristor structures based on the temporal overlapping on pre- and post-synaptic pulses driven by microcontrollers. Mostafa et al. (2016) designed a memristor/CMOS neuron interface constituted by 4 CMOS transistors to drive weight depression and potentiation operations separately within the framework of a generalized version of STDP. Covi et al. (2018) tested the same structure by wire-connecting a single memristor to two 350 nm technology CMOS neurons. The neurons delivered the correct programming pulses to obtain both analoge and digital memristive responses. In Brivio et al. (2019a), a 6 transistor-1 memristor structure is proposed to control synaptic potentiation, depression, and read operations in an implementation of a generalized version of STDP. The system-level simulation implemented in the present work are compatible with such implementation details.

The main weighting property of a synapse is its resolution, i.e., the number of weight values that it can store. The resolution of a synapse has a direct impact on the performances of a network (Bill et al., 2014; Brivio et al., 2019a). However, while the definition of number of levels is straightforward for L-HB dynamics, the same does not hold true in the case of non-linear weight evolution, because in this case the weight values are not evenly spaced. As a matter of principle, for NL-HB case, the resolution could be evaluated equal to the number of update events that are necessary to bring the weight from one boundary value (e.g., 0) to the opposite (i.e., 1), i.e., exactly N_{stop}. However, this is not a proper definition because the weight can be driven from 0 to 1 with the same number of steps but through various and very different trails. In particular, **Figure 2C** reports three NL-HB cases for which the same number of pulses is required to cover the full weight range but show very different dynamics. It is reasonable to associate different resolutions (or effective number of levels) to the three examples. In addition, it is worth pointing out that the number of pulses required to cover the full weight range is not a good definition for the NL-SB, which strictly requires an infinite number of steps to reach the boundary values. Indeed, the non-linear cases reported in **Figure 2** should be associated to different resolutions from a purely mathematical point of view. These considerations are completely independent from the effect of the noise and variability that unavoidably affect any real memristive device (Yu et al., 2013; Frascaroli et al., 2015, 2018; Covi et al., 2016; Brivio et al., 2017, 2019b). The impact of noise and variability has been investigated for some specific networks and some applications, demonstrating a general tolerance of neuromorphic systems against memristive synapse variability and noise (Querlioz et al., 2013; Garbin et al., 2014; Burr et al., 2015; Covi et al., 2016; Bocquet et al., 2018). Since we want to restrict the present study to a purely theoretical basis on the very impact of synaptic dynamics on network performances, the effect of noise and variability are left to a future work.

TABLE 1 | List of the investigated dynamics defined by the values of their parameters α, γ, and N_{stop}.

	$\alpha_\pm$	$\gamma_\pm$	$N_{stop,\pm}$	η	λ
L-HB	0.1	–	–	10	0
	0.02	–	–	50	0
	0.01	–	–	100	0
	0.005	–	–	200	0
	0.002	–	–	500	0
NL-SB	0.02	9	–	500	0.02
	0.016	7	–	500	0.020
	0.008	3	–	500	0.010
	0.004	1	–	500	0.005
NL-HB	0.002	3	500	402	0.006
	0.008	3	500	225	0.015
	0.03	3	500	90	0.047
	0.002	1.16	559	500	0.004
	0.002	4.57	796	500	0.006
	0.002	9.88	1281	500	0.009

The corresponding values for resolution, η, and non-linearity, λ, are also reported.

For the reasons above, we arbitrarily define an estimator for the resolution (effective number of levels) of the memristive device which can be applied to any generic dynamics expressed as a weight variation (dw) per pulse (dn) in the continuous domain, $\frac{dw_\pm}{dn} = f_\pm(w)$ [$f_\pm(w)$ must be differentiable for $w \in (0, 1)$]. The resolution, η, is defined as

$$\eta = \left\{ \int_0^{+\infty} \left[\frac{dw}{dn}\right]^2 dn \right\}^{-1}. \tag{4}$$

Equation (4) returns the correct number of levels for the trivial L-HB case, i.e., equal to the number of pulses to go from one boundary to the other one, and a reasonable estimate for the non-linear cases, as discussed in the **Supplementary Material**. η assumes analytical expressions for the dynamics cases under study, as reported in the **Supplementary Material**. It is just worth noticing that $\eta = {}^1/_{\alpha_\pm}$ for the L-HB case and is proportional to

$^1/_{\alpha_\pm}$ for the NL-SB (in agreement with Fusi and Abbott, 2007) and the NL-HB cases.

According to the discussion above and to the recent literature, a second property of weight dynamics is its non-linearity (λ), which can be defined as the average curvature of the weight evolution as a function of the number of potentiation or depression pulses, $w(n)$:

$$\lambda = \frac{4}{\pi}\int_0^{+\infty} \frac{|w''(n)|}{\{1+[w'(n)]^2\}^{3/2}}dn, \tag{5}$$

where $(\cdot)'$ and $(\cdot)''$ indicate the first and the second derivative with respect to n.

In this work, we investigate the impact of these synaptic properties, namely the resolution, η, and the non-linearity, λ, on the training and performance of the SNN described above. In particular, we consider various L-HB, NL-SB, and NL-HB dynamics, as reported in the **Table 1** and shown in **Figure 2**. Note that the L-HB cases are only characterized by different values of the resolution, because only one free parameter exists. For the NL-SB case, it is possible to investigate different dynamics for the same resolution, i.e., with different non-linearities. NL-HB cases are chosen in a way to have the same $N_{stop,\pm}$ (cases 1–3) or the same resolution, η (cases 4–6). We investigate resolution values up to 500 because this is the one that guarantees good performance on MNIST classification on the linear case, according to previous literature results (La Barbera et al., 2018; Brivio et al., 2019a) and as it will be evident also in the following. The free parameters of the various dynamics cases are generally compatible with experimental data as in Frascaroli et al. (2018) and Brivio and Menzel (2020). For the sake of completeness, it is worth noticing that dynamics of a memristive device depends on the properties of the constitutive materials and on the programming conditions. For instance, memristors featuring double insulating layers have been reported to show more gradual conductance evolution than devices with a single insulating material (Park et al., 2016; Wang et al., 2016; Moon et al., 2018; Brivio and Menzel, 2020). The response speed might depend on the diffusivity of the mobile ionic species as well. It is a property of the insulating materials itself, which can also be slightly tuned by doping, strain, or by changing the atomic structure and porosity (Azghadi et al., 2020; Brivio and Menzel, 2020). Furthermore, the programming scheme influences the dynamics. Indeed, strong programming conditions (high voltage or long pulses) result in large conductance changes with a few pulses (Frascaroli et al., 2018). It is worth specifying that all the parameters (α, γ, N_{stop}) defining the memristive dynamics affect both resolution and non-linearity at the same time. More details can be found in the **Supplementary Material**.

3. RESULTS

As discussed in the previous paragraph, the mathematical formulation of all the investigated dynamics comprises a parameter α, the only parameter that is present in all investigated synaptic dynamics. In **Figure 3**, the classification accuracy, *CA*, is shown to decrease as a function of α for the investigated cases. This observation is expected because $^1/_\alpha$ is equal to the synaptic resolution of the L-HB dynamics and is proportional to that of the non-linear ones. It is already evident that the non-linear cases perform better than the linear ones for a wide range of the parameter α, in agreement with previous publications (La Barbera et al., 2018; Brivio et al., 2019a). However, only the evolution as a function of α does not catch the entire complexity of the problem because, for the NL-SB and NL-HB cases, α affects both resolution and non-linearity. Indeed, the various types of weight dynamics in **Figure 3** follow a different decreasing trend. The maximum reached classification accuracy settles close to 85%, which is lower than the best results on theoretical SNNs (Diehl and Cook, 2015). However, as stated above, the aim of the present work is to test SNNs constituents and architectures that can be possibly realized in hybrid CMOS/memristor technology (Valentian et al., 2019; Regev et al., 2020). As discussed in the Methods section, the inclusion of a homeostatic rule, which is of difficult hardware implementation, would recover a classification accuracy close to the best state of the art results, as demonstrated by Querlioz et al. (2015) with the same network as the one implemented in this work. The values of the collected classification accuracy are reported in the **Supplementary Material**.

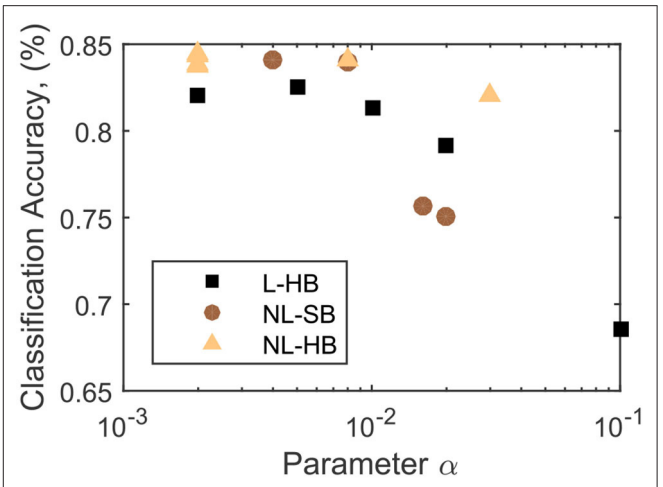


FIGURE 3 | Classification accuracy as a function of the parameter α for the L-HB, NL-SB, and NL-HB cases.

To get a deeper insight on the factors affecting the network performances, the classification accuracy is plotted as a function of η and λ in **Figure 4**, which reports the first of the main results of the paper taking advantage of the mathematical toolkit described in the previous section. **Figure 4A** shows that there is a general trend of increasing *CA* with the synaptic resolution, η. Different resolution values are shown for the dynamics with hard-bounds, linear (squares) and non-linear (triangles), i.e., L-HB and NL-HB. They show a very similar trend with slightly higher *CA* for the non-linear case. The investigated NL-SB dynamics (circles) share the same resolution ($\eta = 500$) but they give different *CA* results. In particular, for the NL-SB, the *CA* is reduced by the increase of non-linearity, λ, as shown in

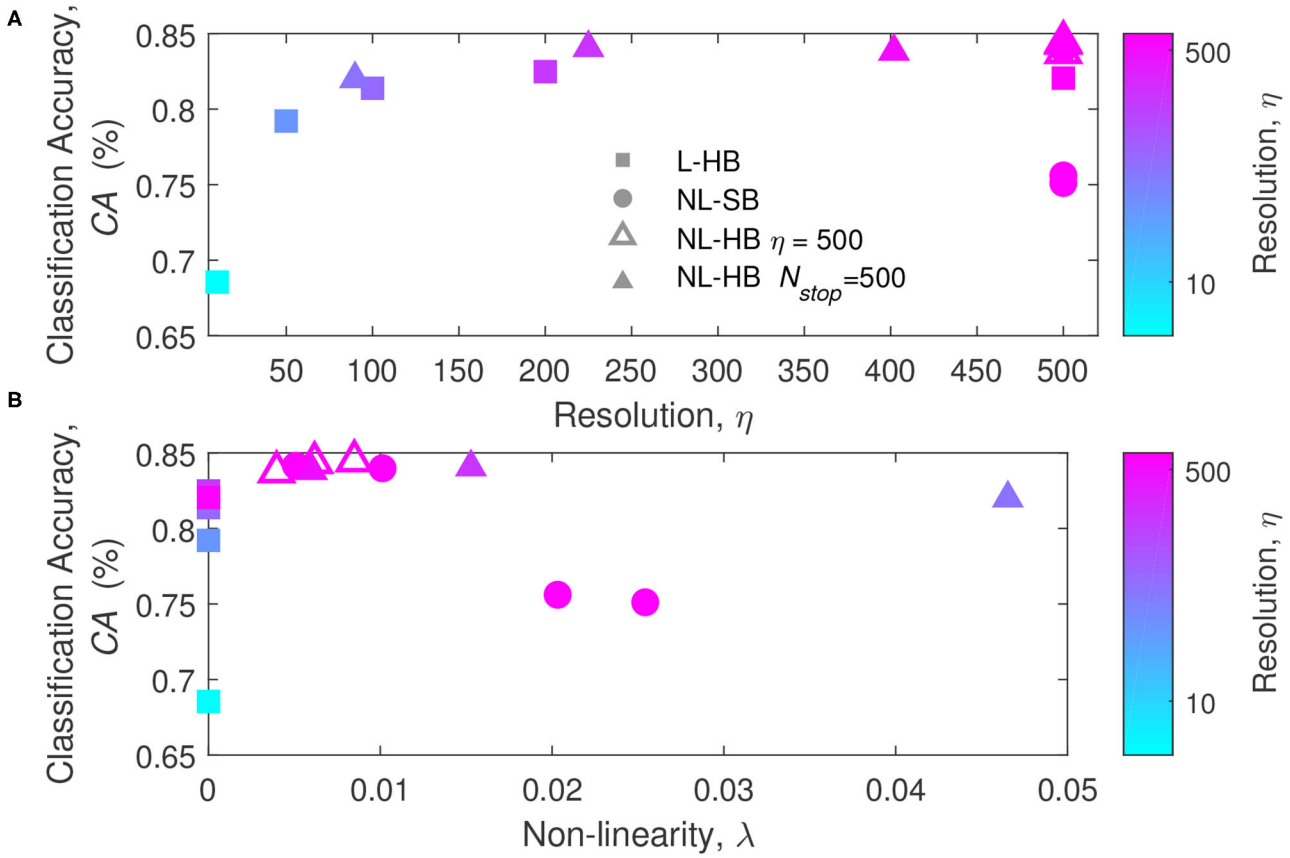


FIGURE 4 | Classification accuracy as a function of the parameters η **(A)** and λ **(B)**, for the L-HB, NL-SB, and NL-HB cases. The symbols color follow the resolution value, η, according to the color bars reported on the right sides of the panels.

Figure 4B. In turn, the non-linearity does not affect significantly the performances of the non-linear synapses with hard bounds, for which the *CA* remains almost stable over a wide range of λ values. It is important to point out that NL-HB synapses with the highest non-linearity $\lambda = 0.047$ are also characterized by the lowest resolution $\eta = 90$ (please consider that the symbol color is indicative of the synaptic resolution η, according to the color reported on the right-hand side of the figure). In any case, both the low resolution and the high non-linearity affect the classification performance only by a small amount. In addition, it is worthless to notice that the L-HB synapses are all described by zero non-linearity ($\lambda = 0$).

In order to understand the previous results we monitor the learning dynamics, i.e., the *CA* as a function of the training time (i.e., number of training digits), which usually displays a growth and a saturation toward the final maximum value. The learning dynamics for all the investigated synaptic models are reported in **Figure 5** (circles, left axis). With training, the synaptic weights evolve in a way that enables the distinction between the digits. In particular, it is well known that linear synapses, i.e., characterized by weight-independent plasticity, tend to develop bi-modal weight distributions after training (Song et al., 2000; van Rossum et al., 2000; Rubin et al., 2001; Billings and van Rossum, 2009). In this case, the weight values accumulate at the edges of the useful weight range, i.e., [0,1] in the present case. On the contrary, non-linear synapses with weight-dependent plasticity tend to result in a uni-modal weight distribution, with weight values accumulating in a value somewhere in the middle of the allowed weight range (Song et al., 2000; van Rossum et al., 2000; Rubin et al., 2001; Morrison et al., 2008; Billings and van Rossum, 2009; Brivio et al., 2019a). This is the result of the fact that strong (weak) synapses with non-linear dynamics are weakly potentiated (weakly depressed), which was shown to improve the memory capacity of the network on one side and, on the other, limit the synaptic specialization of the classification layer (Fusi and Abbott, 2007; Brivio et al., 2019a). As a matter of fact, in general, weight-dependent synapses and uni-modal distributions are considered less informative (Hennequin et al., 2010), because they correspond to a lower degree of specialization than weight-independent synapses and bi-modal distributions. Conversely, uni-modal distributions are considered more biologically realistic (Morrison et al., 2008). The weight distributions of the investigated cases at the end of the training are reported for the sake of completeness in the **Supplementary Figure 2**. In order to monitor the development of a weight specialization that enables the network to classify the

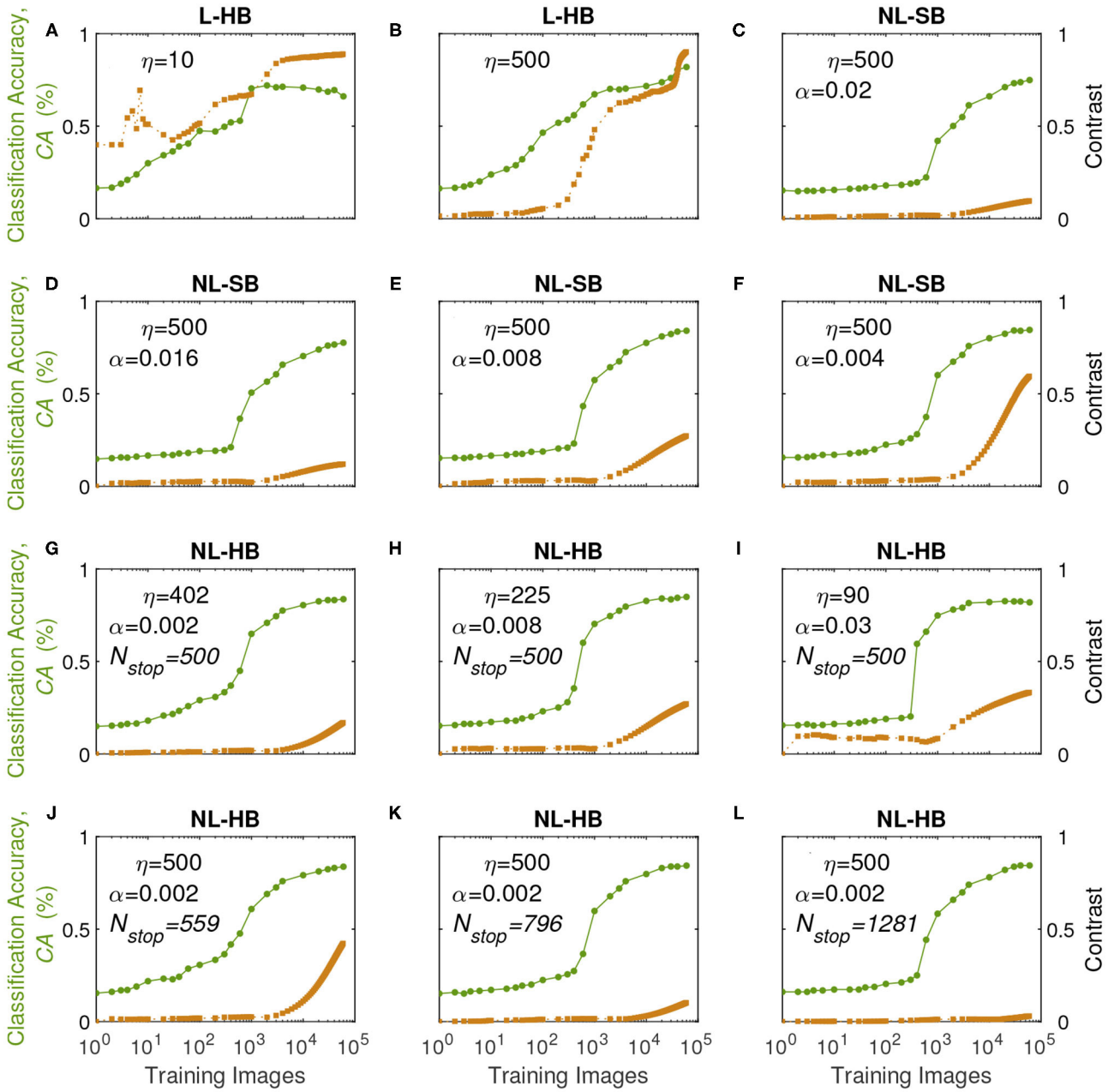


FIGURE 5 | Classification Accuracy (*CA*, left axis) and weight contrast (right axis, as defined in the main text) as a function of the number of training images presented to the SNN for various dynamics in the different panels.

input images, we analyze the clustering of the weights into two groups as a function of training through the *k-means* algorithm and consider the distance between the centers of the two clusters as a measure of the network specialization, which we will call weight contrast. Indeed, intuitively the weight contrast can be considered as the ability to take advantage of a wide portion of the available weight range. Other methods to group the weight values into two clusters are analyzed in the **Supplementary Material** and are in agreement with the *k-means* algorithm results. The weight contrast is reported in **Figure 5** (squares, right axis) for the various dynamics cases. It is possible to notice that the linear cases develop a large weight contrast at the end of training (**Figures 5A,B**) in agreement with the general discussion above. The non-linear cases show lower weight contrast than the linear cases but with significant variations depending on the dynamics parameters (for instance, cf. **Figures 5J,L** for two different NL-HB cases). The weight contrast at the end of the training is plotted against the parameters η and λ in **Figures 6A,B**, respectively.

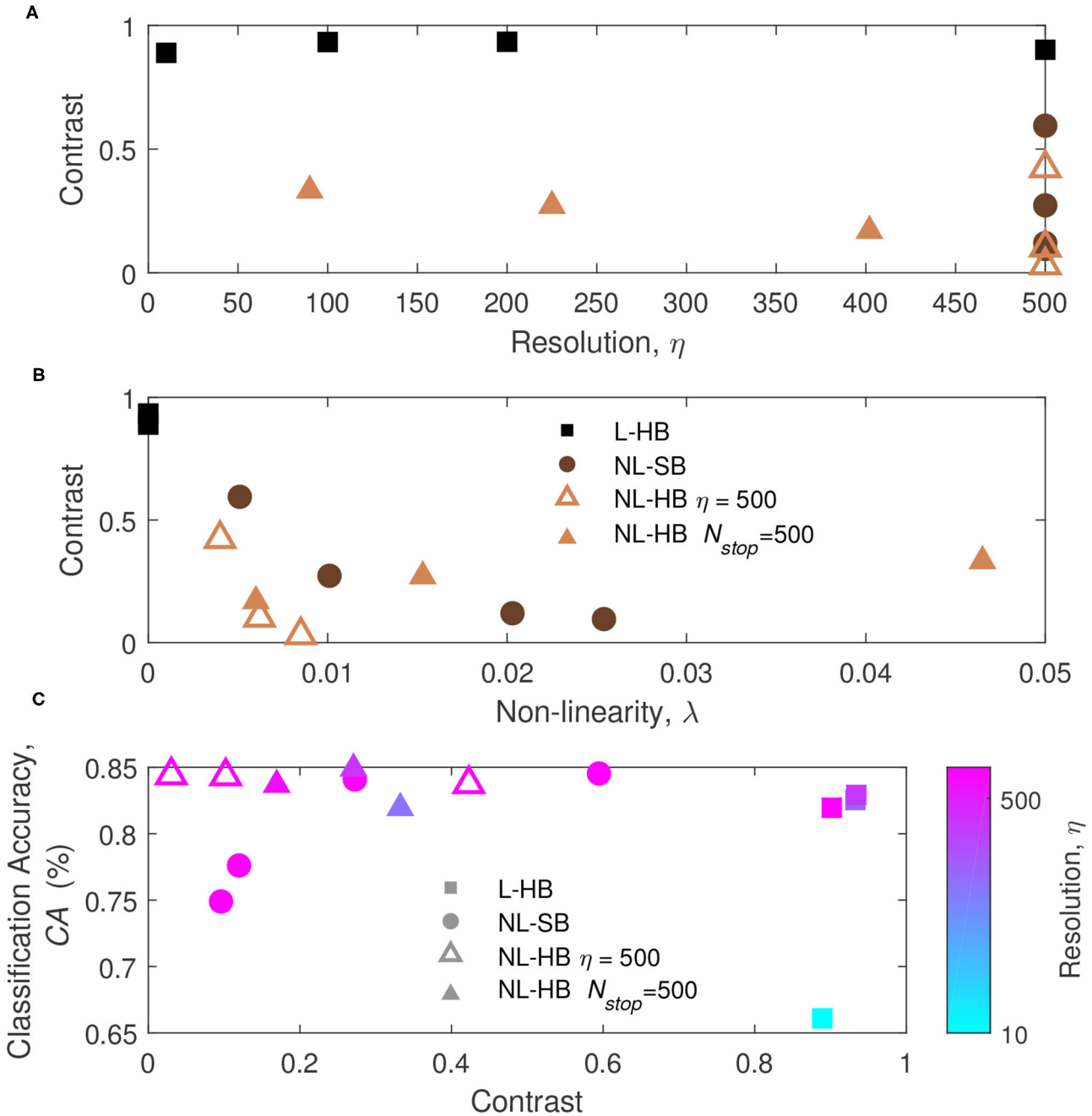


FIGURE 6 | (A,B) Weight contrast at the end of the training as a function of the parameters η and λ, for the L-HB, NL-SB, and NL-HB cases. **(C)** Classification accuracy, *CA*, as a function of the weight contrast. The symbols color follow the resolution value, η, according to the color bar reported on the right side of panel.

Figure 6A shows that the L-HB case results in about the same contrast for every resolution, while in the NL-SB case the same synaptic resolution can give very different weight contrasts, depending on the non-linearity, λ (**Figure 6B**). The NL-HB case is the most interesting, because the additional parameter N_{stop} allows to increase the contrast either by reducing the resolution, as shown by the filled triangles in **Figure 6A**, or by reducing the non-linearity at equal resolution, as shown by the empty triangles in **Figure 6B**. Finally, **Figure 6C** reports the *CA* as a function of the weight contrast. It shows that L-HB synapses (squares) are all characterized by high contrast but only those with high resolution achieve high classification accuracy (please notice, again, that the symbol color is indicative of the synaptic resolution η, according to the color reported on the right-side of the figure). NL-SB synapses (circles) achieve high *CA* only when the weight dynamics develops a high contrast. This is obtained

by reducing the non-linearity (please compare with **Figure 6B**). The classification results of the NL-HB synapses (triangles) are almost independent from the weight contrast obtained at the end of the training.

The results of **Figures 4**, **6** constitute already a relevant result with respect to the state of the art. Indeed, linear synaptic dynamics is often considered as the best solution for any kind of hardware neural network, so that large efforts are spent to improve linearity of memristor dynamics (Wang et al., 2016; Bousoulas et al., 2017; Chen et al., 2019). Such belief may have raised as a generalization of the results of exemplary works on NN accelerators trained by back propagation of the error generalized to other networks and other training protocols (Burr et al., 2015; Fumarola et al., 2018). As a matter of fact, as mentioned above and shown in **Figure 6B**, linearity improves weight contrast and sustains the specialization of the network. However, it has been demonstrated that non-linear synapses improve memory lifetime and memory capacity of a network in which the rates of potentiation and depression events are not perfectly balanced (Fusi and Abbott, 2007). Furthermore, van Rossum et al. (2000) pointed out that STDP tends to make potentiated synapses more and more potentiated. Indeed, as a synapse is strengthened, its correlation with the post-synaptic neurons increases thus leading to a further potentiation. Van Rossum et al. demonstrated that this destabilizing tendency of STDP can be profitably counterbalanced by introducing weight-dependent plasticity (i.e., a non-linear dynamics) which produces a certain competition among synapses. The results in **Figures 4**, **6** can be generically ascribed to a different balance between contrast decrease, increase of memory lifetime, and synaptic competition with increasing non-linearity.

This result marks a difference with respect to memristor-based neural network accelerators trained by global error back-propagation for which the achievement of high weight contrast and bi-modal weight distribution taking advantage of the full weight range is fundamental for a successful training (Sidler et al., 2016; Fumarola et al., 2018).

Another important aspect to consider is the duration of the training process, which for some applications must be reduced to a minimum. To evaluate it, we define the parameter Δ_{train} as the fraction of training images required to reach 99% of the final classification accuracy over the total number of digits available for training, n_{tot} (with n_{tot} = 60,000 here). In symbols,

$$\Delta_{train} = \frac{n_{train}(CA = 99\%CA_{max})}{n_{tot}}, \tag{6}$$

The parameter Δ_{train} is shown as a function of η and λ in **Figures 7A,B**, respectively. **Figure 7A** indicates a correlation between the synapse resolution and Δ_{train}. The correlation is somehow expected in case of a strong tendency to the formation of a bi-modal weight distribution, i.e., linear synapses (squares). Indeed, if the weight values tend to concentrate at the boundary values, the number of steps required to move the weight values from a generic initial one to the boundary scales with the synapse resolution. In agreement with this interpretation, the correlation between Δ_{train} and η is not perfect for the non-linear cases, because for the same resolution very different Δ_{train} values are obtained, as shown in **Figure 7A** in particular for the NL-SB cases (filled circles). Interestingly, the evolution of Δ_{train} as a function of λ follows opposite trends for soft and hard bound cases (also considering only the points at equal resolution, empty triangles and filled circles), as visible in **Figure 7B**. It is worth noticing that NL-SB and NL-HB with 500 levels resolution also show the same evolution of contrast as a function of non-linearity, as shown by filled circles and empty triangles in **Figure 6B**. Therefore, the opposite trends of Δ_{train} as a function of non-linearity cannot be explained by the need to develop, during training, a weight contrast that scales differently with non-linearity for NL-SB and NL-HB dynamics. On the contrary, the classification accuracy of NL-SB and NL-HB dynamics with the same 500 levels resolution follows opposite trends as a function of non-linearity, as indicated in **Figure 4B** (though the change for the NL-HB case is very modest). Therefore, the fact that both accuracy and training time follow opposite trends as a function of non-linearity can be an indication that, for non-linear dynamics, the training time is influenced by the maximum classification accuracy allowed by the particular synaptic dynamics. Finally, also considering the training time, the NL-HB cases (triangles) demonstrate more versatility than the other dynamics in reducing the training duration either by reducing the resolution, η (**Figure 7A**), or increasing the non-linearity, λ (**Figure 7B**).

All the results are summarized in **Figure 8**. **Figure 8A** reports the classification accuracy as a function of the training duration, Δ_{train}, for the various dynamics. The usual increase of *CA* with η is evident for the L-HB case, demonstrating that an increase in synaptic resolution produces a higher classification accuracy at the expense of longer training duration. This fact can be appreciated reminding that the symbol color follows the resolution, η, in agreement with the color bar on the right-hand side of the **Figure 8**. The saturation visible at high Δ_{train} may be just due to the fact that, during training, further *CA* increase takes longer and longer time. In **Figure 8A**, no general trend can be appreciated for NL-SB and NL-HB synapses. For instance, some NL-SB cases present long training times associated to a degraded *CA* as a consequence of the effect of the non-linearity, according to **Figures 4**, **7**. In addition, for the NL-HB cases, the *CA* shows a limited dependence on Δ_{train}. In particular, the point corresponding to the lowest training duration, interestingly, guarantees almost the same classification performances as the points requiring a longer training. This case could be considered as the one realizing the best trade-off between classification accuracy and required training time. As a matter of principle, some applications may require both to maximize the *CA* and to minimize Δ_{train} (i.e., maximize $1 - \Delta_{train}$). For this reason, we can define the SNN efficiency, ϵ, as

$$\epsilon = \frac{CA + (1 - \Delta_{train})}{2}, \tag{7}$$

which is normalized between 0 and 1. ϵ values are shown in **Figures 8B,C** as a function of η and λ, respectively (all the achieved values of the performance metrics and a figure

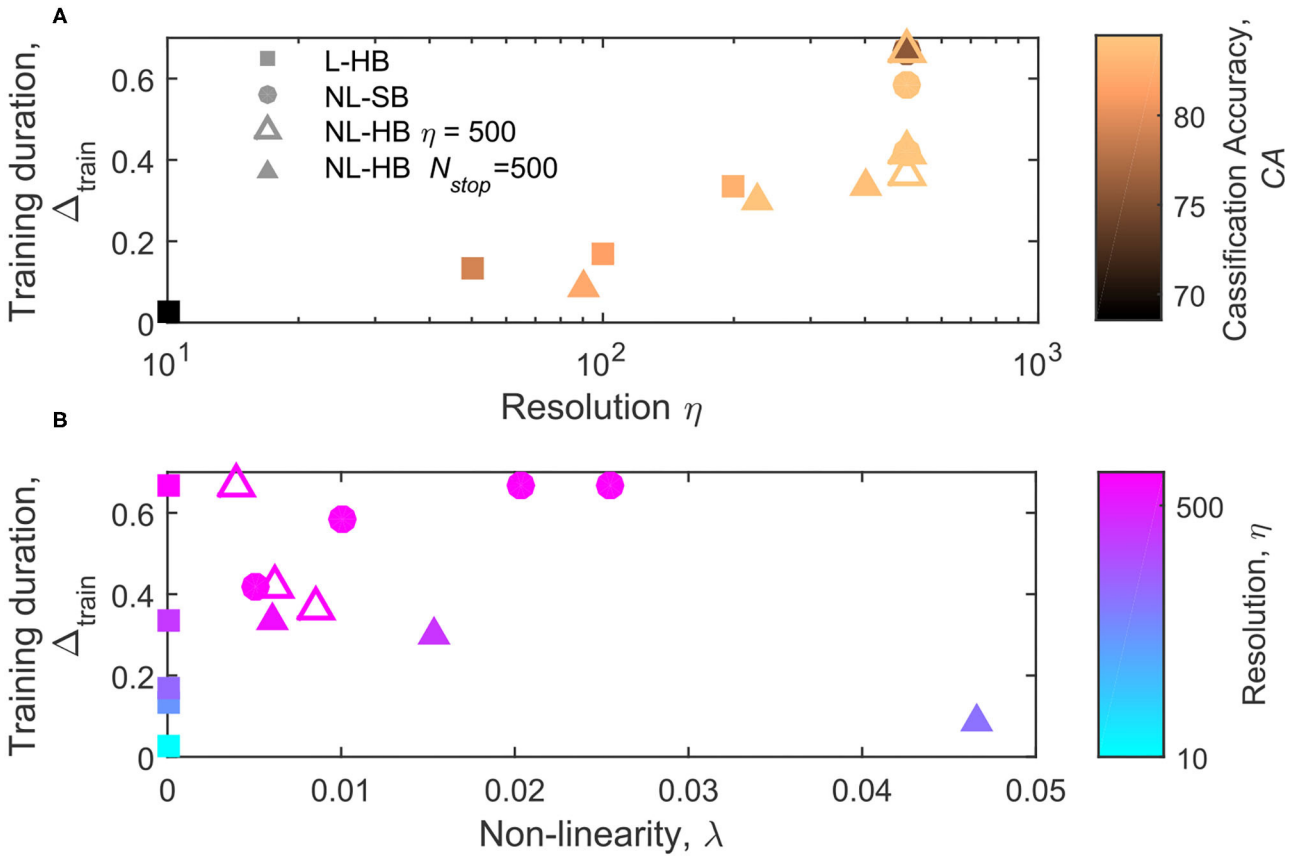


FIGURE 7 | Training duration, N_{train}, as a function of the parameter η **(A)** and λ **(B)**, for the L-HB, NL-SB, and NL-HB cases. The symbols colors in follow the classification accuracy, CA, in **(A)** and the resolution value, η, in **(B)** according to the color bars reported on the right sides of the panels.

reporting the efficiency as a function of accuracy are reported in the **Supplementary Material**). The maximum efficiency is reached by the NL-HB case with the lowest resolution and the highest non-linearity (top- and left-most triangle in **Figure 8B** and top- and right-most triangle in **Figure 8C**, with $\eta = 90$ and $\lambda = 0.047$). It corresponds to the dynamics with $\alpha = 0.03$ in **Figure 2C**, which grants a classification accuracy that is only slightly affected by resolution and non-linearity, as shown in **Figure 4**. Such highly non-linear and highly weight-dependent NL-HB dynamics resembles a NL-SB one and may endow the network with longer memory lifetime (Fusi and Abbott, 2007) and a higher synaptic competition within a STDP training framework (van Rossum et al., 2000), resulting in an improved synaptic contrast (right-most filled triangle in **Figure 6C**). Furthermore, the maximum efficiency dynamics takes advantage of a short training time justified by its low resolution, as shown in **Figure 7A**. In turn, for the L-HB cases (squares), the efficiency is degraded with increasing resolution as a consequence of the increase of the training duration, as shown in **Figure 7A**. The non-linearity, instead, deteriorates the efficiency of the NL-SB dynamics (circles in **Figure 7C**) because it both increases the training duration (**Figure 7B**) and reduces the classification accuracy (**Figure 7**).

4. CONCLUSIONS

In conclusion, we analyzed the impact of the synaptic weight dynamics on the performances of a two-layer fully-connected SNN compatible with a hybrid CMOS/memristive implementation and trained through an unsupervised STDP protocol. We chose weight dynamics that can be realized, at least as a matter of principle, through memristive technology. We found that synapses with non-linear dynamics and hard weight boundary values (NL-HB synapses) give performance advantages for a SNN with STDP-based learning in various aspects. First, NL-HB synapses guarantee the best classification accuracy among the investigated dynamics (see **Figures 3**, **4**, **8A**) over all the investigated range of resolution, η. It is worth noticing that this is a significant result in the context of the present literature. Indeed, it has been extensively demonstrated in several publications (Chen et al., 2015; Ambrogio et al., 2018; Fumarola et al., 2018; Moon et al., 2018) that linear synapses enable the best classification accuracy of neuromorphic systems that implement in hardware the back-propagation of the global error. This result has been extended, as a supposedly natural consequence, as holding true for SNNs. However, few recent works from the present authors (La Barbera et al., 2018; Brivio

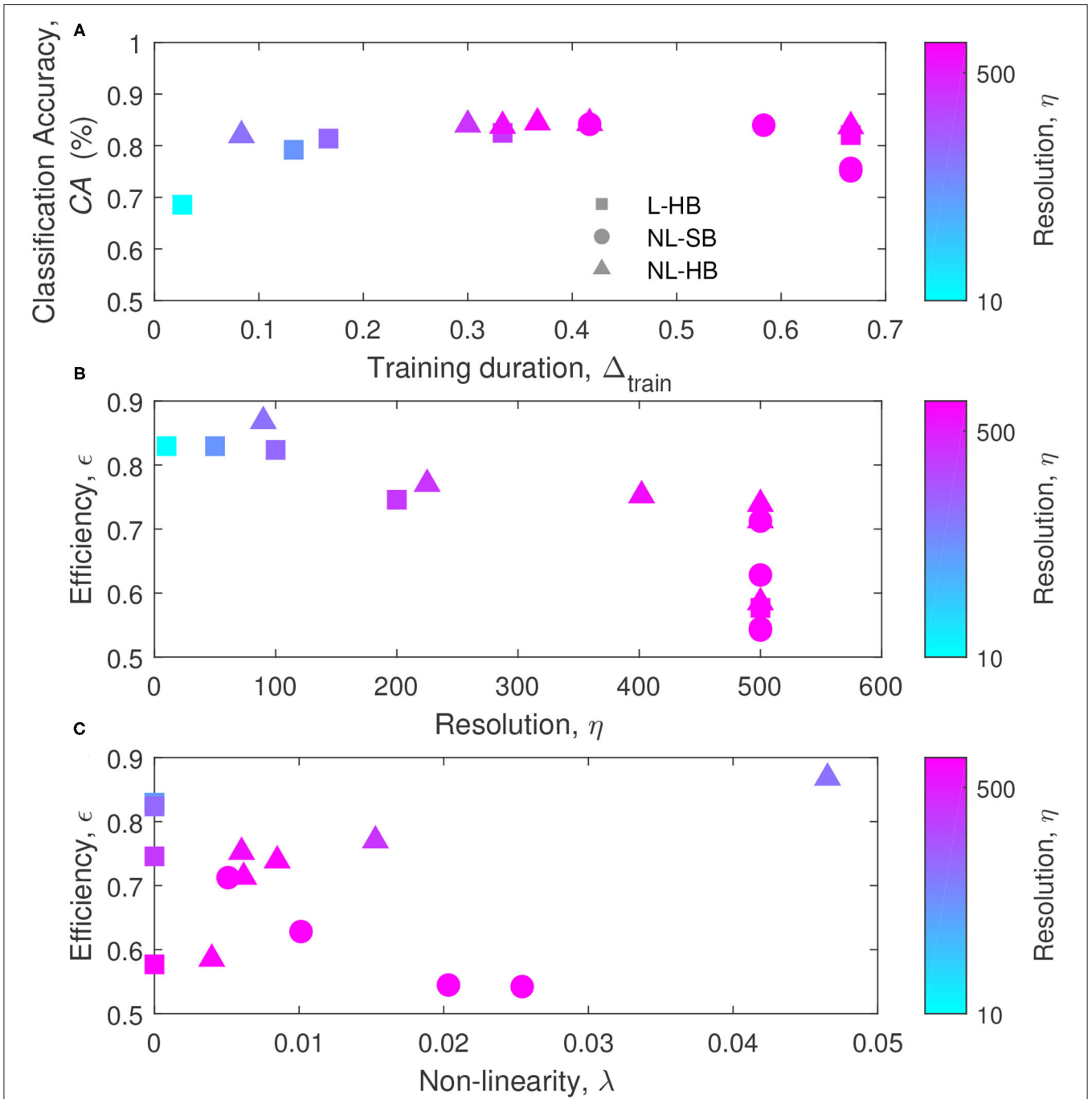


FIGURE 8 | (A) Classification accuracy, *CA*, as a function of the training duration, Δ_{train}, for the various dynamics. **(B)** Efficiency as a function of the resolution, η, and **(C)** efficiency as a function of the non-linearity, λ. The symbols colors follow the resolution value, η, according to the color bars reported on the right sides of the panels.

et al., 2019a) have given indications that non-linear synapses can perform better than linear ones for SNNs, which resulted in an interesting debate (Berg et al., 2019). In the present work, we put on firmer and quantitative basis the role of non-linearity on the performances of unsupervised and STDP-based SNNs.

Furthermore, for applications in which the training duration has to be minimized, the NL-HB dynamics also realized the best trade-off between classification accuracy and training duration, in agreement with the mathematical definition of efficiency given above (see **Figure 8**).

All these results are ascribed to the fact that the NL-HB dynamics produces a distinct behavior of the SNN, with respect to L-HB and NL-SB dynamics. Indeed, in case of hard-bounds, the classification accuracy and the weight contrast (ability to

take advantage of a wide portion of the available weight range) is minimally affected by the non-linearity (compare NL-SB and NL-HB cases in **Figures 4**, **6**). Moreover, the non-linearity of NL-HB synapses tends to reduce SNN training duration, in clear opposition with the trend of the soft-bound synapses (**Figure 7B**). This is the reason for the low training duration for the highly non-linear hard bound synapses, which results in a high efficiency, ϵ, according to the definition above (**Figure 8**).

In addition, it is interesting to make some considerations from a technological point of view. Memristive devices are characterized by an intrinsic non-linear conductance dynamics. More precisely, we have recently shown that the NL-SB dynamics is the model that faithfully describes the behavior of filamentary memristive devices (Frascaroli et al., 2018; Brivio et al., 2019a). On the other hand, technological efforts have been mainly focused on developing memristive synaptic devices with high resolution and low non-linearity because these are the requirement for hardware neural networks relying on back-propagation of the error. The linear dynamics is usually obtained by truncating the non-linear dynamics in the linear regime. This solution however limits the synapse resolution to a lower values with respect to those that can be obtained with a more complete non-linear dynamics. In fact, in the present study, the dynamics free parameters have been set to realistic values in particular for the non-linear cases. On the contrary, resolutions of 200 and 500 levels can hardly be obtained over a linear conductance evolution (Wang et al., 2016; Bousoulas et al., 2017; Chen et al., 2019). For instance, in one of the best literature results, Wang et al. (2016) reports a nearly linear dynamics over 300 pulses, indicating a resolution close to 300 levels. However, their data is best fitted with a NL-HB models with $\alpha = 0.004$, $\gamma = 1.02$ and a resolution of about 266 levels. Therefore, according to our results, in the case of SNNs with STDP-based unsupervised training, higher classification accuracy values, or efficiency values, can be obtained with non-linear hard-bound synapses relaxing the requirements on resolution and non-linearity for memristive devices. Therefore, high performances for STDP-based SNNs can be obtained with moderately challenging device engineering by embracing, instead of facing, their intrinsic non-linear dynamics. It is worth specifying that simulations have intentionally been performed neglecting any source of variability in the synaptic elements in order to isolate the very effect of synaptic dynamics. From an experimental point of view, the various dynamics may be affected more or less seriously by noise and variability. In particular, we can expect the linear dynamics, being the most challenging in real devices as stated above, to be the most affected by noise and variability. However, a methodological experimental investigation on highly optimized devices is required in order to take into account the different role of dynamics-dependent variability in the simulations.

Finally, the present paper defines a methodology to assess the impact of synaptic dynamics on the performances of a neural network and provides the basis for future works applied to different training protocols, network architectures, applications, and different synaptic dynamics features, e.g., asymmetry between weight depression and potentiation processes and potentially different dynamics evolution, size of the readout layer and, as mentioned above, the impact of dynamics-specific noise and variability features, all of which can have an impact on the trade-offs pointed out in the manuscript.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

SB conceived the idea and analyzed the data. DL performed the simulations and contributed to data analysis. EV and SS supervised the activity. All authors contributed to the manuscript writing.

FUNDING

This work was partially supported by the Horizon 2020 European project MeM-Scales—Memory technologies with multi-scale time constants for neuromorphic architectures (grant agreement No. 871371).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2021.580909/full#supplementary-material

REFERENCES

Agarwal, S., Plimpton, S. J., Hughart, D. R., Hsia, A. H., Richter, I., Cox, J. A., et al. (2016). "Resistive memory device requirements for a neura algorithm accelerator," in *2016 International Joint Conference on Neural Networks (IJCNN)* (Vancouver, BC), 929–938.

Ambrogio, S., Ciocchini, N., Laudato, M., Milo, V., Pirovano, A., Fantini, P., and Ielmini, D. (2016). Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapses. *Front. Neurosci*. 10:56. doi: 10.3389/fnins.2016.00056

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R. M., Boybat, I., Di Nolfo, C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558:60. doi: 10.1038/s41586-018-0180-5

Azghadi, R., Chen, Y.-C., Eshraghian, J. K., Chen, J., Lin, C.-Y., Amirsoleimani, A., et al. (2020). Complementary metal-oxide semiconductor and memristive hardware for neuromorphic computing. *Adv. Intell. Syst*. 2:1900189. doi: 10.1002/aisy.201900189

Berg, A., Brivio, S., Brown, S., Burr, G., Deswal, S., Deuermeier, J., Gale, E., et al. (2019). Synaptic and neuromorphic functions: general discussion. *Faraday Discuss*. 213, 553–578. doi: 10.1039/C8FD90065E

Bichler, O., Roclin, D., Gamrat, C., and Querlioz, D. (2013). "Design exploration methodology for memristor-based spiking neuromorphic architectures with the Xnet event-driven simulator," in *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)* (Brooklyn, NY), 7–12. doi: 10.1109/NanoArch.2013.6623029

Bill, J., and Legenstein, R. (2014). A compound memristive synapse model for statistical learning through STDP in spiking neural networks. *Front. Neurosci.* 8:412. doi: 10.3389/fnins.2014.00412

Billings, G., and van Rossum, M. C. W. (2009). Memory retention and spike-timing-dependent plasticity. *J. Neurophysiol.* 101, 2775–2788. doi: 10.1152/jn.91007.2008

Bocquet, M., Hirztlin, T., Klein, J.-O., Nowak, E., Vianello, E., Portal, J.-M., et al. (2018). "In-memory and error-immune differential RRAM implementation of binarized deep neural networks," in *2018 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 20.6.1–20.6.4. doi: 10.1109/IEDM.2018.8614639

Bousoulas, P., Michelakaki, I., Skotadis, E., Tsigkourakos, M., and Tsoukalas, D. (2017). Low-power forming free $TiO_{2-x}/HfO_{2-y}/TiO_{2-x}$-trilayer RRAM devices exhibiting synaptic property characteristics. *IEEE Trans. Electron Devices* 64, 3151–3158. doi: 10.1109/TED.2017.2709338

Bouvier, M., Alexandre, V., Mesquida, T., Rummens, F., Reyboz, M., Vianello, E., et al. (2019). Spiking neural networks hardware implementations and challenges: a survey. *ACM J. Emerg. Technol. Comput. Syst.* 15, 1–35. doi: 10.1145/3304103

Brivio, S., Conti, D., Nair, M. V., Frascaroli, J., Covi, E., Ricciardi, C., et al. (2019a). Extended memory lifetime in spiking neural networks employing memristive synapses with nonlinear conductance dynamics. *Nanotechnology* 30:015102. doi: 10.1088/1361-6528/aae81c

Brivio, S., Frascaroli, J., Covi, E., and Spiga, S. (2019b). Stimulated ionic telegraph noise in filamentary memristive devices. *Sci. Rep.* 9:6310. doi: 10.1038/s41598-019-41497-3

Brivio, S., Frascaroli, J., and Spiga, S. (2017). Role of Al doping in the filament disruption in HfO_2 resistance switches. *Nanotechnology* 28:395202. doi: 10.1088/1361-6528/aa8013

Brivio, S., and Menzel, S. (2020). "Chapter 2 - resistive switching memories," in *Memristive Devices for Brain-Inspired Computing, Woodhead Publishing Series in Electronic and Optical Materials*, eds S. Spiga, A. Sebastian, D. Querlioz, and B. Rajendran (Chennai: Woodhead Publishing), 17–61. doi: 10.1016/B978-0-08-102782-0.00002-2

Burr, G. W., Shelby, R. M., Sidler, S., di Nolfo, C., Jang, J., Boybat, I., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network (165, 000 synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electron Dev.* 62, 3498–3507. doi: 10.1109/TED.2015.2439635

Chen, J., Lin, C., Li, Y., Qin, C., Lu, K., Wang, J., et al. (2019). $LiSiO_X$-based analog memristive synapse for neuromorphic computing. *IEEE Electron Dev. Lett.* 40, 542–545. doi: 10.1109/LED.2019.2898443

Chen, P. Y., Lin, B., Wang, I. T., Hou, T. H., Ye, J., et al. (2015). "Mitigating effects of non-ideal synaptic device characteristics for on-chip learning," in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (Austin, TX), 194–199.

Covi, E., Brivio, S., Serb, A., Prodromakis, T., Fanciulli, M., and Spiga, S. (2016). Analog memristive synapse in spiking networks implementing unsupervised learning. *Front. Neurosci.* 10:482. doi: 10.3389/fnins.2016.00482

Covi, E., George, R., Frascaroli, J., Brivio, S., Mayr, C., Mostafa, H., et al. (2018). Spike-driven threshold-based learning with memristive synapses and neuromorphic silicon neurons. *J. Phys. D Appl. Phys.* 51:344003. doi: 10.1088/1361-6463/aad361

Dalgaty, T., Payvand, M., De Salvo, B., Casas, J., Lama, G., Nowak, E., et al. (2019). "Hybrid cmos-rram neurons with intrinsic plasticity," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)* (Sapporo), 1–5. doi: 10.1109/ISCAS.2019.8702603

Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney), 1–8. doi: 10.1109/IJCNN.2015.7280696

Diehl, P. U., Zarrella, G., Cassidy, A., Pedroni, B. U., and Neftci, E. (2016). "Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware," in *2016 IEEE International Conference on Rebooting Computing (ICRC)* (San Diego, CA), 1–8. doi: 10.1109/ICRC.2016.7738691

Donati, E., Payvand, M., Risi, N., Krause, R., and Indiveri, G. (2019). Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network. *IEEE Trans. Biomed. Circuits Syst.* 13, 795–803. doi: 10.1109/TBCAS.2019.2925454

Frascaroli, J., Brivio, S., Covi, E., and Spiga, S. (2018). Evidence of soft bound behaviour in analogue memristive devices for neuromorphic computing. *Sci. Rep.* 8, 1–12. doi: 10.1038/s41598-018-25376-x

Frascaroli, J., Volpe, F. G., Brivio, S., and Spiga, S. (2015). Effect of Al doping on the retention behavior of HfO_2 resistive switching memories. *Microelectron. Eng.* 147, 104–107. doi: 10.1016/j.mee.2015.04.043

Fumarola, A., Sidler, S., Moon, K., Jang, J., Shelby, R. M., Narayanan, P., et al. (2018). Bidirectional non-filamentary RRAM as an analog neuromorphic synapse, part ii: impact of $Al/Mo/Pr_0.7Ca_0.3MnO_3$ device characteristics on neural network training accuracy. *IEEE J. Electron Devices Soc.* 6, 169–178. doi: 10.1109/JEDS.2017.2782184

Fusi, S., and Abbott, L. F. (2007). Limits on the memory storage capacity of bounded synapses. *Nat. Neurosci.* 10:485. doi: 10.1038/nn1859

Garbin, D., Bichler, O., Vianello, E., Rafhay, Q., Gamrat, C., Perniola, L., et al. (2014). "Variability-tolerant Convolutional Neural Network for Pattern Recognition applications based on OxRAM synapses," in *2014 IEEE International, Electron Devices Meeting (IEDM)* (San Francisco, CA), 28.4.1–28.4.4. doi: 10.1109/IEDM.2014.7047126

Garbin, D., Vianello, E., Bichler, O., Rafhay, Q., Gamrat, C., Ghibaudo, G., et al. (2015). HfO_2-based OxRAM devices as synapses for convolutional neural networks. *IEEE Trans. Electron Devices* 62, 2494–2501. doi: 10.1109/TED.2015.2440102

Gokmen, T., and Vlasov, Y. (2016). Acceleration of deep neural network training with resistive cross-point devices: design considerations. *Front. Neurosci.* 10:333. doi: 10.3389/fnins.2016.00333

Hennequin, G., Gerstner, W., and Pfister, J.-P. (2010). STDP in adaptive neurons gives close-to-optimal information transmission. *Front. Comput. Neurosci.* 4:143. doi: 10.3389/fncom.2010.00143

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24:384010. doi: 10.1088/0957-4484/24/38/384010

Jang, J. W., Park, S., Burr, G. W., Hwang, H., and Jeong, Y. H. (2015). Optimization of conductance Change in $Pr_1 - xCa_xMnO_3$-based synaptic devices for neuromorphic systems. *IEEE Electron Device Lett.* 36, 457–459. doi: 10.1109/LED.2015.2418342

La Barbera, S., Ly, D. R. B., Navarro, G., Castellani, N., Cueto, O., Bourgeois, G., et al. (2018). Narrow heater bottom electrode-based phase change memory as a bidirectional artificial synapse. *Adv. Electron. Mater.* 4:1800223. doi: 10.1002/aelm.201800223

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lee, D., Moon, K., Park, J., Park, S., and Hwang, H. (2015). Trade-off between number of conductance states and variability of conductance change in pr0.7ca0.3mno3-based synapse device. *Appl. Phys. Lett.* 106:113701. doi: 10.1063/1.4915924

Ly, D. R. B., Grossi, A., Fenouillet-Beranger, C., Nowak, E., Querlioz, D., and Vianello, E. (2018). Role of synaptic variability in resistive memory-based spiking neural networks with unsupervised learning. *J. Phys. D* 51:444002. doi: 10.1088/1361-6463/aad954

Moon, K., Fumarola, A., Sidler, S., Jang, J., Narayanan, P., Shelby, R. M., et al. (2018). Bidirectional non-filamentary RRAM as an analog neuromorphic synapse, part I: $Al/Mo/Pr_0.7Ca_0.3MnO_3$ material improvements and device measurements. *IEEE J. Electron Devices Soc.* 6, 146–155. doi: 10.1109/JEDS.2017.2780275

Morrison, A., Diesmann, M., and Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybern.* 98, 459–478. doi: 10.1007/s00422-008-0233-1

Mostafa, H., Mayr, C., and Indiveri, G. (2016). "Beyond spike-timing dependent plasticity in memristor crossbar arrays," in

2016 IEEE International Symposium on Circuits and Systems (ISCAS) (Montreal, QC), 926–929. doi: 10.1109/ISCAS.2016.7527393

Müller, L. K., Nair, M. V., and Indiveri, G. (2017). "Randomized unregulated step descent for limited precision synaptic elements," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (Baltimore, MD), 1–4. doi: 10.1109/ISCAS.2017.8050217

Nair, M. V., Muller, L. K., and Indiveri, G. (2017). A differential memristive synapse circuit for on-line learning in neuromorphic computing systems. *Nano Futures* 1:035003. doi: 10.1088/2399-1984/aa954a

Park, J., Kwak, M., Moon, K., Woo, J., Lee, D., and Hwang, H. (2016). TiO_x-based RRAM synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing. *IEEE Electron Device Lett.* 37, 1559–1562. doi: 10.1109/LED.2016.2622716

Payvand, M., Muller, L. K., and Indiveri, G. (2018). "Event-based circuits for controlling stochastic learning with memristive devices in neuromorphic architectures," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (Florence), 1–5. doi: 10.1109/ISCAS.2018.8351544

Pedretti, G., Milo, V., Ambrogio, S., Carboni, R., Bianchi, S., Calderoni, A., et al. (2017). Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. *Sci. Rep.* 7, 1–10. doi: 10.1038/s41598-017-05480-0

Querlioz, D., Bichler, O., Dollfus, P., and Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* 12, 288–295. doi: 10.1109/TNANO.2013.2250995

Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616

Regev, A., Bricalli, A., Piccolboni, G., Valentian, A., Mesquida, T., Molas, G., et al. (2020). "Fully-integrated spiking neural network using siox-based rram as synaptic device," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Genova), 145–148. doi: 10.1109/AICAS48895.2020.9073840

Rubin, J., Lee, D. D., and Sompolinsky, H. (2001). Equilibrium properties of temporally asymmetric hebbian plasticity. *Phys. Rev. Lett.* 86, 364–367. doi: 10.1103/PhysRevLett.86.364

Sengupta, A., Ye, Y., Wang, R., Liu, C., and Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Front. Neurosci.* 13:95. doi: 10.3389/fnins.2019.00095

Sidler, S., Boybat, I., Shelby, R. M., Narayanan, P., Jang, J., Fumarola, A., et al. (2016). "Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: impact of conductance response," in *2016 46th European Solid-State Device Research Conference (ESSDERC)* (Lausanne), 440–443. doi: 10.1109/ESSDERC.2016.7599680

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3, 919–926. doi: 10.1038/78829

Spiga, S., Sebastian, A., Querlioz, D., and Rajendran, B. (2020). *Memristive Devices for Brain-Inspired Computing*. Chennai: Woodhead Publishing.

Suri, M., Bichler, O., Querlioz, D., Cueto, O., Periola, L., Sousa, V., et al. (2011). "Phase change memory as synapse for ultra-dense neuromorphicsystems: application to complex visual pattern extraction," in *2011 International Electron Devices Meeting* (Washington, DC), 4.4.1–4.4.4.

Urbanczik, R., and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. *Neuron* 81, 521–528. doi: 10.1016/j.neuron.2013.11.030

Valentian, A., Rummens, F., Vianello, E., Mesquida, T., de Boissac, C. L., Bichler, O., et al. (2019). "Fully integrated spiking neural network with analog neurons and RRAM synapses," in *2019 IEEE International Electron Devices Meeting (IEDM)*, 14.3.1–14.3.4. doi: 10.1109/IEDM19573.2019.8993431

van Rossum, M. C. W., Bi, G. Q., and Turrigiano, G. G. (2000). Stable hebbian learning from spike timing-dependent plasticity. *J. Neurosci.* 20, 8812–8821. doi: 10.1523/JNEUROSCI.20-23-08812.2000

Wang, Z., Yin, M., Zhang, T., Cai, Y., Wang, Y., Yang, Y., et al. (2016). Engineering incremental resistive switching in TaO_x based memristors for brain-inspired computing. *Nanoscale* 8, 14015–14022. doi: 10.1039/C6NR00476H

Yu, S. (2018). Neuro-inspired computing with emerging nonvolatile memory. *Proc. IEEE* 106, 260–285. doi: 10.1109/JPROC.2018.2790840

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. (2013). Stochastic learning in oxide binary synaptic device for neuromorphic computing. *Front. Neurosci.* 7:186. doi: 10.3389/fnins.2013.00186

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer HM declared a past co-authorship with two of the authors SB, SS to the handling editor.

Copyright © 2021 Brivio, Ly, Vianello and Spiga. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

REVIEW
published: 11 May 2021
doi: 10.3389/fnins.2021.611300

Adaptive Extreme Edge Computing for Wearable Devices

Erika Covi[1†], Elisa Donati[2*], Xiangpeng Liang[3], David Kappel[4*], Hadi Heidari[3*], Melika Payvand[2*] and Wei Wang[5*]*

[1] NaMLab gGmbH, Dresden, Germany, [2] Institute of Neuroinformatics, University of Zurich, Eidgenössische Technische Hochschule Zürich (ETHZ), Zurich, Switzerland, [3] Microelectronics Lab, James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom, [4] Bernstein Center for Computational Neuroscience, III Physikalisches Institut–Biophysik, Georg-August Universität, Göttingen, Germany, [5] The Andrew and Erna Viterbi Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa, Israel

OPEN ACCESS

Edited by:
Martin Ziegler,
Technische Universität Ilmenau,
Germany

Reviewed by:
Can Li,
The University of Hong Kong,
Hong Kong
Jiyong Woo,
Kyungpook National University,
South Korea
Jason Kamran Jr. Eshraghian,
University of Michigan, United States

***Correspondence:**
Erika Covi
erika.covi@namlab.com
Elisa Donati
elisa@ini.uzh.ch
Hadi Heidari
hadi.heidari@glasgow.ac.uk
David Kappel
david.kappel@phys.uni-goettingen.de
Melika Payvand
melika@ini.uzh.ch
Wei Wang
wangweifcc@gmail.com

†Formerly with Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, IU.NET, Milan, Italy

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: 28 September 2020
Accepted: 24 March 2021
Published: 11 May 2021

Citation:
Covi E, Donati E, Liang X, Kappel D, Heidari H, Payvand M and Wang W (2021) Adaptive Extreme Edge Computing for Wearable Devices. Front. Neurosci. 15:611300. doi: 10.3389/fnins.2021.611300

Wearable devices are a fast-growing technology with impact on personal healthcare for both society and economy. Due to the widespread of sensors in pervasive and distributed networks, power consumption, processing speed, and system adaptation are vital in future smart wearable devices. The visioning and forecasting of how to bring computation to the edge in smart sensors have already begun, with an aspiration to provide adaptive extreme edge computing. Here, we provide a holistic view of hardware and theoretical solutions toward smart wearable devices that can provide guidance to research in this pervasive computing era. We propose various solutions for biologically plausible models for continual learning in neuromorphic computing technologies for wearable sensors. To envision this concept, we provide a systematic outline in which prospective low power and low latency scenarios of wearable sensors in neuromorphic platforms are expected. We successively describe vital potential landscapes of neuromorphic processors exploiting complementary metal-oxide semiconductors (CMOS) and emerging memory technologies (e.g., memristive devices). Furthermore, we evaluate the requirements for edge computing within wearable devices in terms of footprint, power consumption, latency, and data size. We additionally investigate the challenges beyond neuromorphic computing hardware, algorithms and devices that could impede enhancement of adaptive edge computing in smart wearable devices.

Keywords: neuromorphic computing, edge computing, wearable devices, learning algorithms, memristive devices

1. INTRODUCTION

Wearable devices can monitor various human body symptoms ranging from heart, respiration, movement, to brain activities. Such miniaturized devices using different sensors can detect, predict, and analyze the physical performance, physiological status, biochemical composition, and mental alertness of the human body. Despite advances in novel materials that can improve the resolution and sensitivity of sensors, modern wearable devices are facing various challenges, such as low computing capability, high power consumption, high amount of data to be transmitted, and low speed of the data transmission. Conventional wearable sensing solutions mostly transmit the collected data to external servers for off-chip computing and processing. This approach typically creates an information bottleneck acting as one of the major limiting factors in lowering the power consumption and improving the speed of the operation of the sensing systems. In addition, the use of conventional remote servers with conventional signal processing techniques for processing

these temporal real-time sensing data makes it computationally intensive and results in significant power consumption and hardware occupation. In this scenario, the edge computing paradigm, whose definition typically includes all the networks where the computation node is not in the cloud, has become very attractive. Indeed, the closer the computing unit to the sensing one, the more power efficient. In particular, a system is defined able of "extreme edge computing" when the data processing occurs right next to the sensor, on the same device (Rubino et al., 2021). This paradigm calls for a radical shift of perspective. Indeed, general-purpose systems are powerful and versatile, but they do not take the diversity of the quantity and quality of the information generated by different devices into account. In this respect, a custom solution which optimizes the available resources to perform the task at hand might prove to be more advantageous in terms of power, area, and latency than a general-purpose one. Moreover, even when computing is moved to the extreme edge, standard processing units might not provide the ideal solution to the aforementioned issues. Standard von-Neumann architectures feature a physical separation between memory and processing unit, thus further increasing the power consumption to shuttle data between units. Such solutions always need a trade-off between power lifetime and computing capability. Bringing computing at the edge enables faster response times and opens the possibility of personalized always-on wearable devices able for continuously interacting and learning with the environment. However, a radical change of paradigm which uses innovative algorithms, circuits and memory devices is needed to maximize the system performance whilst keeping power and memory budgets at a minimum.

Conventional computers, using Boolean and bit-precise digital representations and executing operations with time-multiplexed and clocked signal, are not optimized for fuzzy inputs and complex cognitive tasks, such as pattern recognition, time series prediction, and decision making. Deep Artificial Neural Networks (ANNs) on the other hand have demonstrated amazing results in a wide range of pattern recognition tasks including machine vision, Natural Language Processing (NLP), and speech recognition (LeCun et al., 2015; Schmidhuber, 2015). Dedicated hardware ANN accelerators, including Graphical Processing Units (GPUs), Tensor Processing Units (TPUs), and custom Application Specific Integrated Circuits (ASICs) with parallel architectures are being developed to execute these algorithms and obtain high accuracy inference results. GPUs provide a substrate well-suited to the parallel processing nature of the ANNs and their very long memory bus is particularly apt for running Vector Matrix Multiplications (VMMs), which are at the core of the processing in deep neural networks. Therefore, GPUs support the parallelism, though still pales in comparison to the scale of parallelism that exists in the brain, but they consume orders of magnitude more power than that of the brain (Silver et al., 2016), since they are clocked and the memory access is not localized. To solve this problem, ASIC accelerators try to reduce the complexity of the structure by making the system more application specific and using clock gating and specific hardware structure which matches best to the structure of the mapped neural network to reduce power consumption through less memory read and data access (Cavigelli and Benini, 2016; Chen et al., 2016; Lee et al., 2019; Song et al., 2019). For a complete survey on the state-of-the-art ASIC accelerators for biomedical signals refer to Azghadi et al. (2020).

To go even further in power savings, there are two problems to be solved: (i) remove clock and (ii) perform computation with co-localization of memory and processor. The first problem calls for the development of event-based systems, where processing is performed "asynchronously," i.e., only when there are input "events." The algorithmic basis for this kind of "asynchronous" processing is Spiking Neural Network (SNN), in which neurons spike asynchronously only to communicate information to each other.

To avoid the data movement between the memory and the processor, the memory element should be not only used to store data but also to perform computation inside the processor. This approach is called "in-memory computing." These two approaches of (i) event-based systems and (ii) in-memory computing, together with (iii) massive parallelism, are the three fundamental principles which have led to the development of neuromorphic computing, and to the realization of highly efficient neuromorphic platforms (Schemmel et al., 2010; Furber et al., 2014; Merolla et al., 2014; Moradi et al., 2017; Davies et al., 2018; Frenkel et al., 2019a). Therefore, in this article, we will refer to event-based highly parallel systems that are able to perform real-time sensory processing.

Despite that current fully Complementary Metal-Oxide-Semiconductor (CMOS) implementations of neuromorphic platforms have shown remarkable performance in terms of power efficiency and classification accuracy, there are still some bottlenecks hindering the design of embedded sensing and processing systems. First, the memory used is typically Static Random Access Memory (SRAM), which has very low static power consumption, but it is a large element (six transistors per cell) and it is volatile. The latter feature implies that the information about the network configuration has to be stored elsewhere and transferred to the system at its startup. For large networks, it may take tens of minutes before the system is ready for normal operation. Second, always-on adaptive systems need to work with time constants that have the same time-span of the task that is being learned (e.g., longer than seconds). Implementing such long time constants in neuromorphic CMOS circuits is impractical, since it requires large area capacitors.

To overcome the limitations of fully CMOS-based approaches, the intrinsic unique physical properties of emerging memristive devices can be exploited for both long-term (non-volatile) weight storage and short-term (volatile) task-relevant timescales. In particular, non-volatile devices feature retention times on a long time scale ($>$10 years, Cheng et al., 2012; Udayakumar et al., 2013; Goux et al., 2014; Golonzka et al., 2018) while showing weight reconfigurability with voltages compatible with typical CMOS circuits ($\leq$3.3 V). Volatile devices, instead, can have time constants on the order of tens of milliseconds to seconds (Jo et al., 2015; Wang et al., 2017; Wang et al., 2019; Wang et al., 2019c; Yang et al., 2017; Covi et al., 2019), thus being able to emulate biological time constants. This feature is especially

useful to implement spatiotemporal recognition (Wang et al., 2021) or to enable brain inspired algorithms which need to keep trace of the recent neural activity. This non-volatile/volatile property of memristive devices, together with a small footprint and power efficiency, has indeed attracted a lot of interest in the last 10 years (Linares-Barranco and Serrano-Gotarredona, 2009; Ielmini and Wong, 2018; Chicca and Indiveri, 2020). However, memristive technology has to be supported by *ad hoc* theoretically sound biologically plausible algorithms enabling continual learning and capable to exploit the intrinsic physical properties of memristive devices, such as stochasticity, to achieve accuracy performance comparable to state-of-the-art ANN whilst reducing the power consumption.

This review discusses the challenges to undertake for designing extreme edge computing wearable devices for healthcare and biomedical applications in four different categories: (i) the state-of-the-art wearable sensors and main restrictions toward low-power and high performance learning capabilities; (ii) different algorithms for modeling biologically plausible continual learning; (iii) CMOS-based neuromorphic processors and signal processing techniques enabling low-power local edge computing strategies; (iv) emerging memristive devices for more efficient and scalable embedded intelligent systems. We focus on neuromorphic systems as key enabler of extreme edge computing paradigms since they offer a very convenient trade-off between computational capability and power consumption. As graphically summarized in **Figure 1**, we argue that a holistic approach which combines and exploits all the strengths of these four categories in a co-designed system is the key factor enabling future generations of smart sensing systems.

2. WEARABLE SENSORS

Sensors act as the information collector of a machine or a system that can respond to its physical ambient environment. They are able to translate a specific type of information from a physical environment, such as the human body, to an electrical signal (Gao et al., 2016). Wearable devices enable mass ambient data collection from humans and surrounding environment and require miniaturized, flexible, and highly sensitive sensors to capture clear information from the body. However, from processing aspect and to make a signal meaningful toward personalized devices, further development is still needed.

Since the sensing signal is relatively weak and noisy, a readout circuit (normally composed by an amplifier, a conditioning circuit and an analog signal processing unit) is necessary to make the signal readable for a system (Kanoun and Tränkler, 2004; Gao et al., 2016). The subsequent high-level system processes the data and sends commands to actuators for a closed-loop control or interaction (Witkowski et al., 2014; Lopez et al., 2018; Nweke et al., 2018). For various applications ranging from human-machine interfaces (Lopez et al., 2018) to health monitoring (Pantelopoulos and Bourbakis, 2010; Herry et al., 2017), different combinations of sensor and systems have been developed over the past decade (Li et al., 2018c; Liang et al., 2019). The use of machine learning empowers sensors to build novel smart applications. The examples will be provided in this section.

2.1. Wearable Sensors With Machine Learning

Recently, the field of artificial intelligence further boosts the possibility of smart wearable sensory systems. The emerging intelligent applications and high-performance systems require more complexity and demand sensory units to accurately describe the physical object. The decision-making unit or algorithm can therefore output a more reliable result (Khezri and Jahed, 2007; Wu et al., 2016; He et al., 2017; Liang et al., 2018, 2019). Depending on the signal acquiring position, **Figure 1** illustrates four biopotential sensors and two widely used wearable sensors along with their learning systems and applications, which have also been summarized in **Table 1**. As evident from **Table 1**, different sensors have very different specifications in terms of bandwidth and signal amplitude, therefore, the front-end interface needs to be designed taking the sensor features into account. The sensors and systems for the biopotential signal will be introduced first, and the other two wearable sensors will be provided separately. The biopotential signal can be extracted from the human body using a sensor with direct electrode contact. The electrochemical activity of the cells in nervous, muscular, and glandular tissue generates ionic currents in the body. An electrode-electrolyte transducer is needed to convert the ionic current to electric current for the front-end circuit. The electrode that is normally made up of metal can be oxidized by the electrolyte, generating metal ions and free electrons. In addition, the anions in the electrolyte can also be oxidized to neutral atoms and free electrons. These free electrons result in current flow through the electrode. Thus, the surface potential generated by the electrochemical activities in cells can be sensed by the electrode. However, the bio-signals sensed by the electrode are weak and noisy. Before digitizing the collected signals by Analog to Digital Converter (ADC), an analog front-end is essential to provide a readable signal. The design requirements of the front-end for the biopotential electrodes can be summarized as follows: (i) high common mode rejection ratio; (ii) high signal-to-noise-ratio; (iii) low-power consumption; (iv) signal filtering, and (v) configurable gain (Yazicioglu et al., 2008).

2.1.1. Electrocardiography (ECG)

ECG sensor measures the electrical activity generated by the electrochemistry around cardiac tissue. Containing morphological or statistical features, ECG provides comprehensive information for analyzing and diagnosing cardiovascular diseases (Luz et al., 2016; Liang et al., 2020). In previous studies, automatic ECG classification has been achieved using machine learning techniques, such as Deep Neural Network (DNN) (Kiranyaz et al., 2016; Rahhal et al., 2016), Support Vector Machine (SVM) (Zhang et al., 2014; Raj et al., 2016), and Recurrent Neural Network (RNN) (Alfaras et al., 2019; Ortín et al., 2019). According to the Association for the Advancement of Medical Instrumentation, there are five classes of ECG type of interest: normal, ventricular, supraventricular, fusion of normal and ventricular, and unknown beats. These

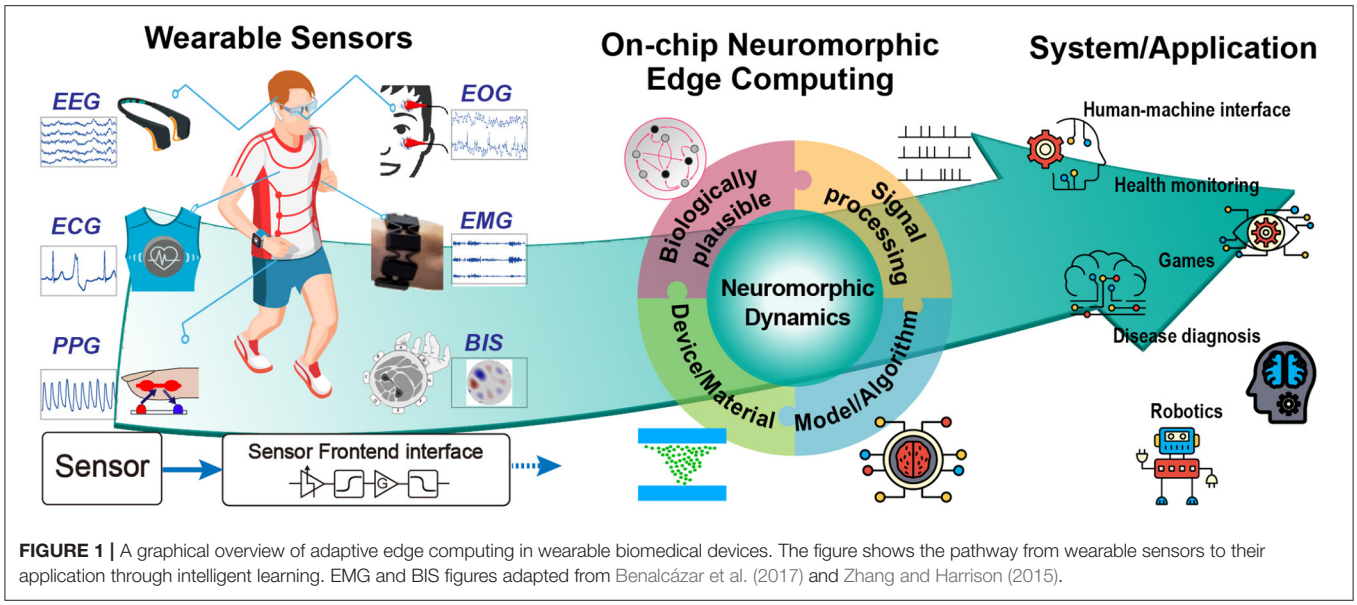


FIGURE 1 | A graphical overview of adaptive edge computing in wearable biomedical devices. The figure shows the pathway from wearable sensors to their application through intelligent learning. EMG and BIS figures adapted from Benalcázar et al. (2017) and Zhang and Harrison (2015).

TABLE 1 | Wearable biomedical signals and sensors.

Signal	Sensor	Position	Signal band (Hz)	Amplitude (mV)	Information	Application
ECG	Electrode	Chest	0.5–200	0.05–3	Heart contraction and relaxation	Heartrate monitoring, cardiovascular disease diagnosis
EMG	Electrode	Forearm surface/implant	20–1,000	0.01–10	Muscle activity	Human-machine interaction
EEG	Electrode	Head surface/implant	0.1–100	0.001–0.1	Brain activity	Brain-computer interface, brain disorder monitoring
EOG	Electrode	Around eye	0.1–10	0.001–0.1	Gaze	Human-machine interaction
BIS	Drive electrodes and measurement electrodes	Body	>0.1	–	Body tissue impedance	Cancer detection, health evaluation, human-machine interaction
PPG	Light emitter and receiver	Body	0.1–10	–	Pulse	Heartrate monitoring, biometric identification

methodologies can be evaluated by available ECG database and yield over 90% accuracy and sensitivity for the five classes, which is essential for future cardiovascular health monitoring. In wearable application, Hossain and Muhammad (2016) and Yang et al. (2016) present systems that measure ECG and send it to the cloud for classification and health monitoring. Furthermore, ECG sensor has been embedded in some of the commercially available devices, such as Apple watch (Apple Inc.), which also enables self-diagnosis for simple cardiovascular disease like atrial fibrillation (Isakadze and Martin, 2020).

2.1.2. Electroencephalography (EEG)

Our brain neurons communicate with each other through electrical impulses. An EEG electrode can help to detect potential information associated with this activity through investigating EEG (Lin et al., 2014; Jebelli et al., 2018) on the surface of the skull. In comparison with other biopotential signals, surface EEG is relatively weak (normally in the range of microvolt-level) and noisy (Gargiulo et al., 2010; Thakor, 2015). Therefore, it requires high input impedance readout circuit and intensive signal pre-processing for clean EEG data (Yazicioglu et al., 2008; Jebelli et al., 2018). While wet-electrode (Ag/AgCl) is more precise and more suitable for clinical purpose, passive dry-electrode is more suitable for daily health monitoring and brain-machine interface (Gargiulo et al., 2010; Li et al., 2015). Besides, the applications also include mental disorder (Shen et al., 2008), driving safety (Lin et al., 2014; Li et al., 2015), and emotion evaluation (Wang et al., 2014b). A commercial biopotential data acquisition system, Biosemi Active Two, provides up to 256 channels for EEG analysis (BioSemi, 2020). For a specific application, we can reduce the number of electrodes to only detect the relevant areas, such as 19 channels for depression diagnosis (Hosseinifard et al., 2013), four channels for evaluating driver vigilance (Lin et al., 2014) and 64 channels for emotional state classification (Wang et al., 2014b). Although EEG is on-body biopotential, most of the existing EEG researches employed offline learning and analysis because of the system complexity and the high number of channels. In wearable real-time applications, a smaller number of channels are usually selected and the data are wirelessly sent to

cloud for further processing (Lin et al., 2014; Li et al., 2015; Xu et al., 2017; Hwang et al., 2018).

2.1.3. Electrooculography (EOG)

The eye movement, which results in potential variations around eyes as EOG, is a combined effect of environmental and psychological changes. It returns relatively weak voltage (0.01–0.1 mV) and low frequency (0.1–10 Hz) (Thakor, 2015). Differently from other eye tracking techniques using a video camera and infrared, EOG provides a lightweight, inexpensive and fully wearable solution to access human's eye movement (Duchowski, 2007). It is the most widely used wearable human-machine interface, especially for assisting quadriplegics (Duchowski, 2007). It has been used to control a wheelchair (Eid et al., 2016), control a prosthesis limb (Duvinage et al., 2011; Witkowski et al., 2014), and evaluate sleeping (Piñero et al., 2004; Zhu et al., 2014; Barua et al., 2019). Additionally, recent studies fuse EEG and EOG to increase the degree of freedom of signal and enhance the system reliability, since they have similar implicit information, such as sleepiness (Martin et al., 1972; Barua et al., 2019) and mental health (Stevens et al., 1979). EOG can also act as a supplement to provide additional functions or commands to an EEG system (Punsawad et al., 2010; Wang et al., 2014a; Witkowski et al., 2014).

2.1.4. Electromyography (EMG)

EMG is an electrodiagnostic method for recording and analyzing the electrical activity generated by skeletal muscles. EMG is generated by skeletal muscle movement, which frequently occurs in arms and legs. It yields higher amplitude (up to 10 mV) and bandwidth (20–1,000 Hz) compared to the other biopotentials (Yazicioglu et al., 2008; Thakor, 2015). Near the active muscle, different oscillation signals can be measured by a dry electrode array, which allows the computer to sense and decode body motion (Rissanen et al., 2008; Wang et al., 2010; Mendez et al., 2017). A prime example is the Myo armband of Thalmic Labs, which is a commercial multi-sensor device that consists of EMG sensors, gyroscope, accelerometer and magnetometer (Rawat et al., 2016). The sensory data is sent to phone or PC via Bluetooth, where various body movements can be classified by feature extraction and machine learning techniques. Moreover, the application of EMG is frequently linked to target control like a wheelchair (Inhyuk et al., 2005) and prosthetic hand (Cipriani et al., 2008; Artemiadis and Kyriakopoulos, 2011) for assisting disabled people. In addition, its application also includes sign language recognition (Mendez et al., 2017), diagnosis of neuromuscular disorders (Rissanen et al., 2008; Subasi, 2013), analysis of walking strides (Wang et al., 2010), and virtual reality (Rincon et al., 2016). Machine learning enables the system to overcome the variation of EMG signals from different users (Rissanen et al., 2008; Mendez et al., 2017).

2.1.5. Photoplethysmography (PPG)

PPG is an non-invasive and low-cost optical measurement method that is often used for blood pressure and heart rate monitoring in wearable devices. The optical properties in skin and tissue are periodically changing due to the blood flow driven by the heartbeat. By using a light emitter toward the skin surface, the photosensor can detect the variations in light absorption, normally from wrist or finger. This signal variation is called PPG, which is highly relevant to the rhythm of the cardiovascular system (Biswas et al., 2019b). Compared with ECG, PPG is easily accessible and low cost, which makes it an ideal intermedia of wearable heart rate measurement. Wrist-PPG has already been deployed in various commercial smartwatches or wristbands, such as Apple Watch, Fitbit Charge, and TomTom Touch, for heart-rate monitoring (Hough et al., 2017). The main disadvantage against ECG is that the PPG is relatively less informative and not unique for different persons and body positions. Thus, further analysis of PPG requires machine learning or other statistics tools for calibrating the signal to different scenarios. For example, it can be used in biometric identification after deep learning (Reşit Kavsaoğlu et al., 2014; Biswas et al., 2019a). It is worth mentioning that PPG can be also a strong supplementary indicator in the application of ECG.

2.1.6. Bioimpedance spectroscopy (BIS)

BIS is another low-cost and powerful sensing technique that provides informative body parameters. The principle is that cell membrane behaves like a frequency-dependent capacitor and impedance. The emitter electrodes generate multifrequency excitation signal (0.1–100 MHz) on the skin while the receiver electrodes collect these currents for demodulating the impedance spectral data of the tissue in between (Matthie, 2008; Caytak et al., 2019). Compared to homogeneous materials, body tissue presents more complicated impedance spectra due to the cell membranes and macromolecules. Therefore, the tissue conditions, such as muscle concentration, structural, and chemical composition, can be analysed through BIS. The BIS can measure body composition, such as fat and water (Matthie, 2008). Based on the different setup in terms of position and frequency, it can also be helpful in the early detection of diseases, such as lymphedema, organ ischemia, and cancer (Sun et al., 2010). Furthermore, multiple pair-wise electrodes can form electrical impedance tomography that describes impedance distribution. By embedding these electrodes in a wristband, the tomography can estimate hand gestures after training, which is another novel solution of inexpensive human-machine interface (Zhang et al., 2016).

2.2. Multisensory Fusion in Wearable Devices

Every sensor has its own limitation. In some demanding cases, a single sensor itself cannot satisfy the system requirement, such as accuracy or robustness (Khaleghi et al., 2013; Alsheikh et al., 2014; Gravina et al., 2017; Liang et al., 2019). The solution involves increasing the number and type of sensors to form a multisensory system or sensor network for one measured target (Khaleghi et al., 2013; Alsheikh et al., 2014; Gravina et al., 2017). Multiple types of sensor synergistically working in a system provide more dimensions of input to fully map an object onto the data stream. Different sensors return different data with respect to sampling rate, number of inputs and the information behind the data. Machine learning models, such as ANN and

SVM, can be designed to combine multiple sources of data. Depending on the application, sensor types and data structure, several approaches have been proposed for multisensory fusion. Generally, in such a system, machine learning is frequently used and plays a vital role in merging different sources of sensory data based on its multidimensional data processing mechanism. The machine learning algorithms enable sensory fusion to occur at the signal, feature or decision level (Khaleghi et al., 2013; Gravina et al., 2017). When dealing with SNN, the multi-sensory features or raw-data need to be encoded and fused in spike sequences in order to fit the input modality of the spike-based neural network. Furthermore, encoding the information in spikes can also further attenuate the risk of catastrophic forgetting issue in conventional neural networks (Azghadi et al., 2020). For decision level fusion, a voting mechanism is typically needed to output the final result after receiving the decisions from different sources of sensors which may be processed by different networks with different algorithms (Li et al., 2017). The results showed that a multisensory system is advantageous in improving system performance. For example, the fusion of ECG and PPG patterns can be an informative physiological parameter for robust medical assessment (Rundo et al., 2018). Counting the peak intervals between PPG and ECG can estimate the arterial blood pressure (He et al., 2014). Interestingly, a recent study shows that the QRS complex of ECG can be reconstructed from PPG by a novel transformed attentional neural network after training (Chiu et al., 2020). This could be beneficial for the accessibility of wearable ECG.

2.3. Challenges Toward Smart Wearable Sensors With Edge Computing

The novel applications using multiple sensors and high learning ability usually require more energy in the wearable computing unit (Pantelopoulos and Bourbakis, 2010). Nevertheless, the power supply in the wearable domain is a difficulty with existing battery technologies. This weakness limits the further development of smart wearable devices (Pantelopoulos and Bourbakis, 2010). The existing solution is to wirelessly transfer the raw data onto a cloud where the computationally intensive algorithm is implemented (Patel et al., 2016). However, this solution is not ideal considering (i) the complexity of using a wireless module, (ii) the non-negligible power consumption, (iii) the amount of data, (iv) the space limitation due to the range of wireless transmission, (v) privacy issues due to the broadcast of signals, and (vi) non-negligible time latency due to communication channel. These technological drawbacks strongly limit the application of wearable sensors.

Implementations of ANN in von Neumann architectures, which have been frequently used in sensors, result therefore in a non-optimized distribution of the energy consumption. Conversely, it has been reported that signal processing activity in the brain is several orders of magnitudes more power-efficient and one order in processing rate better than digital systems (Mead, 2020). Compared to conventional approaches based on a binary digital system, brain-inspired neuromorphic hardware has yet to be advanced in the contexts of data storage and removal as well as their transmission between different units. In this perspective, a neuromorphic chip with a built-in intelligent algorithm can act as a front-end processor next to the sensor. The conventional ADCs could be replaced by a delta encoder or feature extractor converting the sensor analog output to spike-based signal for the hardware (see Section 4). In the end, the output becomes the result of recognition or prediction instead of an intensive data stream. In this way, the computation occurs at the local edge under low power and brain-like architecture. In summary, the research on on-chip neuromorphic edge computing is a multidisciplinary topic involving biologically plausible algorithms, device/material engineering, system modeling/co-design, and signal processing (**Figure 1**). The following sections will provide more comprehensive discussion toward these subjects.

3. ALGORITHMS FOR BIOLOGICALLY PLAUSIBLE CONTINUAL LEARNING

In this section we will highlight some recently introduced methods to port the power of modern machine learning to neuromorphic edge devices. In the last couple of years, machine learning has made big steps forward reaching close-to human performance on a wide range of tasks. Many of the most successful machine learning methods are based on artificial neural networks (ANN), which are inspired by the organization of information processing in the brain. However, somewhat contradictory—mapping modern ANN learning methods to brain-inspired hardware poses considerable challenges to the algorithm and hardware design. The main reason for this is, that the development of machine learning algorithms has been strongly influenced by the development of powerful mainframe computers that perform learning offline in big server farms only eventually sending back results to the user. While this development has paved the ground for today's success of ANNs, it has also lead the field away from following the principles used in biology for efficient learning.

Neuromorphic realizations of on-chip learning have therefore often focused on biologically inspired learning rules, such as Spike-Timing Dependent Plasticity (STDP). In this model, synaptic weight changes only take place if pre-synaptic spikes arrive at the synapse, which makes them very well-suited for event-based algorithms (Diehl and Cook, 2014; Chen et al., 2018; Li et al., 2018b; Lin et al., 2018). In this section we focus on algorithmic advances that combine the efficiency of bio-inspired plasticity rules with modern machine learning approaches. In the following section 3.1 we will review recent approaches to combine the strengths of modern machine learning and brain-inspired algorithms, that are of particular interest for edge computing applications. In section 3.2 we will focus on the problem to cope with extreme memory constraints by exploiting sparsity. In section 3.3 we will highlight additional open challenges and future work.

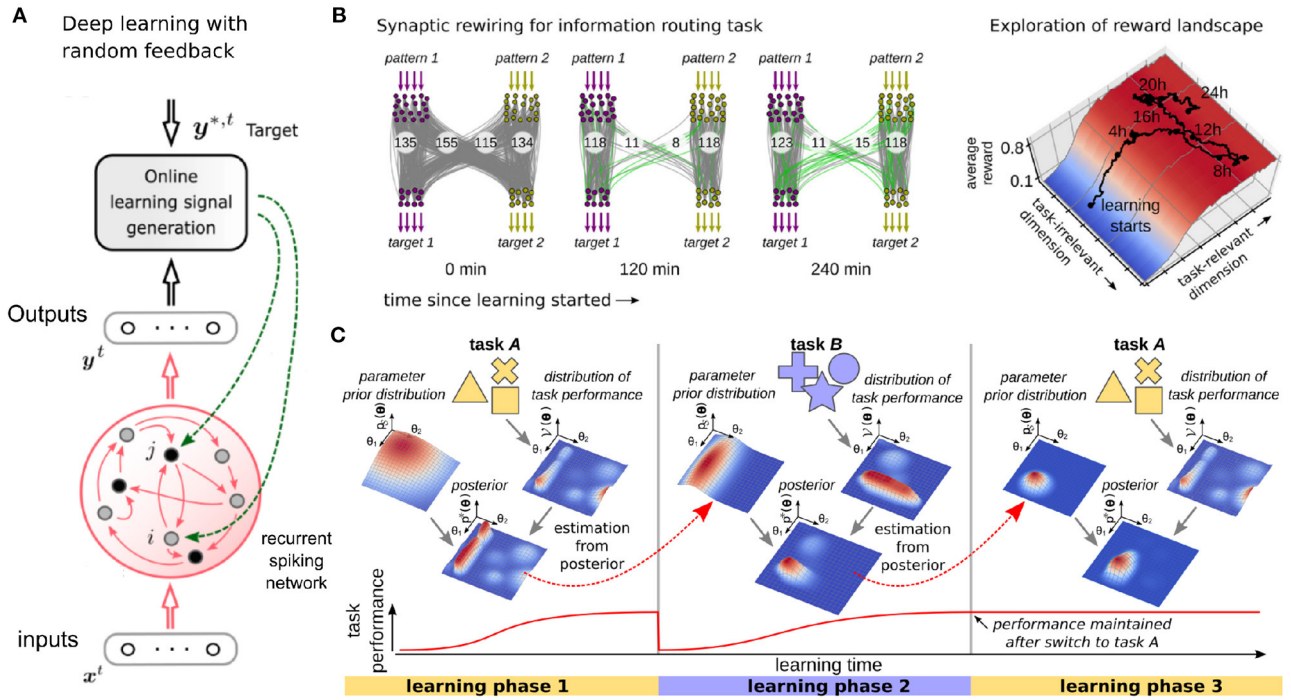


FIGURE 2 | Biologically inspired algorithms of learning in spiking neural networks. **(A)** The e-prop algorithm (Bellec et al., 2019) approximates back-propagation through time using random feedback to propagate error signals to synapses of a recurrent SNN (adapted from Bellec et al., 2020). **(B)** Synaptic sampling (Kappel et al., 2015) exploits the variability of learning rules and redundancy in the task solution space to learn sparse and robust network configurations (adapted from Kappel et al., 2018). **(C)** Overcoming forgetting by selectively slowing down weight changes (Kirkpatrick et al., 2017). After learning a first task A, parameter distributions are absorbed into a prior distribution that confines the motility of synaptic weights in subsequent tasks (task B).

3.1. Brain-Inspired Learning Algorithms for Neuromorphic Hardware

Today, the dominating method for training artificial neural networks is the error backpropagation (Backprop) algorithm (Rumelhart et al., 1986), which provides an efficient and scalable solution to adapting the network parameters to a set of training data. Backprop is an iterative, gradient-based, supervised learning algorithm that operates in three phases. First, a given input activation is propagated through the network to generate the output based on the current set of parameters. Then, the mismatch between the generated outputs and target values is computed using a loss function, and propagated backwards through the network architecture to compute suitable weight changes. Finally, the network parameters are updated to reduce the loss. We will not go into the details behind Backprop here, but see Schmidhuber (2015) for an excellent review and historical survey of the development of the algorithm. The problem of porting Backprop to neuromorphic hardware stems from a well-known shortcoming of the algorithm known as *locking* (Czarnecki et al., 2017). The weights of a network can only be updated after a full forward propagation of the data through the network, followed by loss evaluation. A learning cycle ends after waiting for the back-propagation of error gradients, which makes an efficient implementation of Backprop on online distributed architectures challenging. Also, Backprop is not well-suited for spiking neural networks which have non-differentiable output functions. These problems have been recently addressed in brain-inspired variants of the Backprop algorithm.

3.1.1. Brain-Inspired Alternatives to Error Backpropagation

In recent years a number of methods have been proposed to approximate the gradient computation performed by Backprop in order to prevent locking (see Richards et al., 2019 for a recent review). Lillicrap et al. (2016) and Samadi et al. (2017) proposed to replace the non-local error back-propagating term of the Backprop algorithm by sending the loss through a fixed feedback network with random weights that are excluded from training. In this approach, named *random feedback alignment* the back-propagating error signal acts as a local feedback to each synapse, similar to a reward signal in reinforcement learning. The fixed random feedback network de-correlates the error signals providing individual feedback to each synapse. Lillicrap et al. could show that this simple approach already provides a viable approximation to the exact Backprop algorithm and performs well for practical machine learning problems of moderate size. In Neftci et al. (2017) an event-based version of random feedback alignment, that is well-suitable for neuromorphic hardware, was introduced. This approach was further generalized in Payvand

et al. (2020a) to include a larger class of algorithms that use error feedback signals.

An efficient model for learning complex sequences in spiking neural networks, named *Superspike*, was introduced in Zenke and Ganguli (2018). The model also uses a learning rule that is modulated by error feedback signals and locally minimizes the mismatch between the network output and a target spike train. To overcome the problem of non-differentiable output, Superspike uses a surrogate gradient approach that replaces the infinitely steep spike events with a finite auxiliary function at the time points of network spike events (Bengio et al., 2013). As in random feedback alignment, learning signals are communicated to the synapses via a feedback network with fixed weights. Using this approach Zenke and others could demonstrate efficient learning of complex sequences in spiking networks.

Another approach to approximate Backprop in spiking neural networks uses an anatomical detail of Cortical neurons. Sacramento et al. (2017) introduced a biologically inspired two-compartment neuron model that approximates the error backpropagation algorithm by minimizing a local dendritic prediction error. Göltz et al. (2019) port learning by Backprop to neuromorphic hardware by incorporating dynamics with finite time constants and by optimizing the backward pass with respect to substrate variability. They demonstrate the algorithm on the BrainScaleS analog neuromorphic architecture.

3.1.2. Brain-Inspired Alternatives to Backpropagation Through Time

Recurrent neural network (RNN) architectures often show superior learning results for tasks that involve a temporal dimension, which is often the case for edge computing applications. Porting learning algorithms for RNNs is therefore of utmost importance for efficient machine learning on the edge. Backpropagation through time (BPTT)—the standard RNN learning method used in most GPU implementations—unfolds the network in time and keeps this extended structure in memory to propagate information forward and backward which poses a severe challenge to the power and area constraints of edge computing. Recent theoretical results (Bellec et al., 2018, 2019) show that the power of BPTT can be brought to biologically inspired spiking neural networks (SNN) while at the same time the unfolding can be prevented in an approximation that operates only forward in time, enabling *online, always-on* learning. This algorithm operates at every synapse in parallel and incrementally updates the synaptic weights. As for random feedback alignment and Superspike discussed above, the weight update depends only on three factors, where the first two are determined by the states of the two related input/output neurons, and the third is given by synapse-specific feedback conveying the mismatch between the target and the actual output (see **Figure 2A** for an illustration). The temporal gap between these factors is mitigated by an *eligibility trace* describing a transient dynamic. Eligibility traces, have been theoretically predicted for a long time (Williams, 1992; Izhikevich, 2007), and have also recently been observed experimentally in the brain (Yagishita et al., 2014; Brzosko et al., 2015; He et al., 2015; Bittner et al., 2017).

3.2. Efficient Learning Under Stringent Memory Constraints

The amount of available resources in neuromorphic systems is kept low to increase energy efficiency. Memory elements are especially impactful on the energy budget. Therefore, algorithms are needed that make efficient use of the available memory resources. The largest amount of memory in a network is usually consumed by the synaptic weights. Since in practice, the weights of many connections in a network converge to values close to zero, several methods have been proposed to reduce the memory footprint of machine learning algorithms by exploiting sparsity in the network connectivity. Also in many applications the bit precision per synapse can be reduced without significant performance loss which further reduces the memory footprint. We will discuss here three types of algorithms that work under stringent memory constraints: (i) those that are based on *pruning connections after learning*, (ii) *online* learning with *sparse* networks and (iii) *quantization-aware training* that implements learning algorithms in networks with reduced bit precision per weight.

3.2.1. Pruning

Many approaches to exploit sparsity in learning algorithms focus on pruning the network after training (see Gale et al., 2019 for a recent review). Simple methods rely on pruning by magnitude, simply by eliminating the weakest (closest to zero) weights in the network (Ström, 1997; Collins and Kohli, 2014; Han et al., 2015). Some methods based on this idea have reported impressive sparsity rates of over 95% for standard machine learning benchmarks with negligible performance loss (Guo et al., 2016; Zhu and Gupta, 2017). Other methods are based on theoretical motivations and classical sparsification and regularization techniques (Louizos et al., 2017; Molchanov et al., 2017; Ullrich et al., 2017). These models reach high compression rates. Dai et al. (2019) proposed a method to iteratively grow and prune a network in order to generate a compact yet precise solution. They provide a detailed comparison with state of the art dense networks and other pruning methods and reaching sparsity above 99% for the LeNet-5 benchmark.

3.2.2. Online Learning in Sparse Networks

A number of authors also introduced methods that work directly with sparse networks during training, which is often the more interesting case for neuromorphic applications with online training. Bellec et al. (2017) introduced an algorithm for online stochastic rewiring in deep neural networks that works with a fixed number of synaptic connections throughout learning. The algorithm showed close-to state of the art performance at up to 98% sparsity. Sparse evolutionary training (SET) (Mocanu et al., 2018) introduced a heuristic approach that prunes the smallest weights and regrows new weights in random locations. Dynamic Sparse Reparameterization (Mostafa and Wang, 2019) introduces a prune-redistribute-regrowth cycle. They demonstrated compelling performance levels also for very deep neural network architectures. Lee et al. (2018) introduced a single shot pruning algorithm that yields sparse networks based on a saliency criterion prior to the actual training. Dettmers

and Zettlemoyer (2019) introduced a refined method for online pruning and redistribution that surpasses the previous methods in terms of sparsity and learning performance.

3.2.3. Quantization-Aware Training

Quantization-aware training is today a common method applied in commercial and practical settings to port machine learning to hardware with reduced bit precision per synapse. Several approaches have been proposed. *Stochastic rounding* translates the weight update into a probability and flips the weights to the closest quantized value. This method has been applied to online and offline learning with very low bit resolutions of down to 2 bits per synapse (Müller and Indiveri, 2015; Müller et al., 2017). (Hubara et al., 2016) introduced a *binary deep neural network* architecture that uses only two weight values (+1 and −1), achieving compelling learning performance. The weight quantization was implemented with smooth functions so that widely available implementations of error Backpropagation could be used. Wang et al. (2018a) and Sun et al. (2019) demonstrated deep learning in large state-of-the-art networks with 8-bit precision floating point weights. Finally in recent work (Choi et al., 2020) regularization, quantization and pruning was combined to train compressed deep learning models and a detailed performance analysis was provided.

3.3. Open Challenges and Future Work

As outlined above, edge computing poses quite specific challenges to learning algorithms that are substantially different from requirements of classical applications. Some of the algorithms outlined above have already been successfully ported to neuromorphic hardware. For example, the e-prop algorithm of Bellec et al. (2018) has been implemented on the SpiNNaker 2 chip yielding an additional energy reduction by two orders of magnitude compared to a X86 implementation (Liu et al., 2018). See the next section 4 for more details on available neuromorphic hardware and their applications.

In the remainder of this section we will highlight open challenges that remain to be solved for efficient learning in edge computing applications. In addition to the stringent memory and power constraints learning at the edge also has to function in an online scenario where data arrive in a continuous stream. Some dedicated hardware resources, e.g., like memristive devices discussed in section 5, may also show high levels in intrinsic variability, so the learning algorithm should be robust against these noise sources. In this section we discuss recent advances in this line of research and provide food for thought on how these specific challenges can be approached in future work.

3.3.1. Robust Learning Algorithms for Neuromorphic Devices Exploiting Device Noise

Here we review recent advances in using inspiration from biology to make learning algorithms robust against device variability. Several authors have suggested that device noise and variability should not be seen as a nuisance, but rather can serve as a computational resource for network simulation and learning algorithms (see Maass, 2014 for a thorough discussion). Pecevski and Maass (2016) have shown that variability in neuronal outputs can be exploited to learn complex statistical dependencies between sensory stimuli. The stochastic behavior of the neurons is used in this model to compute probabilistic inference, while biologically motivated learning rules, that only require local information at the synapses can be used to update the synaptic weights. A theoretical foundation of the model shows that the spiking network performs a Markov chain Monte Carlo sampling process, that allows the network to "reason" about statistical problems.

This idea is taken one step further in Neftci et al. (2015) by showing that also the variability of synaptic transmission can be used for stochastic computing. The intrinsic noise of synaptic release is used to drive a sampling process that can be implemented in an event-based fashion. In Kappel et al. (2015) it was shown that the variability of learning rules and weight parameters gives rise to a biologically plausible model of online learning. The intrinsic noise of synaptic weight changes drives a sampling process that can be used to exploit redundancies in the task solution space (see **Figure 2B** for an illustration). This model was applied to unsupervised learning in spiking neural networks, and to closed-loop reinforcement learning problems (Kappel et al., 2018; Kaiser et al., 2019). In Yan et al. (2019) this model was also ported to the SpiNNaker 2 neuromorphic many-core system.

3.3.2. Biologically Motivated Mechanisms to Combat Forgetting in Always-on Learning Scenarios

Neuromorphic systems often operate in an environment where they are permanently on and learning a continuous stream of data. This mode of operation is quite different from most other machine learning applications that work with hand-labeled batches of training data. Always-on learning inevitably leads to forgetting previously learned sensory experiences as a necessary consequence of applying weight updates over time (Fusi et al., 2005; Benna and Fusi, 2016). Inspiration to solve the associated stability-plasticity problem by protecting relevant information comes from biology. The mammalian brain seems to combat forgetting relevant memories by actively protecting previously acquired knowledge in neocortical circuits (Pan and Yang, 2009; Yang et al., 2009, 2014; Cichon and Gan, 2015; Hayashi-Takagi et al., 2015). When a new skill is acquired, a subset of synapses is strengthened, stabilized and persists despite the subsequent learning of other tasks (Yang et al., 2009).

A theoretical treatment of the forgetting problem was conducted in the *cascade model* of Stefano Fusi and others (Fusi et al., 2005; Benna and Fusi, 2016). They could show that learning an increasing number of patterns in a single neural network leads unavoidably to a state which they called catastrophic forgetting. Trying to train more patterns into the network will interfere with all previously learned ones, effectively wiping out the information stored in the network. The proposed cascade model to overcome this problem uses multiple parameters per synapse that are linked through a cascade of local interactions. This cascade of parameters selectively slows down weight changes, thus stabilizes synapses when required and effectively combats effects of catastrophic forgetting. A related model, that uses multiple parameters per synapse to combat forgetting was used

in Kirkpatrick et al. (2017) (see also Huszár, 2018 for a recently introduced variation of the model). They used a Bayesian approach that infers a prior distribution over parameter values at each synapse. Synapses that stabilize during learning (converge to a fixed solution) will be considered relevant in subsequent learning and Bayesian priors help to maintain their values (see **Figure 2C** for an illustration).

Another promising biologically inspired method that has recently gained attention in machine learning, and which may enable a system to benefit from large amounts of unlabeled data, is *self-supervised learning*. This technique augments the learning problem with pretext tasks which can be formulated using only unlabeled data, but do require higher-level semantic understanding in order to be solved (Hendrycks et al., 2019; Zhai et al., 2019). These pretext tasks typically involve a simple manipulation of the input, such as image rotation, for which a target objective can be computed without supervision (Kolesnikov et al., 2019). A promising recent approach combines self-supervised learning and semi-supervised learning where sparse labeled data is used to enhance the model performance (Zhai et al., 2019). This method that incorporates sparse feedback from a supervisor might be of particular interest for edge devices.

3.3.3. Biologically Motivated Mechanisms to Enhancing Transfer and Sensor Fusion

Distributed computing architectures at the edge need to make decisions by integrate information from different sensors and sensor modalities and should be able to best make use of the sensory information across a wide range of tasks. It is clearly not very efficient to learn from scratch when confronted with a new task. Therefore, to boost the performance of edge computing, we consider here two aspects of transferring information to new situations: transfer of knowledge between sensors (*sensor fusion*), which has been treated in section 2.2, and transfer of knowledge between multiple different tasks (*transfer learning*).

Transfer learning denotes the improvement of learning in a new task through the use of knowledge from a related task that has already been learned previously (Caruana, 1997; Torrey and Shavlik, 2010). This contrasts most other of today's machine learning applications that focus on one very specific task. In transfer learning, when a new task is learned, knowledge from previous skills can be reused without interfering with them. For example, the ability to perform a tennis swing can be transferred to playing ping pong, while maintaining the ability to do both sports. The literature on transfer learning is extensive and many different strategies have been developed depending on the relationship between the different task domains (see Lu et al., 2015 and Weiss et al., 2016 for systematic reviews). In machine learning a number of approaches have been applied to a wide range of problems, including classification of images (Kulis et al., 2011; Zhu et al., 2011; Duan et al., 2012; Long et al., 2017), text (Prettenhofer and Stein, 2010; Wang and Mahadevan, 2011; Zhou et al., 2014a,b), or human activity (Harel and Mannor, 2010).

A very general approach to learn across multiple domains is followed in the *learning to learn* framework of Schmidhuber (1992, 1993). Their model features networks that are able to modify their own weights through the network activity. These network are therefore able to tinker with their own processing properties. This approach has been taken to its most extreme form where a network leans to implement an optimization algorithm by itself (Andrychowicz et al., 2016). This model consists of an outer-loop learning network (*the optimizer*) that controls the parameters of an inner-loop network (*the optimizee*). The training algorithm of the inner-loop network works on single tasks that are presented sequentially, whereas the outer-loop learner operates across tasks and can acquire strategies to transfer knowledge. This learning-to-learn framework was recently applied to SNNs to obtain properties of LSTM networks and use them to solve complex sequence learning tasks (Bellec et al., 2018). In Bohnstingl et al. (2019), the learning-to-learn framework was also applied to a neuromorphic hardware platform.

4. SIGNAL PROCESSING FOR WEARABLE DEVICES ON NEUROMORPHIC CHIP

Neuromorphic engineering is a branch of electrical engineering dedicated to the design of analog/digital data processors that aims to emulate biological neurons and synapses. The key technological advantage of neuromorphic chips lies in (i) their power efficiency as a result of reducing data movement through co-location of memory and processor and sparsifying the temporal information through events (spikes); (ii) their low latency since they enable the real-time processing of signals through temporal dynamics and (iii) their adaptive properties which enable adjusting their parameters to the environment they are being employed.

This increasing interest in neuromorphic engineering shows that hardware SNNs are considered a key future technology with high potential in key application, such as edge computing, and wearable devices.

Neuromorphic technologies have sparked interest from universities (Furber et al., 2014; Qiao et al., 2015; Moradi et al., 2017; Neckar et al., 2018; Schemmel et al., 2020) and companies, such as IBM (Merolla et al., 2014) and Intel (Davies et al., 2018). There are two main approaches of fully-digital and analog/digital mixed-signal that have been taken to design event-driven neuromorphic chips. The similarities between the two types are the employment of events and sending packets for communicating information between different computational cores. The employed communication scheme is Address-Event Representation (AER), where the communicating neurons place their address on a shared communication bus whenever they spike. The difference between the two approach is the way the computation is done. In the digital approach, the VMM and the dynamics are calculated using bit-precise and time-stepped approach, whereas in the mixed-signal approach the physics of the computational substrate is used.

In this section, we will provide an overview of the neuromorphic platforms, that to the best of our knowledge were deployed for biomedical signal processing, showing promising results to be exploited in wearable devices.

4.1. Neuromorphic Processors

4.1.1. TrueNorth

TrueNorth (Merolla et al., 2014) is IBM's neuromorphic chip that uses a digital approach for both processing and communication. One million neurons arranged in a tiled array of 4,096 neurosynaptic cores enable *massive parallel processing*. Each core contains 13 kB of *local SRAM memory* to keep neurons and synapse's states along with the axonal delays and information on the fan-out destination. There are 256 Leaky-Integrator and Fire (LIF) neurons implemented by time-multiplexing and 256 million synapses are designed in the form of SRAM memory. Each core can support up to 256 fan-in and fan-out, and this connectivity can be configured such that a neuron in any core can communicate its spikes to any other neuron in any other core.

Thanks to the *event-driven nature*, the co-location of memory and processing units in each core, and the use of low-leakage silicon CMOS technology, TrueNorth can perform 46 billion synaptic operations per second (SOPS) per watt for real-time operation, with 26 pJ per synaptic event. Its power density of 20 mW/cm^2 is about three orders of magnitude smaller than that of typical CPUs.

4.1.2. SpiNNaker

The SpiNNaker machine (Furber et al., 2014), designed by the University of Manchester, is a custom-designed ASIC based on *massively parallel architecture* that has been designed to efficiently simulate large spiking neural networks. It consists of ARM968 processing cores arranged in a 2D array where the precise details of the neurons and their dynamics can be programmed. Although the processing cores are synchronous microprocessors, the *event-based* aspect of SpiNNaker is apparent in its message-handling paradigm. A message (event) gets delivered to a core generating a request for being processed. The communications infrastructure between these nodes is specially optimized to carry very large numbers of very small packets, optimal for spiking neurons.

A second generation of SpiNNaker was designed by Technical University of Dresden (Mayr et al., 2019). Spinnaker2 continues the line of dedicated digital neuromorphic chips for brain simulation increasing the simulation capacity by a factor >10 while staying in the same power budget (i.e., $10\times$ better power efficiency). The full-scale SpiNNaker2 consists of 10 Million ARM cores distributed across 70,000 Chips in 10 server racks. This system takes advantage of advanced 22 nm FDSOI technology node with Adaptive Body Biasing enabling reliable and ultra-low power processing. It also features incorporating numerical accelerators for the most common operations.

4.1.3. Loihi

Loihi (Davies et al., 2018) is Intel's neuromorphic chip with many-core processing incorporating on-line learning designed in 14 nm FinFET technology. The chip supports about 130,000 neurons and 130 million synapses distributed in 128 cores. Spikes are transported between the cores in the chip using packetized messages by an asynchronous network on chip. It includes three embedded $\times 86$ processors and provides a very flexible learning engine on which diverse online learning algorithms, such as STDP, different three factor and trace-based learning rules can be implemented. The chip also provides hierarchical connectivity, dendritic compartments, synaptic delays as different features that can enrich a spiking neural network. The synaptic weights are stored on local SRAM memory and the bit precision can vary between 1 and 9 bits. All logic in the chip is digital, functionally deterministic, and implemented in an asynchronous bundled data design style.

4.1.4. DYNAP-SE

DYNAP-SE implements a multi-core neuromorphic processor with scalable architecture fabricated using a standard 0.18 μm CMOS technology (Moradi et al., 2017). It is a full-custom asynchronous mixed-signal processor, with a fully asynchronous inter-core and inter-chip hierarchical routing architecture. Each core comprises 256 adaptive exponential integrate-and-fire (AEI&F) neurons for a total of 1k neurons per chip. Each neuron has a Content Addressable Memory (CAM) block, containing 64 addresses representing the pre-synaptic neurons that the neuron is subscribed to. Rich synaptic dynamics are implemented on the chip by using Differential Pair Integrator (DPI) circuits (Bartolozzi and Indiveri, 2007). These circuits produce EPSCs and IPSCs (Excitatory/Inhibitory Post-Synaptic Currents), with time constants that can range from a few μs to hundreds of ms. The analog circuits are operated in the sub-threshold domain, thus minimizing the dynamic power consumption, and enabling implementations of neural and synaptic behaviors with biologically plausible temporal dynamics. The asynchronous CAMs on the synapses are used to store the tags of the source neuron addresses connected to them, while the SRAM cells are used to program the address of the destination core/chip that the neuron targets.

4.1.5. ODIN/MorphIC

Online-learning DIgital spiking Neuromorphic (ODIN) processor occupies an area of only 0.086 mm^2 in 28 nm FDSOI CMOS (Frenkel et al., 2019a). It consists of a single neurosynaptic core with 256 neurons and 256^2 synapses. Each neuron can be configured to phenomenologically reproduce the 20 Izhikevich behaviors of spiking neurons (Izhikevich, 2004). The synapses embed a 3-bit weight and a mapping table bit that allows enabling or disabling Spike-Dependent Synaptic Plasticity (SDSP) locally (Brader et al., 2007), thus allowing for the exploration of both off-chip training and on-chip online learning setups.

MorphIC is a quad-core digital neuromorphic processor with 2k LIF neurons and more than 2M synapses in 65 nm CMOS (Frenkel et al., 2019b). MorphIC was designed for high-density large-scale integration of multi-chip setups. The four 512-neuron crossbar cores are connected with a hierarchical routing infrastructure that enables neuron fan-in and fan-out values of 1k and 2k, respectively. The synapses are binary and can be either programmed with offline-trained weights or trained online with a stochastic version of SDSP.

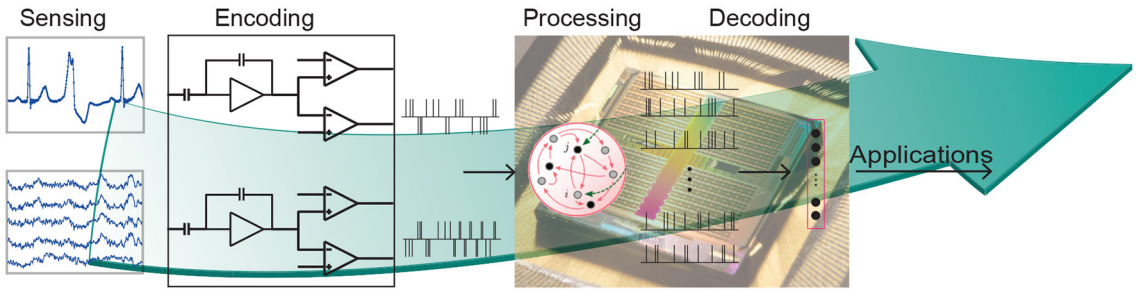


FIGURE 3 | Biomedical signal processing on neuromorphic hardware, from sensors to applications.

4.2. Biomedical Signal Processing on Neuromorphic Hardware

Given the low latency and low power properties of these neuromorphic chips, they are promising candidates for on-edge processing of biomedical signals. **Figure 3** illustrates the different stages of biomedical processing using a neuromorphic system pipeline. The sensory signals should first be encoded to spikes or events which are fed to a neuromorphic SNN processor. Depending on the application, appropriate SNN architecture is mapped onto the chip and the output (e.g., anomaly detection, or gesture recognition) is read out.

4.2.1. Encoding

In SNNs a single spike by itself does not carry any information. However, the number and the timing of spikes produced by a neuron are important. Just as their biological counterpart, silicon neurons in neuromorphic devices produce spike trains at a rate that is proportional to their input current. At the input side, synapse circuits integrate the spikes they receive to produce analog currents, with temporal dynamics and time constants that can be made equivalent to their biological counterparts. The sum of all the positive (excitatory) and negative (inhibitory) synaptic currents afferent to the neuron is then injected into the neuron.

To provide biomedical signals to the synapses of the SNN input layer, it is necessary to first convert them into spikes. A common way to do this is to use a delta-modulator circuit (Corradi and Indiveri, 2015; Sharifshazileh et al., 2019) functionally equivalent to the one used in the Dynamic Vision Sensor (DVS) (Lichtsteiner et al., 2008). This circuit, in practice, is an ADC that produces two asynchronous digital pulse outputs (UP or DOWN) for every biosignal channel in the input. The UP (DOWN) spikes are generated every time the difference between the current and previous value exceeds a pre-defined threshold. The sign of the difference corresponds to the UP or DOWN channel where the spike is produced. This approach was used to convert EMG signals, used in mixed-signal neuromorphic chips (Donati et al., 2018, 2019) and in digital ones (Behrenbeck et al., 2019; Ceolini et al., 2020), ECG signals (Bauer et al., 2019; Corradi et al., 2019), and EEG and High Frequency Oscillation (HFO) ones (Corradi and Indiveri, 2015; Sharifshazileh et al., 2019).

4.2.2. Processing and Decoding

Table 2 shows the summary of neuromorphic processors described previously where biomedical signal processing applications were used. These works show promising results for always-on embedded biomedical systems.

The first chip presented in this table is DYNAP-SE, used to implement SNNs for the classification or detection of EMG (Donati et al., 2018, 2019; Ma et al., 2020a,b) and ECG (Bauer et al., 2019; Corradi et al., 2019) and to implement a simple spiking perceptron as part of a design to detect HFO in human intracranial EEG (Sharifshazileh et al., 2019). In particular, in Donati et al. (2018), Bauer et al. (2019), and Ma et al. (2020a,b) a spiking RNN is deployed for EMG/ECG signal separation to facilitate the classification with a linear read-out. SVM and linear least square approximation is used in the read out layer for Bauer et al. (2019) and Corradi et al. (2019) and overall accuracy of 91% and 95% for anomaly detection were reached, respectively. In Ma et al. (2020a) a RNN was implemented for discriminating three hand gesture using sEMG. Two hardware-friendly spike-based read-out models were used to evaluate the network performances: a rate-based state distance model, and a STDP model. The results show classification accuracy of the state distance method above 75%, better than the SVM approach, whereas the STDP learning rule only achieved 60% accuracy. The system was further expanded in Ma et al. (2020b), where an adapting spike conversion was introduced, improving the performances to 85%. In Donati et al. (2018), the state property of the spiking RNN on EMG was investigated for different hand-gestures. In Donati et al. (2019) the performance of a feedforward SNN and a hardware-friendly spike-based learning algorithm was investigated for hand-gesture recognition using superficial EMG and compared to traditional machine learning approaches, such as SVM. Results show that applying SVM and the spiking learning method on the spiking output of the hidden layer achieved a classification rate of 84% and 74%, respectively. Nevertheless, the latter show a power consumption of about 0.05 mW, two orders of magnitude more power-efficient than the state-of-the-art embedded system (Benatti et al., 2015; Montagna et al., 2018).

Recently, the hand-gesture classification benchmark was implemented and compared on two digital neuromorphic platforms, i.e., Loihi (Davies et al., 2018)

and ODIN/MorphIC (Frenkel et al., 2019a,b) and an embedded GPU, Nvidia Jetson Nano. The systems were using two different sensor modalities, event-driven sensors and EMG to perform sensor fusion. In particular, for processing vision inputs, a spiking Convolutional Neural Network (CNN) was implemented on Loihi and a spiking Multilayer Perceptron (MLP) was implemented on ODIN/MorphIC (Ceolini et al., 2020) while both the platforms used MLP for EMG processing. The difference in the two pipelines is due to the design properties of the neuromorphic systems (i.e., number of neurons, fan-in). However, in both cases, the fusion was performed on the layer before the one of classification, combining the output from the spiking CNN and the spiking MLP for Loihi, and from the two spiking MLPs on ODIN/MorphIC hardware. The same structure was implemented on the embedded GPU and the comparison was performed in terms of accuracy, power consumption, and latency showing that the neuromorphic chips are able to achieve the same accuracy with significantly smaller energy-delay product, 30× and 600× more efficient for Loihi and ODIN/MorphIC, respectively (Ceolini et al., 2020). The comparison was further extended in Azghadi et al. (2020), where the same task was applied to Field Programmable Gate Array (FPGA) and memristive implementations. Results show that neuromorphic hardware presents approximately two orders of magnitude improvement in the energy-delay product when compared to their FPGA counterparts, which highlights the prospective use of such architectures in edge computing.

4.3. Adaptation in Neuromorphic Processor

Local adaptation is an important aspect in extreme edge computing, specially for wearable devices. The current methods for training networks for biomedical signals rely on large datasets collected from different patients. However, when it comes to biological data, there is no "one size fits all." Each patient and person has their own unique biological signature. Therefore, the field of Personalized Medicine (PM) has gained lots of attention in the past few years and the online on-edge adaptation feature of neuromorphic chips can be a game changer for PM.

As was discussed in section 3.1, there is on-going effort in designing spike-based online learning algorithms which can be implemented on neuromorphic chips.

Example of today's state of the art for on-chip learning are Intel's Loihi (Davies et al., 2018), DynapSEL and ROLLS chip from UZH/ETHZ (Qiao et al., 2015; Qiao and Indiveri, 2016), BrainScales from Heidelberg (Schemmel et al., 2010) and ODIN from UC Louvain (Frenkel et al., 2019a). Intel's Loihi includes a learning engine which can implement different learning rules, such as simple pairwise STDP, triplet STDP, reinforcement learning with synaptic tag assignments or any three factor learning rule implementation. DynapSEL, ROLLS and ODIN encompass the SDSP, also known as the Fusi learning rule, which is a form of semi-supervised learning rule that can support both unsupervised clustering applications and supervised learning with labels for shallow networks (Brader et al., 2007). Brainscales chip implements the STDP rule. Moreover, Spinnaker 1 and 2 (Furber et al., 2013; Mayr et al., 2019) can implement a wide variety of on-chip learning algorithms since their designs make use of ARM microcontrollers providing lots of configurability for the users. **Table 2** summarizes the learning algorithms implemented on the neuromorphic chips that have been used for biomedical signal processing. Synaptic bit precision is an important parameter for online learning which is limited on chip due to the memory footprint.

4.4. Open Challenges

4.4.1. System Integration

One of the main challenge in developing a device for Edge Computing is the integration of the sensors with the processor, which is generally valid, but even more in neuromorphic systems. In heterogeneous systems, where sensor and processor are not integrated in the same substrate, the main challenge is due to the lack of a standard in the protocol of communication. Although most of neuromorphic systems, both sensors and processors, implement AER protocol, they present slightly different implementations, i.e., parallel, serial, different AER address width, which makes the integration difficult. Another approach consists of designing sensors and processors on the same substrate. This solution is preferable for wearable solutions where edge computing is required, but it is currently not the case for any neuromorphic chips. Any neuromorphic system, in fact, comprises not only of the neuromorphic core but a digital infrastructure that surrounds the core, i.e., FPGAs and microcontrollers that allow the communication with the external world and the network configuration.

4.4.2. Locality

The learning information for updating the weights of any on-chip network should be locally available to the synapse since otherwise this information should be "routed" to the synapse by wires which will take a significant amount of area on chip. The simplest form of learning which satisfies this requirement is Hebbian learning which has been implemented on a variety of neuromorphic chips in forms of unsupervised/semi-supervised learning (Schemmel et al., 2010; Qiao et al., 2015; Qiao and Indiveri, 2016; Frenkel et al., 2019a). However, Hebbian-based algorithms are limited in the tasks they can learn and to the best of our knowledge no large-scale task has been demonstrated using this rule. Since gradient descent-based algorithms, such as Backprop has had lots of success in deep learning, there are increasingly more spike-based error Backprop rules that are being developed as was discussed in section 3.1. These types of learning algorithms have recently been custom designed in the form of spike-based delta rule as back-bone of the Backprop algorithm. For example, single layer implementation of the delta rule has been designed in Payvand and Indiveri (2019) and employed for EMG classification (Donati et al., 2019). Expanding this to multi-layer networks involves non-local weight updates which limits its on-chip implementation. Making the Backprop algorithm local is a topic of on-going research which we have discussed in section 3.1.

TABLE 2 | Summary of neuromorphic platforms and biomedical applications.

Neuromorphic chip	DYNAP-SE	SpiNNaker	Loihi	TrueNorth	ODIN
CMOS technology	180 nm	ARM968, 130 nm	14 nm FinFET	28 nm	28 nm FDSOI
Implementation	Mixed-signal	Digital	Digital ASIC	Digital ASIC	Digital ASIC
Energy per SOP	17 pJ @ 1.8 V	Peak power 1 W per chip	23.6 pJ @ 0.75 V	26 pJ @ 0.775	12.7 pJ@0.55 V
Size	38.5 mm^2	102 mm^2	60 mm^2	0.093 mm^2 (core)	0.086 mm^2
On-chip learning	No	Yes (configurable)	Yes (configurable)	No	Yes (SDSP)
Synaptic bit precision	2	Configurable	1–9	1	3
Applications	EMG, ECG, HFO	EMG and EEG	EMG	EEG and Local Field Potential (LFP)	EMG

4.4.3. Weight Storage

The ideal weight storage for online on-chip learning should have the following properties: (i) non-volatility to keep the state of the learnt weights even when the power shuts down to reduce the time and energy footprints of reloading the weights to the chip. (ii) Linear update which allows the state of the memory to change linearly with the calculated update. (iii) Analog states which allows a full-precision for the weights. Non-volatile memristive devices have been proposed as a great potential for the weight storage and there is a large body of work combining the CMOS technology with that of the memristive devices to get the best of two worlds.

In the next section we provide a thorough review on the state of the art for the emerging memory devices and the efforts to integrate and use them in conjunction with neuromorphic chips.

5. MEMRISTIVE DEVICES AND COMPUTING

The severe power and area constraints under which a neuromorphic processor for edge computing must work opened ways toward the investigation of beyond-CMOS solutions. Despite remaining in the early phase of its technological development, memristive devices have been drawing attention in the last decade thanks to their scalability, low-power operation, compatibility with CMOS chip power supply and CMOS fabrication process, and volatile/non-volatile properties. In section 5.1, we will introduce memristive devices and the properties that are appealing for adaptive extreme edge computing paradigms. In section 5.2, we will explore the role of memristive devices in neuromemristive systems and give examples of possible applications. In section 5.3, we will discuss the current challenges and the future perspectives of memristive technology.

5.1. Conventional and Wearable Memristive Devices

Memristive devices, as the name suggested, are devices which can change and memorize their resistance states. They are usually two-terminal devices, however, can be implemented with various physical mechanisms, resulting in versatile existing forms, e.g., resistive random access memory (RRAM, **Figures 4A,B**) (Ielmini and Wong, 2018), phase change memory (PCM, **Figure 4C**) (Zhang et al., 2019), magnetic random access memory (MRAM, **Figures 4D,E**) (Miron et al., 2011), ferroelectric tunneling junction (FTJ, **Figure 4F**) (Wen et al., 2013), etc. The resistance memory of these devices can mimic the memory effect of the basic components of biological neural system, while the resistance changing can mimic the plasticity of biological synapse. Facilitated with their simplicity of two-terminal configuration and scalability to nanoscale, they are inherently suitable for the hardware implementation of brain-inspired computation materializing an artificial neural network, i.e., neuromorphic computation (Jo et al., 2010; Wang et al., 2016a).

This notation, in recent years, has incited wide investigations on the various memristive devices and on their applications in neural network learning and recognition, or, in short, memristive learning (Ohno et al., 2011; Kuzum et al., 2012; Alibart et al., 2013; Yang et al., 2013; Eryilmaz et al., 2014; Ambrogio et al., 2018). The memristive learning can enable energy efficient and low latency information process within a reduced size of systems abandoning the conventional von-Neumann architecture. Among other benefits, this will also make it possible to process information where they are acquired, i.e., within sensors, and reduce the bandwidth needed for transferring the sensor data to data center, accelerating the coming of the era of Internet-of-Things (IOT). **Table 3** summarizes the key features of the main memristive device technologies for neuromorphic/wearable applications in terms of cell area, electrical characteristics, main advantages and challenges. It is worth noticing that some figures of merit in this context are radically different with respect to standard memory requirements. Indeed, while in the memory scenario higher read currents enable faster reading speed, in neuromorphic applications currents as low as possible are preferred, since the current is a limiting factor for neurons' fan-out. Similarly, SET and RESET times should be as fast as possible in memory applications, while in our applications this requirement can be relaxed thanks to the lower operating frequency of the neurons (20–100 Hz). Moreover, the number of achievable conductance levels has to be increased (Ielmini and Pedretti, 2020). Some non-idealities which are usually detrimental for memory applications, for instance, stochasticity of switching parameters, are even beneficial for the neural networks. It is also

worth noticing that the figures of merits in **Table 3** are the best results extracted from different devices. There are no devices that simultaneously show all these best merits. For instance, if the RRAM and PCM devices are engineered to have multilevel states for multilevel synaptic application, lower endurance would be expected. However, in another aspect, devices with only binary states can also be used with dedicated binarized neural networks and stochastic algorithms.

In addition to the commonly referred non-volatile type of memristive switching, the RRAM device can also show volatile behavior, which usually occurs when active materials, such as silver or copper are used as electrode. The relatively long retention time of the volatile behavior [tens of milliseconds to seconds (Covi et al., 2019)] is then found to be similar to the timescale of short term memory, and naturally was proposed to mimic the short term memory effect of biological synapses (Wang et al., 2017, 2019a). Practical examples where volatile devices can be useful are voice (Zhong et al., 2021) and spatiotemporal (Wang et al., 2021) recognition. In the latter case, thanks to device volatility, the network does not need any training and is naturally configured to detect events which occur in time (Du et al., 2017; Wang et al., 2018c, 2019a; Moon et al., 2019). Moreover, it should be mentioned that volatile devices have also shown potential when used as reservoir in a computing system for temporal information processing and time-series prediction, and solver of second-order non-linear dynamic tasks (Du et al., 2017; Moon et al., 2019).

Although most researches on memristive devices are carried on rigid silicon substrates, the simple structure of memristive devices can also be realized on flexible substrates (Shi et al., 2020), which opens new interesting possibilities for realizing local computation within wearable devices (Shang et al., 2017; Dang et al., 2019).

The conventional floating gate non-volatile memories could also be used for synaptic and neuromorphic application. For instance, Malavena et al. (2019) show that floating gate memories in NOR Flash array can be used for pattern learning via STDP weight update algorithms. Floating gate transistors can also be fabricated in two-terminal configuration, which can behave like a memristive device and be used for various neuromorphic applications (Danial et al., 2019). The mature fabrication process and increasing integration capability of floating gate transistors pose great advantages over emerging non-volatile memories.

5.2. Memristive Devices for Neuromorphic Computing

5.2.1. Memristive Neural Components

As mentioned in section 5.1, the primary function of memristive devices is the usage as synaptic devices to implement the memory and plasticity of biological synapses. However, there are increasing interests for these devices to be utilized to implement nanoscale artificial neurons.

On the neuron side, the memristive device gradual internal state change and its consequently abrupt switching closely mimic the integrate-and-fire behavior of biological neurons (Mehonic and Kenyon, 2016; Tuma et al., 2016; Suresh et al., 2019, **Figures 5A–C**). Due to the sample structure and nanometer level scalability, memristive neurons can be much more compact than current CMOS neurons which might consist of current sensor, ADC, Digital to Analog Converter (DAC), and capacitors, all of which are expensive to implement in current CMOS technology in terms of area and/or power consumption (Kwon et al., 2018). The implementation of memristive neurons will also enable full memristive neuromorphic computing (Wang et al., 2018c), which promises further increases in the integration of the hardware neuromorphic computing.

On the synaptic side, the key feature of the biological synapses is their plasticity, i.e., tunable weight, which can be generally implemented by resistance or conductance modification in the memristive devices (**Figure 5D**). Fundamental learning rules based on STDP have already been widely explored (Kuzum et al., 2012; Wang et al., 2015; Covi et al., 2016, 2018; Mulaosmanovic et al., 2017). Spatial spiking pattern recognition (Pedretti et al., 2017), spiking co-incidence detection (Sebastian et al., 2017; Prezioso et al., 2018), and spatial-temporal correlation (Wang et al., 2018b, 2019b) has been reported recently. Synaptic metaplasticity, such as paired-pulse facilitation, can also be achieved via various device operation mechanism (Wang et al., 2017; Zhu et al., 2017b; Wu et al., 2018).

5.2.2. Memristive Neural Networks

There are generally two approaches for a hardware neuromorphic system utilizing memristive devices as synapses: (i) deep learning accelerator, accelerating the artificial neural network computing with multiple layer and error back-propagation, as well as it's variations, like convolutional neural network, recurrent neural network, etc.; (ii) brain-like computing, attempting to closely mimicking the behaviors of biological neural system, like spike representation (**Figure 5D**) and collective decision making behavior. In the deep learning accelerator approach, on-line training places more requirements for the memristive synapses. For instance, linear and symmetrical weight update is crucial for the on-line training (Burr et al., 2015; Ambrogio et al., 2018), while off-line training ignores it since the synaptic weight can be programmed to the memristive device with fine tuning and iterative verify (Yao et al., 2020). In deep learning, therefore, the minimization of device variability becomes of utmost importance to enable online training, as already proposed in some works (Shafiee et al., 2016; Cheng et al., 2017; Song et al., 2017; Imani et al., 2019; Ankit et al., 2020).

Collective decision making is an important feature of the brain computing, which requires high parallelism and, consequently, low current devices. For instance, this feature is the essential for Hopfield neural network (Hopfield, 1982), cellular neural network (Duan et al., 2015), and coupled oscillators (Romera et al., 2018). In the Hopfield neural network, the system automatically evolves to its energy minimization points leading the functionality of associative memory. The use of Hopfield like recurrent neural networks (RNNs) with memristive devices has already been successfully demonstrated in a variety of tasks (Milo et al., 2017; Wang et al., 2020b). As an example of memristive based coupled oscillator network, Ignatov et al. (2017) used a network of self-sustained van der Pol oscillators coupled

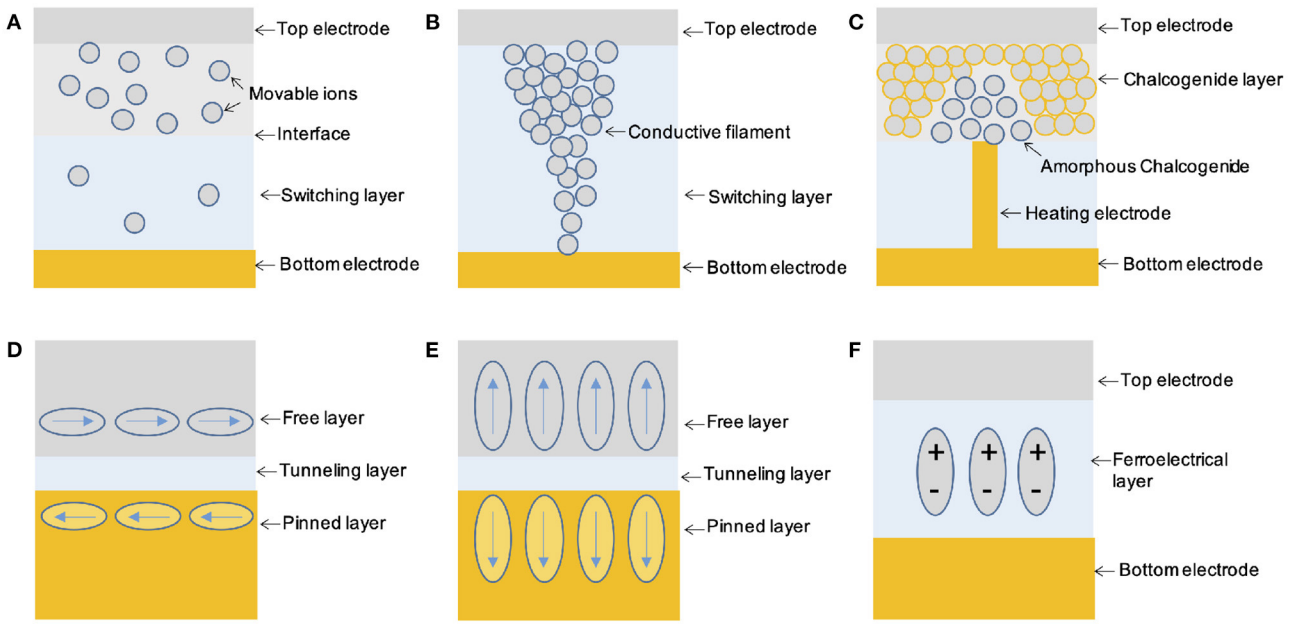


FIGURE 4 | Memristive devices for neuromorphic computing. **(A)** Interface type RRAM device; **(B)** Filamentary RRAM device; **(C)** Phase change memory device; **(D)** MRAM device with in-plane spin polarization; **(E)** MRAM device with perpendicular spin polarization; **(F)** FTJ device.

TABLE 3 | Key features of non-volatile memristive devices.

	RRAM	PCM	MRAM	FTJ
Cell area [min. feature size]	$4F^2$ (IRDS, 2020)	$4F^2$ (IRDS, 2020)	$9F^2$ (Rho et al., 2017)	$4F^2$ (IRDS, 2020)
Retention	>10 years (Goux et al., 2014)	>10 years (Cheng et al., 2012)	>10 years (Golonzka et al., 2018)	>10 years (Udayakumar et al., 2013)
Endurance	10^{12} (Kim et al., 2011; Lee et al., 2011)	10^{11} (Kim et al., 2010)	10^{12} (Saida et al., 2017)	$> 10^{15}$ (Udayakumar et al., 2013)
SET/RESET time	100 ps (Torrezan et al., 2011) 85 ps (Choi et al., 2016)	>100 ns, 10 ns (IRDS, 2020)	20 ns (Jan et al., 2018) 3 ns (Kitagawa et al., 2012)	30 ns, 30 ns (Francois et al., 2019)
Read current	100 pA (Luo et al., 2016)	25 μA (De Sandre et al., 2010)	20 μA (Kitagawa et al., 2012)	0.8 nA (Bruno et al., 2016), device diameter 300 nm)
Write energy per bit	20 fJ (Kang et al., 2015)	~100 fJ (Xiong et al., 2011)	90 fJ (Kitagawa et al., 2012)	<10 fJ (Francois et al., 2019)
Main features	Scalability, multilevel, speed, low energy	Scalability, multilevel, low voltage	Endurance, low power	Endurance, low power, speed
Challenges	Variability	RESET current, temperature stability, resistance drift	Density, scalability, variability	Scalability

with oxide-based memristive devices to investigate the temporal binding problem, which is a well-known issue in the field of cognitive neuroscience. In this experiment, the network is able to emulate an optical illusion which shows two patterns depending on the influence of attention. This means that the network is able to select relevant information from a pool of inputs, as in the case of a system collecting signals from multiple sensors.

5.2.3. Applications of Memristive Neural Networks

At present, Backprop has already exploited for offline training of moderate size memristive neural networks (Valentian et al., 2019). Backpropagation based on online training schemes has also been implemented in several memristive deep learning accelerators (Li et al., 2018a; Wang et al., 2019d; Yao et al., 2020), showing great success of memristive array on accelerating the deep learning training and adaptive to some device non-ideal characteristics. The readers can refer to more comprehensive review papers for more details (Wang et al., 2020a; Zhang et al., 2020; Berggren et al., 2021). In these works, however, the error backpropagation—a backward vector matrix multiplication, and the gradient descent calculation—a vector-vector out-product, are both conducted in hosting computer. The implementation of these two operations in memristive array will further improve the performance of the deep learning accelerators, while Hebbian-based learning algorithms could potentially bypass these operations.

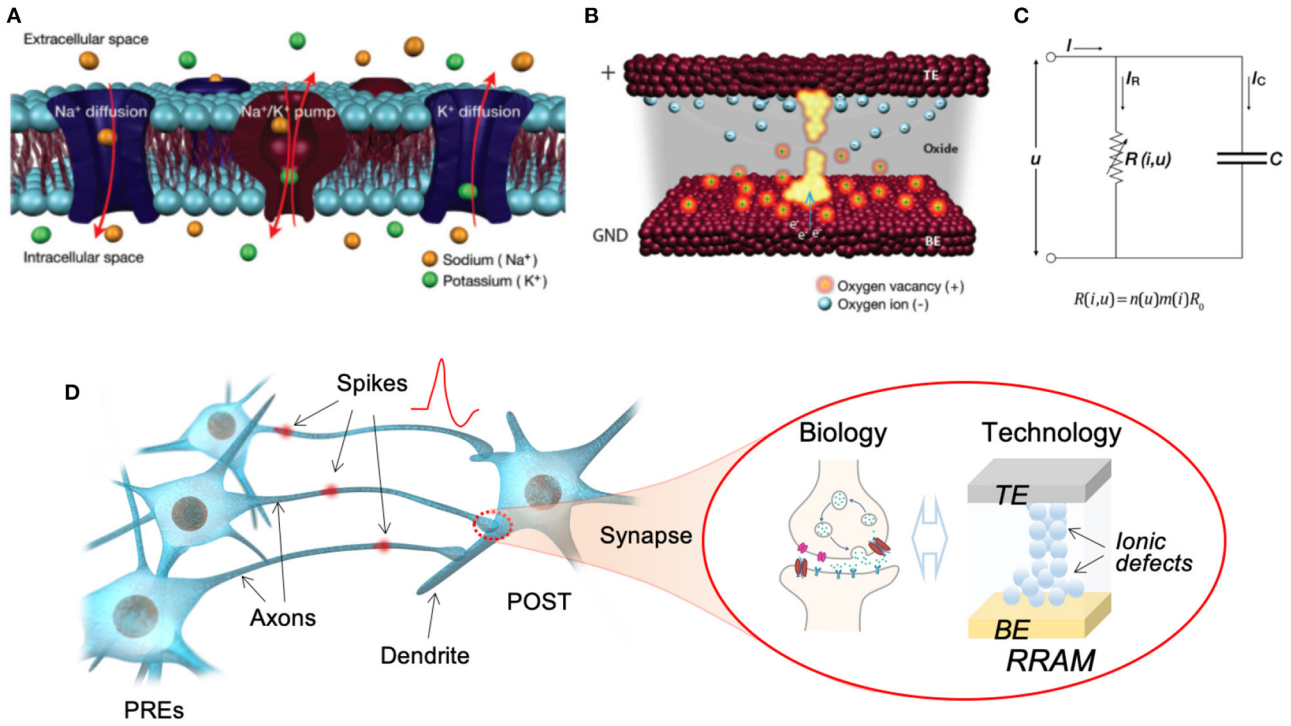


FIGURE 5 | Memristive devices as synapse or neuron for neuromorphic computing. **(A–C)** Memristive device act as threshold device for the firing function of biological neuron (Mehonic and Kenyon, 2016), reproduced under the CC BY license. **(D)** Conceptual illustration of memristive device as artificial synapse for brain-like neuromorphic computing (Wang et al., 2018b), reproduced under the CC BY-NC license.

Online versions of Backprop, as discussed in section 3, are very recent and a memristive-based hardware demonstration is not yet available, despite some work in this direction is being done (Payvand et al., 2020b). To implement adaptation, biologically plausible algorithms able to cope with the non-ideal characteristics of memristive devices are needed. Hebbian-based algorithms are expected to fulfill all these requirements. However, memristive technology with Hebbian-based learning algorithms has been so far mainly used in relatively simple networks. More recently, systems able of solving different tasks, such as speech recognition (Park et al., 2015), and exploring different architectures and learning algorithms are being investigated. In particular, the benefits of exploiting sparsity, mentioned in section 3.2, are demonstrated for feature extraction and image classification in networks trained with stochastic gradient descend and winner-take-all learning algorithms (Sheridan et al., 2016), as well as in hierarchical temporal memory, which does not need training (Krestinskaya and James, 2018).

In the latest years, memristive devices have been used in applications closer to biology, enabling hybrid biological-artificial systems (Serb et al., 2020) and investigating biomedical applications, ranging from speech and emotion recognition (Saleh et al., 2015) to biosignal (Kudithipudi et al., 2016) and medical image (Zhu et al., 2017a) processing. An interesting application is the one of memristive biosensors, which Tzouvadaki et al. (2018) used to implement a system for cancer diagnostic. The innovative use of memristive properties was demonstrated in hardware and opens the way to a broader use of memristive technology where sensors and computing co-exist in the same system or, possibly, in the same device. Finally, a recent work utilizes memristor array for neural signal processing which shows three-orders-of-magnitude improvements in power efficiency compared with literature of CMOS ASIC technology (Liu et al., 2020).

5.3. Open Challenges and Future Work

5.3.1. Device Non-idealities

Implementation of mainstream deep learning algorithms with Backprop learning rule and memristive synapses imposes some requirements for the memristive device, including linear current-voltage relation for reading, analog conductance tuning, linear and symmetric weight update, long retention time, high endurance, etc. (Gokmen and Vlasov, 2016). However, no single device can fulfill all these requirements simultaneously.

Various techniques have been proposed to compensate the device non-idealities. For instance, to compensate the non-linear current-voltage relation for reading, fixed read voltage with variable pulse width or pulse number can be used for synaptic weight reading, and the readout is represented by the charge accumulation in the output nodes (Cai et al., 2019). Linear and symmetric weight update is crucial for accurate online learning of a memristive multilayer neural network with Backprop learning rule (Burr et al., 2015). However, PCM devices usually only show gradual switching in set direction (weight potentiation),

while RRAM devices show gradual switching in reset direction (weight depression). To achieve linear and symmetric weight update, differential pair with two of these devices are usually used. For a differential pair with two PCM devices, the potentiation is achieved by applying set pulses on the positive part and the depression is achieved by applying set pulses on the negative part, thus gradual weight update in both potentiation and depression can be achieved. To further enhance the linearity of weight update, a minor conductance pair consisting of capacitors can be used for frequent but smaller weight update, and finally transferred to the major pair periodically (Ambrogio et al., 2018). Another option to improve device linearity is limiting the device dynamic range in a region far from saturation and where the weight update is linear (Wang et al., 2016b; Woo et al., 2016).

In addition to mitigate the non-idealities of memristive devices, more and more research efforts are made to exploit these non-idealities for brain-like computations. For instance, the stochasticity or noise in reading of memristive device can be used for the probability computation for restricted Boltzmann machine (Mahmoodi et al., 2019), or escape for local minimization points in a Hopfield neural network (Cai et al., 2020). The Ag filament based resistive switching device shows short retention time and high switching dynamics, thus was proposed for reservoir computing (Midya et al., 2019) and spatiotemporal computing (Wang et al., 2019a) to process time-encoded information.

5.3.2. Co-integration of Hybrid CMOS-Memristive Neuromorphic Systems

The main steps to be taken to exploit the full potential of an ASIC for end-to-end processing system go through the integration of memristive devices and sensors with CMOS technology. Indeed, the works presented so far are based either on simulations or on real device data, or on memristive chips interfaced with some standard digital hardware. Despite integration of CMOS technology has been demonstrated for non-volatile resistive switching devices already at a commercial level (Yang-Scharlotta et al., 2014; Hayakawa et al., 2015), the design of co-integrated memristive-based neuromorphic processors is still under development. We envisage a three-phase process to achieve a fully integrated system.

The first one is the co-integration of non-volatile memristive devices with some peripheral circuits (Hirtzlin et al., 2020) and to implement some logic and multiply-and-accumulate (MAC) operations (Chen et al., 2019), which reaches the maturity with the demonstration of a fully co-integrated SNN with analog neurons and memristive synapses (Valentian et al., 2019). The second phase is the co-integration of different technologies. Despite this approach results in higher fabrication costs, it presents several advantages in terms of system performance, which can be more compact and potentially more power efficient. In particular, the co-integration of non-volatile and volatile memristive devices can lead to a fully memristive approach. As an example, Wang et al. (2018c) exploit volatile memristive devices to emulate stochastic neurons and non-volatile memristive devices to store the synaptic weights on the same chip, thus demonstrating the feasibility and the advantages of the dual technology co-integration process. Eventually, the final step which has to be taken in the development of a dedicated ASIC for wearable edge computing is the co-integration of sensors and memristive-based systems. Shulaker et al. (2017) tackled this challenge by designing and fabricating a gas sensing system able of gas classification. The system uses RRAM arrays as memory, Carbon Nanotube Field Effect Transistor (CNFET) for computation and gas sensing, both 3D monolithically integrated on CMOS circuits, which carry out computation and allow memory access.

Finally, there are some further aspects to be considered in order to ensure a successful co-integration. At advanced technological nodes, the power supply of the chip might be lower than the voltages required to operate memristive devices, especially when a forming operation is required. To avoid the use of charge pump circuits, as it is necessary in Flash technology, a possible solution is investigating forming-free devices (Hansen et al., 2018) and low-voltage operation devices with programming voltages <1 V (Gilbert et al., 2013; Guo et al., 2020).

5.3.3. Learning With Memristive Devices

Adaptability is a feature of paramount importance in smart wearable devices, which need to be able to learn the unique feature of their user. This calls for the implementation of lifelong learning paradigms, i.e., the ability of continuously learning new features from experience. Typically, a network has a limited memory capacity dependent on the network size and architecture. Once the maximum number of experiences is recorded, new features learned will erase old ones, thus originating the phenomenon of catastrophic forgetting.

The problem of an efficient implementation of continual learning has been thoroughly investigated (Parisi et al., 2019). In the current scenario, a dichotomy exist between backprop-based ANNs, which have very high accuracy but a limited memory capacity, and brain-inspired SNNs, which feature higher memory capacity thanks to their higher flexibility, but at the cost of lower accuracy. Models used to reduce the effect of forgetting stability-plasticity problem are described in section 3.3. The use of memristive devices in such networks is still an open point. It is possible that memristive device will be beneficial to increase the network capacity (Brivio et al., 2018) at no extra computational cost thanks to their slow approach to the boundaries (Frascaroli et al., 2018), but so far this topic is still quite unexplored. An interesting approach is proposed by Muñoz-Martín et al. (2019), where the key strengths of supervised convolutional ANNs, unsupervised SNNs, and memristive devices are combined in a single system. The results indicate that this approach is robust against catastrophic forgetting, whilst reaching 93% accuracy when tested with both trained and non-trained classes.

6. DISCUSSION AND CONCLUSIONS

In this study, we presented the state-of-the-art core elements that enable the development of wearable devices for healthcare and biomedical applications with extreme edge adaptive computing capability. Various sensors that can collect different bio-signals from the human body are investigated. There is a variety of

sensing specifications in terms of size, resolution, mechanical flexibility and output signals that needs to be considered along with their analog readout circuit at a limited amount of power consumption. However, when the real-time processing of these signals is deployed on edge, severe constraints raise in terms of power efficiency, fast response times, and accuracy in the data classification. The widely-used solution is to find a trade-off between the energy and computational capacity, or send the data to the cloud. However, these strategies are not ideal and slow down the development of wearable smart sensing. Another important aspect to be considered is the matching of the time constants with the intended application. Indeed, electronic systems are intrinsically much faster than real-time events. This property can be exploited to carry out accelerated-time simulations, which are extremely appealing to investigate processes occurring in very long time scales (Schemmel et al., 2020). In systems interacting with the environment, instead, the time constants should be slowed down to match real-time ones in order to optimize energy utilization and enable a seamless processing of biological signals. To meet all the requirements, the development of a platform needs to be optimized in synergy with the other elements and every aspect of the design, from the learning algorithms to the architecture.

Continual learning is required for adaptive wearable devices. In this respect, brain-inspired algorithms promise to be valid alternatives to standard machine learning approaches, such as Backprop and BPTT. The exploitation of sparsity in network connectivity increases the power efficiency by optimizing the use of the available memory. However, the problem of algorithmic robustness to non-ideal hardware (such as noise and variability) and the problems of forgetting and information transfer between tasks still persist and have to be solved in combination with neuromorphic and emerging technologies. SNNs are conceptually ideal for low-power in-memory computing. Their event-based approach, which exploits the low latency of electronics to route the spikes to the correct neuron (Moradi et al., 2017), together with the use of analog subthreshold circuits to reproduce biological timescales, allows fast response times of the network while enabling smooth real-time processing of data. The encoding of the incoming signals into spikes is however still challenging. Moreover, a fully CMOS-based approach has two major technological issues. First, capacitors used to implement biological time constants are massive and may consume up to 60% of the chip area. Memristive technology can be beneficial in this respect, as volatile devices offer a compact alternative to CMOS capacitors. Second, the network configuration and the synaptic weights are usually stored in Ternary Content-Addressable Memory (TCAM)s and in SRAMs, respectively, which hold the state only in the presence of a power supply. This implies that (i) power supply cannot be switched off during normal system operation unless the relevant information is first stored somewhere else and (ii) at every start up of the system, the information on the network has to be uploaded, which may take tens of minutes. Non-volatile memristive device-based versions of TCAM dramatically reduce the initialization times, since the information is already stored in the network. Moreover, memristive-based synapses can also enable normally-off computing paradigms, thus further improving power efficiency.

Besides low-power operation in a small footprint, memristive devices also offer noisy properties, which—if exploited in the right way—might facilitate the implementation of stochastic learning algorithms. However, the technology is still at its infancy and fabrication processes are still under development, yielding high device variability, which makes it difficult to produce reliable multi-bit memory.

The focus of this study is describing the technological challenges and possible solutions to bring computing abilities on the edge. However, there are other practical aspects that may pose a hurdle for the deployment of the envisaged high performance edge biomedical systems (**Figure 1**). (i) *Data set.* The available biomedical data sets may not represent uniformly the human population, since they are mainly collected in countries with a granted basic healthcare system. In this context, online adaptation enables the biomedical device to learn directly from the signal of the user, which should mitigate data set related issues. (ii) *Need for interpretability.* Especially in high-risk scenarios, such as in medicine, where a false positive or negative can have a huge impact on the patient, having transparent Artificial Intelligence (AI) models and systems is of paramount importance to support medical doctors in a decision (Barredo Arrieta et al., 2020). (iii) *Legal responsibility.* Machine Learning (ML) is not unerring. When it does, for example in automotive or healthcare scenarios, our current legal systems lack laws that can clearly define responsibilities (Eshraghian, 2020). (iv) *Generalization performance.* Human intelligence outperforms AI when dealing with generalization tasks, even though some efforts are already devoted to improving this aspect (McKinney et al., 2020). If successful, AI can provide a valid instrument for medical doctors for an early detection of a pathological abnormality (even before the patient displays symptoms), an early start of appropriate therapy, and an overall improvement of prognosis. However, these aspects lie well-beyond the scope of this study and deserve an extensive review on their own.

In summary, the ultimate goal toward smart wearable sensing with edge computing capabilities relies on a bespoke platform embedding sensors, front-end circuit interface, neuromorphic processor and memristive devices. This platform requires high-compatibility of existing sensing technologies with CMOS circuitry and memristive devices to move the intelligent algorithm into the wearable edge without significantly increase the cost in energy. New solutions are needed to enhance the performance of local adaptive learning rules to be competitive with the accuracy of Backprop. Novel encoding techniques to allow seamless communication from sensors to neuromorphic chips have to be developed and flanked by efficient event-based algorithms. So far there is not a uniquely ideal solution, but we envisage that a holistic approach where all the elements of the system are co-designed as a whole is the key to build low-power

end-to-end real-time adaptive systems for next-generation smart wearable devices.

AUTHOR CONTRIBUTIONS

XL and HH: wearable sensors. DK: biologically plausible algorithms. MP and ED: signal processing and neuromorphic computing. EC and WW: memristive devices. EC: led and coordinated the cooperative writing and all discussions. All authors equally contributed to the manuscript, actively participating to the discussions and to the writing.

FUNDING

This work was partially supported by the UK EPSRC under grants EP/R511705/1 and EP/W522168/1. EC and MP acknowledge funding by the European Union's Horizon 2020 research and innovation programme under grant agreement No 871737.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Thomas Mikolajick and Dr. Stefan Slesazeck for useful discussion on ferroelectric and memristive devices.

REFERENCES

Alfaras, M., Soriano, M. C., and Ortín, S. (2019). A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection. *Front. Phys.* 7:103. doi: 10.3389/fphy.2019.00103

Alibart, F., Zamanidoost, E., and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training. *Nat. Commun.* 4:2072. doi: 10.1038/ncomms3072

Alsheikh, M. A., Lin, S., Niyato, D., and Tan, H. P. (2014). Machine learning in wireless sensor networks: algorithms, strategies, and applications. *IEEE Commun. Surv. Tutor.* 16, 1996–2018. doi: 10.1109/COMST.2014.2320099

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R. M., Boybat, I., Di Nolfo, C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558, 60–67. doi: 10.1038/s41586-018-0180-5

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., et al. (2016). "Learning to learn by gradient descent by gradient descent," in *Advances in Neural Information Processing Systems* (Barcelona), 3981–3989.

Ankit, A., El Hajj, I., Chalamalasetti, S. R., Agarwal, S., Marinella, M., Foltin, M., et al. (2020). Panther: a programmable architecture for neural network training harnessing energy-efficient reram. *IEEE Trans. Comput.* 69, 1128–1142. doi: 10.1109/TC.2020.2998456

Artemiadis, P. K., and Kyriakopoulos, K. J. (2011). A switching regime model for the EMG-based control of a robot arm. *IEEE Trans. Syst. Man Cybernet. B Cybernet.* 41, 53–63. doi: 10.1109/TSMCB.2010.2045120

Azghadi, M. R., Lammie, C., Eshraghian, J. K., Payvand, M., Donati, E., Linares-Barranco, B., et al. (2020). Hardware implementation of deep network accelerators towards healthcare and biomedical applications. *IEEE Trans. Biomed. Circuits Syst.* 14, 1138–1159. doi: 10.1109/TBCAS.2020.3036081

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inform. Fusion* 58, 82–115. doi: 10.1016/j.inffus.2019.12.012

Bartolozzi, C., and Indiveri, G. (2007). Synaptic dynamics in analog VLSI. *Neural Comput.* 19, 2581–2603. doi: 10.1162/neco.2007.19.10.2581

Barua, S., Ahmed, M. U., Ahlström, C., and Begum, S. (2019). Automatic driver sleepiness detection using EEG, EOG and contextual information. *Expert Syst. Appl.* 115, 121–135. doi: 10.1016/j.eswa.2018.07.054

Bauer, F. C., Muir, D. R., and Indiveri, G. (2019). Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor. *IEEE Trans. Biomed. Circuits Syst.* 13, 1575–1582. doi: 10.1109/TBCAS.2019.2953001

Behrenbeck, J., Tayeb, Z., Bhiri, C., Richter, C., Rhodes, O., Kasabov, N., et al. (2019). Classification and regression of spatio-temporal signals using NeuCube and its realization on SpiNNaker neuromorphic hardware. *J. Neural Eng.* 16:026014. doi: 10.1088/1741-2552/aafabc

Bellec, G., Kappel, D., Maass, W., and Legenstein, R. (2017). Deep rewiring: training very sparse deep networks. *arXiv* 1711.05136.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., and Maass, W. (2018). "Long short-term memory and learning-to-learn in networks of spiking neurons," in *Advances in Neural Information Processing Systems*, eds S. Bengio, H. M. Wallach, H. Larochelle, K. L. Grauman, and N. C. Cesa-Bianchi (Montreal, QC: Curran Associates Inc), 787–797.

Bellec, G., Scherr, F., Hajek, E., Salaj, D., Legenstein, R., Maass, W., et al. (2019). Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv* 1901.09049.

Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., et al. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *bioRxiv* 738385. doi: 10.1101/738385

Benalcázar, M. E., Jaramillo, A. G., Zea, A., Páez, A., and Andaluz, V. H. (2017). "Hand gesture recognition using machine learning and the myo armband," in *2017 25th European Signal Processing Conference (EUSIPCO)* (Kos), 1040–1044.

Benatti, S., Casamassima, F., Milosevic, B., Farella, E., Schönle, P., Fateh, S., et al. (2015). A versatile embedded platform for EMG acquisition and gesture recognition. *IEEE Trans. Biomed. Circuits Syst.* 9, 620–630. doi: 10.1109/TBCAS.2015.2476555

Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv* 1308.3432.

Benna, M. K., and Fusi, S. (2016). Computational principles of synaptic memory consolidation. *Nat. Neurosci.* 19, 1697–1706. doi: 10.1038/nn.4401

Berggren, K., Xia, Q., Likharev, K. K., Strukov, D. B., Jiang, H., Mikolajick, T., et al. (2021). Roadmap on emerging hardware and technology for machine learning. *Nanotechnology* 32:012002. doi: 10.1088/1361-6528/aba70f

BioSemi (2020). *Biosemi Systems*. Available online at: https://www.biosemi.com/products.htm

Biswas, D., Everson, L., Liu, M., Panwar, M., Verhoef, B., Patki, S., et al. (2019a). CorNET: deep learning framework for PPG-based heart rate estimation and biometric identification in ambulant environment. *IEEE Trans. Biomed. Circuits Syst.* 13, 282–291. doi: 10.1109/TBCAS.2019.2892297

Biswas, D., Sim?es-Capela, N., Hoof, C. V., and Helleputte, N. V. (2019b). Heart rate estimation from wrist-worn photoplethysmography: a review. *IEEE Sens. J.* 19, 6560–6570. doi: 10.1109/JSEN.2019.2914166

Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S., and Magee, J. C. (2017). Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science* 357, 1033–1036. doi: 10.1126/science.aan3846

Bohnstingl, T., Scherr, F., Pehle, C., Meier, K., and Maass, W. (2019). Neuromorphic hardware learns to learn. *Front. Neurosci.* 13:483. doi: 10.3389/fnins.2019.00483

Brader, J. M., Senn, W., and Fusi, S. (2007). Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Comput.* 19, 2881–2912. doi: 10.1162/neco.2007.19.11.2881

Brivio, S., Conti, D., Nair, M. V., Frascaroli, J., Covi, E., Ricciardi, C., et al. (2018). Extended memory lifetime in spiking neural networks employing memristive synapses with nonlinear conductance dynamics. *Nanotechnology* 30:015102. doi: 10.1088/1361-6528/aae81c

Bruno, F. Y., Boyn, S., Fusil, S., Girod, S., Carr?t?ro, C., Marinova, M., et al. (2016). Millionfold resistance change in ferroelectric tunnel junctions based on nickelate electrodes. *Adv. Electron. Mater.* 2:1500245. doi: 10.1002/aelm.201500245

Brzosko, Z., Schultz, W., and Paulsen, O. (2015). Retroactive modulation of spike timing-dependent plasticity by dopamine. *Elife* 4:e09685. doi: 10.7554/eLife.09685

Burr, G. W., Shelby, R. M., Sidler, S., Di Nolfo, C., Jang, J., Boybat, I., et al. (2015). Experimental demonstration and tolerancing of a large-scale neural network

(165,000 synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electron Dev*. 62, 3498–3507. doi: 10.1109/TED.2015.2439635

Cai, F., Correll, J. M., Lee, S. H., Lim, Y., Bothra, V., Zhang, Z., et al. (2019). A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nat. Electron*. 2, 290–299. doi: 10.1038/s41928-019-0270-x

Cai, F., Kumar, S., Van Vaerenbergh, T., Sheng, X., Liu, R., Li, C., et al. (2020). Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron*. 3, 1–10. doi: 10.1038/s41928-020-0436-6

Caruana, R. (1997). Multitask learning. *Mach. Learn*. 28, 41–75. doi: 10.1023/A:1007379606734

Cavigelli, L., and Benini, L. (2016). Origami: a 803-GOp/s/W convolutional network accelerator. *IEEE Trans. Circuits Syst. Video Technol*. 27, 2461–2475. doi: 10.1109/TCSVT.2016.2592330

Caytak, H., Boyle, A., Adler, A., and Bolic, M. (2019). "Bioimpedance spectroscopy processing and applications," in *Encyclopedia of Biomedical Engineering*, ed R. Narayan (Oxford: Elsevier), 265–279. doi: 10.1016/B978-0-12-801238-3.10884-0

Ceolini, E., Frenkel, C., Shrestha, S. B., Taverni, G., Khacef, L., Payvand, M., et al. (2020). Hand-gesture recognition based on emg and event-based camera sensor fusion: a benchmark in neuromorphic computing. *Front. Neurosci*. 14:637. doi: 10.3389/fnins.2020.00637

Chen, G. K., Kumar, R., Sumbul, H. E., Knag, P. C., and Krishnamurthy, R. K. (2018). A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS. *IEEE J. Solid State Circuits* 54, 992–1002. doi: 10.1109/JSSC.2018.2884901

Chen, W. H., Dou, C., Li, K. X., Lin, W. Y., Li, P. Y., Huang, J. H., et al. (2019). Cmos-integrated memristive non-volatile computing-in-memory for AI edge processors. *Nat. Electron*. 2, 420–428. doi: 10.1038/s41928-019-0288-0

Chen, Y. H., Krishna, T., Emer, J. S., and Sze, V. (2016). Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE J. Solid State Circuits* 52, 127–138. doi: 10.1109/JSSC.2016.2616357

Cheng, H. Y., Wu, J. Y., Cheek, R., Raoux, S., BrightSky, M., Garbin, D., et al. (2012). "A thermally robust phase change memory by engineering the Ge/N concentration in $(Ge, N)_xSb_yTe_z$ phase change material," in *2012 International Electron Devices Meeting* (San Francisco, CA), 31.1.1–31.1.4. doi: 10.1109/IEDM.2012.6479141

Cheng, M., Xia, L., Zhu, Z., Cai, Y., Xie, Y., Wang, Y., et al. (2017). "TIME: a training-in-memory architecture for memristor-based deep neural networks," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 1–6. doi: 10.1145/3061639.3062326

Chicca, E., and Indiveri, G. (2020). A recipe for creating ideal hybrid memristive-cmos neuromorphic processing systems. *Appl. Phys. Lett*. 116:120501. doi: 10.1063/1.5142089

Chiu, H. Y., Shuai, H. H., and Chao, P. C. P. (2020). Reconstructing QRS complex from PPG by transformed attentional neural networks. *IEEE Sens. J*. 20, 12374–12383. doi: 10.1109/JSEN.2020.3000344

Choi, B. J., Torrezan, A. C., Strachan, J. P., Kotula, P. G., Lohn, A. J., Marinella, M. J., et al. (2016). High-speed and low-energy nitride memristors. *Adv. Funct. Mater*. 26, 5290–5296. doi: 10.1002/adfm.201600680

Choi, Y., El-Khamy, M., and Lee, J. (2020). Learning sparse low-precision neural networks with learnable regularization. *IEEE Access* 8, 96963–96974. doi: 10.1109/ACCESS.2020.2996936

Cichon, J., and Gan, W. B. (2015). Branch-specific dendritic Ca^{2+} spikes cause persistent synaptic plasticity. *Nature* 520:180. doi: 10.1038/nature14251

Cipriani, C., Zaccone, F., Micera, S., and Carrozza, M. C. (2008). On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction. *IEEE Trans. Robot*. 24, 170–184. doi: 10.1109/TRO.2007.910708

Collins, M. D., and Kohli, P. (2014). Memory bounded deep convolutional networks. *arXiv* 1412.1442.

Corradi, F., and Indiveri, G. (2015). A neuromorphic event-based neural recording system for smart brain-machine-interfaces. *IEEE Trans. Biomed. Circuits Syst*. 9, 699–709. doi: 10.1109/TBCAS.2015.2479256

Corradi, F., Pande, S., Stuijt, J., Qiao, N., Schaafsma, S., Indiveri, G., et al. (2019). "ECG-based heartbeat classification in neuromorphic hardware," in *2019 International Joint Conference on Neural Networks (IJCNN)* (Budapest: IEEE), 1–8. doi: 10.1109/IJCNN.2019.8852279

Covi, E., Brivio, S., Serb, A., Prodromakis, T., Fanciulli, M., and Spiga, S. (2016). Analog memristive synapse in spiking networks implementing unsupervised learning. *Front. Neurosci*. 10:482. doi: 10.3389/fnins.2016.00482

Covi, E., George, R., Frascaroli, J., Brivio, S., Mayr, C., Mostafa, H., et al. (2018). Spike-driven threshold-based learning with memristive synapses and neuromorphic silicon neurons. *J. Phys. D Appl. Phys*. 51:344003. doi: 10.1088/1361-6463/aad361

Covi, E., Lin, Y. H., Wang, W., Stecconi, T., Milo, V., Bricalli, A., et al. (2019). "A volatile RRAM synapse for neuromorphic computing," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (Genoa), 903–906. doi: 10.1109/ICECS46596.2019.8965044

Czarnecki, W. M., Świrszcz, G., Jaderberg, M., Osindero, S., Vinyals, O., and Kavukcuoglu, K. (2017). Understanding synthetic gradients and decoupled neural interfaces. *arXiv* 1703.00522.

Dai, X., Yin, H., and Jha, N. K. (2019). Nest: a neural network synthesis tool based on a grow-and-prune paradigm. *IEEE Trans. Comput*. 68, 1487–1497. doi: 10.1109/TC.2019.2914438

Dang, B., Wu, Q., Sun, J., Zhao, M., Wang, S., Song, F., et al. (2019). Physically transient memristor synapse based on embedding magnesium nanolayer in oxide for security neuromorphic electronics. *IEEE Electron. Device Lett*. 40, 1265–1268. doi: 10.1109/LED.2019.2921322

Danial, L., Pikhay, E., Herbelin, E., Wainstein, N., Gupta, V., Wald, N., et al. (2019). Two-terminal floating-gate transistors with a low-power memristive operation mode for analogue neuromorphic computing. *Nat. Electron*. 2, 596–605. doi: 10.1038/s41928-019-0331-1

Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

De Sandre, G., Bettini, L., Pirola, A., Marmonier, L., Pasotti, M., Borghi, M., et al. (2010). "A 90 nm 4 Mb embedded phase-change memory with 1.2 V 12 ns read access time and 1 MB/s write throughput," in *2010 IEEE International Solid-State Circuits Conference (ISSCC)* (San Francisco, CA), 268–269. doi: 10.1109/ISSCC.2010.5433911

Dettmers, T., and Zettlemoyer, L. (2019). Sparse networks from scratch: faster training without losing performance. *arXiv* 1907.04840.

Diehl, P. U., and Cook, M. (2014). "Efficient implementation of STDP rules on SpiNNaker neuromorphic hardware," in *2014 International Joint Conference on Neural Networks (IJCNN)* (Beijing: IEEE), 4288–4295. doi: 10.1109/IJCNN.2014.6889876

Donati, E., Payvand, M., Risi, N., Krause, R., Burelo, K., Indiveri, G., et al. (2018). "Processing EMG signals using reservoir computing on an event-based neuromorphic system," in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (Cleveland, OH: IEEE), 1–4. doi: 10.1109/BIOCAS.2018.8584674

Donati, E., Payvand, M., Risi, N., Krause, R. B., and Indiveri, G. (2019). Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network. *IEEE Trans. Biomed. Circuits Syst*. 13, 795–803. doi: 10.1109/TBCAS.2019.2925454

Du, C., Cai, F., Zidan, M. A., Ma, W., Lee, S. H., and Lu, W. D. (2017). Reservoir computing using dynamic memristors for temporal information processing. *Nat. Commun*. 8:2204. doi: 10.1038/s41467-017-02337-y

Duan, L., Xu, D., and Tsang, I. (2012). Learning with augmented features for heterogeneous domain adaptation. *arXiv* 1206.4660.

Duan, S., Hu, X., Dong, Z., Wang, L., and Mazumder, P. (2015). Memristor-based cellular nonlinear/neural network: design, analysis, and applications. *IEEE Trans. Neural Netw. Learn. Syst*. 26, 1202–1213. doi: 10.1109/TNNLS.2014.2334701

Duchowski, A. (2007). *Eye Tracking Methodology–Theory and Practice*. Cham: Springer.

Duvinage, M., Castermans, T., and Dutoit, T. (2011). "Control of a lower limb active prosthesis with eye movement sequences," in *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)* (Paris), 1–7. doi: 10.1109/CCMB.2011.5952116

Eid, M. A., Giakoumidis, N., and Saddik, A. E. (2016). A novel eye-gaze-controlled wheelchair system for navigating unknown environments: case study with a person with ALS. *IEEE Access* 4, 558–573. doi: 10.1109/ACCESS.2016.2520093

Eryilmaz, S. B., Kuzum, D., Jeyasingh, R., Kim, S., BrightSky, M., Lam, C., et al. (2014). Brain-like associative learning using a nanoscale

non-volatile phase change synaptic device array. *Front. Neurosci.* 8:205. doi: 10.3389/fnins.2014.00205

Eshraghian, J. K. (2020). Human ownership of artificial creativity. *Nat. Mach. Intell.* 2, 157–160. doi: 10.1038/s42256-020-0161-x

Francois, T., Grenouillet, L., Coignus, J., Blaise, P., Carabasse, C., Vaxelaire, N., et al. (2019). "Demonstration of BEOL-compatible ferroelectric $Hf_{0.5}Zr_{0.5}O_2$ scaled FeRAM co-integrated with 130 nm CMOS for embedded NVM applications," in *2019 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 15.7.1–15.7.4. doi: 10.1109/IEDM19573.2019.8993485

Frascaroli, J., Brivio, S., Covi, E., and Spiga, S. (2018). Evidence of soft bound behaviour in analogue memristive devices for neuromorphic computing. *Sci. Rep.* 8:7178. doi: 10.1038/s41598-018-25376-x

Frenkel, C., Lefebvre, M., Legat, J. D., and Bol, D. (2019a). A 0.086-mm^2 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS. *IEEE Trans. Biomed. Circuits Syst.* 13, 145–158. doi: 10.1109/TBCAS.2018.2880425

Frenkel, C., Legat, J. D., and Bol, D. (2019b). Morphic: a 65-nm 738k-synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE Trans. Biomed. Circuits Syst.* 13, 999–1010. doi: 10.1109/TBCAS.2019.2928793

Furber, S. B., Galluppi, F., Temple, S., and Plana, L. A. (2014). The spinnaker project. *Proc. IEEE* 102, 652–665. doi: 10.1109/JPROC.2014.2304638

Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., et al. (2013). Overview of the SpiNNaker system architecture. *IEEE Trans. Comput.* 62, 2454–2467. doi: 10.1109/TC.2012.142

Fusi, S., Drew, P. J., and Abbott, L. F. (2005). Cascade models of synaptically stored memories. *Neuron* 45, 599–611. doi: 10.1016/j.neuron.2005.02.001

Gale, T., Elsen, E., and Hooker, S. (2019). The state of sparsity in deep neural networks. *arXiv* 1902.09574.

Gao, W., Emaminejad, S., Nyein, H. Y. Y., Challa, S., Chen, K., Peck, A., et al. (2016). Fully integrated wearable sensor arrays for multiplexed *in situ* perspiration analysis. *Nature* 529, 509–514. doi: 10.1038/natur e16521

Gargiulo, G., Calvo, R. A., Bifulco, P., Cesarelli, M., Jin, C., Mohamed, A., et al. (2010). A new EEG recording system for passive dry electrodes. *Clin. Neurophysiol.* 121, 686–693. doi: 10.1016/j.clinph.2009.12.025

Gilbert, N., Zhang, Y., Dinh, J., Calhoun, B., and Hollmer, S. (2013). "A 0.6 V 8 pJ/write non-volatile CBRAM macro embedded in a body sensor node for ultra low energy applications," in *2013 Symposium on VLSI Circuits* (Kyoto: IEEE), C204–C205.

Gokmen, T., and Vlasov, Y. (2016). Acceleration of deep neural network training with resistive cross-point devices: design considerations. *Front. Neurosci.* 10:333. doi: 10.3389/fnins.2016.00333

Golonzka, O., Alzate, J.-G., Arslan, U., Bohr, M., Bai, P., Brockman, J., et al. (2018). "MRAM as embedded non-volatile memory solution for 22FFL FinFET technology," in *2018 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 18.1.1–18.1.4. doi: 10.1109/IEDM.2018.8614620

Göltz, J., Baumbach, A., Billaudelle, S., Breitwieser, O., Dold, D., Kriener, L., et al. (2019). Fast and deep neuromorphic learning with time-to-first-spike coding. *arXiv* 1912.11443. doi: 10.1145/3381755.3381770

Goux, L., Fantini, A., Redolfi, A., Chen, C. Y., Shi, F. F., Degraeve, R., et al. (2014). "Role of the Ta scavenger electrode in the excellent switching control and reliability of a scalable low-current operated TiN/Ta_2O_5/Ta RRAM device," in *2014 Symposium on VLSI Technology (VLSI-Technology)*: *Digest of Technical Papers* (Honolulu, HI), 1–2. doi: 10.1109/VLSIT.2014.6894401

Gravina, R., Alinia, P., Ghasemzadeh, H., and Fortino, G. (2017). Multi-sensor fusion in body sensor networks: state-of-the-art and research challenges. *Inform. Fusion* 35, 1339–1351. doi: 10.1016/j.inffus.2016.09.005

Guo, X., Wang, Q., Lv, X., Yang, H., Sun, K., Yang, D., et al. (2020). SiO_2/Ta_2O_5 heterojunction ECM memristors: physical nature of their low voltage operation with high stability and uniformity. *Nanoscale* 12, 4320–4327. doi: 10.1039/C9NR09845C

Guo, Y., Yao, A., and Chen, Y. (2016). "Dynamic network surgery for efficient DNNs," in *Advances in Neural Information Processing Systems*, eds D. D. Lee, U. von Luxburg, R. Garnett, M. Sugiyama, and I. Guyon (Barcelona: Curran Associates Inc.), 1379–1387.

Han, S., Pool, J., Tran, J., and Dally, W. (2015). "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, eds C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Cambridge, MA: MIT Press). 1135–1143.

Hansen, M., Zahari, F., Kohlstedt, H., and Ziegler, M. (2018). Unsupervised Hebbian learning experimentally realized with analogue memristive crossbar arrays. *Sci. Rep.* 8:8914. doi: 10.1038/s41598-018-27033-9

Harel, M., and Mannor, S. (2010). Learning from multiple outlooks. *arXiv* 1005.0027.

Hayakawa, Y., Himeno, A., Yasuhara, R., and Boullart, W. (2015). "Highly reliable TaOx ReRAM with centralized filament for 28-nm embedded application," in *VLSI Technology* (Kyoto), T14–T15. doi: 10.1109/VLSIT.2015.7223684

Hayashi-Takagi, A., Yagishita, S., Nakamura, M., and Shirai, F. (2015). Labelling and optical erasure of synaptic memory traces in the motor cortex. *Nature* 525:333. doi: 10.1038/nature15257

He, K., Huertas, M., and Hong, S. Z. (2015). Distinct eligibility traces for LTP and LTD in cortical synapses. *Neuron* 88, 528–538. doi: 10.1016/j.neuron.2015.09.037

He, S., Yang, C., Wang, M., Cheng, L., and Hu, Z. (2017). "Hand gesture recognition using myo armband," in *Proceedings–2017 Chinese Automation Congress, CAC 2017* (Jinan), 4850–4855. doi: 10.1109/CAC.2017.8243637

He, X., Goubran, R. A., and Liu, X. P. (2014). Secondary peak detection of PPG signal for continuous cuffless arterial blood pressure measurement. *IEEE Trans. Instr. Meas.* 63, 1431–1439. doi: 10.1109/TIM.2014.2299524

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. *arXiv* 1906.12340.

Herry, C. L., Frasch, M., Seely, A. J. E., and Wu, H. T. (2017). Heart beat classification from single-lead ECG using the synchrosqueezing transform. *Physiol. Meas.* 38, 171–187. doi: 10.1088/1361-6579/aa5070

Hirtzlin, T., Bocquet, M., Penkovsky, B., Klein, J. O., Nowak, E., Vianello, E., et al. (2020). Digital biologically plausible implementation of binarized neural networks with differential hafnium oxide resistive memory arrays. *Front. Neurosci.* 13:1383. doi: 10.3389/fnins.2019.01383

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hossain, M. S., and Muhammad, G. (2016). Cloud-assisted industrial internet of things (IIoT)–enabled framework for health monitoring. *Comput. Netw.* 101, 192–202. doi: 10.1016/j.comnet.2016.01.009

Hosseinifard, B., Moradi, M. H., and Rostami, R. (2013). Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from EEG signal. *Comput. Methods Programs Biomed.* 109, 339–345. doi: 10.1016/j.cmpb.2012.10.008

Hough, P., Glaister, M., and Pledger, A. (2017). The accuracy of wrist-worn heart rate monitors across a range of exercise intensities. *J. Phys. Activity Res.* 2, 112–116. doi: 10.12691/jpar-2-2-8

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks: training neural networks with weights and activations constrained to+ 1 or −1. *arXiv* 1602.02830.

Huszár, F. (2018). Note on the quadratic penalties in elastic weight consolidation. *Proc. Natl. Acad. Sci. U.S.A.* 115, E2496–E2497. doi: 10.1073/pnas.1717042115

Hwang, S., Jebelli, H., Choi, B., Choi, M., and Lee, S. (2018). Measuring workers emotional state during construction tasks using wearable EEG. *J. Construct. Eng. Manage.* 144:04018050. doi: 10.1061/(ASCE)CO.1943-7862.0001506

Ielmini, D., and Pedretti, G. (2020). Device and circuit architectures for in-memory computing. *Adv. Intell. Syst.* 2:2000040. doi: 10.1002/aisy.202000040

Ielmini, D., and Wong, H. S. P. (2018). In-memory computing with resistive switching devices. *Nat. Electron.* 1, 333–343. doi: 10.1038/s41928-018-0092-2

Ignatov, M., Ziegler, M., Hansen, M., and Kohlstedt, H. (2017). Memristive stochastic plasticity enables mimicking of neural synchrony: memristive circuit emulates an optical illusion. *Sci. Adv.* 3:e1700849. doi: 10.1126/sciadv.1700849

Imani, M., Gupta, S., Kim, Y., and Rosing, T. (2019). "Floatpim: in-memory acceleration of deep neural network training with high precision," in *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)* (Phoenix, AZ: IEEE), 802–815. doi: 10.1145/3307650.3322237

Inhyuk, M., Myungjoon, L., Junuk, C., and Museong, M. (2005). "Wearable EMG-based HCI for electric-powered wheelchair users with motor disabilities," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (Barcelona), 2649–2654.

IRDS (2020). *International Roadmap for Devices and Systemstrademark*. Avaialable online at: https://irds.ieee.org/

Isakadze, N., and Martin, S. S. (2020). How useful is the smartwatch ECG? *Trends Cardiovasc. Med*. 30, 442–448. doi: 10.1016/j.tcm.2019.10.010

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw*. 15, 1063–1070. doi: 10.1109/TNN.2004.832719

Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb. Cortex* 17, 2443–2452. doi: 10.1093/cercor/bhl152

Jan, G., Thomas, L., Le, S., Lee, Y., Liu, H., Zhu, J., et al. (2018). "Demonstration of ultra-low voltage and ultra low power STT-MRAM designed for compatibility with 0x node embedded LLC applications," in *2018 IEEE Symposium on VLSI Technology* (Honolulu, HI), 65–66. doi: 10.1109/VLSIT.2018.8510672

Jebelli, H., Hwang, S., and Lee, S. (2018). EEG signal-processing framework to obtain high-quality brain waves from an off-the-shelf wearable EEG device. *J. Comput. Civil Eng*. 32:04017070. doi: 10.1061/(ASCE)CP.1943-5487.0000719

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett*. 10, 1297–1301. doi: 10.1021/nl904092h

Jo, S. H., Kumar, T., Narayanan, S., and Nazarian, H. (2015). Cross-point resistive RAM based on field-assisted superlinear threshold selector. *IEEE Trans. Electron Devices* 62, 3477–3481. doi: 10.1109/TED.2015.2426717

Kaiser, J., Hoff, M., Konle, A., Vasquez Tieck, J. C., Kappel, D., Reichard, D., et al. (2019). Embodied synaptic plasticity with online reinforcement learning. *Front. Neurorobot*. 13:81. doi: 10.3389/fnbot.2019.00081

Kang, C. F., Kuo, W. C., Bao, W., Ho, C. H., Huang, C. W., Wu, W. W., et al. (2015). Self-formed conductive nanofilaments in (Bi, Mn)Ox for ultralow-power memory devices. *Nano Energy* 13, 283–290. doi: 10.1016/j.nanoen.2015.02.033

Kanoun, O., and Tränkler, H. R. (2004). Sensor technology advances and future trends. *IEEE Trans. Instr. Meas*. 53, 1497–1501. doi: 10.1109/TIM.2004.834613

Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. (2015). Network plasticity as bayesian inference. *PLoS Comput. Biol*. 11:e1004485. doi: 10.1371/journal.pcbi.1004485

Kappel, D., Legenstein, R., Habenschuss, S., Hsieh, M., and Maass, W. (2018). A dynamic connectome supports the emergence of stable computational function of neural circuits through reward-based learning. *eNeuro* 5:ENEURO.0301-17.2018. doi: 10.1523/ENEURO.0301-17.2018

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: a review of the state-of-the-art. *Inform. Fusion* 14, 28–44. doi: 10.1016/j.inffus.2011.08.001

Khezri, M., and Jahed, M. (2007). Real-time intelligent pattern recognition algorithm for surface EMG signals. *Biomed. Eng. Online* 6, 1–12. doi: 10.1186/1475-925X-6-45

Kim, I. S., Cho, S. L., Im, D. H., Cho, E. H., Kim, D. H., Oh, G. H., et al. (2010). "High performance PRAM cell scalable to sub-20nm technology with below $4F^2$ cell size, extendable to DRAM applications," in *2010 Symposium on VLSI Technology* (Honolulu, HI), 203–204. doi: 10.1109/VLSIT.2010.5556228

Kim, Y., Lee, S. R., Lee, D., Lee, C. B., Chang, M., Hur, J. H., et al. (2011). "Bi-layered RRAM with unlimited endurance and extremely uniform switching," in *2011 Symposium on VLSI Technology–Digest of Technical Papers* (Kyoto), 52–53.

Kiranyaz, S., Ince, T., and Gabbouj, M. (2016). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Trans. Biomed. Eng*. 63, 664–675. doi: 10.1109/TBME.2015.2468589

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., and Veness, J. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U.S.A*. 114, 3521–3526. doi: 10.1073/pnas.1611835114

Kitagawa, E., Fujita, S., Nomura, K., Noguchi, H., Abe, K., Ikegami, K., et al. (2012). "Impact of ultra low power and fast write operation of advanced perpendicular MTJ on power reduction for high-performance mobile CPU," in *2012 International Electron Devices Meeting* (San Francisco, CA), 29.4.1–29.4.4. doi: 10.1109/IEDM.2012.6479129

Kolesnikov, A., Zhai, X., and Beyer, L. (2019). "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 1920–1929. doi: 10.1109/CVPR.2019.00202

Krestinskaya, O., and James, A. P. (2018). Feature extraction without learning in an analog spatial pooler memristive-CMOS circuit design of hierarchical temporal memory. *Analog Integr. Circuits Signal Process*. 95, 457–465. doi: 10.1007/s10470-018-1161-1

Kudithipudi, D., Saleh, Q., Merkel, C., Thesing, J., and Wysocki, B. (2016). Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing. *Front. Neurosci*. 9:502. doi: 10.3389/fnins.2015.00502

Kulis, B., Saenko, K., and Darrell, T. (2011). "What you saw is not what you get: domain adaptation using asymmetric kernel transforms," in *CVPR 2011* (Colorado Springs, CO: IEEE), 1785–1792. doi: 10.1109/CVPR.2011.5995702

Kuzum, D., Jeyasingh, R. G. D., Yu, S., and Wong, H. S. P. (2012). Low-energy robust neuromorphic computation using synaptic devices. *IEEE Trans. Electron Devices* 59, 3489–3494. doi: 10.1109/TED.2012.2217146

Kwon, M. W., Baek, M. H., Hwang, S., Park, K., Jang, T., Kim, T., et al. (2018). Integrate-and-fire neuron circuit using positive feedback field effect transistor for low power operation. *J. Appl. Phys*. 124:152107. doi: 10.1063/1.5031929

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Lee, J., Lee, J., Han, D., Lee, J., Park, G., and Yoo, H. J. (2019). "LNPU: a 25.3 TFLOPS/W sparse deep-neural-network learning processor with fine-grained mixed precision of FP8-FP16," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)* (San Francisco, CA), 142–144. doi: 10.1109/ISSCC.2019.8662302

Lee, M. J., Lee, C. B., Lee, D., Lee, S. R., Chang, M., Hur, J. H., et al. (2011). A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures. *Nat. Mater*. 10, 625–630. doi: 10.1038/nmat3070

Lee, N., Ajanthan, T., and Torr, P. H. (2018). Snip: Single-shot network pruning based on connection sensitivity. *arXiv* 1810.02340.

Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., et al. (2018a). Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks. *Nat. Commun*. 9:2385. doi: 10.1038/s41467-018-04484-2

Li, C., Wang, Y., Zhang, J., Cui, X., and Huang, R. (2018b). "A compact and accelerated spike-based neuromorphic VLSI chip for pattern recognition," in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (Cleveland, OH: IEEE), 1–4.

Li, G., Lee, B., and Chung, W. (2015). Smartwatch-based wearable EEG system for driver drowsiness detection. *IEEE Sens. J*. 15, 7169–7180. doi: 10.1109/JSEN.2015.2473679

Li, H., Shrestha, A., Fioranelli, F., Kernec, J. L., Heidari, H., Pepa, M., et al. (2017). "Multisensor data fusion for human activities classification and fall detection," in *2017 IEEE Sensors* (Glasgow), 1–3.

Li, H., Shrestha, A., Heidari, H., Le Kernec, J., and Fioranelli, F. (2018c). A multisensory approach for remote health monitoring of older people. *IEEE J. Electromagn. RF Microwaves Med. Biol*. 2, 102–108. doi: 10.1109/JERM.2018.2827099

Liang, X., Fan, H., Mercer, J., and Heidari, H. (2020). "A delay-based neuromorphic processor for arrhythmias detection," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5. doi: 10.1109/ISCAS45731.2020.9181032

Liang, X., Ghannam, R., and Heidari, H. (2018). Wrist-worn gesture sensing with wearable intelligence. *IEEE Sens. J*. 19, 1082–1090. doi: 10.1109/JSEN.2018.2880194

Liang, X., Li, H., Wang, W., Liu, Y., Ghannam, R., Fioranelli, F., et al. (2019). Fusion of wearable and contactless sensors for intelligent gesture recognition. *Adv. Intell. Syst*. 1:1900088. doi: 10.1002/aisy.201900088

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128 × 128 120 db 15 us latency asynchronous temporal contrast vision sensor. *IEEE J. Solid State Circuits* 43, 566–576. doi: 10.1109/JSSC.2007.914337

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun*. 7:13276. doi: 10.1038/ncomms13276

Lin, C., Chuang, C., Huang, C., Tsai, S., Lu, S., Chen, Y., et al. (2014). Wireless and wearable EEG system for evaluating driver vigilance. *IEEE Trans. Biomed. Circuits Syst*. 8, 165–176. doi: 10.1109/TBCAS.2014.2316224

Lin, C. K., Wild, A., Chinya, G. N., Cao, Y., Davies, M., Lavery, D. M., et al. (2018). Programming spiking neural networks on Intel's Loihi. *Computer* 51, 52–61. doi: 10.1109/MC.2018.157113521

Linares-Barranco, B., and Serrano-Gotarredona, T. (2009). Memristance can explain spike-time-dependent-plasticity in neural synapses. *Nature Precedings*. doi: 10.1038/npre.2009.3010.1

Liu, C., Bellec, G., Vogginger, B., Kappel, D., Partzsch, J., Neumärker, F., et al. (2018). Memory-efficient deep learning on a SpiNNaker 2 prototype. *Front. Neurosci*. 12:840. doi: 10.3389/fnins.2018.00840

Liu, Z., Tang, J., Gao, B., Li, X., Yao, P., Lin, Y., et al. (2020). Multichannel parallel processing of neural signals in memristor arrays. *Sci. Adv*. 6:eabc4797. doi: 10.1126/sciadv.abc4797

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017). "Deep transfer learning with joint adaptation networks," in *34th International Conference on Machine Learning*, Vol. 70 (Sydney, NSW: JMLR.org), 2208–2217.

López, A., Fernández, M., Rodríguez, H., Ferrero, F., and Postolache, O. (2018). Development of an EOG-based system to control a serious game. *Meas. J. Int. Meas. Conf.* 127, 481–488. doi: 10.1016/j.measurement.2018.06.017

Louizos, C., Welling, M., and Kingma, D. P. (2017). Learning sparse neural networks through *l*_0 regularization. *arXiv* 1712.01312.

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. (2015). Transfer learning using computational intelligence: a survey. *Knowl. Based Syst*. 80, 14–23. doi: 10.1016/j.knosys.2015.01.010

Luo, Q., Xu, X., Liu, H., Lv, H., Gong, T., Long, S., et al. (2016). Super non-linear RRAM with ultra-low power for 3D vertical nano-crossbar arrays. *Nanoscale* 8, 15629–15636. doi: 10.1039/C6NR02029A

Luz, E. J. S., Schwartz, W. R., Cámara-Chávez, G., and Menotti, D. (2016). ECG-based heartbeat classification for arrhythmia detection: a survey. *Comput. Methods Programs Biomed*. 127, 144–164. doi: 10.1016/j.cmpb.2015.12.008

Ma, Y., Chen, B., Ren, P., Zheng, N., Indiveri, G., and Donati, E. (2020a). EMG-based gestures classification using a mixed-signal neuromorphic processing system. *IEEE J. Emerg. Select. Top. Circuits Syst*. 10, 578–587. doi: 10.1109/JETCAS.2020.3037951

Ma, Y., Donati, E., Chen, B., Ren, P., Zheng, N., and Indiveri, G. (2020b). "Neuromorphic implementation of a recurrent neural network for EMG classification," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Genova: IEEE), 69–73.

Maass, W. (2014). Noise as a resource for computation and learning in networks of spiking neurons. *Proc. IEEE* 102, 860–880. doi: 10.1109/JPROC.2014.2310593

Mahmoodi, M. R., Prezioso, M., and Strukov, D. B. (2019). Versatile stochastic dot product circuits based on nonvolatile memories for high performance neurocomputing and neurooptimization. *Nat. Commun*. 10:5113. doi: 10.1038/s41467-019-13103-7

Malavena, G., Filippi, M., Spinelli, A. S., and Monzio Compagnoni, C. (2019). Unsupervised learning by spike-timing-dependent plasticity in a mainstream NOR flash memory array—part II: array learning. *IEEE Trans. Electron Devices* 66, 4733–4738. doi: 10.1109/TED.2019.2940599

Martin, W. B., Johnson, L. C., Viglione, S. S., Naitoh, P., Joseph, R. D., and Moses, J. D. (1972). Pattern recognition of EEG-EOG as a technique for all-night sleep stage scoring. *Electroencephalogr. Clin. Neurophysiol*. 32, 417–427. doi: 10.1016/0013-4694(72)90009-0

Matthie, J. R. (2008). Bioimpedance measurements of human body composition: critical analysis and outlook. *Expert Rev. Med. Devices* 5, 239–261. doi: 10.1586/17434440.5.2.239

Mayr, C., Hoeppner, S., and Furber, S. (2019). Spinnaker 2: a 10 million core processor system for brain simulation and machine learning. *arXiv* 1911.02385.

McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., et al. (2020). International evaluation of an AI system for breast cancer screening. *Nature* 577, 89–94. doi: 10.1038/s41586-019-1799-6

Mead, C. (2020). How we created neuromorphic engineering. *Nat. Electron*. 3, 434–435. doi: 10.1038/s41928-020-0448-2

Mehonic, A., and Kenyon, A. J. (2016). Emulating the electrical activity of the neuron using a silicon oxide RRAM cell. *Front. Neurosci*. 10:57. doi: 10.3389/fnins.2016.00057

Mendez, I., Hansen, B. W., Grabow, C. M., Smedegaard, E. J. L., Skogberg, N. B., Uth, X. J., et al. (2017). "Evaluation of the myo armband for the classification of hand motions," in *2017 International Conference on Rehabilitation Robotics (ICORR)* (Moradabad), 1211–1214. doi: 10.1109/ICORR.2017.8009414

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Midya, R., Wang, Z., Asapu, S., Zhang, X., Rao, M., Song, W., et al. (2019). Reservoir computing using diffusive memristors. *Adv. Intell. Syst*. 1:1900084. doi: 10.1002/aisy.201900084

Milo, V., Ielmini, D., and Chicca, E. (2017). "Attractor networks and associative memories with STDP learning in RRAM synapses," in *2017 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 11.2.1–11.2.4. doi: 10.1109/IEDM.2017.8268369

Miron, I. M., Garello, K., Gaudin, G., Zermatten, P. J., Costache, M. V., Auffret, S., et al. (2011). Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection. *Nature* 476, 189–193. doi: 10.1038/nature10309

Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., and Liotta, A. (2018). Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nat. Commun*. 9:2383. doi: 10.1038/s41467-018-04316-3

Molchanov, D., Ashukha, A., and Vetrov, D. (2017). "Variational dropout sparsifies deep neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 (Sydney, NSW: JMLR.org), 2498–2507.

Montagna, F., Rahimi, A., Benatti, S., Rossi, D., and Benini, L. (2018). "PULP-HD: accelerating brain-inspired high-dimensional computing on a parallel ultra-low power platform," in *Proceedings of the ACM/ESDA/IEEE Design Automation Conference (DAC)* (San Francisco, CA), 1–6. doi: 10.1145/3195970.3196096

Moon, J., Ma, W., Shin, J. H., Cai, F., Du, C., Lee, S. H., et al. (2019). Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron*. 2, 480–487. doi: 10.1038/s41928-019-0313-3

Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G. (2017). A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE Trans. Biomed. Circuits Syst*. 12, 106–122. doi: 10.1109/TBCAS.2017.2759700

Mostafa, H., and Wang, X. (2019). Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. *arXiv* 1902.05967.

Mulaosmanovic, H., Ocker, J., Müller, S., Noack, M., Müller, J., Polakowski, P., et al. (2017). "Novel ferroelectric FET based synapse for neuromorphic systems," in *2017 Symposium on VLSI Technology* (Kyoto: IEEE), T176–T177. doi: 10.23919/VLSIT.2017.7998165

Müller, L. K., and Indiveri, G. (2015). Rounding methods for neural networks with low resolution synaptic weights. *arXiv* 1504.05767.

Müller, L. K., Nair, M. V., and Indiveri, G. (2017). "Randomized unregulated step descent for limited precision synaptic elements," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (Baltimore, MD: IEEE), 1–4. doi: 10.1109/ISCAS.2017.8050217

Muñoz-Martín, I., Bianchi, S., Pedretti, G., Melnic, O., Ambrogio, S., and Ielmini, D. (2019). Unsupervised learning to overcome catastrophic forgetting in neural networks. *IEEE J. Explor. Solid State Comput. Devices Circuits* 5, 58–66. doi: 10.1109/JXCDC.2019.2911135

Neckar, A., Fok, S., Benjamin, B. V., Stewart, T. C., Oza, N. N., Voelker, A. R., et al. (2018). Braindrop: a mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proc. IEEE* 107, 144–164. doi: 10.1109/JPROC.2018.2881432

Neftci, E. O., Augustine, C., Paul, S., and Detorakis, G. (2017). Event-driven random back-propagation: enabling neuromorphic deep learning machines. *Front. Neurosci*. 11:324. doi: 10.3389/fnins.2017.00324

Neftci, E. O., Pedroni, B. U., Joshi, S., Al-Shedivat, M., and Cauwenberghs, G. (2015). Unsupervised learning in synaptic sampling machines. *arXiv* 1511.04484.

Nweke, H. F., Teh, Y. W., Al-garadi, M. A., and Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. *Expert Syst. Appl*. 105, 233–261. doi: 10.1016/j.eswa.2018.03.056

Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J. K., and Aono, M. (2011). Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater*. 10, 591–595. doi: 10.1038/nmat3054

Ortín, S., Soriano, M. C., Alfaras, M., and Mirasso, C. R. (2019). Automated real-time method for ventricular heartbeat classification. *Comput. Methods Programs Biomed*. 169, 1–8. doi: 10.1016/j.cmpb.2018.11.005

Pan, S. J., and Yang, Q. (2009). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 1345–1359. doi: 10.1109/TKDE.2009.191

Pantelopoulos, A., and Bourbakis, N. G. (2010). A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Trans. Syst. Man Cybernet. C Appl. Rev*. 40, 1–12. doi: 10.1109/TSMCC.2009.2032660

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Netw*. 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Park, S., Chu, M., Kim, J., Noh, J., Jeon, M., Hun Lee, B., et al. (2015). Electronic system with memristive synapses for pattern recognition. *Sci. Rep*. 5:10123. doi: 10.1038/srep10123

Patel, S., McGinnis, R. S., Silva, I., DiCristofaro, S., Mahadevan, N., Jortberg, E., et al. (2016). "A wearable computing platform for developing cloud-based machine learning models for health monitoring applications," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Orlando, FL), 5997–6001. doi: 10.1109/EMBC.2016.7592095

Payvand, M., Fouda, M. E., Kurdahi, F., Eltawil, A., and Neftci, E. O. (2020a). "Error-triggered three-factor learning dynamics for crossbar arrays," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Genova: IEEE), 218–222. doi: 10.1109/AICAS48895.2020.9073998

Payvand, M., Fouda, M. E., Kurdahi, F., Eltawil, A. M., and Neftci, E. O. (2020b). On-chip error-triggered learning of multi-layer memristive spiking neural networks. *IEEE J. Emerg. Select. Top. Circuits Syst*. 10, 522–535. doi: 10.1109/JETCAS.2020.3040248

Payvand, M., and Indiveri, G. (2019). "Spike-based plasticity circuits for always-on on-line learning in neuromorphic systems," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)* (Sapporo), 1–5. doi: 10.1109/ISCAS.2019.8702497

Pecevski, D., and Maass, W. (2016). Learning probabilistic inference through spike-timing-dependent plasticity. *eNeuro* 3:ENEURO.0048-15.2016. doi: 10.1523/ENEURO.0048-15.2016

Pedretti, G., Milo, V., Ambrogio, S., Carboni, R., Bianchi, S., Calderoni, A., et al. (2017). Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. *Sci. Rep*. 7:5288. doi: 10.1038/s41598-017-05480-0

Pi nero, P., Garcia, P., Arco, L., Álvarez, A., Garcíá, M. M., and Bonal, R. (2004). Sleep stage classification using fuzzy sets and machine learning techniques. *Neurocomputing* 58–60, 1137–1143. doi: 10.1016/j.neucom.2004.01.178

Prettenhofer, P., and Stein, B. (2010). "Cross-language text classification using structural correspondence learning," in *48th Annual Meeting of the Association for Computational Linguistics* (Uppsala), 1118–1127.

Prezioso, M., Mahmoodi, M. R., Bayat, F. M., Nili, H., Kim, H., Vincent, A., et al. (2018). Spike-timing-dependent plasticity learning of coincidence detection with passively integrated memristive circuits. *Nat. Commun*. 9:5311. doi: 10.1038/s41467-018-07757-y

Punsawad, Y., Wongsawat, Y., and Parnichkun, M. (2010). "Hybrid EEG-EOG brain-computer interface system for practical machine control," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (Buenos Aires), 1360–1363. doi: 10.1109/IEMBS.2010.5626745

Qiao, N., and Indiveri, G. (2016). "Scaling mixed-signal neuromorphic processors to 28 nm FD-SOI technologies," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)* (Shanghai: IEEE), 552–555. doi: 10.1109/BioCAS.2016.7833854

Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., et al. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Front. Neurosci*. 9:141. doi: 10.3389/fnins.2015.00141

Rahhal, M. M. A., Bazi, Y., Alhichri, H., Alajlan, N., Melgani, F., and Yager, R. R. (2016). Deep learning approach for active classification of electrocardiogram signals. *Inform. Sci*. 345, 340–354. doi: 10.1016/j.ins.2016.01.082

Raj, S., Ray, K. C., and Shankar, O. (2016). Cardiac arrhythmia beat classification using DOST and PSO tuned SVM. *Comput. Methods Programs Biomed*. 136, 163–177. doi: 10.1016/j.cmpb.2016.08.016

Rawat, S., Vats, S., and Kumar, P. (2016). "Evaluating and exploring the myo armband," in *2016 International Conference System Modeling & Advancement in Research Trends (SMART)* (Moradabad), 115–120. doi: 10.1109/SYSMART.2016.7894501

Reşit Kavsaoğlu, A., Polat, K., and Recep Bozkurt, M. (2014). A novel feature ranking algorithm for biometric recognition with PPG signals. *Comput. Biol. Med*. 49, 1–14. doi: 10.1016/j.compbiomed.2014.03.005

Rho, K., Tsuchida, K., Kim, D., Shirai, Y., Bae, J., Inaba, T., et al. (2017). "23.5 A 4Gb LPDDR2 STT-MRAM with compact 9F^2 1T1MTJ cell and hierarchical bitline architecture," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)* (San Francisco, CA), 396–397. doi: 10.1109/ISSCC.2017.7870428

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., et al. (2019). A deep learning framework for neuroscience. *Nat. Neurosci*. 22, 1761–1770. doi: 10.1038/s41593-019-0520-2

Rincon, A. L., Yamasaki, H., and Shimoda, S. (2016). "Design of a video game for rehabilitation using motion capture, EMG analysis and virtual reality," in *2016 International Conference on Electronics, Communications and Computers (CONIELECOMP)* (Cholula), 198–204. doi: 10.1109/CONIELECOMP.2016.7438575

Rissanen, S. M., Kankaanpää, M., Meigal, A., Tarvainen, M. P., Nuutinen, J., Tarkka, I. M., et al. (2008). Surface EMG and acceleration signals in Parkinson's disease: feature extraction and cluster analysis. *Med. Biol. Eng. Comput*. 46, 849–858. doi: 10.1007/s11517-008-0369-0

Romera, M., Talatchian, P., Tsunegi, S., Abreu Araujo, F., Cros, V., Bortolotti, P., et al. (2018). Vowel recognition with four coupled spin-torque nano-oscillators. *Nature* 563, 230–234. doi: 10.1038/s41586-018-0632-y

Rubino, A., Livanelioglu, C., Qiao, N., Payvand, M., and Indiveri, G. (2021). Ultra-low-power FDSOI neural circuits for extreme-edge neuromorphic intelligence. *IEEE Trans. Circuits Syst. I Reg. Pap*. 68, 45–56. doi: 10.1109/TCSI.2020.3035575

Rumelhart, D., Hinton, G., and Williams, R. (1986). "Learning internal representations by error propagation," in *Parallel Distributed Processing*, Vol. 1 (Cambridge, MA: MIT Press), 318–362. doi: 10.21236/ADA164453

Rundo, F., Conoci, S., Ortis, A., and Battiato, S. (2018). An advanced bio-inspired PhotoPlethysmoGraphy (PPG) and ECG pattern recognition system for medical assessment. *Sensors* 18:405. doi: 10.3390/s18020405

Sacramento, J., Costa, R. P., Bengio, Y., and Senn, W. (2017). Dendritic error backpropagation in deep cortical microcircuits. *arXiv* 1801.00062.

Saida, D., Kashiwada, S., Yakabe, M., Daibou, T., Fukumoto, M., Miwa, S., et al. (2017). 1×- to 2×-nm perpendicular MTJ switching at Sub-3-ns pulses below 100μA for high-performance embedded STT-MRAM for Sub-20-nm CMOS. *IEEE Trans. Electron Devices* 64, 427–431. doi: 10.1109/TED.2016.2636326

Saleh, Q., Merkel, C., Kudithipudi, D., and Wysocki, B. (2015). "Memristive computational architecture of an echo state network for real-time speech-emotion recognition," in *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)* (Verona, NY), 1–5. doi: 10.1109/CISDA.2015.7208624

Samadi, A., Lillicrap, T. P., and Tweed, D. B. (2017). Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural Comput*. 29, 578–602. doi: 10.1162/NECO_a_00929

Schemmel, J., Billaudelle, S., Dauer, P., and Weis, J. (2020). Accelerated analog neuromorphic computing. *arXiv* 2003.11996.

Schemmel, J., Bruderle, D., Grubl, A., Hock, M., Meier, K., and Millner, S. (2010). "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (Paris), 1947–1950. doi: 10.1109/ISCAS.2010.5536970

Schmidhuber, J. (1992). Learning to control fast-weight memories: an alternative to dynamic recurrent networks. *Neural Comput*. 4, 131–139. doi: 10.1162/neco.1992.4.1.131

Schmidhuber, J. (1993). "A neural network that embeds its own meta-levels," in *IEEE International Conference on Neural Networks* (San Francisco, CA: IEEE), 407–412.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Netw*. 61, 85–117. doi: 10.1016/j.neunet.2014.09.003

Sebastian, A., Tuma, T., Papandreou, N., Le Gallo, M., Kull, L., Parnell, T., et al. (2017). Temporal correlation detection using computational phase-change memory. *Nat. Commun*. 8:1115. doi: 10.1038/s41467-017-01481-9

Serb, A., Corna, A., George, R., Khiat, A., Rocchi, F., Reato, M., et al. (2020). Memristive synapses connect brain and silicon spiking neurons. *Sci. Rep*. 10:2590. doi: 10.1038/s41598-020-58831-9

Shafiee, A., Nag, A., Muralimanohar, N., Balasubramonian, R., Strachan, J. P., Hu, M., et al. (2016). ISAAC: a convolutional neural network accelerator with *in-situ* analog arithmetic in crossbars. *ACM SIGARCH Comput. Archit. News* 44, 14–26. doi: 10.1145/3007787.3001139

Shang, J., Xue, W., Ji, Z., Liu, G., Niu, X., Yi, X., et al. (2017). Highly flexible resistive switching memory based on amorphous-nanocrystalline hafnium oxide films. *Nanoscale* 9, 7037–7046. doi: 10.1039/C6NR08687J

Sharifshazileh, M., Burelo, K., Fedele, T., Sarnthein, J., and Indiveri, G. (2019). "A neuromorphic device for detecting high-frequency oscillations in human iEEG," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (Genoa: IEEE), 69–72. doi: 10.1109/ICECS46596.2019.8965192

Shen, K. Q., Li, X. P., Ong, C. J., Shao, S. Y., and Wilder-Smith, E. P. V. (2008). EEG-based mental fatigue measurement using multi-class support vector machines with confidence estimate. *Clin. Neurophysiol.* 119, 1524–1533. doi: 10.1016/j.clinph.2008.03.012

Sheridan, P. M., Du, C., and Lu, W. D. (2016). Feature extraction using memristor networks. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 2327–2336. doi: 10.1109/TNNLS.2015.2482220

Shi, Q., Wang, J., Aziz, I., and Lee, P. S. (2020). Stretchable and wearable resistive switching random access memory. *Adv. Intell. Syst.* 2:2000007. doi: 10.1002/aisy.202000007

Shulaker, M. M., Hills, G., Park, R. S., Howe, R. T., Saraswat, K., Wong, H. S. P., et al. (2017). Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. *Nature* 547, 74–78. doi: 10.1038/nature22994

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961

Song, J., Cho, Y., Park, J. S., Jang, J. W., Lee, S., Song, J. H., et al. (2019). "An 11.5TOPS/W 1024-MAC butterfly structure dual-core sparsity-aware neural processing unit in 8 nm flagship mobile SoC," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)* (San Francisco, CA), 130–132. doi: 10.1109/ISSCC.2019.8662476

Song, L., Qian, X., Li, H., and Chen, Y. (2017). "Pipelayer: a pipelined ReRAM-based accelerator for deep learning," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)* (Austin, TX: IEEE), 541–552. doi: 10.1109/HPCA.2017.55

Stevens, J. R., Bigelow, L., Denney, D., Lipkin, J., Livermore, Arthur H., J., et al. (1979). Telemetered EEG-EOG during psychotic behaviors of schizophrenia. *Archiv. Gen. Psychiatry* 36, 251–262. doi: 10.1001/archpsyc.1979.01780030017001

Ström, N. (1997). "Sparse connection and pruning in large dynamic artificial neural networks," in *5th European Conference on Speech Communication and Technology* (Rhodes), 2807–2810.

Subasi, A. (2013). Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. *Comput. Biol. Med.* 43, 576–586. doi: 10.1016/j.compbiomed.2013.01.020

Sun, T. P., Ching, C. T. S., Cheng, C. S., Huang, S. H., Chen, Y. J., Hsiao, C. S., et al. (2010). The use of bioimpedance in the detection/screening of tongue cancer. *Cancer Epidemiol.* 34, 207–211. doi: 10.1016/j.canep.2009.12.017

Sun, X., Choi, J., Chen, C. Y., Wang, N., Venkataramani, S., Srinivasan, V., et al. (2019). "Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 32.

Suresh, B., Bertele, M., Breyer, E. T., Klein, P., Mulaosmanovic, H., Mikolajick, T., et al. (2019). "Simulation of integrate-and-fire neuron circuits using HfO_2-based ferroelectric field effect transistors," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (Genoa), 229–232. doi: 10.1109/ICECS46596.2019.8965004

Thakor, N. V. (2015). "Biopotentials and electrophysiology measurements," in *Telehealth and Mobile Health* (CRC Press), 595–614. doi: 10.1201/b19147-36

Torrey, L., and Shavlik, J. (2010). "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques* (IGI Global), 242–264. doi: 10.4018/978-1-60566-766-9.ch011

Torrezan, A. C., Strachan, J. P., Medeiros-Ribeiro, G., and Williams, R. S. (2011). Sub-nanosecond switching of a tantalum oxide memristor. *Nanotechnology* 22:485203. doi: 10.1088/0957-4484/22/48/485203

Tuma, T., Pantazi, A., Le Gallo, M., Sebastian, A., and Eleftheriou, E. (2016). Stochastic phase-change neurons. *Nat. Nanotechnol.* 11, 693–699. doi: 10.1038/nnano.2016.70

Tzouvadaki, I., Tuoheti, A., De Micheli, G., Demarchi, D., and Carrara, S. (2018). "Portable memristive biosensing system as effective point-of-care device for cancer diagnostics," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (Florence), 1–5. doi: 10.1109/ISCAS.2018.8351438

Udayakumar, K. R., San, T., Rodriguez, J., Chevacharoenkul, S., Frystak, D., Rodriguez-Latorre, J., et al. (2013). "Low-power ferroelectric random access memory embedded in 180 nm analog friendly CMOS technology," in *2013 5th IEEE International Memory Workshop* (Monterey, CA), 128–131. doi: 10.1109/IMW.2013.6582115

Ullrich, K., Meeds, E., and Welling, M. (2017). Soft weight-sharing for neural network compression. *arXiv* 1702.04008.

Valentian, A., Rummens, F., Vianello, E., Mesquida, T., Lecat-Mathieu de Boissac, C., Bichler, O., et al. (2019). "Fully integrated spiking neural network with analog neurons and RRAM synapses," in *2019 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 14.13.1–14.13.4. doi: 10.1109/IEDM19573.2019.8993431

Wang, C., and Mahadevan, S. (2011). "Heterogeneous domain adaptation using manifold alignment," in *Twenty-Second International Joint Conference on Artificial Intelligence* (Pasadena), 1541–1546.

Wang, H., Li, Y., Long, J., Yu, T., and Gu, Z. (2014a). An asynchronous wheelchair control by hybrid EEG–EOG brain–computer interface. *Cogn. Neurodyn.* 8, 399–409. doi: 10.1007/s11571-014-9296-y

Wang, N., Choi, J., Brand, D., Chen, C. Y., and Gopalakrishnan, K. (2018a). Training deep neural networks with 8-bit floating point numbers. *arXiv* 1812.08011.

Wang, Q., Zhang, X., Chen, X., Chen, R., Chen, W., and Chen, Y. (2010). "A novel pedestrian dead reckoning algorithm using wearable EMG sensors to measure walking strides," in *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service* (Kirkkonummi), 1–8. doi: 10.1109/UPINLBS.2010.5653821

Wang, W., Covi, E., Lin, Y., Ambrosi, E., and Ielmini, D. (2019). "Modeling of switching speed and retention time in volatile resistive switching memory by ionic drift and diffusion," in *2019 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 32.3.1–32.3.4. doi: 10.1109/IEDM19573.2019.8993625

Wang, W., Covi, E., Milozzi, A., Farronato, M., Ricci, S., Sbandati, C., et al. (2021). Neuromorphic motion detection and orientation selectivity by volatile resistive switching memories. *Adv. Intell. Syst.* 2000224. doi: 10.1002/aisy.202000224

Wang, W., Laudato, M., Ambrosi, E., Bricalli, A., Covi, E., Lin, Y. H., et al. (2019a). Volatile resistive switching memory based on Ag ion drift/diffusion—part II: compact modeling. *IEEE Trans. Electron Devices* 66, 3802–3808. doi: 10.1109/TED.2019.2928888

Wang, W., Li, Y., Wang, M., Wang, L., Liu, Q., Banerjee, W., et al. (2016a). "A hardware neural network for handwritten digits recognition using binary RRAM as synaptic weight element," in *2016 IEEE Silicon Nanoelectronics Workshop (SNW)* (Honolulu, HI: IEEE), 50–51. doi: 10.1109/SNW.2016.7577980

Wang, W., Pedretti, G., Milo, V., Carboni, R., Calderoni, A., Ramaswamy, N., et al. (2018b). Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses. *Sci. Adv.* 4:eaat4752. doi: 10.1126/sciadv.aat4752

Wang, W., Pedretti, G., Milo, V., Carboni, R., Calderoni, A., Ramaswamy, N., et al. (2019b). Computing of temporal information in spiking neural networks with ReRAM synapses. *Faraday Discuss.* 213, 453–469. doi: 10.1039/C8FD00097B

Wang, W., Song, W., Yao, P., Li, Y., Van Nostrand, J., Qiu, Q., et al. (2020a). Integration and co-design of memristive devices and algorithms for artificial intelligence. *iScience* 23:101809. doi: 10.1016/j.isci.2020.101809

Wang, W., Wang, M., Ambrosi, E., Bricalli, A., Laudato, M., Sun, Z., et al. (2019c). Surface diffusion-limited lifetime of silver and copper nanofilaments in resistive switching devices. *Nat. Commun.* 10:81. doi: 10.1038/s41467-018-07979-0

Wang, X. W., Nie, D., and Lu, B. L. (2014b). Emotional state classification from EEG data using machine learning approach. *Neurocomputing* 129, 94–106. doi: 10.1016/j.neucom.2013.06.046

Wang, Y., Yu, L., Wu, S., Huang, R., and Yang, Y. (2020b). Memristor based biologically plausible memory based on discrete and continuous attractor networks for neuromorphic systems. *Adv. Intell. Syst.* 2:2000001. doi: 10.1002/aisy.202000001

Wang, Z., Ambrogio, S., Balatti, S., and Ielmini, D. (2015). A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning in neuromorphic systems. *Front. Neurosci.* 8:438. doi: 10.3389/fnins.2014.00438

Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., et al. (2018c). Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron*. 1, 137–145. doi: 10.1038/s41928-018-0023-2

Wang, Z., Joshi, S., Savel'ev, S. E., Jiang, H., Midya, R., Lin, P., et al. (2017). Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater*. 16, 101–108. doi: 10.1038/nmat4756

Wang, Z., Li, C., Lin, P., Rao, M., Nie, Y., Song, W., et al. (2019d). *In situ* training of feed-forward and recurrent convolutional memristor networks. *Nat. Mach. Intell*. 1, 434–442. doi: 10.1038/s42256-019-0089-1

Wang, Z., Yin, M., Zhang, T., Cai, Y., Wang, Y., Yang, Y., et al. (2016b). Engineering incremental resistive switching in TaOx based memristors for brain-inspired computing. *Nanoscale* 8, 14015–14022. doi: 10.1039/C6NR00476H

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *J. Big Data* 3:9. doi: 10.1186/s40537-016-0043-6

Wen, Z., Li, C., Wu, D., Li, A., and Ming, N. (2013). Ferroelectric-field-effect-enhanced electroresistance in metal/ferroelectric/semiconductor tunnel junctions. *Nat. Mater*. 12, 617–621. doi: 10.1038/nmat3649

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn*. 8, 229–256. doi: 10.1007/BF00992696

Witkowski, M., Cortese, M., Cempini, M., Mellinger, J., Vitiello, N., and Soekadar, S. R. (2014). Enhancing brain-machine interface (BMI) control of a hand exoskeleton using electrooculography (EOG). *J. Neuroeng. Rehabil*. 11, 1–6. doi: 10.1186/1743-0003-11-165

Woo, J., Moon, K., Song, J., Lee, S., Kwak, M., Park, J., et al. (2016). Improved synaptic behavior under identical pulses using AlO_x/HfO_2 bilayer RRAM array for neuromorphic systems. *IEEE Electron Device Lett*. 37, 994–997. doi: 10.1109/LED.2016.2582859

Wu, Q., Wang, H., Luo, Q., Banerjee, W., Cao, J., Zhang, X., et al. (2018). Full imitation of synaptic metaplasticity based on memristor devices. *Nanoscale* 10, 5875–5881. doi: 10.1039/C8NR00222C

Wu, W., Nagarajan, S., and Chen, Z. (2016). Bayesian machine learning: EEG/MEG signal processing measurements. *IEEE Signal Process. Mag*. 33, 14–36. doi: 10.1109/MSP.2015.2481559

Xiong, F., Liao, A. D., Estrada, D., and Pop, E. (2011). Low-power switching of phase-change materials with carbon nanotube electrodes. *Science* 332, 568–570. doi: 10.1126/science.1201938

Xu, J., Mitra, S., Hoof, C. V., Yazicioglu, R. F., and Makinwa, K. A. A. (2017). Active electrodes for wearable EEG acquisition: review and electronics design methodology. *IEEE Rev. Biomed. Eng*. 10, 187–198. doi: 10.1109/RBME.2017.2656388

Yagishita, S., Hayashi-Takagi, A., Ellis-Davies, G. C., Urakubo, H., Ishii, S., and Kasai, H. (2014). A critical time window for dopamine actions on the structural plasticity of dendritic spines. *Science* 345, 1616–1620. doi: 10.1126/science.1255514

Yan, Y., Kappel, D., Neumärker, F., Partzsch, J., Vogginger, B., Höppner, S., et al. (2019). Efficient reward-based structural plasticity on a SpiNNaker 2 prototype. *IEEE Trans. Biomed. Circuits Syst*. 13, 579–591. doi: 10.1109/TBCAS.2019.2906401

Yang, G., Lai, C. S. W., Cichon, J., Ma, L., Li, W., and Gan, W. B. (2014). Sleep promotes branch-specific formation of dendritic spines after learning. *Science* 344, 1173–1178. doi: 10.1126/science.1249098

Yang, G., Pan, F., and Gan, W. B. (2009). Stably maintained dendritic spines are associated with lifelong memories. *Nature* 462, 920–924. doi: 10.1038/nature08577

Yang, H., Hao, X., Wang, Z., Malmhall, R., Gan, H., Satoh, K., et al. (2017). "Threshold switching selector and 1S1R integration development for 3D cross-point STT-MRAM," in *2017 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 38.1.1–38.1.4. doi: 10.1109/IEDM.2017.8268513

Yang, J. J., Strukov, D. B., and Stewart, D. R. (2013). Memristive devices for computing. *Nat. Nanotechnol*. 8, 13–24. doi: 10.1038/nnano.2012.240

Yang, Z., Zhou, Q., Lei, L., Zheng, K., and Xiang, W. (2016). An IoT-cloud based wearable ECG monitoring system for smart healthcare. *J. Med. Syst*. 40:286. doi: 10.1007/s10916-016-0644-9

Yang-Scharlotta, J., Fazio, M., Amrbar, M., White, M., and Sheldon, D. (2014). "Reliability characterization of a commercial TaOx-based ReRAM," in *IEEE IIRW* (South Lake Tahoe, CA), 131–134. doi: 10.1109/IIRW.2014.7049528

Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., et al. (2020). Fully hardware-implemented memristor convolutional neural network. *Nature* 577, 641–646. doi: 10.1038/s41586-020-1942-4

Yazicioglu, R. F., Van Hoof, C., and Puers, R. (2008). *Biopotential Readout Circuits for Portable Acquisition Systems*. Springer Science & Business Media.

Zenke, F., and Ganguli, S. (2018). Superspike: supervised learning in multilayer spiking neural networks. *Neural Comput*. 30, 1514–1541. doi: 10.1162/neco_a_01086

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. (2019). "S4l: self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 1476–1485.

Zhang, W., Mazzarello, R., Wuttig, M., and Ma, E. (2019). Designing crystallization in phase-change materials for universal memory and neuro-inspired computing. *Nat. Rev. Mater*. 4, 150–168. doi: 10.1038/s41578-018-0076-x

Zhang, Y., and Harrison, C. (2015). "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognitiion," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology* (Charlotte, NC), 167–173.

Zhang, Y., Wang, Z., Zhu, J., Yang, Y., Rao, M., Song, W., et al. (2020). Brain-inspired computing with memristors: challenges in devices, circuits, and systems. *Appl. Phys. Rev*. 7:011308. doi: 10.1063/1.5124027

Zhang, Y., Xiao, R., and Harrison, C. (2016). "Advancing hand gesture recognition with high resolution electrical impedance tomography," in *UIST 2016–Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo), 843–850. doi: 10.1145/2984511.2984574

Zhang, Z., Dong, J., Luo, X., Choi, K. S., and Wu, X. (2014). Heartbeat classification using disease-specific feature selection. *Comput. Biol. Med*. 46, 79–89. doi: 10.1016/j.compbiomed.2013.11.019

Zhong, Y., Tang, J., Li, X., Gao, B., Qian, H., and Wu, H. (2021). Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat. Commun*. 12:408. doi: 10.1038/s41467-020-20692-1

Zhou, J. T., Pan, S. J., Tsang, I. W., and Yan, Y. (2014a). "Hybrid heterogeneous transfer learning through deep learning," in *Twenty-eighth AAAI Conference on Artificial Intelligence* (Québec), 2213–2219.

Zhou, J. T., Tsang, I. W., Pan, S. J., and Tan, M. (2014b). "Heterogeneous domain adaptation for multiple classes," in *Artificial Intelligence and Statistics* (Reykjavik), 1095–1103.

Zhu, M., and Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv* 1710.01878.

Zhu, S., Wang, L., and Duan, S. (2017a). Memristive pulse coupled neural network with applications in medical image processing. *Neurocomputing* 227, 149–157. doi: 10.1016/j.neucom.2016.07.068

Zhu, X., Du, C., Jeong, Y., and Lu, W. D. (2017b). Emulation of synaptic metaplasticity in memristors. *Nanoscale* 9, 45–51. doi: 10.1039/C6NR08024C

Zhu, X., Zheng, W., Lu, B., Chen, X., Chen, S., and Wang, C. (2014). "EOG-based drowsiness detection using convolutional neural networks," in *2014 International Joint Conference on Neural Networks (IJCNN)* (Beijing), 128–134. doi: 10.1109/IJCNN.2014.6889642

Zhu, Y., Chen, Y., Lu, Z., Pan, S. J., Xue, G. R., Yu, Y., et al. (2011). "Heterogeneous transfer learning for image classification," in *Twenty-Fifth AAAI Conference on Artificial Intelligence* (San Francisco, CA), 1304–1309.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer JE declared a past co-authorship with one of the authors MP to the handling editor.

Copyright © 2021 Covi, Donati, Liang, Kappel, Heidari, Payvand and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 29 November 2021
doi: 10.3389/fnins.2021.698635

Markov Chain Abstractions of Electrochemical Reaction-Diffusion in Synaptic Transmission for Neuromorphic Computing

Margot Wagner[1,2,3]*, *Thomas M. Bartol*[3,4], *Terrence J. Sejnowski*[1,2,3,4,5] and *Gert Cauwenberghs*[1,2,5]

[1] Department of Bioengineering, University of California, San Diego, La Jolla, CA, United States, [2] Institute for Neural Computation, University of California, San Diego, La Jolla, CA, United States, [3] Computational Neurobiology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA, United States, [4] Center for Theoretical Biological Physics, University of California, San Diego, La Jolla, CA, United States, [5] Division of Biological Sciences, University of California, San Diego, La Jolla, CA, United States

Progress in computational neuroscience toward understanding brain function is challenged both by the complexity of molecular-scale electrochemical interactions at the level of individual neurons and synapses and the dimensionality of network dynamics across the brain covering a vast range of spatial and temporal scales. Our work abstracts an existing highly detailed, biophysically realistic 3D reaction-diffusion model of a chemical synapse to a compact internal state space representation that maps onto parallel neuromorphic hardware for efficient emulation at a very large scale and offers near-equivalence in input-output dynamics while preserving biologically interpretable tunable parameters.

Keywords: neuromorphic, synapse, Markov chain, Monte Carlo, synaptic transmission, computational efficiency

OPEN ACCESS

Edited by:
Erika Covi,
NaMLab gGmbH, Germany

Reviewed by:
Stanisław Woźniak,
IBM Research-Zurich, Switzerland
Fabien Alibart,
UMR8520 Institut d'électronique, de microélectronique et de nanotechnologie (IEMN), France

***Correspondence:**
Margot Wagner
mwagner@ucsd.edu

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *21 April 2021*
Accepted: *04 November 2021*
Published: *29 November 2021*

Citation:
Wagner M, Bartol TM, Sejnowski TJ and Cauwenberghs G (2021) Markov Chain Abstractions of Electrochemical Reaction-Diffusion in Synaptic Transmission for Neuromorphic Computing. Front. Neurosci. 15:698635. doi: 10.3389/fnins.2021.698635

1. INTRODUCTION

It has been known since the pioneering of computer architecture by John von Neumann that brains are far more effective and efficient in processing sensory information than digital computers, owing to the massively parallel distributed organization of neural circuits in the brain that tightly couple synaptic memory and computing at a fine grain scale (von Neumann, 1958). Modern day computers still follow the "von Neumann" architecture where computing and memory are kept separate, incurring severe penalties in computing bandwidth due to the bottleneck in data flow between centralized processing and vast memory. Moore's law's relentless scaling of semiconductor technology, with a doubling of integration density every 2 years, has allowed the von Neumann architecture to remain fundamentally unchanged since its advent. As the shrinking dimensions of transistors supporting the progression of Moore's law are approaching fundamental limits, it has become essential to consider alternative novel computing architectures to meet increasing computational needs in this age of the deep learning revolution, which itself is driven by advances rooted in a deeper understanding of brain function (Sejnowski, 2020). At the forefront of this movement are neuromorphic systems, introduced by Mead (1990) as a solution to these limitations. Neuromorphic engineering looks toward human brains as inspiration for hardware systems due to their highly efficient computational nature. The human brain is regarded as the pinnacle of efficient computing, operating at an estimated rate of 10^{16} complex

operations per second while consuming less than 20 W of power (Churchland and Sejnowski, 1992; Cauwenberghs, 2013). Therefore, neuromorphic engineering looks to mimic the function and organization of neural structures using hybrid analog and digital systems. This is possible because there is significant overlap in the physics of computation between the brain and neuromorphic engineering (**Figure 1**). In both systems, information is carried in the form of charge, and, in hardware, neuronal membrane dynamics are represented using metal-oxide-semiconductor field-effect transistors (MOSFETs) (Mead, 1989). In the MOSFET sub-threshold region of operation, electrons and holes are the carriers of current between *n*- or *p*-type channels and behave akin to ions flowing through ion channels that mediate current across the neuronal cell membrane. Fundamentally, these hardware systems share analogous properties to their biological counterparts, including charge stochasticity, diffusion as the primary mechanism of carrier transport, and energy barriers modulated by gating voltage. Paired with Boltzmann distributions of charge, these systems are able to emulate current as an exponential function of the applied voltage, capturing the same biophysics underlying the neuronal dynamics (Mead, 1989; Broccard et al., 2017).

Since the introduction of neuromorphic engineering, computational models of different complexity have been introduced to describe neuronal dynamics, typically ranging from more detailed and realistic conductance-based Hodgkin-Huxley models to simpler integrate-and-fire models allowing for better scalability. Synaptic connectivity between neurons is of primary concern in the field currently because synaptic strength and plasticity are fundamental to learning and memory in both biological and artificial representations of neural networks (Indiveri et al., 2011; Broccard et al., 2017). In neuromorphic architectures, synapses instantiate both computation and memory, and a new focus on compact electronic implementations of this computational memory has been emerging recently including the use of memristors (Boybat et al., 2018). Efficient representation of synapses is a crucial topic of concern as there are roughly 10^4 synapses for each neuron, totalling approximately 10^{16} in the human brain. They are diverse in nature and have highly complex temporal and spatial dynamics, which further complicates their representations (Broccard et al., 2017). Currently, there is a push for efficient synaptic models while maintaining the intricate dynamical behavior exhibited biophysiologically. Current models include time-multiplexing synapses, analog bistable synapses, and binary synapses to name a few, but the need for scalable and dynamically complex models of synaptic function and transmission is still existent and critical (Bartolozzi and Indiveri, 2007; Broccard et al., 2017).

Modeling synapses is a challenging task due to their intricacy and sheer quantity. As noted above, there are an estimated 10^{16} synapses in the human brain. They vary in function and type, including both chemical and electrical synapses and exhibit behavior spanning multiple different temporal and spatial scales, as well as being highly stochastic in nature (van Rossum et al., 2003; Wang et al., 2020). Additionally, synaptic plasticity causes changes in synaptic strength over time associated with learning and memory. Synaptic transmission involves a multitude of mechanisms and molecular components, making simulations including all components not readily scalable. In order to capture the sophisticated dynamics of synapses in a scalable manner, abstractions have to be made according to the research problem in question. The stochastic nature of synapses also makes large scale simulations more complicated as modeling stochastic processes is typically more computationally demanding. It has been shown in multiple instances that the noise present due to the stochastic variability in synapses is highly integral to synaptic transmission, so this becomes an important feature to maintain (Malagon et al., 2016). For example, Moezzi et al. (2014) proved that models including ion channel noise in calcium channels paired with the existence of a presynaptic mechanism causing random delays in synaptic vesicle availability best capture the interspike interval behavior of auditory nerve fiber models. Additionally, multiple experimental works have found the existence of presynaptic vesicles that are released into the synaptic cleft with some probability (Castillo and Katz, 1954; Korn and Faber, 1991). There are multiple similar conclusions found in modeling and experimental results as recently discussed by McDonnell et al. (2016).

Synapses form the connections between neurons and the strength of these connections changes over time, forming the basis of learning and memory in both biological and artificial neural networks. The computations involved in accurately modeling the biophysics of synapses are complex due to the highly nonlinear nature of their dynamics, yet most of the neural network models in use today abstract synaptic strength to a single or small number of scalar values, tuned to a specific task. The learning rule for updating synaptic strength is then typically applied using abstractions of synaptic plasticity such as spike-time dependent plasticity and its causal extensions for scalable real-time hardware implementation (Pedroni et al., 2019). Physical constraints and limitations in VLSI implementations restrict the functional form of synaptic representation. In turn, these abstractions restrict the potential computing power of neuromorphic systems and restrain achievable benchmarks in approaching the functional flexibility, resilience, and efficiency of neural computation in the biological brain. Our work addresses the need for a more biophysically realistic model of the synapse with biologically tunable parameters to represent synaptic dynamics while offering a path toward efficient real-time implementation in neuromorphic hardware.

Synaptic transmission is dictated by a series of events initiated by presynaptic stimulation in the form of action potentials. An action potential causes membrane depolarization which leads to stochastic opening and closing of voltage-dependent calcium channels (VDCCs) lying on the presynaptic membrane and a resulting influx of calcium to the presynaptic terminal. Neurotransmitter release is modulated by calcium binding to calcium sensors near the neurotransmitter filled vesicles at the active zone, but calcium has other fates as it diffuses from the VDCCs. In addition to binding to the calcium sensors, it can bind to calbindin, which acts as a buffer, or it can be removed by plasma membrane calcium ATPase (PMCA) pumps. If enough calcium is able to bind to the calcium sensors, though,

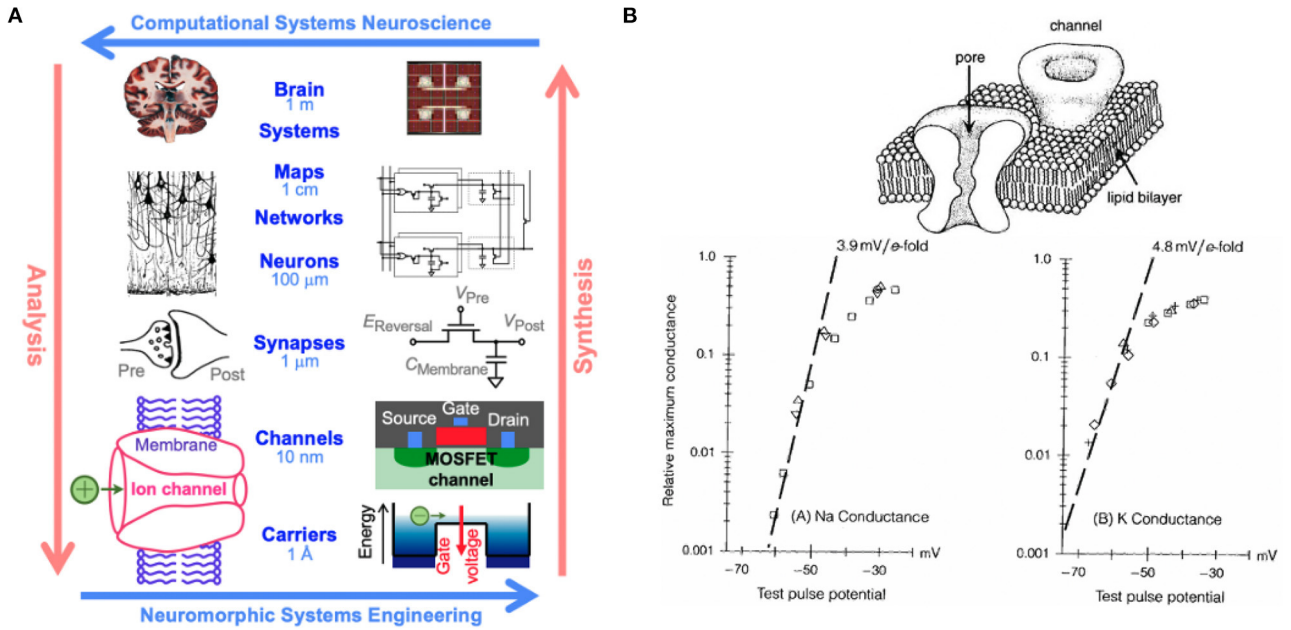


FIGURE 1 | (A) Multiscale correspondence between the fields of computational neuroscience and neuromorphic engineering. Reproduced with permission from Churchland and Sejnowski (1992) and Cauwenberghs (2013) and **(B)** equivalence in Boltzmann thermodynamics between metal-oxide-semiconductor field-effect transistors (MOSFETs) and ion channels resulting in current as an exponential function of applied voltage in both systems for sodium *(bottom left)* and potassium *(bottom right)* (Hodgkin and Huxley, 1952; Mead, 1989).

then neurotransmitters are released across the synaptic cleft and initiate downstream effects at the postsynaptic membrane (Bartol et al., 2015). This process of synaptic transmission is the basis of communication in the brain.

Abstracting this for computational efficiency, we created a series of Markov state transitions to realize the system with multiple internal states allowing for a biophysically tunable model of synaptic connectivity implementable in neuromorphic architectures. Markov models have a history of use as a stochastic discrete state alternative to Hodgkin-Huxley type formulations since their introduction (Hodgkin and Huxley, 1952; Armstrong, 1971; Colquhoun, 1973). Additional stochastic models have been introduced, including the Gillespie method (1977), which has been used to model neural channel noise (Gillespie, 1977; Skaugen and Walloe, 1979; Chow and White, 1996). Markov models have also found use in whole-cell models (Winslow et al., 1999). Further extensions utilize a particle model (Koch, 1999). The importance of the inclusion of stochasticity in ion channel behavior and synaptic transmission generally cannot be understated. Its inclusion has been demonstrated time and time again in experimental work and is thought to be integral in the form and function of synaptic transmission (McDonnell et al., 2016). This provides an additional complication in modeling synapses and has been handled at various different stages of transmission, including the stochastic models of vesicle release using probabilistically generated quantal components, stochastic models of transmitter diffusion, and stochastic models of receptors (Castillo and Katz, 1954; van Rossum et al., 2003; Bartol et al., 2015). These simulations are computationally expensive due to the high transition rates paired with the small number of transitions necessitating a small timepoint. Specifically, Markov models have shown to be an effective method of modeling ion channels but require high computational cost to effectively do so.

This paper looks to abstract the computationally complex and nonlinear nature of synaptic transmission dynamics in a manner that is efficient and readily scalable for implementation in neuromorphic silicon very large-scale integrated (VLSI) circuits. This is done by introducing an efficient stochastic sampling scheme within a Markov chain representation of the components integral to stochastic presynaptic quantal transmission.

2. MATERIALS AND METHODS

2.1. Markov Chain Models

The cascade of events from the action potential stimulus input to the presynaptic neurotransmitter release output can be equivalently modeled as a Markov chain to realize the system with multiple internal states instead of directly tracking all molecules and their kinetics in a computationally complex spatiotemporal 3D reaction-diffusion model. Each internal Markov state is assumed to be dependent solely on the state at the previous timepoint and is conditionally independent of all previous timepoints, simplifying simulations. Therefore, the fully biophysically complex system of synaptic transmission can be abstracted and sampled to create a Markov Chain Monte Carlo (MCMC) simulation which answers the same

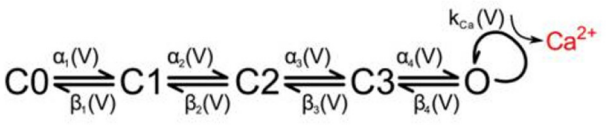


FIGURE 2 | State diagram for voltage-dependent calcium channels and resulting calcium influx in the presynaptic membrane. Reproduced with permission from Bartol et al. (2015).

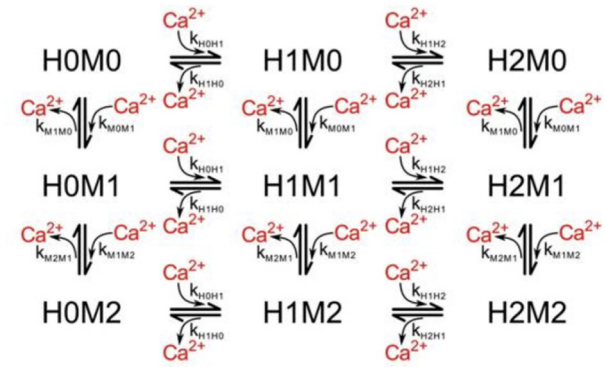


FIGURE 3 | State diagram for calbindin binding where H*a*M*b* describes the a^{th} high-affinity binding state and the b^{th} medium-affinity binding state. Reproduced with permission from Bartol et al. (2015).

question of neurotransmitter release utilizing tunable biophysical parameters while providing scalability for implementation in neuromorphic architectures.

For any given stimulus input, the VDCCs are assigned transition probabilities between states based on a five-state kinetic model (**Figure 2**) found experimentally and validated computationally with four closed states and a single open state (Church and Stanley, 1996; Bischofberger et al., 2002; Bartol et al., 2015). Prior to the stimulus, all VDCCs begin in the initial closed state, C0, and the concentration of calcium in the presynaptic terminal is at steady-state. The transition probabilities are voltage dependent akin to a Hodgkin-Huxley model where $\alpha_i(V) = \alpha_{io} \exp\left(\frac{V_m}{V_i}\right)$ and similarly $\beta_i(V) = \beta_{io} \exp\left(\frac{V_m}{V_i}\right)$ with parameter values from Bischofberger et al. (2002). The number of open VDCCs at any given moment is used to determine the number of calcium entering the presynaptic terminal based on experimental I-V curves and the resulting I-V equation found in Bischofberger et al. (2002) and used in Bartol et al. (2015), which gives the value for k_{Ca}. Calcium influx is captured by including transitions from the final closed VDCC state, C3, to the open VDCC state and an internal calcium generation. Using this, influx of calcium is modeled over the entire stimulus input due to the VDCCs opening.

Once calcium has entered the presynaptic terminal, much of it binds to calbindin, which acts as a buffer and primarily modulates the amount of calcium that is able to reach the calcium sensors at the active zone. The state transitions are reversible first-order reactions, thus transition probabilities are dependent on the free calcium in the system and updated as that amount changes over time. Calbindin has four binding sites, two of high affinity and two of medium affinity, leading to a nine-state calcium concentration-dependent kinetic model (**Figure 3**; Nagerl et al., 2000). By modeling the binding and unbinding of calcium to calbindin as a loss or gain of free calcium, respectively, calcium transients can also be elucidated.

Our Markov chain is a discrete-state chain in discrete time. Markov chains are modeled by a probability that the chain will move to another state given its current state and is conditionally independent of all previous timesteps. The probabilities are by nature only dependent on the current state of the Markov chain. The probability of the state of a molecule X can typically be predicted for a certain timepoint $t + \Delta t$ as some particular state x_j using the states at all previous timepoints from the start of the simulation, $t = 0$, to the timepoint just before that in question, t. For a Markov chain simulation solely dependent on the previous timepoint, it is possible to predict the probability that a molecule is in a given state, x_j at the timepoint $t + \Delta t$ using solely the state of the single timepoint just before, X_t, which is known to be a particular state x_i. Thus, the probability of the molecule being in state x_j given that at the previous timepoint it was in state x_i is given as P_{ij}. Succinctly, this is written as

$$\begin{aligned} P(X_{t+\Delta t} = x_j | X_t = x_i, X_{t-\Delta t}, ..., X_0) &= P(X_{t+\Delta t} = x_j | X_t = x_i) \\ &= P_{ij} \end{aligned} \tag{1}$$

For state transitions, the probability of transitioning to an adjacent state is the transition rate inherent in the system (k_{ij} for the transition from state i to state j, and k_{ij} is not necessarily equal to k_{ji}) times the timepoint, Δt. In the case of calbindin transitions, this is further multiplied by the amount of free unbound calcium for forward reactions as it is a first-order reaction. For the VDCCs, the transition rates are the α, β, and k_{Ca}. The probability that a molecule stays in its current state is the sum of the probabilities it transitions to an adjacent state subtracted from unity. For a multi-state system, this gives a transition probability matrix for the likelihood of transition from a given state at the current timepoint to any other state at the next timepoint. This matrix is sparse, with nonzero probabilities only for adjacent states to which a transition is possible. In the case of the five-state VDCC system, this gives the probability of a transition from state i to state j as

$$P_{ij} = \begin{cases} k_{ij}\,\Delta t & j = i \pm 1 \\ 1 - \sum_{k=0}^{N_{adj}} P_{ik} & j = i, k \neq i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where transitions to adjacent states are given by the transition rate k_{ij} times the timepoint, Δt; the probability of staying in the current state is the sum of probabilities of adjacent state transitions subtracted from unity, where N_{adj} is the number of possible adjacent states. The probability of transitioning to a non-adjacent state is set to zero.

Typically Markov state transitions are modeled via a discrete inverse transform method, where given a random variable X, the

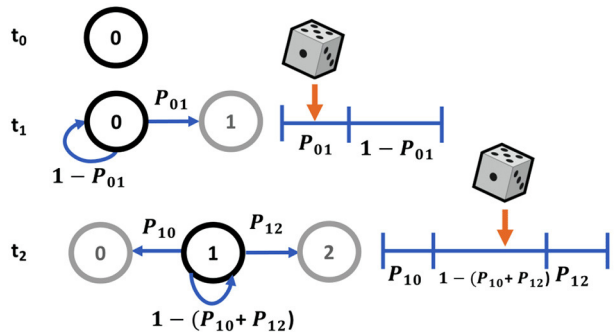


FIGURE 4 | Markov sampling scheme for state transitions using partitions of unity.

transition probabilities P_{ij} describe a partition of unity (**Figure 4**). Therefore, we can generate a random number uniformly, $R \sim U(0,1)$ and map it onto discrete values of X. For example, in a two state system, $X_j = 0$ if $R \leq P_{i0}$ or $X_j = 1$ if $P_{i0} < R \leq P_{i0} + P_{i1} = 1$. This involves searching the state space for the next state given the current state for each molecule in the system at each timepoint, which can be a slow process for systems with a large number of states and molecules.

Here we have implemented a more efficient MCMC sampling strategy involving sampling from a multinomial distribution. Therefore, instead of sampling from a uniform distribution for each of n molecules, we sample from a multinomial distribution once for each state, using n molecules as the number of experiments, where $X \sim \text{Multi}(n, p_1, \ldots p_k)$. For simulations where the number of possible states is less than the number of molecules, this is a more efficient sampling strategy. Since we are particularly interested in the number of molecules in each state at each timepoint, this is an effective approach. Multinomial sampling thus describes the distribution of the n experiments across k possible outcomes each with a probability of p_k, where n_k is the number of experiments falling into the k^{th} outcome following a probability mass function of

$$f(n_1, \ldots, n_k; n, p_1, \ldots p_k) = \frac{n!}{n_1! \ldots n_k!} p_1^{n_1} \ldots p_k^{n_k} \quad (3)$$

In our model, for each state i, we have an initial number of molecules in that state at a given timepoint t, or $n_{i,t}$. As previously described, there exists a probability that the molecules will transition to any state at the next timepoint, including staying in the original state given by P_{ij}. Thus, to determine the distribution of molecules $n_{i,t}$ across all states at the next timepoint, we sample from a multinomial distribution according to

$$X_{i,t+\Delta t} \sim \text{Multi}(n_{i,t}, P_{i1}, \ldots P_{ik}) \quad (4)$$

for k possible states. We do this sampling for each state at each timepoint and sum accordingly. This expedites computation by only requiring a single computation at each timepoint, sampling the distribution of all n molecules at once. **Algorithm 1** highlights the pseudocode for this process.

Algorithm 1: Markov Multinomial Reaction Sampling.

```
Result: The number of molecules in each state at each
        timepoint for a simulation.
initialize number of states;
initialize matrix of number of molecules per state per
timepoint;
for each timepoint do
    for each state do
        calculate transition probability according to Equation
        (1);
        sample multinomial distribution with current
        number of molecules in the state and transition
        probability;
    end
    update number of molecules in each state using samples;
end
```

Markov simulations for the VDCCs were run for 65 VDCCs all starting in the closed state, C0. Calbindin molecules were initiated in the different binding states according to the steady-state concentration of calcium and at a baseline concentration of $4.5\,10^{-5}$ M. All simulations were run for 10 ms with a timestep of 1 μs. The simulations were repeated 1,000 times to obtain an average and standard deviation. Markov simulations were implemented using Python.

2.2. MCell Models

MCell is a modeling software that uses spatially realistic 3D geometries and Monte Carlo reaction-diffusion modeling algorithms, which allows for biophysically realistic simulations of high complexity as it specifically tracks the state of every molecules in space and time (Bartol et al., 2015). Due to the accuracy and specificity, it provides a ground truth for biological simulations but does so at the cost of computational complexity.

To validate and compare our Markov models of synaptic transmission, we built a biophysically realistic stochastic 3D reaction-diffusion system with all major components for presynaptic vesicle release variability in response to a stimulus input (**Figure 5**) based on the models of Nadkarni et al. (2010) and Bartol et al. (2015). The model includes realistic geometry for a CA3-CA1 *en passant* synapse focusing primarily on the presynaptic Schaffer collateral axon of a CA3 pyramidal cell found in the hippocampus with parameters set from experimental data (Nadkarni et al., 2010; Bartol et al., 2015). The CA3-CA1 synapse was chosen for the simulations as it is highly studied experimentally and is important for learning and memory. Furthermore, CA3-CA1 synapses are relatively small, containing one to two neurotransmitter release zones. Release from this region is also known to be highly stochastic in nature, necessitating the inclusion of stochasticity in biologically realistic models (Nadkarni et al., 2010). All kinetics and parameters match those used for the equivalent Markov models.

The MCell model includes the canonical presynaptic geometry for an average CA3-CA1 synaptic terminal as a rectangular box

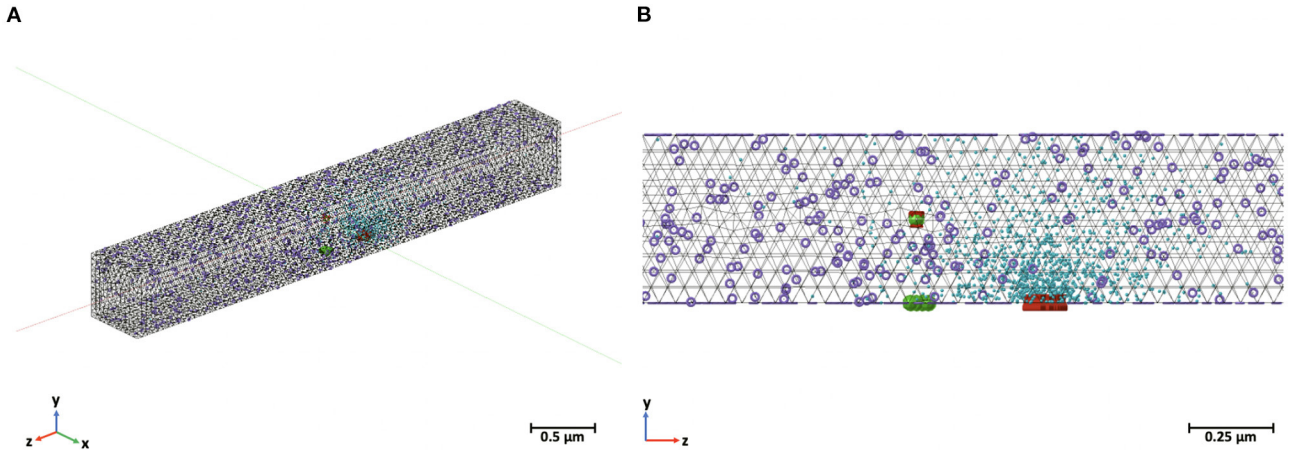


FIGURE 5 | MCell model for synaptic transmission containing voltage-dependent calcium channels (red), calcium (blue), calcium sensors (green), and plasma membrane calcium ATPase pumps (purple). Calbindin not pictured here due to their large number. **(A)** Entire 0.5 μm by 0.5 μm by 4 μm box representing one vesicular release site in a Schaffer collateral axon in the CA3 region and **(B)** a close-up of the release site.

measuring 0.5μm by 0.5μm by 4μm. This box captures the dynamics of a single synaptic active zone, referring to the region on the presynaptic membrane specialized for neurotransmitter release. Initially, the terminal contains the calbindin buffer, steady-state calcium concentration, PMCA pumps, VDCCs and calcium sensors modulating neurotransmitter release (Nadkarni et al., 2010). The detailed diffusion dynamics and kinetics of these systems are based on experimental data and have been discussed in further detail in Bartol et al. (2015). The active zone is based on that of an average presynaptic active zone containing seven docked neurotransmitter vesicle release sites. The VDCCs, of type P/Q, are stationed at a biophysically realistic distance from the active zone. They transition states in response to the membrane depolarization. The location, number, and calcium conductance of the VDCCs is replicated from experimental data (Nadkarni et al., 2010). PMCAs are homogenously placed across the presynaptic membrane while calbindin molecules are in a uniform concentration within the volume. This is a flexible architecture that can respond to any stimulus input and allows for monitoring of the states of each molecule in the system. The MCell CA3-CA1 synaptic transmission models were originally created and validated in Nadkarni et al. (2010) and Bartol et al. (2015). To compare with the Markov models, we used the same single action potential stimulus.

MCell models were also run 1,000 times for 10 *ms* with a timestep of 1 μs.

3. RESULTS

3.1. Voltage-Dependent Calcium Channels

The efficient Markov chain implementation has strong agreement with the full MCell model in terms of the internal state transients in response to an external stimulus. The number of closed VDCCs (state C0) decreases over the duration of the stimulus (**Figure 6A**). The internal states (C1-C3) subsequently increase and decrease as the membrane voltage increases and the forward rates for the VDCCs increase (**Figures 6B–D**), leading to an exponential increase in the open VDCCs while the membrane depolarizes. **Figure 6E** shows the fraction of open VDCCs over time in response to the action potential, which controls the amount of calcium influx to the system. At the maximum membrane potential, almost all VDCCs are in the open state. As the membrane repolarizes, the reverse reaction rate constants increase, and the VDCCs close. This leads to another increase and decrease in the internal VDCC states as the receptors go from their open to resting closed state (C0).

In its open state, VDCCs allow for the probabilistic influx of calcium through the channels into the presynaptic bouton. This is exemplified in **Figure 6F**, where there is an increase in the calcium influx through the open VDCCs over the course of the stimulus. Again, there is strong agreement between the more computationally complex MCell model and the computationally efficient Markov equivalent model.

3.2. Calbindin Buffer

Simulations of homogeneous calcium and calbindin were run using the MCell, Markov and deterministic simulation schemes. In the presence of calcium, the forward binding reaction is heavily favored, and this is highlighted in **Figure 7A** where free calcium exponentially decreases. A similar transient is apparent for the unbound state of calbindin, as it quickly transitions to different stages of high and medium binding **Figure 7B**. Over the course of the simulation, all the free calcium is removed from the system, and calbindin states reach a new steady-state where there is still unbound calbindin. Similarly, the fully bound state, H2M2, rapidly increases and reaches a new steady state that is still only 1% of all calbindin **Figure 7C**. This is due to the high concentration of calbindin in the presynaptic bouton. Even once all the calcium is in a bound state, there is still plenty of unbound or partially bound calbindin remaining in the

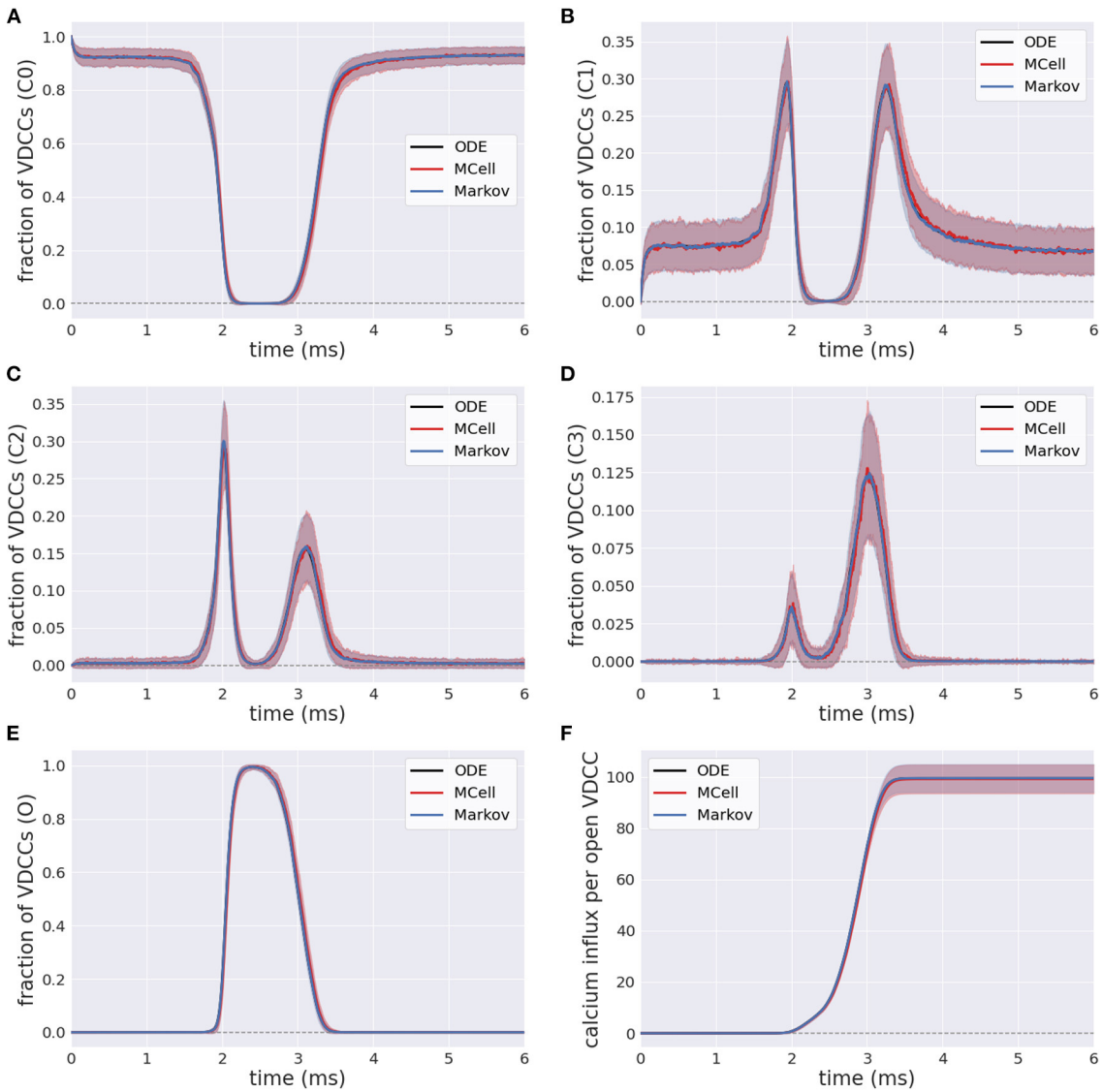


FIGURE 6 | Fraction of voltage-dependent calcium channels (VDCCs) in each state: **(A–D)** Internal closed channel states, C0-C3, and **(E)** The open channel state, O. **(F)** Calcium influx through open VDCCs in response to action potential stimulus for stochastic MCell, Markov, and deterministic ODE forward Euler simulation.

system. Calbindin acts as a strong buffer allowing for calcium storage and asynchronous neurotransmitter release, so this and slow unbinding rates become an important feature of calbindin. The rapid extent to which calcium binds to calbindin shows the impact of buffering on calcium's ability to diffuse and bind to the calcium sensors in the active zone. The inclusion of calbindin at such high concentrations becomes a key feature of maintaining the relatively low release rates of neurotransmitters even in the presence of a stimulus.

3.3. Complexity Analysis

MCell uses a scheduler which allows for only making changes to the scheduled particles, though in the worst-case, this still scales with the total number of particles in the simulation, n, where n_{VDCC} is 65 and n_{calb} is 2.7×10^4. It also scales with the length of the simulation, t, described by the number of time points for a discrete simulation. The simulations for the VDCC and calbindin both use 10k timepoints. At each timepoint, a particle can transition to any of its adjacent or branched states, b, which is similarly described by a fan-out factor in electronic implementation. From the VDCC kinetic model described in **Figure 2**, b_{VDCC} is 1–2 depending on the state while the calbindin kinetic model in **Figure 3** gives b_{calb} of 2-4. The overall time complexity for MCell is $\mathcal{O}(bnt)$. The classical Markov representation tracks every particle. It also searches through the space of each adjacent state for potential state transitions at each time point. Therefore, classical Markov implementation similarly results in an $\mathcal{O}(bnt)$ time complexity, or $\mathcal{O}(nt \log_2 b)$ at best

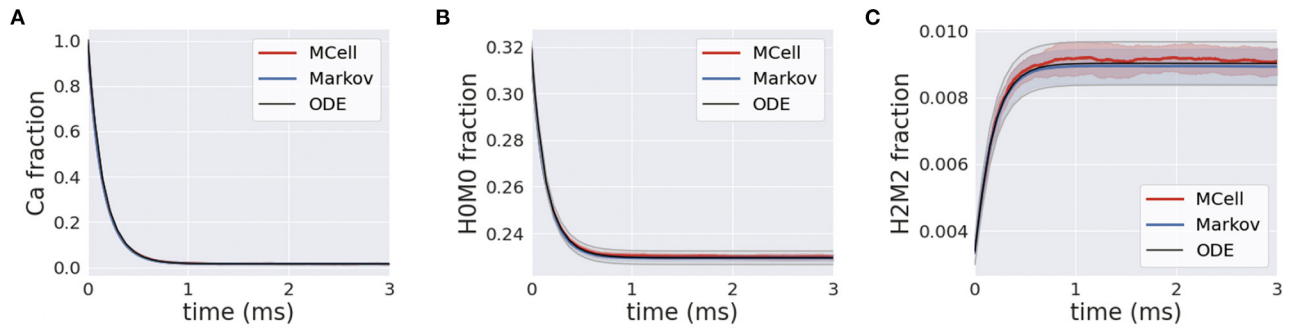


FIGURE 7 | (A) Transients for homogeneous calcium-calbindin buffer binding in the presynaptic bouton for free calcium; **(B)** the unbound calbindin state, H0M0; and **(C)** the fully bound calbindin state, H2M2 in all simulation types.

for implementation with an efficient search algorithm. Both the multinomial Markov model and the Euler ODE implementation describe the system in terms of the number of molecules in each state leading to a dependence on the total number of states, $s \geq b$, rather than the total number of particles. The total number of states for VDCC is 5 (**Figure 2**) while the number of states for calbindin is 9 (**Figure 3**). Due to efficient sampling methods, the multinomial Markov method is independent of the number of adjacent states, leading to a time complexity of $\mathcal{O}(bst)$ for both the multinomial Markov and Euler ODE methods. Thus, our stochastic multinomial Markov model is equally amenable to large scale simulations as the deterministic ODE method that is typically used in simulations involving more synapses.

The traditional Markov sampling model and the MCell representation store the molecular states in bits for each particle as well as the states adjacent to the current state, leading to a space complexity of $\mathcal{O}(bn \log_2 s)$ The efficient Markov model and ODE solution both simply store the number of molecules represented by bits in each state at each timepoint as well as the branched states resulting in a space complexity of $\mathcal{O}(bs \log_2 n)$. There exists a trade off here between the number of particles in each state compared to the number of states where one is stored directly and one is stored as an index. Thus, for simulations where the number of states is less than the number of particles, the multinomial Markov model is an efficient representation of the system, which is typically the case for biochemical simulation. MCell is more efficient with large state-space systems, but the number of states could be sparsified in a multinomial Markov representation by implementing dynamic instantiation and annihilation of states. Additionally, unseen or rarely seen states could be ignored by truncating based on probability of a particle being in that state. This would functionally decrease the number of states in the system allowing for use of the multinomial Markov simulation method.

3.4. Benchmarks

Runtime and total floating point operations were used as metrics for comparison between the simulation methods (**Table 1**). We also looked at the number of pseudorandom number generator calls (n_{PRNG}) between the simulations as this provides a metric to elucidate the differences observed in execution time between the simulations. Here we compare MCell, the standard Markov model, and the multinomial Markov stochastic models. The deterministic Euler solution is included as well for a non-stochastic comparison. Again, it is valuable to note the importance of stochasticity in these models. Significant work has shown the necessity of stochasticity in models of synaptic transmission in order to match experimental work. It has been demonstrated that deterministic models at this scale generally underestimate quantal release as concentration fluctuations are not captured (van Rossum et al., 2003; Bartol et al., 2015; McDonnell et al., 2016). Thus, while deterministic ODE models provide efficient simulation techniques, they are not able to capture the full complexity of the dynamics of synaptic transmission, hence motivating the need for an efficient stochastic model.

In the VDCC simulations, the multinomial sampling MCMC model has a runtime on the order of the forward Euler deterministic solution. The MCell and the standard Markov stochastic models exemplify a runtime an order of magnitude higher. The number of operations is also higher for the MCell and the standard Markov models compared to the multinomial Markov model. The standard Markov case generates a pseudorandom number for each molecule and each timestep, so n_{PRNG} is equivalent to the number of molecules multiplied by the number of timesteps, $n_{VDCC}t$. In the multinomial Markov simulation, a pseudorandom number is generated for each occupied state and possible branching points at each timepoint, which gives $(bs)_{VDCC}t$ in the worst-case scenario. Therefore, n_{PRNG} is smaller for the multinomial case as long as $(bs)_{VDCC} < n_{VDCC}$, which is always the case here.

For the calbindin model, the multinomial Markov method is again an order of magnitude faster than the MCell model although it is also an order of magnitude slower than the deterministic model. The standard Markov model is an order of magnitude slower than the multinomial model. The operations are also fewer for the multinomial case than the standard case. Again, the standard Markov case gives n_{PRNG} equal to $n_{calb}t$ while the multinomial Markov simulation is $(bs)_{calb}t$. Again we see a smaller n_{PRNG} in the mulinomial case because

TABLE 1 | Benchmarks for different simulation types for both voltage-dependent calcium channel and calbindin binding simulations.

VDCC	Runtime (sec)	No. Pseudorandom number generator calls (n_{PRNG})	Total floating-point operations (FLOPs)
MCell	139.20	1.5×10^9	1.7×10^{11}
Standard markov	109.16	6.5×10^5	7.0×10^7
Muttinomial markov	9.40	1.0×10^5	1.3×10^7
ODE (Euler)	10.05	0	7.7×10^6
Calbindin	**Runtime**	**No. Pseudorandom number generator calls (n_{PRNG})**	**Total floating-point operations (FLOPs)**
MCell	29.34	6.9×10^7	8.7×10^9
Standard Markov	76.78	2.7×10^8	1.4×10^{10}
Muttinomial Markov	2.37	3.6×10^5	2.7×10^7
ODE (Euler)	0.72	0	1.6×10^6

$(bs)_{calb} < n_{calb}$ even in the worst-case scenario where b is at its maximum value. Simulations are not currently optimized on hardware suggesting opportunities for further decreases in runtime. Overall, the multinomial Markov simulation provides a computationally efficient alternative to stochastic MCell simulations while maintaining the biological accuracy.

3.5. Neuromorphic Implementation

Thermodynamic foundations of neuromorphic engineering suggest direct biophysical implementation of populations of ion channels with individual stochastic opening and closing of gating variables driven by thermal noise fluctuations (Mead, 1989). So it seems only natural to consider implementations using stochastic ODEs describing the rates of reaction kinetics under additive white Gaussian noise (AWGN):

$$\frac{dX_{i,t}}{dt} = \sum_j k_{ij} X_{j,t} + \xi_{i,t} \tag{5}$$

where $\xi_{i,t}$ is normally distributed with zero mean and variance dependent on the magnitude of $X_{i,t}$. Fully parallel, continuous-time analog implementation of reaction kinetic rate equations of the type (Equation 5) have been demonstrated in micropower integrated circuits, e.g.,cytomorphic chips in BiCMOS integrated silicon technology (Woo et al., 2018). Abundant intrinsic noise present in these micropower cytomorphic circuits can serve as AWGN, although its magnitude is determined by thermal processes that are hard to control and other non-white Gaussian sources of intrinsic noise contribute strongly colored low-frequency spectra. Thus, discrete-time implementation of the ODEs (Equation 5) through Euler integration on a digital computer offers greater control over the shape and amplitude of the AWGN distribution, limited by the quality of pseudo-random number generation by deterministic algorithms.

Although purely digital algorithmic implementations go against foundational principles of neuromorphic engineering rooted in the physics of computation (Mead, 1989), the convenience of their programmability and reproducibility have made ODE-based digital emulation platforms such as Loihi a popular choice among more software-focused neuromorphic computer scientists (Davies et al., 2018). The computation involved in such discrete-time ODEs (Equation 5) can be performed at varying degrees of parallelism in custom or reconfigurable digital hardware, with the variables X_i being updated in sequence through time-multiplexing a single processing core in one extreme case, or all X_i updated in parallel with dedicated processing elements for each in the other extreme case. Ultimately in practice, the energy efficiency is relatively independent of the compute implementation, and depends more critically on the available memory bandwidth in accessing the rate parameters defining network connectivity (Pedroni et al., 2020). In essence, discrete-time Euler-integration ODE implementation of Equation (5) amounts to sampling from a normal distribution

$$X_{i,t+\Delta t} \sim \mathcal{N}(n_{i,t+\Delta t}, \sigma_{i,t}) \tag{6}$$

with mean and standard deviation

$$n_{i,t+\Delta t} = \sum_j P_{ij}\, n_{j,t} \tag{7}$$

$$\sigma_{i,t} = \sqrt{n_{i,t}\,(n - n_{i,t})} \tag{8}$$

incurring computational complexity $\mathcal{O}(bst)$ (section 3.3).

More fundamentally, the main disadvantage of implementing stochastic ODEs (Equation 5) or their discrete-time digital versions (Equation 6) is that they are primarily based on the Central Limit Theorem for very large number of variables, $n \to \infty$. As such, they have limited accuracy in approximating the reaction kinetics in systems with smaller numbers of molecular variables. While one may be tempted to assume that molecules are always excessively abundant, this is not typically the case since reactions are rate limited by the least abundant of reagents. Low numbers in molecular dynamics are prevalent in biologically relevant settings, giving rise to significant amounts of biological noise that are critical in neural dynamics, e.g., the highly stochastic quantal release of neurotransmitter in synaptic transmission. Thus, there is need for a mathematical description of stochastic synaptic transmission dynamics able to capture the accuracy in simulations with relatively small numbers of variables. Here we have shown that our multinomial Markov alternative, which directly samples the variables from the multinomial distribution (Equation 4) rather than the limiting normal distribution (Equation 6), produces accurate results for any value of n while offering

TABLE 2 | Space and time complexity for the various simulation strategies.

Complexity	Time complexity	Space complexity
MCell	$O(b\ n\ t)$	$O(b\ n\ \log_2 s)$
Standard markov	$O(b\ n\ t)$	$O(b\ n\ \log_2 s)$
Multinomial markov	$O(b\ s\ t)$	$O(b\ s\ \log_2 n)$
ODE (Euler)	$O(b\ s\ t)$	$O(b\ s\ \log_2 n)$

n (total number of molecules), s (total number of states), b (number of branching states), t (number of timepoints).

nearly identical implementation complexity $\mathcal{O}(bst)$ (section 3.3). Hence we see the Markov chain abstractions of reaction kinetics not only as a means to approach biophysical realism in modeling molecular cellular dynamics without molecular-scale representation, but also as a means toward efficient neuromorphic hardware without biophysical compromise. The key point is that the computational complexity of implementing our multinomial Markov model is essentially identical to that of stochastic ODEs (see **Table 2**), whether in software executing serially on a von Neumann programmable digital computer or in massively parallel digital or analog hardware. Hence, the neuromorphic circuit designer tasked to implement brain-inspired models of information processing faces an easy choice: more bio-realistic models that account for detailed stochasticity in reaction kinetics incur the same resource utilization and energy costs, and use similar design principles, as their stochastic ODE approximations.

In addition to field-programmable gate array (FPGA) reconfigurable (Pedroni et al., 2020) or custom-integrated neuromorphic programmable (Davies et al., 2018) instantiations in digital hardware, we envision physically neuromorphic instantiations in micropower analog continuous-time compute-in-memory hardware that obviate sampling from posterior distributions and directly implement Markov state transitions through parallel implementation of sum-product rules with self-normalizing probabilities (Chakrabartty and Cauwenberghs, 2004, 2005), at throughput density and energy efficiency that are orders of magnitude higher than today's most advanced general-programmable computational platforms.

4. DISCUSSION

The goal of this work was to create a more computationally efficient model of biologically realistic synaptic transmission for use in large-scale neuromorphic systems. We created a multinomial MCMC sampling strategy for capturing the internal states of vital molecules in the system in response to stimulus where transition probabilities could be voltage- or concentration-dependent, and the next timestep could be predicted solely using the current timestep. This scheme was implemented to capture the dynamics of the stochastic opening and closing of VDCCs through multiple internal states as well as the resulting calcium influx into the presynaptic bouton through the open VDCCs. Once calcium has entered the presynaptic terminal, we also simulated calcium binding to the calbindin buffer which modulates calcium levels in the bouton, directly impacting the amount of calcium that reaches the calcium sensors in the active zone. This amount impacts the neurotransmitter release from the presynaptic side and the resulting effects on the postsynaptic side.

All simulations were modeled using the multinomial Markov sampling method as well as a typical Markov sampling method and compared to highly detailed 3D geometric stochastic reaction-diffusion simulations done using MCell. The Markov simulations show agreement with the MCell simulations for the system dynamics including the number of open VDCCs and calcium influx in response to an action potential stimulus as well as the binding of calcium to the calbindin buffer. Differences are observed from the deterministic solution to the stochastic simulations implying the importance of stochasticity in these simulations to capture more biologically-realistic systems.

Exemplified by runtime and total number of operations, the multinomial MCMC method of simulations was shown to be more efficient than the standard Markov model while also being faster than the MCell equivalents. This is hopeful for scaling these biologically-realistic models to large-scale systems while maintaining biological tunability.

Next steps involve modeling the remaining kinetics in a similar fashion including the binding and removal of calcium by the plasma membrane calcium (PMCA) pumps as well as binding to the calcium sensors. In addition, to capture the diffusion of calcium through the presynaptic terminal but specifically to the calcium sensors at the active zone, a diffusive kernel must be included to the system. Upon inclusion of these elements, the entire process from stimulus to neurotransmitter release can be captured as a series of Markov chains leading to powerful implications for synaptic transmission modeling. The whole synapse can be included as well with the inclusion of a diffusive kernel across the synaptic cleft as well as downstream effects on the postsynaptic side, of which many mirror similar kinetics and dynamics as the presynaptic side leading to a natural extension of this modeling framework. The resulting system would be a biologically tunable model of synaptic transmission for any stimulus input in a highly efficient manner. This opens the door for large-scale implementations of synaptic transmission and learning readily implementable into neuromorphic architectures with strong biological realism.

Through the utilization of Markov-based abstractions applied to biophysically realistic 3D reaction-diffusion models of a chemical synapse, we have created a compact and efficient internal state space representation of synaptic transmission. This is in response to the challenge presented by the high dimensionality and complex nature of molecular-scale interactions in synapses and across scales making implementation in very large-scale systems previously unattainable. The model is directly amenable to efficient emulation in parallel neuromorphic hardware systems while maintaining biophysically relevant and interpretable parameters that are readily tunable. This opens the door toward neuromorphic circuits and systems on very large scale that strike a greater balance between integration density and biophysical accuracy in modeling neural function at the whole-brain level.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/margotwagner/markov-mcell-data.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

This work was supported by a NSF Graduate Research Fellowship to MW, the National Science Foundation, and the Office of Naval Research.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2021.698635/full#supplementary-material

REFERENCES

Armstrong, C. M. (1971). Interaction of tetraethylammonium ion derivatives with the potassium channels of giant axons. *J. Gen. Physiol.* 58, 413–437. doi: 10.1085/jgp.58.4.413

Bartol, T. M., Keller, D. X., Kinney, J. P., Bajaj, C. L., Harris, K. M., Sejnowski, T. J., et al. (2015). Computational reconstitution of spine calcium transients from individual proteins. *Front. Synaptic Neurosci.* 7:17. doi: 10.3389/fnsyn.2015.00017

Bartolozzi, C., and Indiveri, G. (2007). Synaptic dynamics in analog VLSI. *Neural Comput.* 19, 2581–2603. doi: 10.1162/neco.2007.19.10.2581

Bischofberger, J., Geiger, J. R. P., and Jonas, P. (2002). Timing and efficacy of Ca^2+ channel activation in hippocampal mossy fiber boutons. *J. Neurosci.* 22:0593-602. doi: 10.1523/JNEUROSCI.22-24-10593.2002

Boybat, I., Gallo, M. L., Nandakumar, S. R., Moraitis, T., Parnell, T., Tuma, T., et al. (2018). Neuromorphic computing with multi-memristive synapses. *Nat. Commun.* 9:2514. doi: 10.1038/s41467-018-04933-y

Broccard, F. D., Joshi, S., Wang, J., and Cauwenberghs, G. (2017). Neuromorphic neural interfaces: from neurophysiological inspiration to biohybrid coupling with nervous systems. *J. Neural Eng.* 14, 041002. doi: 10.1088/1741-2552/aa67a9

Castillo, J., and Katz, B. (1954). Quantal components of the end-plate potential. *J. Physiol.* 124, 560–573. doi: 10.1113/jphysiol.1954.sp005129

Cauwenberghs, G. (2013). Reverse engineering the cognitive brain. *Proc. Natl. Acad. Sci. U.S.A.* 110, 15512–15513. doi: 10.1073/pnas.1313114110

Chakrabartty, S., and Cauwenberghs, G. (2004). "Margin normalization and propagation in analog VLSI," in *2004 IEEE International Symposium on Circuits and Systems (ISCAS), Vol.* 1 (Vancouver, BC: IEEE), I–901. doi: 10.1109/ISCAS.2004.1328341

Chakrabartty, S., and Cauwenberghs, G. (2005). "Sub-microwatt analog VLSI support vector machine for pattern classification and sequence estimation," in *Advances in Neural Information Processing Systems, Vol. 17*, eds L. Saul, Y. Weiss, and L. Bottou (Cambridge, MA: MIT Press).

Chow, C. C., and White, J. A. (1996). Spontaneous action potentials due to channel fluctuation. *Biophys. J.* 71, 3013–3021. doi: 10.1016/S0006-3495(96)79494-8

Church, P., and Stanley, E. (1996). Single L-type calcium channel conductance with physiological levels of calcium in chick ciliary ganglion neurons. *J. Physiol.* 496 (Pt 1):59–68. doi: 10.1113/jphysiol.1996.sp021665

Churchland, P. S., and Sejnowski, T. J. (1992). The Computational Brain. *The MIT Press*. doi: 10.7551/mitpress/2010.001.0001

Colquhoun, D. (1973). *The Relation Between Classical and Cooperative Models for Drug Action. Drug Receptors*. (London: Macmillan Press).

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* 81, 2340–2361. doi: 10.1021/j100540a008

Hodgkin, A., and Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073

Koch, C. (1999). *Biophysics of Computation*. (New York: Oxford University Press).

Korn, H., and Faber, D. (1991). Quantal analysis and synaptic efficacy in the CNS. *Trends Neurosci.* 14, 439–445. doi: 10.1016/0166-2236(91)90042-S

Malagon, G., Miki, T., Llano, I., Neher, E., and Marty, A. (2016). Counting vesicular release events reveals binomial release statistics at single glutamatergic synapses. *J. Neurosci.* 36, 4010–4025. doi: 10.1523/JNEUROSCI.4352-15.2016

McDonnell, M. D., Goldwyn, J. H., and Lindner, B. (2016). Neuronal stochastic variability: Influences on spiking dynamics and network activity. *Front. Comput. Neurosci.* 10:38. doi: 10.3389/fncom.2016.00038

Mead, C. (1989). *Analog VLSI and Neural Systems*. (Boston, MA: Addison-Wesley).

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636. doi: 10.1109/5.58356

Moezzi, B., Iannella, N., and McDonnell, M. D. (2014). Modeling the influence of short term depression in vesicle release and stochastic calcium channel gating on auditory nerve spontaneous firing statistics. *Front. Comput. Neurosci.* 8:163. doi: 10.3389/fncom.2014.00163

Nadkarni, S., Bartol, T. M., Stevens, C. F., Sejnowski, T. J., and Levine, H. (2010). Modelling vesicular release at hippocampal synapses. *PLoS Comput. Biol.* 6:e1000983. doi: 10.1371/journal.pcbi.1000983

Nagerl, U. B., Novo, D., Mody, I., and Vergara, J. L. (2000). Binding kinetics of calbindin-D28k determined by flash photolysis of caged Ca^2+. *Biophys. J.* 79, 3009–3018. doi: 10.1016/S0006-3495(00)76537-4

Pedroni, B. U., Deiss, S. R., Mysore, N., and Cauwenberghs, G. (2020). "Design principles of large-scale neuromorphic systems centered on high bandwidth memory," in *2020 International Conference on Rebooting Computing (ICRC)*, (Atlanta, GA) 90–94.

Pedroni, B. U., Joshi, S., Deiss, S. R., Sheik, S., Detorakis, G., Paul, S., et al. (2019). Memory-efficient synaptic connectivity for spike-timing-dependent plasticity. *Front. Neurosci.* 13:357. doi: 10.3389/fnins.2019.00357

Sejnowski, T. J. (2020). *The Deep Learning Revolution*. (Cambridge, MA: The MIT Press).

Skaugen, E., and Walloe, L. (1979). Firing behavior in a stochastic nerve membrane model based upon the Hodgkin-Huxley equations. *Acta Physiol. Scand.* 107, 343–363. doi: 10.1111/j.1748-1716.1979.tb06486.x

van Rossum, M. C., O'Brien, B., and Smith, R. (2003). Effects of noise on the spike timing precision of retinal ganglion cells. *Am. Physiol. Soc.* 89, 2406–2419. doi: 10.1152/jn.01106.2002

von Neumann, J. (1958). *The Computer and the Brain*. Yale University Press.

Wang, J., Cauwenberghs, G., and Broccard, F. D. (2020). Neuromorphic dynamical synapses with reconfigurable voltage-gated kinetics.

IEEE Trans. Biomed. Eng. 67, 1831–1840. doi: 10.1109/TBME.2019.2948809

Winslow, R. L., Rice, J., Jafri, S., Marbán, E., and O'Rourke, B. (1999). Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, ii. *Circ. Res.* 84, 571–586. doi: 10.1161/01.RES.84.5.571

Woo, S. S., Kim, J., and Sarpeshkar, R. (2018). A digitally programmable cytomorphic chip for simulation of arbitrary biochemical reaction networks. *IEEE Trans. Biomed. Circ. Syst.* 12, 360–378. doi: 10.1109/TBCAS.2017.2781253

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Wagner, Bartol, Sejnowski and Cauwenberghs. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership