



OPEN ACCESS

EDITED BY

Claudio Castellano,
Istituto dei Sistemi Complessi (ISC-CNR), Italy

REVIEWED BY

Gabriele Di Antonio,
Santa Lucia Foundation, Italy
Tobias Kühn,
University of Bern, Switzerland

*CORRESPONDENCE

Patrick Krauss,
patrick.krauss@fau.de

RECEIVED 27 May 2025

REVISED 28 November 2025

ACCEPTED 30 November 2025

PUBLISHED 05 January 2026

CITATION

Metzner C, Schilling A, Maier A and Krauss P (2026) Organizational regularities in recurrent neural networks.

Front. Complex Syst. 3:1636222.

doi: 10.3389/fcpxs.2025.1636222

COPYRIGHT

© 2026 Metzner, Schilling, Maier and Krauss. 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.

Organizational regularities in recurrent neural networks

Claus Metzner, Achim Schilling, Andreas Maier and
Patrick Krauss*

Cognitive Computational Neuroscience Group, Pattern Recognition Lab, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Erlangen, Germany

Previous work has shown that the dynamical regime of Recurrent Neural Networks (RNNs)—ranging from oscillatory to chaotic and fixed point behavior—can be controlled by the global distribution of weights in connection matrices with statistically independent elements. However, it remains unclear how network dynamics respond to organizational regularities in the weight matrix, as often observed in biological neural networks. Here, we investigate three such regularities: (1) monopolar output weights per neuron, in accordance with Dale's principle, (2) reciprocal symmetry between neuron pairs, as in Hopfield networks, and (3) modular structure, where strongly connected blocks are embedded in a background of weaker connectivity. These regularities are studied independently, but as functions of the RNN's general connection strength and its excitatory/inhibitory bias. For this purpose, we construct weight matrices in which the strength of each regularity can be continuously tuned via control parameters, and analyze how key dynamical signatures of the RNN evolve as a function of these parameters. Moreover, using the RNN for actual information processing in a reservoir computing framework, we study how each regularity affects performance. We find that Dale monopolarity and modularity significantly enhance task accuracy, while Hopfield reciprocity tends to reduce it by promoting early saturation, limiting reservoir flexibility.

KEYWORDS

dale's principle, hopfield network, modularity, recurrent neural networks, reservoir computing

1 Introduction

Over the past decades, deep learning has achieved remarkable progress (LeCun et al., 2015; Alzubaidi et al., 2021), notably through the rise of large language models (Min et al., 2023). These models are typically based on feedforward architectures, where information flows unidirectionally from input to output layers. In contrast, Recurrent Neural Networks (RNNs) include feedback connections, enabling them to function as autonomous dynamical systems (Maheswaranathan et al., 2019) that sustain neural activity even without ongoing external input.

RNNs exhibit certain “universal” properties—such as the ability to approximate arbitrary functions (Maximilian et al., 2006) or general dynamical systems (Aguiar et al., 2023)—which, alongside other strengths, have spurred interest in their fine-grained behavior. For example, they can preserve information from temporally extended input sequences (Jaeger, 2001; Schuecker et al., 2018; Büsing et al., 2010; Dambre et al., 2012; Wallace et al., 2013; Gonon and Ortega, 2021) and learn effective internal representations by balancing compression and expansion of information (Farrell et al., 2022).

A further key research theme concerns the control of RNN dynamics, including how internal and external noise shape network behavior (Rajan et al., 2010; Jaeger, 2014; Haviv et al., 2019; Lutz et al., 1992; Ikemoto et al., 2018; Krauss et al., 2019a; Bönsel et al., 2021; Metzner and Krauss, 2022). RNNs have also been proposed as models for neural computation in the brain (Barak, 2017). Notably, sparse RNNs—with a low average node degree, resembling biological circuits (Song et al., 2005)—have shown improved capacity for information storage (Brunel, 2016; Narang et al., 2017; Gerum et al., 2020; Folli et al., 2018).

In our earlier work, we systematically explored the interplay between network structure and dynamics, beginning with three-neuron motifs (Krauss et al., 2019b). We later showed how the weight distribution's width w , the connection density d , and the balance b between excitation and inhibition can be tuned to control the dynamics of large, autonomously active RNNs (Krauss et al., 2019c; Metzner and Krauss, 2022). We also investigated noise-induced resonance phenomena in such systems (Bönsel et al., 2021; Schilling et al., 2022; Krauss et al., 2016; Krauss et al., 2019a; Schilling et al., 2021; Schilling et al., 2023; Metzner et al., 2024).

Most studies—including our own—have assumed statistically independent weight matrix elements, drawn from fixed distributions and assigned randomly. While analytically convenient, this assumption does not reflect the structural regularities seen in biological networks. Real neural systems exhibit highly non-random connectivity, shaped by development, functional demands, and evolutionary constraints.

The potential benefits of incorporating structural regularities into neural architectures have increasingly been recognized in the literature. Several recent developments have explored this direction, aiming to move beyond fully random or uniform connectivity. For example, Capsule Networks introduced by Hinton and colleagues (Sabour et al., 2017; Hinton et al., 2018) implement local groups of neurons—capsules—that preserve part-whole relationships via structured routing mechanisms. In transformer models, architectural variants have been proposed that introduce explicit modularity or routing constraints to enhance interpretability and scalability (Rosenbaum et al., 2018; Shazeer et al., 2017; Rosenbaum et al., 2019). Moreover, a number of studies have examined recurrent networks with biologically inspired topology, exploring the impact of modularity, reciprocity, or Dale-like constraints on network dynamics and learning performance (Zador, 2019; Cornford et al., 2020; Rodriguez et al., 2019).

Recent theoretical work has further deepened our understanding of how structured connectivity influences recurrent network dynamics. For instance, studies have examined how the number of effective degrees of freedom and the resulting low-dimensional organization of neural trajectories depend on architectural constraints and coupling statistics (Hwang et al., 2019; Hwang et al., 2020). Other analyses have characterized dynamical regimes in structured or partially symmetric networks, including glassy attractor states and transitions between ordered and chaotic phases (Berlemont and Mongillo, 2022; Fournier et al., 2025). Together, these works highlight that architectural regularities do not merely stabilize dynamics, but can fundamentally shape the computational landscape of recurrent systems.

These efforts highlight a growing consensus that structural features—long regarded as biological idiosyncrasies—may in fact

play a functional role in shaping the computational behavior of artificial neural systems. Against this background, we systematically examine the isolated effects of three such regularities:

First, biological neurons follow Dale's Principle, meaning each neuron is either excitatory or inhibitory, but not both (Strata and Harvey, 1999; Somogyi et al., 1998).

Second, neural circuits exhibit an increased likelihood of reciprocal connections: if neuron A projects to B, B is more likely to project back to A (Song et al., 2005; Perin et al., 2011). This bidirectional coupling introduces local symmetries that may stabilize attractor states and support mutual reinforcement.

Third, the brain is modular, comprising groups of neurons more densely connected within than between groups (Sporns and Betzel, 2016; Meunier et al., 2010). Such organization appears across scales, from cortical microcircuits to large-scale areas, and enables specialized yet integrative processing.

In the following, we study how each of these biologically inspired regularities affects the dynamical regime of RNNs, using a minimal implementation in which their respective strength can be continuously tuned (compare Figure 1).

The first regularity, called Dale Homogeneity, is controlled via a continuous parameter $h \in [0, 1]$, indicating the degree to which a neuron's outputs are consistently excitatory or inhibitory. At $h = 0$, polarities are mixed; at $h = 1$, all neurons are strictly monopolar.

The second regularity, Hopfield Reciprocity, is controlled by a parameter $r \in [0, 1]$. For $r = 0$, weights w_{ij} and w_{ji} are independent; for $r = 1$, they are identical, as in Hopfield networks.

The third regularity is Modularity, parameterized by $m \in [0, 1]$. A network with $m = 0$ lacks modular structure, whereas $m > 0$ introduces strongly connected quadratic blocks with standard deviation $w_S > w$, embedded in a weaker background with $w_W < w$. As the degree of modularity m increases, w_S and w_W are adjusted so that the global width w remains constant.

While h , r , and m control the strength of structural regularities, we assess their impact on network dynamics using deterministic neurons with tanh activation and three dynamical measures: F , C , and N .

The fluctuation $F \in [0, 1]$ is the standard deviation of neuron outputs, averaged over neurons and time. It quantifies spontaneous activity; excessively large F would hinder reliable computation.

The covariance $C \in [-1, +1]$ is defined as the product $y_m(t) \cdot y_n(t+1)$, averaged over neuron pairs and time. In a globally oscillatory state, $C \approx -1$, in a chaotic state $C \approx 0$, and in a fixed point state $C \approx +1$.

Finally, the nonlinearity $N \in [-1, +1]$ is the average operating regime of neurons. At $N \approx -1$, neurons operate linearly; at $N \approx +1$, they are saturated and behave digitally.

While the primary focus of this study lies on how organizational regularities shape the intrinsic dynamics of RNNs, it is natural to ask whether these structural features also affect the network's information processing performance. To address this, we embed the RNN into a reservoir computing framework and examine how the accuracy A in simple benchmark tasks varies as a function of the regularity control parameters.

The following sections describe the construction of the weight matrices, the simulation setup for measuring dynamical indicators, and the test tasks used to evaluate computational performance.

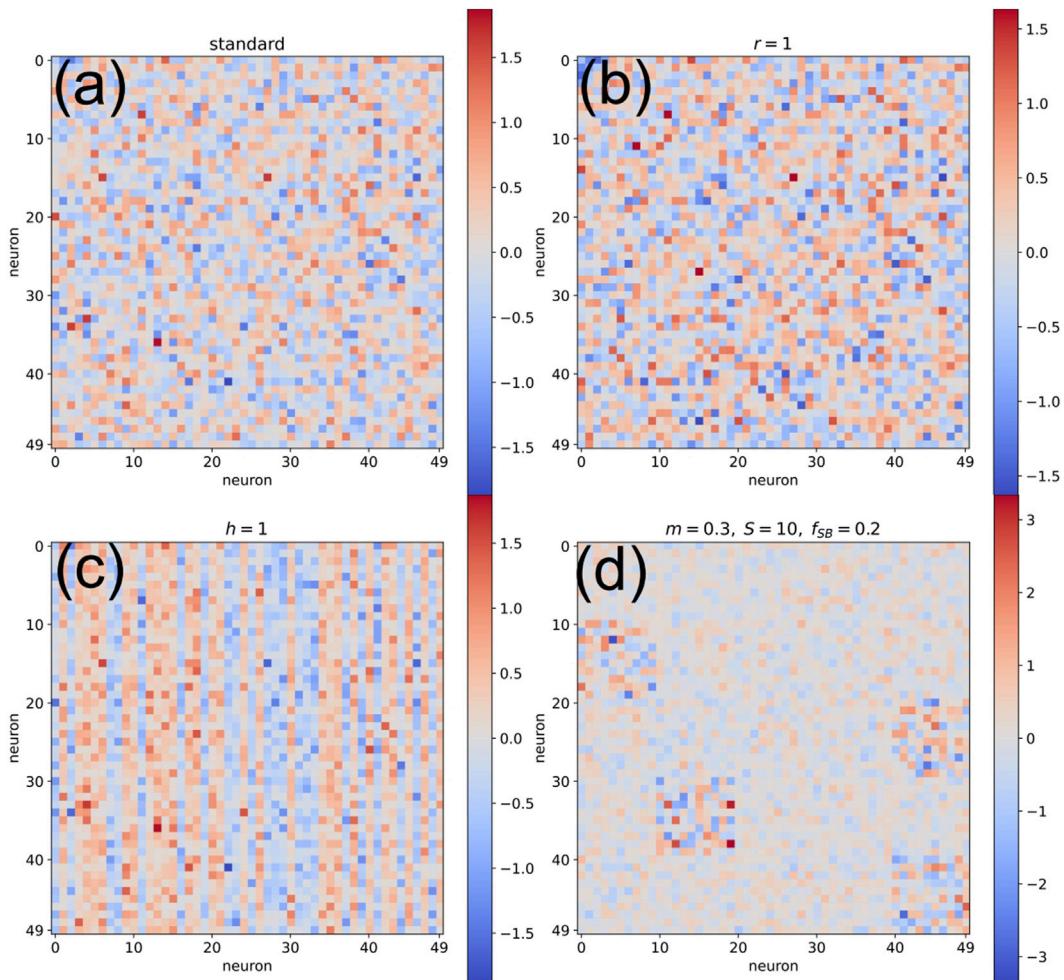


FIGURE 1

Example weight matrices. (a) A standard weight matrix of size 50×50 , with density $d = 1$, balance $b = 0$, width $w = 0.5$, and all organizational regularity parameters set to zero: $r = h = m = 0$. (b) With maximal Hopfield reciprocity $r = 1$ and all other parameters unchanged, the matrix becomes symmetric about the diagonal, while preserving density, balance, and width. (c) With maximal Dale homogeneity $h = 1$, each matrix column (representing a neuron's output weights) adopts a uniform sign. (d) With modularity $m = 0.3$, block size $S = 10$, and probability $f_{SB} = 0.2$, some blocks exhibit significantly increased fluctuation width, while others become weaker. Note the different color bar in (d).

2 Methods

2.1 General simulation setup

The overall workflow of our investigation consists of the following steps:

First, we set the control parameters, including the distribution parameters—width w , density d , and excitatory-inhibitory balance b —as well as the regularity parameters: Dale homogeneity h , Hopfield reciprocity r , and modularity m .

Second, we generate a set of N_R random square weight matrices that satisfy these specifications. Their properties are verified by empirically evaluating their statistical features.

Third, for each matrix, we simulate the spontaneous dynamics of the corresponding RNN. The network is initialized in a random state and then run freely for a large number of time steps. We refer to the resulting time series of neural activations as the output stream.

Fourth, we compute the dynamical measures—fluctuation F , covariance C , and nonlinearity N —for each output stream, resulting

in N_R samples per measure. For visualization, we average these values across the ensemble of N_R networks.

In addition to analyzing intrinsic dynamics, we also examine how the regularity parameters affect information processing. For this, the RNN is embedded into a reservoir computing framework: it receives input via a fixed input matrix, and its output stream is passed to a trainable readout layer. However, in the first part of the study, we set the input matrix to zero, so the RNN runs autonomously without external input, and the readout is not used. Nonetheless, we describe the full reservoir architecture in the following for completeness.

2.2 Design of reservoir computer (RC)

The RC consists of an input layer, a recurrent reservoir, and a readout layer. The input data comprises E consecutive episodes, each corresponding, for example, to a pattern to be classified. This

data stream is fed into the reservoir and circulates through the system while propagating toward the output.

At each time step t , the input layer receives M parallel signals $x_m^{(t)} \in [-1, +1]$. These are linearly transformed by the input matrix \mathbf{I} of size $N \times M$ and injected into the reservoir, as described by [Equation 1](#). Each input episode spans T time steps.

The input layer consists solely of the matrix \mathbf{I} and is therefore purely linear. Its elements I_{mn} are drawn independently from a normal distribution with zero mean and standard deviation w_I . To study autonomous RNN dynamics, we set $w_I = 0$, effectively decoupling the reservoir from external input.

The reservoir comprises N recurrently connected neurons with \tanh activation. At each time step, all neuron states y_n are updated in parallel. Each neuron receives a bias term $b_{w,n}$, input from the external signals via \mathbf{I} , and recurrent input via the weight matrix \mathbf{W} (see [Equation 1](#)). Initial states $y_n^{(0)}$ are drawn uniformly from $[-1, +1]$ and kept fixed for repeated simulations of the same reservoir. Different weight matrices receive independent initial states.

The readout layer performs an affine-linear transformation of the reservoir states y_n using a $K \times N$ output matrix \mathbf{O} and a bias vector \mathbf{b}_o , as in [Equation 2](#). These parameters are trained via the pseudoinverse method (see below).

In the sequence generation task, which serves as the main information processing benchmark in this work, the continuous outputs z_k of the readout layer directly form the vectors of the output sequence. In classification tasks, by contrast, the z_k represent soft votes that are converted into discrete predicted class labels c using the argmax function (see [Equation 3](#)). This final step introduces a nonlinearity that sharpens the class boundaries in the output space.

In summary, the RC is governed by the following equations.

$$y_n^{(t)} = \tanh \left(b_{w,n} + \sum_m I_{mn} x_m^{(t-1)} + \sum_{n'} W_{mn'} y_{n'}^{(t-1)} \right) \quad (1)$$

$$z_k^{(t)} = b_{o,k} + \sum_n O_{kn} y_n^{(t)} \quad (2)$$

$$c^{(t)} = \arg \max \{ z_k^{(t)} \} \quad (3)$$

2.3 Sequence generation task

In this task, the reservoir computer functions as a deterministic system that maps an input sequence X of real-valued vectors onto a corresponding output sequence Z :

$$X \in [-1, +1]^{TI \times M} \rightarrow Z \in [-1, +1]^{TO \times K}$$

Here, TI and TO denote the temporal lengths of the input and output sequences, and M and K their respective vector dimensions.

At the beginning of each episode, a randomly chosen input sequence X from one of N_{DC} discrete classes is fed into the network, driving it into a class-specific internal configuration or ‘priming state’. After the input ends, the reservoir evolves autonomously through a sequence of internal states Y , which the linear readout layer is trained to map onto the corresponding target sequence Z .

Since the system is strictly deterministic, it will always produce the same trajectory for each distinct priming state. Provided that the

induced trajectory is not a cyclic attractor with a period shorter than TO , the readout should in principle be able to convert the trajectory into the desired output sequence.

However, because the reservoir is not reset at the beginning of each episode, residues of previous states may persist, such that the priming state is not exactly identical every time a given class is selected. If the reservoir operates in a chaotic regime, these small differences can be amplified, producing an effectively unpredictable trajectory that cannot be mapped to the correct target.

In practice, the target mapping also fails if the reservoir neurons enter the saturated ‘digital’ state, since the resulting trajectory is insufficiently rich and lacks the necessary high-dimensional diversity.

Successful performance therefore requires a balance between stability and richness: the reservoir must forget prior excitation rapidly enough to respond reproducibly to identical inputs, while still maintaining sufficient temporal and spatial diversity across neurons.

To systematically explore the influence of network parameters under controlled conditions, we employ a minimal task configuration with $M = K = 2$, $TI = 1$, $TO = 2$, and $N_{DC} = 2$. The class-specific input sequences X_c and their target sequences Z_c are predefined, with all vector components drawn independently from a uniform distribution over $[-1, +1]$. In each episode, one of the N_{DC} cases is selected at random.

2.4 Optimal readout layer using pseudoinverse

The optimal weights and biases of the readout layer can be efficiently computed with the method of the pseudoinverse, based on the sequence of reservoir states and the target output (Compare, for example, [Section 3.4](#). in ([Cucchi et al., 2022](#))). Following these ideas, we proceed as follows:

Let $Y \in \mathbb{R}^{(E-1) \times N}$ be the matrix of reservoir states directly after each input episode, where E is the total number of episodes and N is the number of reservoir neurons. Let $Z \in \mathbb{R}^{(E-1) \times K}$ be the matrix of target output states, where K is the number of output units.

To account for biases in the readout layer, a column of ones is appended to Y , resulting in the matrix $Y_{bias} \in \mathbb{R}^{(E-1) \times (N+1)}$:

$$Y_{bias} = [Y \ 1_{E-1}]$$

where $1_{E-1} \in \mathbb{R}^{(E-1) \times 1}$ is a column vector of ones.

The weights and biases of the readout layer are computed by solving the following equation using the pseudoinverse of Y_{bias} :

$$W_{bias} = Y_{bias}^+ Z \quad (4)$$

where Y_{bias}^+ is the Moore-Penrose pseudoinverse of Y_{bias} , and $W_{bias} \in \mathbb{R}^{(N+1) \times K}$ contains both the readout weights and the biases.

To compute the pseudoinverse, we first perform a singular value decomposition (SVD) of Y_{bias} :

$$Y_{bias} = USV^T$$

where $U \in \mathbb{R}^{(E-1) \times (E-1)}$ is a unitary matrix, $S \in \mathbb{R}^{(E-1) \times (N+1)}$ is a diagonal matrix containing the singular values, and $V^T \in \mathbb{R}^{(N+1) \times (N+1)}$ is the transpose of a unitary matrix.

The pseudoinverse of Y_{bias} is computed as:

$$Y_{\text{bias}}^+ = VS^+U^\top$$

where $S^+ \in \mathbb{R}^{(N+1) \times (E-1)}$ is the pseudoinverse of the diagonal matrix S . The pseudoinverse S^+ is obtained by taking the reciprocal of all non-zero singular values in S and leaving zeros unchanged.

Finally, after inserting Y_{bias}^+ into [Equation 4](#), the optimal readout weights $W \in \mathbb{R}^{K \times N}$ and biases $b \in \mathbb{R}^K$ are extracted from the extended matrix W_{bias} as

$$\begin{aligned} W &= (W_{\text{bias}}^\top)_{1:N}, \quad : \\ b_w &= (W_{\text{bias}}^\top)_{N+1}, \quad : \end{aligned}$$

where the first N rows of W_{bias}^\top define the readout weights and the last row defines the biases.

2.5 Generation of weight matrices with homogeneity, reciprocity, or modularity

The generation of weight matrices is based on the six control parameters defined in the Introduction. Depending on the selected values of homogeneity h , reciprocity r , and modularity m , different structural features are incorporated into otherwise random matrices.

We begin by generating a matrix of magnitudes m_{ij} , where each entry is drawn from a normal distribution with zero mean and standard deviation w , then made positive via $m_{ij} := |m_{ij}|$.

A binary mask matrix $n_{ij} \in \{0, 1\}$ is created, where each element is set to 1 with probability d . This defines the sparsity structure.

A sign matrix $s_{ij} \in \{-1, +1\}$ is generated, with +1 assigned with probability $(b+1)/2$ to control the excitatory-inhibitory balance.

The elementwise product of magnitude, mask, and sign matrices yields the pure weight matrix $W^{(\text{pure})} = m_{ij} \cdot n_{ij} \cdot s_{ij}$, which serves as the default base.

To implement homogeneity ($h > 0$, $r = 0$), we generate a Dale-conform matrix $W^{(\text{Dale})}$ in which each column (corresponding to one sending neuron) has a uniform sign. The final weight matrix W is then obtained by stochastic interpolation: $w_{ij} := w_{ij}^{(\text{Dale})}$ with probability h , and $w_{ij}^{(\text{pure})}$ with probability $1 - h$.

The Dale matrix uses the same magnitudes m_{ij} and mask n_{ij} as $W^{(\text{pure})}$, but assigns each column a fixed sign (either all +1 or all -1), again with $(b+1)/2$ controlling the excitatory fraction.

To implement reciprocity ($r > 0$, $h = 0$), we generate a symmetric Hopfield-like matrix $W^{(\text{Hopf})}$ by copying the upper triangle and diagonal of $W^{(\text{pure})}$ into the lower triangle. The final matrix is again obtained via stochastic interpolation: $w_{ij} := w_{ij}^{(\text{Hopf})}$ with probability r , and $w_{ij}^{(\text{pure})}$ otherwise.

To implement modularity ($m > 0$), we divide the $N \times N$ matrix into regular square blocks of size $S \times S$, grouping the N neurons into N/S modules. Each block is randomly assigned to the “strong” or “weak” class, with the probability of “strong” given by f_{SB} . The realized fractions f_s and $f_w = 1 - f_s$ are determined empirically after assignment.

Weak blocks are filled with Gaussian values of standard deviation $w \cdot q_w$, where $q_w = 1 - m$. Strong blocks use standard deviation $w \cdot q_s$, with

$$q_s = \sqrt{\frac{1 - f_w q_w^2}{f_s}} = \sqrt{\frac{1 - f_w (1 - m)^2}{1 - f_w}}$$

to ensure that the total matrix standard deviation remains w .

For $m = 0$, both scaling factors are equal to one; for $m = 1$, the weak blocks vanish and $q_s = \sqrt{1/f_s}$.

Except for the extreme case $m = 1$, where the entries in all weak blocks vanish, the resulting matrix is fully dense ($d = 1$). To impose the desired balance b , all elements are made positive and then flipped to negative with probability $(1 - b)/2$.

For clarity, we never apply homogeneity (h), reciprocity (r) and modularity (m) simultaneously.

2.6 Evaluation of weight matrices

In addition to the control parameters d, b, h, r , we define corresponding empirical parameters D, B, H, R that quantify the actual density, balance, homogeneity, and reciprocity of a given weight matrix W .

The empirical density D is computed as the fraction of non-zero elements:

$$D = \frac{n_{\text{nonzero}}}{n_{\text{total}}}, \quad \text{where } n_{\text{nonzero}} = \#\{w_{ij} \neq 0\}$$

The empirical balance B measures the ratio of excitatory to inhibitory weights:

$$B = \frac{n_{\text{pos}} - n_{\text{neg}}}{n_{\text{pos}} + n_{\text{neg}}}, \quad \text{with } n_{\text{pos}} = \#\{w_{ij} > 0\}, \quad n_{\text{neg}} = \#\{w_{ij} < 0\}$$

The empirical homogeneity H is computed column-wise. For each column c , we calculate

$$H_c = \frac{|n_{\text{pos}} - n_{\text{neg}}|}{n_{\text{pos}} + n_{\text{neg}}}$$

where n_{pos} and n_{neg} refer to the number of positive and negative entries in column c . If all elements in the column are zero, we set $H_c := 0$. The global homogeneity is the mean over all columns:

$$H = \langle H_c \rangle_c$$

The empirical reciprocity R is computed from all off-diagonal pairs (i, j) in the upper triangle:

$$R_{ij} = 1 - \frac{|w_{ij} - w_{ji}|}{|w_{ij}| + |w_{ji}|}$$

If both w_{ij} and w_{ji} are zero, we define $R_{ij} := 1$. The matrix-level reciprocity is the average over all such pairs:

$$R = \langle R_{ij} \rangle_{i < j}$$

For each set of control parameters (d, b, h, r) , we generate an ensemble of weight matrices and compute the empirical parameters D, B, H, R as averages over the corresponding values from each individual matrix.

2.7 Fluctuation measure

The neural fluctuation measure F quantifies the average temporal variability of reservoir activations. For each neuron n , we compute the standard deviation σ_n of its activation time series $y_n^{(t)}$. The global fluctuation is defined as the mean over the standard deviations of the individual neurons:

$$F = \langle \sigma_n \rangle_n$$

Since tanh-neurons produce outputs in $[-1, +1]$, the fluctuation F lies in $[0, 1]$. A value of $F = 0$ indicates a resting or fixed point state, while $F = 1$ corresponds to perfect two-state oscillation (e.g., alternating between $+1$ and -1).

2.8 Covariance measure

To assess temporal covariances, we compute the average product of the activation of neuron m at time t and neuron n at time $t + 1$:

$$C_{mn} = \langle y_m^{(t)} \cdot y_n^{(t+1)} \rangle_t$$

Unlike the Pearson correlation coefficient, we deliberately avoid subtracting the mean or normalizing by the standard deviations. This ensures that the matrix elements C_{mn} remain well-defined even when one or both signals are constant, as in a fixed point state.

The global covariance measure is defined as the average over all neuron pairs, without differentiating between diagonal and off-diagonal elements:

$$C = \langle C_{mn} \rangle_{mn}$$

Owing to the bounded output of the tanh neurons, the covariance values C always lie within the range $[-1, +1]$.

2.9 Nonlinearity measure

The shape of the activation distribution $p(y)$ reflects whether the reservoir operates in a linear or nonlinear regime. A central peak at $y = 0$ indicates a linear regime; two peaks near ± 1 indicate saturation and thus nonlinearity.

We define a nonlinearity measure

$$N = f_A - f_B + f_C$$

based on the fractions of neural activations falling into the following intervals:

$$\begin{aligned} f_A &\in [-1, -0.5] \\ f_B &\in [-0.5, +0.5] \\ f_C &\in (+0.5, +1] \end{aligned}$$

The resulting measure $N \in [-1, +1]$ distinguishes three regimes: $N \approx -1$ for linear operation, $N \approx 0$ for intermediate or flat activation, and $N \approx +1$ for saturated, digital-like behavior.

This intuitive yet robust definition proved most effective among several tested alternatives. It captures the essential qualitative transition in $p(y)$ from unimodal (linear) to bimodal (nonlinear) distributions, as highlighted in earlier studies.

2.10 Accuracy measure

In the sequence generation task, we evaluate performance by comparing the actual output sequences Z_{act} of the readout layer with the corresponding target sequences Z_{tar} , and compute the root-mean-square error E_{RMS} . This error is normalized by the standard deviation ΔZ_{tar} of the target sequences.

To obtain an accuracy measure $A \in [0, 1]$, we define:

$$A = \frac{1}{1 + (E_{\text{RMS}} / \Delta Z_{\text{tar}})}$$

Note that $A \approx 0.5$ when the RMS error is comparable to the variability of the target data, and $A = 1$ when the output Z_{act} matches the target Z_{tar} exactly.

In classification tasks, the accuracy A is simply defined as the fraction of correctly predicted class labels.

3 Results

3.1 Validation of control parameters

As described in the Methods section, we generate weight matrices with prescribed values for connection density d , excitatory/inhibitory balance b , Hopfield reciprocity r , Dale homogeneity h , and modularity m . To verify that the generation procedures operate as intended, we compute the corresponding empirical parameters directly from the generated matrices. Throughout this paper, lowercase letters d, b, r, h, m denote the prescribed control parameters, while uppercase letters D, B, R, H refer to their empirically measured counterparts. Validation of the modular structure—additionally dependent on block size S and the fraction f_{SB} of ‘strong’ blocks—is addressed separately (see below).

For validation, we first fix all control parameters d, b, r, h, m to standard values. Then, one parameter x is varied across its entire admissible range while the others remain fixed. During this one-dimensional scan, we compute all empirical quantities D, B, R, H as functions of x (Figures 2a–e). Ideally, each control parameter x should primarily affect its corresponding empirical statistic X , without significantly altering the others. However, certain interdependencies are inevitable due to shared structural properties.

As shown in panel (a), the prescribed density d directly controls the empirical density D , with negligible effect on the balance B . However, it also influences reciprocity R and homogeneity H to a minor extent.

Similarly, the prescribed balance b determines the empirical balance B , while leaving D unaffected (panel b). Nonetheless, R and H increase as the system becomes more unbalanced. Even for perfectly balanced matrices at $b = 0$, there exists a minimal, unavoidable level of reciprocity and homogeneity due to the Gaussian weight distribution.

Varying the prescribed Hopfield reciprocity r results in a nearly linear increase of the empirical reciprocity R , while the other measures remain unaffected (panel c).

Increasing the prescribed Dale homogeneity h leads to a monotonic rise in empirical homogeneity H , along with a slight

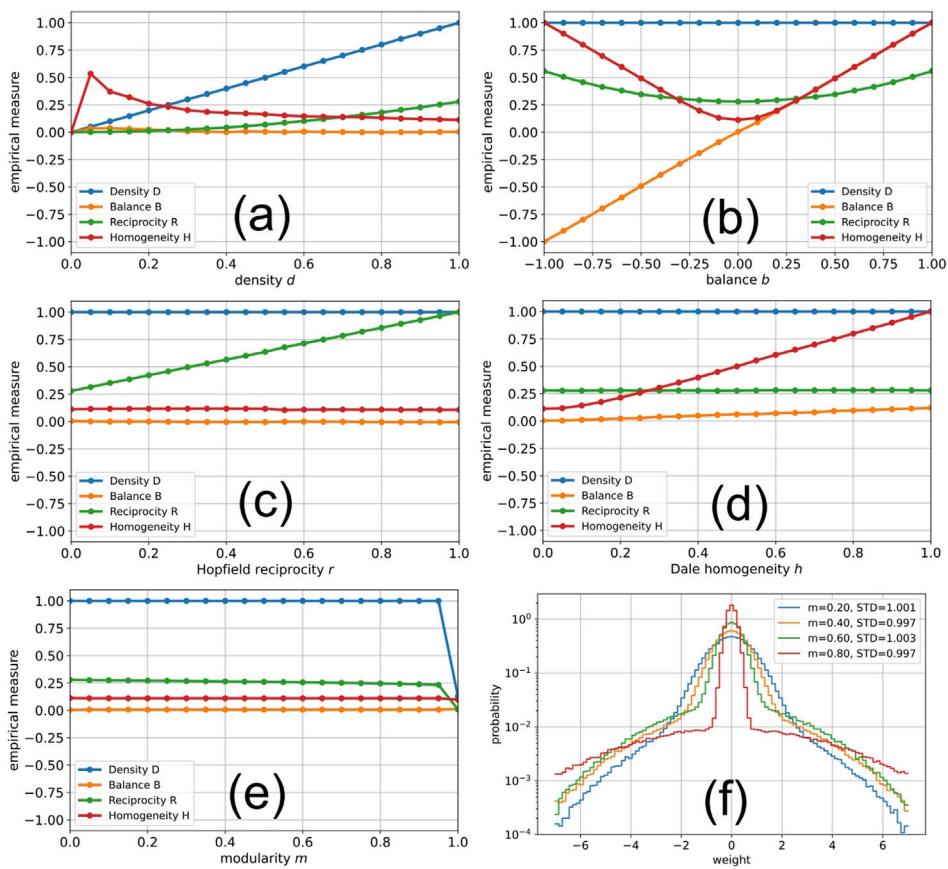


FIGURE 2

Prescribed and empirical control parameters. We use a weight matrix of size 50×50 with parameters initially set to standard values $d = 1$ and $b = r = h = m = 0$. In each of the plots (a–e), one control parameter x is scanned through its full permissible range, while all others remain at their standard values. The empirical measures D, B, R, H are evaluated as a function of the scanned parameter x . (a) Scan of the density d . (b) Scan of the balance b . (c) Scan of the reciprocity r . (d) Scan of the homogeneity h . (e) Scan of the modularity m . (f) Probability distributions of weight matrix elements for different degrees of modularity m , using a 1000×1000 matrix with $w = d = 1$ and $b = 0$. Block size was $S = 100$, with a fraction of strong blocks $f_{SB} = 0.1$. The legend shows that the standard deviations of the distributions remain constant.

increase in balance B . Reciprocity R and density D remain essentially unchanged (panel d).

Finally, the prescribed modularity m has virtually no effect on any of the empirical measures D, B, R, H , except at the extreme value $m = 1$ (panel e).

To validate the modularity construction, we consider a 1000×1000 weight matrix with parameters $w = d = 1$ and $b = 0$. The block size is set to $S = 100$, and the fraction of strong blocks to $f_{SB} = 0.1$. We then gradually increase the modularity parameter m from 0.2 to 0.8, and compute the histogram of matrix elements for each value (panel f).

A semi-logarithmic plot reveals that the resulting distributions are mixtures of two Gaussians with distinct standard deviations. As expected, the mixture preserves the global distribution width (STD), which is explicitly shown in the legend of panel (f).

Note that the weight distribution of the connectivity matrix becomes a true Gaussian mixture only for intermediate modularity parameters $0 < m < 1$. In the one extreme case $m = 0$, the distributions of the weak and strong blocks have identical Gaussian widths, and thus no block structure exists. As m increases, the distribution width of the weak blocks gradually

decreases relative to that of the strong blocks. In the opposite extreme case $m = 1$, the weak blocks have zero distribution width, which effectively means that these connections are no longer present.

3.2 RNN phase diagrams

In this section, we examine an RNN consisting of 50 neurons, each randomly connected to all others with a full connection density of $d = 1$. We analyze how the network's dynamical and information-processing properties vary as a function of the excitatory–inhibitory balance b and the width w of the Gaussian distribution of connection strengths. The results are presented as “phase diagrams”, where selected quantities are color-coded in the b – w plane (see Figure 4).

The fluctuation measure $F \in [0, 1]$ quantifies the mean amplitude of the temporal variations in neural activation.

The nonlinearity $N \in [-1, 1]$ indicates whether neurons operate predominantly in the linear regime of the sigmoidal transfer function ($N \approx -1$) or in the saturated nonlinear regime ($N \approx +1$).

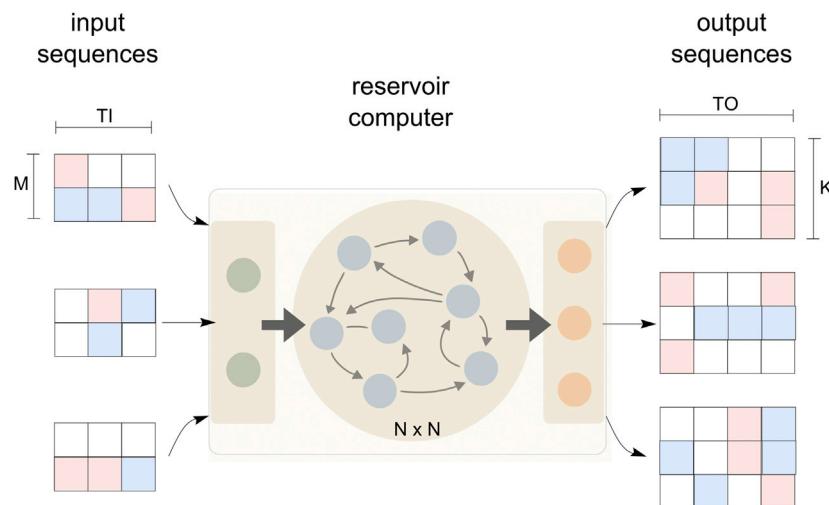


FIGURE 3

Reservoir Computer (RC) and Sequence Generation Task. The RC is treated as a trainable mapping between sequences of real-valued vectors. In the model task, all input sequences belong to N_{DC} distinct classes. Center: The reservoir computer consists of a random input matrix (green nodes), a random recurrent network (blue nodes), and a trained output matrix (orange nodes). Left: Three example input sequences, each comprising M parallel channels (color-coded) and a duration of T_I time steps. Right: The corresponding output sequences, each with width K and duration T_O , which the RC should generate in response to the respective inputs.

The covariance measure $C \in [-1, 1]$ is not a Pearson correlation coefficient, since we deliberately avoid subtracting the means and do not normalize by the variances of the signals. For our specific network of \tanh neurons, this definition of C leads to a smooth transition from $C \approx -1$ in a globally oscillatory regime, through $C \approx 0$ in a quiescent or irregularly fluctuating regime, to $C \approx +1$ in a global fixed point regime (compare Figure 8f). In combination with the fluctuation measure F , this allows us to identify the dominant dynamical regime of the reservoir, as shown previously (Figure 1 in (Metzner et al., 2025)).

The accuracy $A \in [0, 1]$ quantifies the performance of the RNN when used as the recurrent core of a reservoir computer (RC). Although the result naturally depends on the specific task, we restrict our analysis here to a single example (Figure 3), in which the RC is driven by different input sequences and required to generate predefined target output sequences. This task is particularly sensitive to the dynamical regime of the RNN, since the state trajectory must remain input-controlled and sufficiently regular over multiple updates, without entering the saturation regime or being dominated by spontaneous irregular fluctuations.

3.2.1 Free-running RNN

In the free-running RNN (upper row in Figure 4), we identify four characteristic dynamical regimes, most clearly visible in the nonlinearity phase diagram $N(b, w)$:

In the lower central part of the phase plane lies the quiescent region QR, roughly shaped like ‘Mt Fuji’. In this region, at $w \approx 0$, the neurons are virtually unconnected. Therefore all activations remain at very small values, determined only by the individual biases. As indicated by the nonlinearity measure $N \approx -1$, the neurons operate here in the linear regime, near the center of the \tanh activation function. The temporal constancy of the activations is reflected in the fluctuation measure $F \approx 0$. While the Pearson correlation coefficient would diverge for constant zero signals, our non-normalized covariance measure simply yields $C \approx 0$ in this case.

In the right wing of the phase plane lies the fixed point regime FR. Also marked by temporally constant activations, it exhibits fluctuations $F \approx 0$. However, due to strong coupling strengths ($w > 0$) and predominantly excitatory weights ($b > 0$), a positive feedback loop drives the network into high-activation global fixed points, where each neuron becomes trapped in either the positive or negative saturation of the activation function. As a result, the nonlinearity measure $N \approx +1$ indicates that the neurons now operate in a digital regime, and the covariance measure yields $C \approx +1$, showing that the neurons retain the same digital value over time.

In the left wing of the phase plane lies the oscillatory regime OR. Here, strong ($w > 0$), predominantly inhibitory ($b < 0$) coupling leads to global periodic flips of all neurons between the two saturation states. Consequently, the covariance $C \approx -1$ reflects activation values of opposite sign between successive time steps. The fluctuation $F \approx 1$ indicates high-amplitude temporal variation, and the nonlinearity $N \approx +1$ shows that the neurons operate digitally.

In the upper central part of the phase plane lies the chaotic regime CR. The dynamics here are also driven by strong mutual couplings ($w > 0$), but now the approximate balance between excitatory and inhibitory connections ($b \approx 0$) leads to much more complex and irregular behavior—both over time and across neurons. This is reflected in vanishing covariances $C \approx 0$. The neurons operate in a mixed linear, intermediate, or digital regime, such that the nonlinearity measure N lies somewhere between -1 and $+1$. The corresponding temporal fluctuations F , on average, are smaller than in the oscillatory regime.

3.2.2 RNN with input signals

Next, while the RNN is continuously updating, we feed in two time-dependent input signals related to a computational task. For this purpose, we use a dense 50×2 input matrix \mathbf{I} , so that every

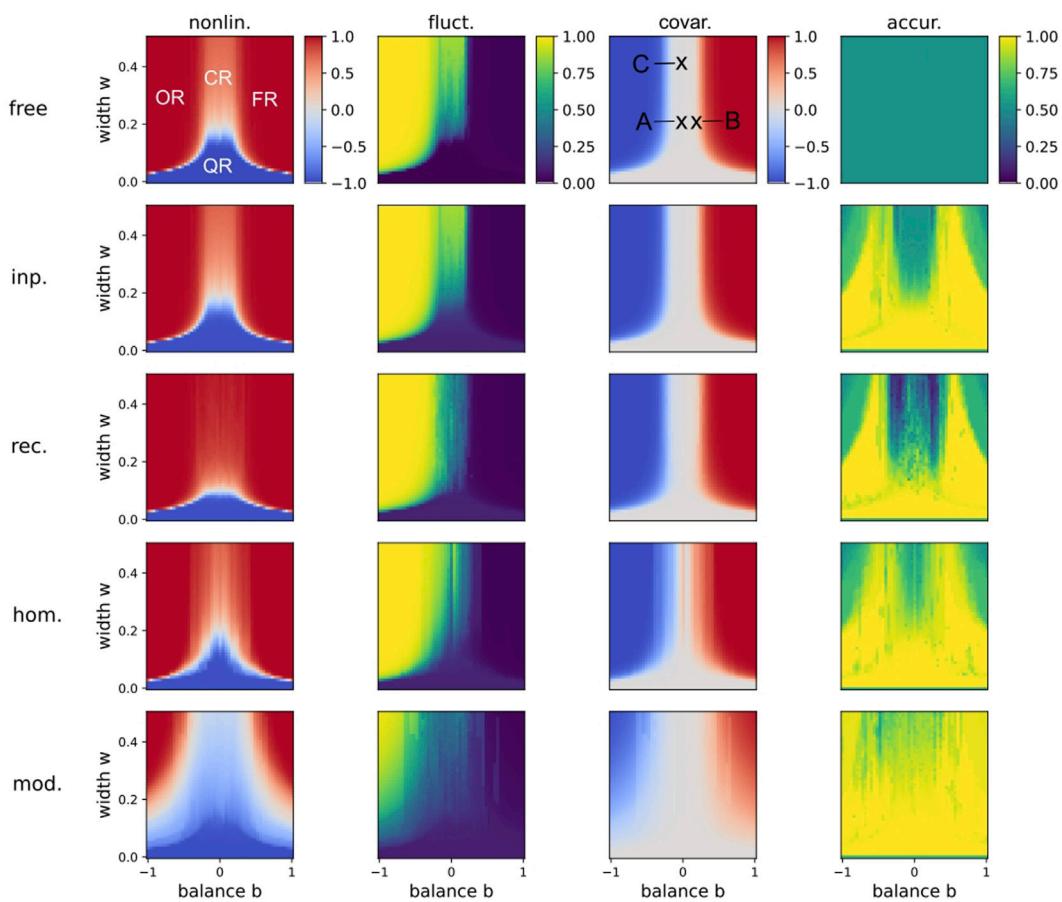


FIGURE 4

Phase diagrams of RNN dynamics and computational performance, as functions of the balance b and the width w . In all cases, the RNN consists of 50 tanh-neurons, fully connected (density $d = 1$). The four columns of phase diagrams, from left to right, correspond to the nonlinearity N , the fluctuation F , the covariance C , and the accuracy A in a sequence generation task (For details see Methods and Results). The top row corresponds to a free-running (no input) RNN, with all regularity parameters set to zero. All plots below correspond to RNNs driven by inputs and used for computations. Second row: All regularity parameters set to zero. Third row: Hopfield reciprocity parameter set to $r = 0.9$. Fourth row: Dale homogeneity parameter set to $h = 0.9$. Fifth row: modularity parameter set to $m = 0.9$. Note that the phase diagram of the nonlinearity parameter in the free-running system (upper left) shows four main dynamical regions characterized by quiescence (QR, blue dome at the bottom), chaos (CR, pale red stripe around the upper center), oscillations (OR, red flank at the left side) and fixed points (FR, red flank at the right side). Turning on the regularity parameters r , h and m has a clear effect on the dynamical variables N , F , C , as well on the accuracy A . In the third panel of the top row, three specific points in the b - w phase plane have been marked with crosses. These parameter combinations (later called phase points A,B,C) are investigated in more detail in Figures 5, 6.

neuron receives both inputs. The elements of \mathbf{I} are drawn from a Gaussian distribution with zero mean and a standard deviation of 0.3. The two input signals range between -1 and $+1$, and all further details are described in the Methods section.

We find that the injection of inputs has only a very weak effect on the phase diagrams of N , F , and C (see second row in Figure 4). Only the fluctuation level F in the quiescent regime QR is slightly elevated compared to the input-free case, which is to be expected.

We also add a readout layer, optimized by the method of the pseudo-inverse, which transforms the global time-dependent states of the RNN into output signals. In our case, the readout layer is optimized for a sequence generation task (see Methods section for details), and the performance of the resulting reservoir computer is measured by an accuracy value A that ranges between zero and one (right-most phase diagram in row 2).

We find that the accuracy remains close to one throughout the entire quiescent regime QR. As shown in a previous publication, this

regime is well-suited for many types of tasks, as the RNN states are then primarily determined by the input, not by spontaneous internal dynamics.

The accuracy drops considerably within the chaotic regime CR, where the irregular and unpredictable dynamics of the RNN interfere with the execution of the computational task.

A strong reduction in accuracy is also observed in the upper and outer parts of the oscillatory (OR) and fixed point (FR) regimes. There, the autonomous RNN dynamics are predictable, but the neurons are driven so deeply into the saturation regime that task-related computations can no longer be carried out.

Remarkably, the accuracy remains close to one in the narrow regions between the chaotic regime CR and the neighboring oscillatory OR or fixed point FR regimes. These two ‘edges of chaos’ thus prove suitable for task-related computation; however, they become increasingly narrow as the recurrent coupling strength w is increased.

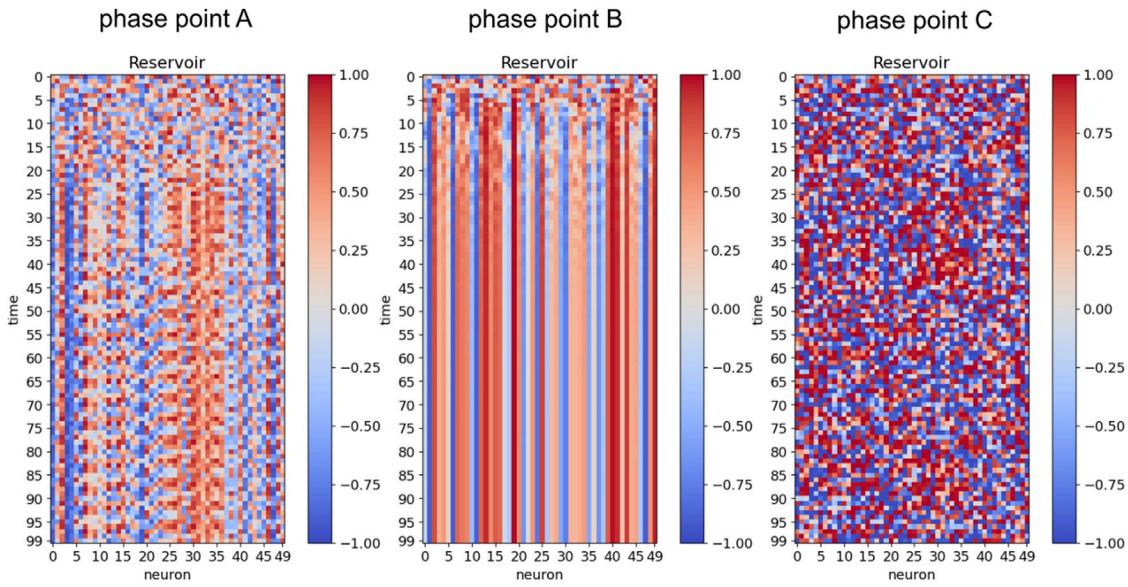


FIGURE 5

Neural activations in three selected points of phase space. Color-coded activation levels of all 50 neurons (horizontal) in the free-running RNN without input or reset at the beginning of each episode, shown as a function of time step (vertical). The three plots correspond to the selected points in the b - w phase plane, marked by crosses in the third panel of the top row in Figure 4. Phase point A (at $w = 0.2$, $b = 0$) represents a balanced network located near the transition between quiescent and chaotic dynamics. After a short transient, the system settles into a periodic attractor. Phase point B (at $w = 0.2$, $b = 0.2$) shares the same moderate connection width as point A but lies close to the border of the fixed point regime. Here, the system converges to a fixed point attractor where each neuron remains frozen at an individual activation level. Phase point C (at $w = 0.4$, $b = 0$) is again balanced but located deeper within the chaotic regime, showing irregular fluctuations of large amplitude.

3.2.3 Effect of strong hopfield reciprocity

Next, we leave the reservoir computer unchanged, with the only modification being the introduction of a relatively strong degree of Hopfield reciprocity, $r = 0.9$, into the RNN's weight matrix (see third row in Figure 4).

Comparing the nonlinearity phase diagram at $r = 0.9$ with the corresponding diagram at $r = 0$, we observe a significant shrinking of the quiescent regime QR and an increase in nonlinearity within the chaotic regime CR.

No strong differences are observed between the fluctuation phase diagrams $F(b, w | r = 0)$ and $F(b, w | r = 0.9)$ or between the covariance phase diagrams $C(b, w | r = 0)$ and $C(b, w | r = 0.9)$.

Finally, the accuracy phase diagram $A(b, w | r = 0.9)$ shows that performance in the chaotic regime has dropped to levels even below those of the $r = 0$ system. This suggests that Hopfield reciprocity can, at least in certain computational tasks, have a detrimental effect on the performance of a reservoir computer.

3.2.4 Effect of strong dale homogeneity

Starting again from the standard case without any structural regularities ($r = h = m = 0$), we now set the Dale homogeneity parameter to $h = 0.9$ and recompute the four phase diagrams (see fourth row in Figure 4).

The diagrams for nonlinearity, fluctuation, and covariance at $h = 0.9$ resemble those observed at $r = 0.9$ more closely than the standard case.

However, the accuracy diagram $A(b, w | h = 0.9)$ shows that Dale homogeneity improves task performance even beyond the

standard case $A(b, w | r = h = m = 0)$. In particular, within the chaotic regime CR, the reservoir computer can now tolerate significantly higher coupling strengths w .

3.2.5 Effect of strong modularity

Finally, we set the modularity parameter to $m = 0.9$, using block sizes of $SNB = 10$ neurons and a strong block fraction of $f_{SB} = 0.1$ (see Methods for details). Modularity exerts a pronounced influence on all phase diagrams (see fifth row in Figure 4):

The nonlinearity and covariance diagrams indicate that the oscillatory and fixed point regimes now occupy only a narrow region of phase space, restricted to highly unbalanced weights $|b| \approx 1$ and strong coupling $w > 0.25$.

Meanwhile, the previously chaotic regime is replaced by a broad region in which both nonlinearity and fluctuations remain moderate, while low covariance values still suggest irregular (non-repetitive) dynamics.

Most strikingly, the accuracy now reaches very high levels across almost the entire phase diagram. This indicates that modularity renders even formerly unproductive regimes—oscillatory, fixed point, and chaotic—computationally useful.

3.3 Effect of gradually increasing regularity

We now examine how a gradual increase of the organizational regularity parameters r, h, m affects the dynamical quantities N, F, C and the computational performance A (Figure 6). All simulations

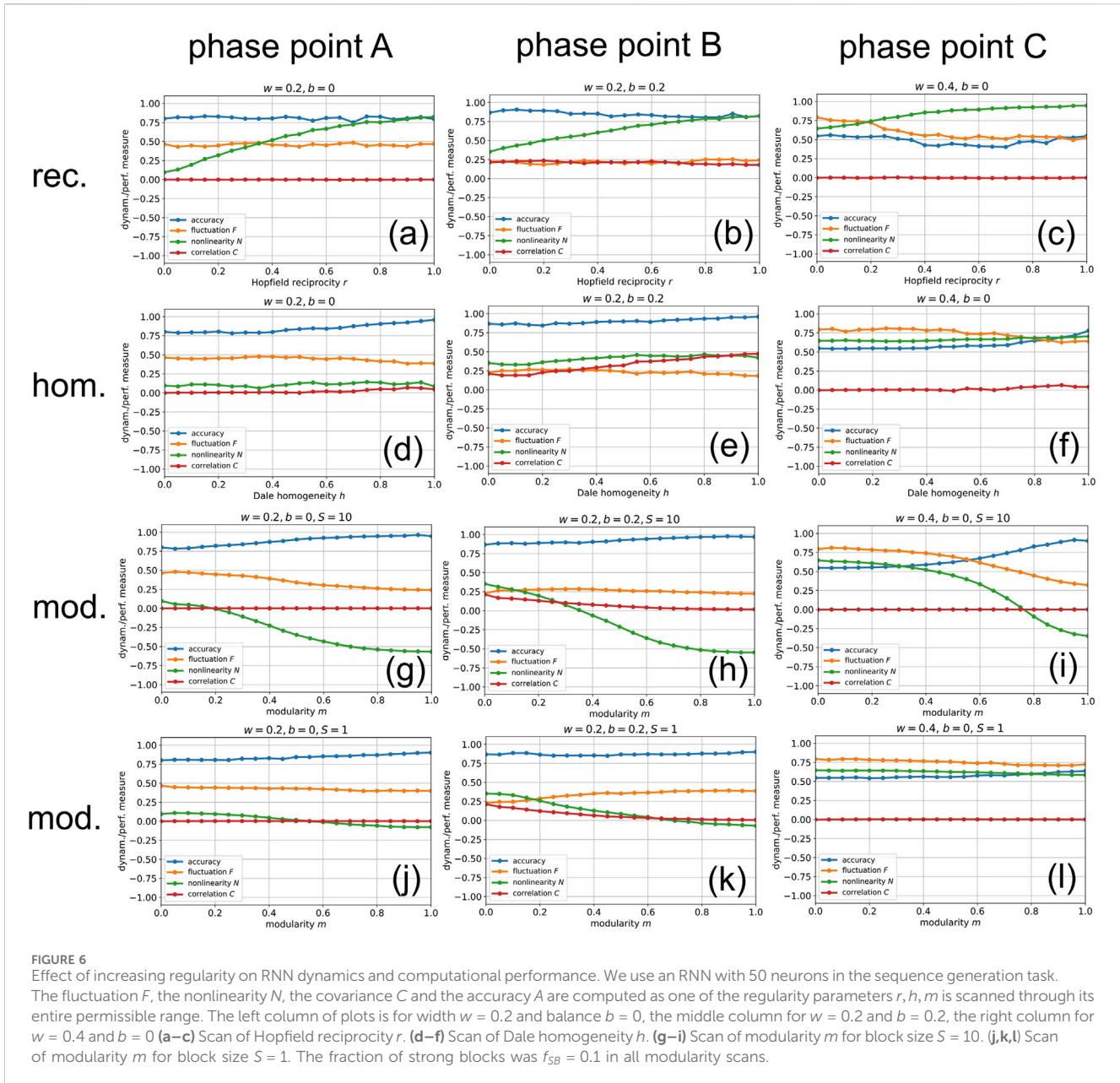


FIGURE 6

Effect of increasing regularity on RNN dynamics and computational performance. We use an RNN with 50 neurons in the sequence generation task. The fluctuation F , the nonlinearity N , the covariance C and the accuracy A are computed as one of the regularity parameters r, h, m is scanned through its entire permissible range. The left column of plots is for width $w = 0.2$ and balance $b = 0$, the middle column for $w = 0.2$ and $b = 0.2$, the right column for $w = 0.4$ and $b = 0$ (a–c) Scan of Hopfield reciprocity r . (d–f) Scan of Dale homogeneity h . (g–i) Scan of modularity m for block size $S = 10$. (j,k,l) Scan of modularity m for block size $S = 1$. The fraction of strong blocks was $f_{SB} = 0.1$ in all modularity scans.

were performed using an RNN with $N = 50$ neurons. We selected three representative combinations of the fundamental control parameters, width w and balance b , corresponding to three specific points in the w – b plane (indicated by crosses in the third panel of the top row in Figure 4):

Phase point A (at $w = 0.2$, $b = 0$) represents a balanced network located near the transition between quiescent and chaotic dynamics.

Phase point B (at $w = 0.2$, $b = 0.2$) shares the same moderate connection width as point A but is positioned close to the border of the fixed point regime.

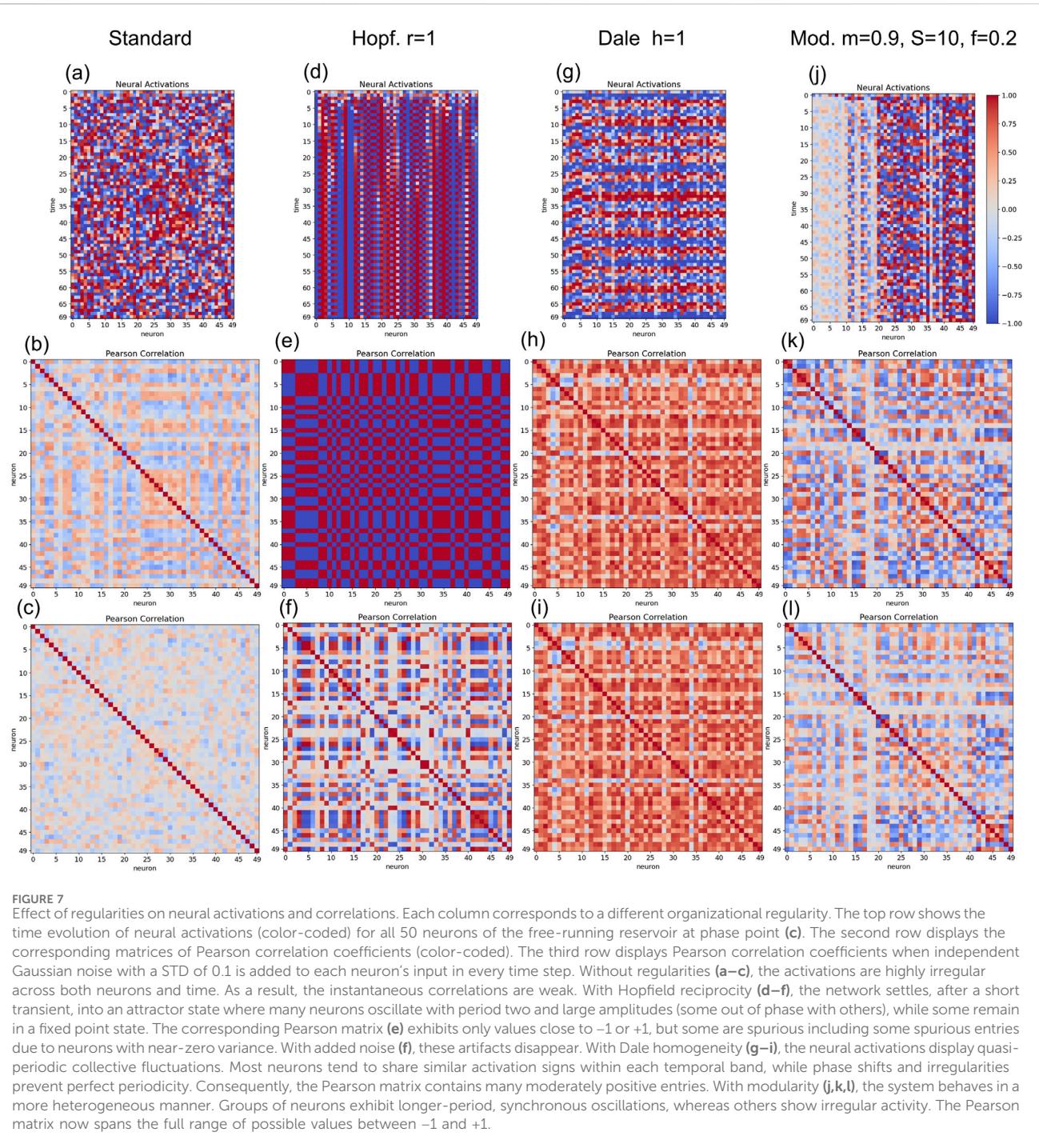
Phase point C (at $w = 0.4$, $b = 0$) is again balanced but lies deeper within the chaotic regime.

The RNN is used throughout as a reservoir in the sequence generation task. While one regularity parameter is varied over its full range [0,1], the others are held at zero.

When the Hopfield reciprocity r is increased (panels a,b,c), the most prominent effects—consistent across all three phase points—are a pronounced rise in nonlinearity N (green) and a small but significant decline in accuracy A (blue). It is also noteworthy that in phase point C, the fluctuation F (orange) decreases while the nonlinearity N (green) increases, demonstrating that these two measures are not trivially related. The covariance C (red) shows hardly any change with reciprocity r .

Increasing Dale homogeneity h (panels d,e,f) leads to a clear gain in accuracy A (blue) across all phase points, indicating that this organizational regularity supports certain types of information processing. The other dynamical quantities show minor, but not entirely consistent, changes as a function of h .

Increasing modularity m with a block size $S = 10$ (panels g,h,i) produces particularly strong effects: the nonlinearity N (green) is



markedly reduced across all three phase points. At the same time, the fluctuation F (orange) decreases in phase points A and B. Most importantly, the accuracy A is significantly enhanced by modularity in all cases.

When the block size is reduced to $S = 1$ (panels j,k,l), the weight matrix no longer contains distinct blocks. In this case, increasing m merely changes the weight distribution from a single Gaussian to a mixture with unchanged total standard deviation. In phase point A, this still yields a modest increase of accuracy A (blue), though the dynamical changes are less pronounced compared to the block-

structured case. For the other two phase points, no further improvement of accuracy is observed.

3.4 Effect of regularities on neuron activations and pearson correlations

To examine how the regularity parameters influence reservoir dynamics, we use the system in phase point C (without input) and simulate the time series of neural activations (Figure 7, first row). We

then analyze the correlations between activations across the reservoir. In contrast to the covariance measure C , which was defined as the unnormalized average product of activations at subsequent time steps ($\Delta t = 1$), we now compute the matrix of instantaneous ($\Delta t = 0$) Pearson correlation coefficients. This means that the mean activity is subtracted and the resulting products are normalized by the variance. We perform the same computation for systems with weak additive noise (third rows). The analysis is carried out for the standard network as well as for each of the three organizational regularities.

Without regularities, the activations are highly irregular across both neurons and time (a). As a result, the instantaneous correlations are weak (b).

With maximal Hopfield reciprocity, the network settles, after a short transient, into an attractor state where many neurons oscillate with period two and large amplitudes (some out of phase with others), while others remain in a fixed point state (d). All corresponding Pearson matrix elements exhibit values close to -1 or $+1$, including some spurious entries when neurons have near-zero variance (e).

With maximal Dale homogeneity, the neural activations display quasi-periodic collective fluctuations. Most neurons tend to share similar activation signs within each temporal band, while phase shifts and irregularities prevent perfect periodicity (g). Consequently, the Pearson matrix contains many moderately positive entries (h).

Finally, with strong modularity, the system behaves in a more heterogeneous manner. Groups of neurons exhibit longer-period, synchronous oscillations, whereas others show irregular activity (j). The Pearson matrix (k) now spans the full range of possible values between -1 and $+1$.

We now repeat the correlation analysis while adding noise to the reservoir. As we have demonstrated in earlier work (Metzner et al., 2024), this prevents the reservoir from becoming permanently trapped in a single attractor throughout the simulation. For this purpose, statistically independent Gaussian random values with zero mean and a standard deviation of 0.1 are added to the total input of each individual neuron in each time step.

The added noise has hardly any effect on the system with Dale homogeneity (i) and causes only a slight reduction in the amplitudes of the correlation coefficients in the system without regularities (c) and in the modular system (L). In contrast, for the Hopfield-like system, the variability introduced by the noise eliminates the spurious perfect correlations previously seen in (f). It now becomes apparent that this regularity induces strong correlations—of either sign—between specific pairs and blocks of neurons.

3.5 Supplemental analyses

3.5.1 Patches classification task

We also tested how modularity affects performance in a completely different type of task. For this purpose, 36 patches from two distinct classes were randomly distributed within a two-dimensional input plane (Figure 8a). The coordinates of a random point (x_1, x_2) in this plane were fed into the reservoir

at the beginning of each episode. The goal of the reservoir computer was to predict the class label of the continuous input point, and the accuracy was computed as the fraction of correctly classified labels.

Although the performance gain is not as strong as in the sequence generation task, the accuracy increases monotonically from near chance level to a significantly higher value when the modularity parameter m (at block size $S = 10$) is tuned from zero to one (Figure 8b), focusing on phase point C.

3.5.2 Larger reservoir

Returning to the standard sequence generation task, we verified the performance gain due to modularity also in a larger reservoir, again focusing on phase point C. As the number of neurons N was doubled from 50 to 100, we made both cases more comparable by reducing at the same time the connection width w by a factor of $1/\sqrt{2}$.

Recomputing the dynamical quantities and the accuracy as functions of the modularity parameter m , we found the same trends as in the smaller reservoir: the nonlinearity N (green) and the fluctuation F (orange) are again strongly reduced, while the accuracy A is significantly enhanced by modularity (Figure 8c).

3.5.3 Chaotic regimes

To demonstrate that within the ‘chaotic regime’ (CR) of phase point C, the temporal evolution of the reservoir exhibits sensitive dependence on initial conditions, we compare the neural activations for two slightly different starting states (Figure 8d). In the second run (middle panel), the initial activation of neuron 0 is increased by 10^{-6} , while all other activations remain identical to those in the first run (left panel). The first noticeable differences (right panel) appear after approximately 45 time steps, after which the two state trajectories diverge rapidly.

3.5.4 Sequential updates

The original motivation of Hopfield networks was to create an attractor landscape of fixed points, each representing a stored pattern. In contrast, our networks with Hopfield-like reciprocal connections predominantly exhibit oscillatory behavior. This difference arises from our use of simultaneous updates of all neuron states at each time step, whereas the original Hopfield model employs sequential updates. To demonstrate this effect, we repeated the simulation shown in Figure 7d, where the RNN was at phase point C with a Hopfield reciprocity parameter of $h = 1$, yet this time with sequential updates. As expected, changing the update scheme results in fixed point rather than oscillatory attractors (Figure 8e)

3.5.5 Diagonal and off-diagonal covariances

In our definition of the covariance measure C , we do not distinguish between diagonal and off-diagonal contributions. To illustrate that both behave similarly, we simulated a reservoir of 50 neurons with coupling width $w = 0.4$ and varied the excitatory/inhibitory balance b from -1 to $+1$, thereby moving the system from the oscillatory through the chaotic into the fixed-point regime. As expected, the covariance measure C shifts from a

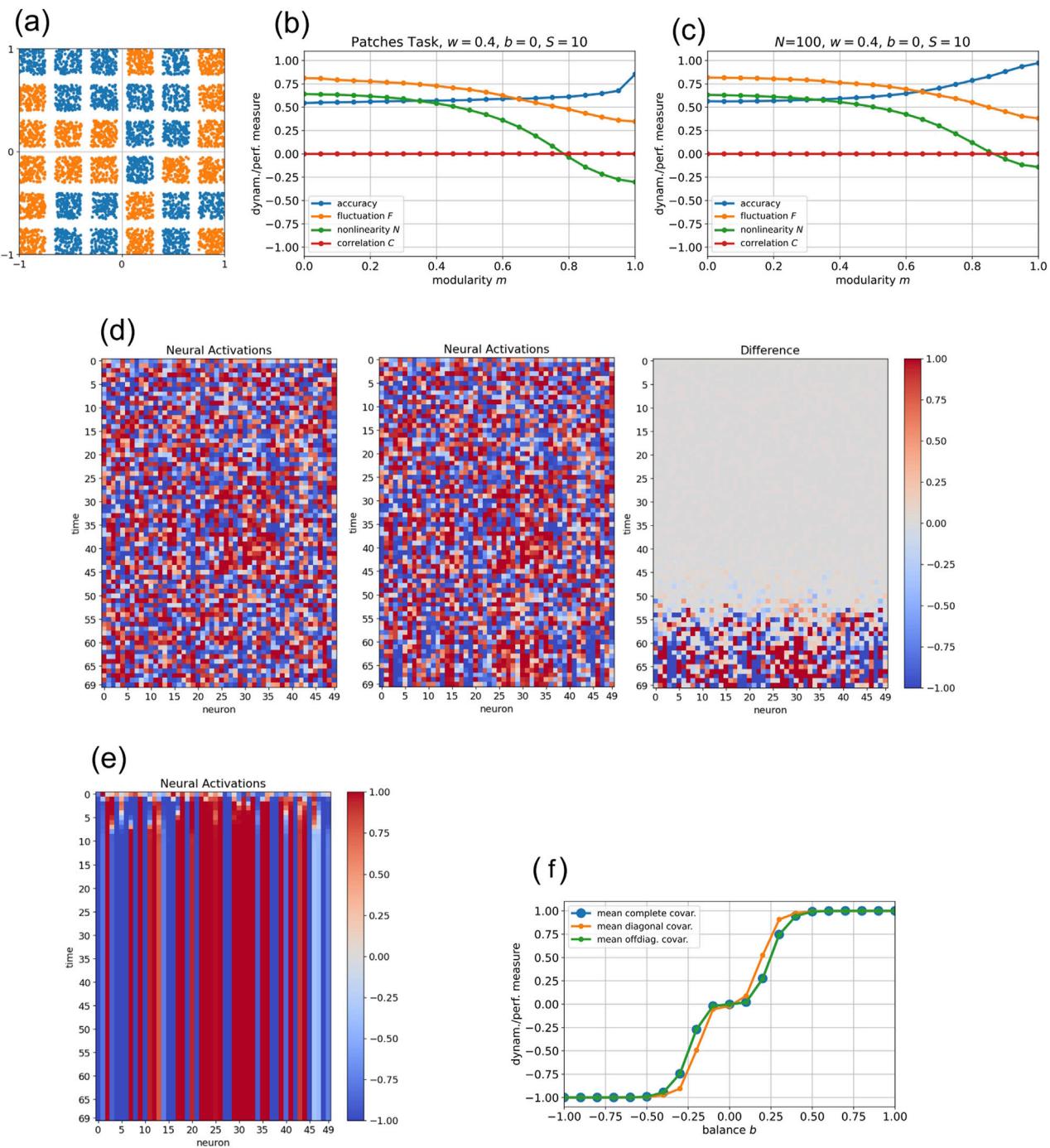


FIGURE 8

Supplemental Analyses. (a) Patches task: 36 patches from two different classes are randomly distributed within the 2D input plane. (b) Dynamic measures and accuracy versus the degree of modularity in the patches classification task. (c) Dynamic measures and accuracy versus the degree of modularity in the sequence generation task, using a rescaled 100-neuron network. (d) Comparing neural activations in phase point C, starting from minimally different starting conditions: In the middle plot, the initial activation of neuron 0 was increased by 10^{-6} . Significant differences between the two system evolutions (right plot) appear after about 45 time steps. (e) Neural activations of a network with Hopfield reciprocity $h = 1$, as in Figure 7d, but with neurons updating sequentially rather than simultaneously. This leads to fixed point attractors. (f) Covariance C as a function of the excitatory/inhibitory bias b , evaluated in a reservoir of 50 neurons with coupling width $w = 0.4$. The purely diagonal contributions in C (orange) show the same basic trend as the off-diagonal contributions (green) or the full covariance (blue).

plateau near -1 , through one around 0 , to one near $+1$ (Figure 8f). The covariance computed only from diagonal terms (orange) follows the same trend as that based on off-diagonal (green) or all terms combined (blue).

4 Discussion

In this work, we investigated how three distinct organizational regularities—Hopfield reciprocity, Dale homogeneity, and

modularity—affect the dynamical behavior and computational performance of recurrent neural networks (RNNs).

4.1 Prior expectations and numerical findings

4.1.1 Hopfield reciprocity

Hopfield-type reciprocity introduces symmetry into the weight matrix by increasing the probability that any connection from neuron A to B is mirrored by an equal connection from B to A. In classical Hopfield networks, such symmetry is instrumental in stabilizing fixed point attractors, corresponding to stored patterns. With some notable exceptions (Kühn and Bös, 1993), these networks normally use binary units and asynchronous updates, enabling the system to settle gradually into one of the memorized configurations. The symmetric weights ensure that the energy landscape has defined minima, guiding the dynamics toward stable fixed points.

In our model, the situation differs in several respects. The neurons are continuous-valued with tanh activation, and all updates occur synchronously. Nonetheless, it was natural to expect that increasing Hopfield reciprocity r might exert an ordering influence on the network dynamics - possibly reducing chaotic fluctuations and thus enhancing information processing capabilities.

The numerical results only partially confirm this expectation. As r increases, the most consistent effect across the three phase points is a strong rise in nonlinearity N . This indicates that the reservoir is increasingly driven into a saturated, quasi-digital regime, which is generally disadvantageous for many computational tasks. The neural activations reveal subpopulations of neurons that settle into fixed points, while others exhibit period-two oscillations. Moreover, the instantaneous Pearson correlation matrices (Figures 7d–f) show that reciprocal coupling induces strongly correlated neuron pairs or groups, often displaying nearly perfect positive or negative correlations that persist even under added noise. Overall, these effects tend to reduce the computational performance.

4.1.2 Dale homogeneity

Dale's principle, a key feature of biological neural networks, stipulates that each neuron maintains a fixed output polarity—either excitatory or inhibitory—across all its targets. In our model, this principle is implemented by the homogeneity parameter $h \in [0, 1]$, which controls the consistency of output signs within each column of the weight matrix. At $h = 0$, neurons send mixed excitatory and inhibitory outputs; at $h = 1$, every neuron acts strictly as a monopolar sender.

From a theoretical perspective, one might expect that Dale homogeneity introduces a more directional and interpretable signal flow through the reservoir. Specifically, we anticipated that increasing h would suppress high-frequency or erratic fluctuations by enforcing more coherent influence patterns among neurons, which might also be reflected in their mutual instantaneous correlations.

Indeed, as the homogeneity parameter h increases, the initially very small instantaneous Pearson correlations observed in the

chaotic phase point C are replaced by much stronger, predominantly positive correlations. An unexpected feature was that neural activations now exhibit quasi-periodic collective oscillations with longer periods and noticeable phase shifts between individual neurons. Nevertheless, the accuracy in our test task clearly improves when Dale regularity is introduced.

4.1.3 Modularity (Block size $S = 1$)

In the case $S = 1$, the modularity parameter m does not produce distinct structural blocks in the weight matrix. Instead, increasing m gradually transforms the underlying weight distribution from a single Gaussian to a mixture of two Gaussians—one narrower (weaker weights) and one broader (stronger weights)—with the total standard deviation kept constant. This results in a heterogeneous distribution of connection strengths with a much broader tail.

We expected that such broader-tailed weight distributions might have a beneficial effect on the network's information-processing capacity. In biological neural circuits, especially within the human cortex, synaptic connection strengths are known to follow heavy-tailed, approximately log-normal distributions, where a small subset of strong connections coexists with a majority of weak ones (Song et al., 2005). This structural heterogeneity has been suggested to support both robustness and dynamic richness by combining stable, high-impact pathways with a flexible background of weaker connections.

Indeed, we find in our simulations a clear rise in task accuracy A , despite the absence of actual modular structure.

4.1.4 Modularity (Block size $S = 10$)

In biological neural networks, modular organization is a well-established principle observed across spatial scales—from cortical microcircuits to large brain regions. Modules, typically defined as groups of neurons with strong internal connectivity and weaker coupling to the rest of the network, are thought to support functional specialization while preserving global integration. From this perspective, introducing modularity into artificial RNNs could plausibly enhance their computational capacity by enabling localized processing and buffering against global instabilities.

In our model, modularity is implemented via the control parameter $m \in [0, 1]$, which adjusts the variance of intra- and inter-module connection strengths while keeping the overall standard deviation constant. When the block size is set to $S = 10$, the resulting matrix exhibits clearly defined patches of stronger weights embedded in a weaker background. However, these high-variance patches may lie on or off the diagonal. Diagonal modules imply recurrently coupled local groups, whereas off-diagonal modules encode directional connections from one group of neurons to another, without necessarily reciprocal feedback. While the former can stabilize local loops of activity, the latter may implement feedforward-like interactions across functionally distinct subnetworks. Though less intuitively interpretable, these asymmetric modules still contribute to shaping constrained, layered dynamics within the recurrent architecture.

Prior to numerical investigation, we expected that such modular organization could lead to partial functional segregation, a dampening of chaotic fluctuations and possibly more reproducible activity patterns within modules.

Indeed, these expectations are supported by the simulations. As m increases, we observe a pronounced reduction in both fluctuation F and nonlinearity N , indicating that the reservoir dynamics become more stable and less saturated. This effect is especially clear in balanced networks ($b = 0$), which otherwise tend to exhibit chaotic behavior. The drop in F suggests that spontaneous overactivation is strongly suppressed, while the lower N points to a shift away from digital-like saturation toward a more linear or intermediate regime.

The Pearson correlation matrices (Figures 7j–l) provide a complementary view of this effect: they reveal substructures of internally coherent neuron clusters with weaker, mixed-sign couplings between modules. The correlations are neither too weak (as in chaotic regimes) nor too strong (as in fixed point or short-period states) but distributed across the full $[-1, +1]$ range. This balanced structure appears to create particularly favorable conditions for reservoir computing.

Most strikingly, both in reservoirs of 50 and 100 neurons, the accuracy A in the sequence generation task increases sharply and reaches near-maximal levels across a wide range of m . The improvement in computational performance surpasses that achieved with Dale homogeneity. A very similar general effect of this organizational regularity was found in the patches classification task.

4.2 Work limitations and future perspectives

The present study has focused almost exclusively on one specific computational task: the generation of predefined output sequences from class-specific input stimuli. While this task is well-suited for evaluating the internal stability and reproducibility of reservoir trajectories, it represents only one of many possible functional challenges an RNN might face. Future studies will systematically explore how the three structural regularities—Hopfield reciprocity, Dale homogeneity, and modularity—affect other task types, including classification, prediction, temporal integration, and generative modeling. These tasks may impose different demands on the reservoir, potentially favoring entirely different dynamical regimes.

Several further limitations of the current setup should be noted. Our networks use a fixed, pointwise tanh activation function throughout, with no mechanism for adapting nonlinear response properties during training or evolution. Moreover, the recurrent connections are entirely fixed, without plasticity or learning mechanisms. While this simplification allows us to isolate the impact of architectural regularities, it remains unclear how the observed effects extend to systems with ongoing synaptic adaptation. Likewise, our networks are relatively small and shallow, lacking hierarchical depth or multi-scale processing pathways. It is an open question whether similar regularities exert comparable influences in large-scale architectures with layered structure, where modularity and reciprocity might interact differently with gradient propagation and representational abstraction.

A further important direction concerns the interaction between structural regularities. In this study, we varied each regularity parameter in isolation while keeping the others fixed. However,

real biological systems typically exhibit several regularities at once. It remains an open question whether combinations of regularities act synergistically or interfere with each other. For instance, it is conceivable that modularity and Dale homogeneity together enhance performance more than either alone—or that reciprocity counteracts the benefits of modular organization.

A separate line of inquiry could explore the relevance of the observed effects for biological computation, although in this context, it would be more realistic to interpret each of our RNN units not a single neuron, but as a homogeneous neuronal assembly of a cortical network (Knight, 2000; Mattia and Del Giudice, 2002). For example, it could be further investigated whether biological circuits show anything resembling Hopfield symmetry at the level of neuronal assemblies. So far, while exact synaptic reciprocity is not observed, cortical microcircuits consistently exhibit an overrepresentation of bidirectional connections (Song et al., 2005; Felix and Triesch, 2017) and population dynamics with attractor-like stability. These features suggest that, on a coarse-grained assembly level, biological networks may display approximate or statistical reciprocity, even though the precise Hopfield topology is not realized.

Since the structural regularities studied here are motivated by neurobiological observations, it is worthwhile to compare their dynamical implications to actual brain circuits. This could involve applying the same dynamical and performance metrics to empirically derived connectomes—such as those of *C. elegans*, *Drosophila*, or the zebrafish larva—and seeing how they perform on comparable tasks.

Beyond empirical validation, the theoretical understanding of how structural regularities shape network dynamics remains incomplete. For instance, it is still unclear why modularity so reliably suppresses fluctuation and nonlinearity, or why Dale homogeneity improves accuracy without driving the system into linearity. While it may be tempting to invoke classical tools such as the spectral radius or eigenvalue spectra of the weight matrix, such linear measures often fail to capture the complex behavior of nonlinear, recurrent systems. In our view, a more fruitful approach would be to characterize how the structural parameters influence the geometry of the state space, the stability of trajectories, or the repeatability of state sequences under repeated input. Concepts such as state convergence, divergence under perturbations, and the reproducibility of internal trajectories may provide more robust and interpretable metrics than traditional eigenvalue-based criteria. A related approach was proposed by Legenstein and Maass (Legenstein and Maass, 2007), who linked the computational performance of neural microcircuits to their dynamical regime near the edge of chaos, using measures of kernel quality and generalization capability to predict functional performance. Developing such diagnostics could help formulate a more general understanding of how structural constraints give rise to functional dynamics in RNNs.

Another promising direction lies in allowing structural regularities to vary dynamically over time. Instead of statically imposed homogeneity or modularity, one could investigate networks in which these properties emerge or change through learning or adaptation. This would connect structural regularities more directly to plasticity rules and functional demands, potentially offering new models of task-dependent reconfiguration in neural circuits.

Data availability statement

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

Author contributions

CM: Writing – original draft, Visualization, Investigation, Methodology, Formal Analysis, Validation, Conceptualization, Data curation. AS: Project administration, Funding acquisition, Supervision, Writing – original draft, Conceptualization. AM: Resources, Writing – original draft, Data curation, Validation, Funding acquisition. PK: Methodology, Writing – original draft, Supervision, Investigation, Conceptualization, Funding acquisition, Project administration, Validation, Resources.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): grants KR 5148/3-1 (project number 510395418), KR 5148/5-1 (project number 542747151), KR 5148/10-1 (project number 563909707) and GRK 2839 (project number 468527017) to PK, and grants SCHI 1482/3-1 (project number 451810794) and SCHI 1482/6-1 (project number 563909707) to AS.

References

Aguiar, M., Das, A., and Johansson, K. H. (2023). “Universal approximation of flows of control systems by recurrent neural networks,” in *2023 62nd IEEE conference on decision and control (CDC)* (IEEE), 2320–2327.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J. Big Data* 8 (1), 1–74. doi:10.1186/s40537-021-0044-8

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Curr. Opinion Neurobiology* 46, 1–6. doi:10.1016/j.conb.2017.06.003

Berlemont, K., and Mongillo, G. (2022). *Glassy phase in dynamically-balanced neuronal networks*, 2022–2023. bioRxiv.

Bönsel, F., Krauss, P., Metzner, C., and Yamakou, M. E. (2021). *Control of noise-induced coherent oscillations in time-delayed neural motifs*. arXiv preprint arXiv: 2106.11361.

Brunel, N. (2016). Is cortical connectivity optimized for storing information? *Nat. Neuroscience* 19 (5), 749–755. doi:10.1038/nn.4286

Büsing, L., Schrauwen, B., and Legenstein, R. (2010). Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural Computation* 22 (5), 1272–1311. doi:10.1162/neco.2009.01-09-947

Cornford, J., Kalajdzieski, D., Leite, M., Lamarquette, A., Kullmann, D. M., and Richards, B. (2020). Learning to live with Dale’s principle: ANNs with separate excitatory and inhibitory units. *BioRxiv*, 2011–2020. doi:10.1101/2020.11.02.364968

Cucchi, M., Abreu, S., Ciccone, G., Brunner, D., and Kleemann, H. (2022). Hands-on reservoir computing: a tutorial for practical implementation. *Neuromorphic Comput. Eng.* 2 (3), 032002. doi:10.1088/2634-4386/ac7db7

Dambre, J., Verstraeten, D., Schrauwen, B., and Massar, S. (2012). Information processing capacity of dynamical systems. *Sci. Reports* 2 (1), 1–7. doi:10.1038/srep00514

Farrell, M., Recanatesi, S., Moore, T., Lajoie, G., and Shea-Brown, E. (2022). Gradient-based learning drives robust representations in recurrent neural networks by balancing compression and expansion. *Nat. Mach. Intell.* 4 (6), 564–573. doi:10.1038/s42256-022-00498-0

Felix, Z., and Triesch, J. (2017). Hoffmann and jochen triesch. Nonrandom network connectivity comes in pairs. *Netw. Neuroscience* 1 (1), 31–41. doi:10.1162/NETN_a_00004

Folli, V., Gosti, G., Leonetti, M., and Ruocco, G. (2018). Effect of dilution in asymmetric recurrent neural networks. *Neural Netw.* 104, 50–59. doi:10.1016/j.neunet.2018.04.003

Fournier, S. J., Pacco, A., Ros, V., and Urbani, P. (2025). *Non-reciprocal interactions and high-dimensional chaos: comparing dynamics and statistics of equilibria in a solvable model*. arXiv preprint arXiv:2503.20908.

Gerum, R. C., Erpenbeck, A., Krauss, P., and Schilling, A. (2020). Sparsity through evolutionary pruning prevents neuronal networks from overfitting. *Neural Netw.* 128, 305–312. doi:10.1016/j.neunet.2020.05.007

Gonon, L., and Ortega, J.-P. (2021). Fading memory echo state networks are universal. *Neural Netw.* 138, 10–13. doi:10.1016/j.neunet.2021.01.025

Haviv, D., Rivkind, A., and Barak, O. (2019). “Understanding and controlling memory in recurrent neural networks,” in *International conference on machine learning*. Cambridge, MA: Proceedings of Machine Learning Research (PMLR), 2663–2671.

Hinton, G. E., Sabour, S., and Frosst, N. (2018). “Matrix capsules with em routing,” in *International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, April 30 - May 3, 2018.

Hwang, S., Folli, V., Lanza, E., Parisi, G., Ruocco, G., and Zamponi, F. (2019). On the number of limit cycles in asymmetric neural networks. *J. Stat. Mech. Theory Exp.* 2019 (5), 053402. doi:10.1088/1742-5468/ab11e3

Hwang, S., Lanza, E., Parisi, G., Rocchi, J., Ruocco, G., and Zamponi, F. (2020). On the number of limit cycles in diluted neural networks. *J. Stat. Phys.* 181 (6), 2304–2321. doi:10.1007/s10955-020-02664-3

Ikemoto, S., DallaLibera, F., and Koh, H. (2018). Noise-modulated neural networks as an application of stochastic resonance. *Neurocomputing* 277, 29–37. doi:10.1016/j.neucom.2016.12.111

Jaeger, H. (2001). *The “echo state” approach to analysing and training recurrent neural networks-with an erratum note*. Bonn, Germany: German National Research Center for Information Technology. GMD Technical Report 13.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Generative AI statement

The author(s) declared that generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

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.

Jaeger, H. (2014). *Controlling recurrent neural networks by conceptors*. arXiv preprint arXiv:1403.3369.

Knight, B. W. (2000). Dynamics of encoding in neuron populations: some general mathematical features. *Neural Comput.* 12 (3), 473–518. doi:10.1162/089976600300015673

Krauss, P., Tziridis, K., Metzner, C., Schilling, A., Ulrich, H., and Schulze, H. (2016). Stochastic resonance controlled upregulation of internal noise after hearing loss as a putative cause of tinnitus-related neuronal hyperactivity. *Front. Neuroscience* 10, 597. doi:10.3389/fnins.2016.00597

Krauss, P., Prebeck, K., Schilling, A., and Metzner, C. (2019a). “Recurrence resonance” in three-neuron motifs. *Front. Computational Neuroscience* 13, 64. doi:10.3389/fncom.2019.00006

Krauss, P., Zankl, A., Schilling, A., Schulze, H., and Metzner, C. (2019b). Analysis of structure and dynamics in three-neuron motifs. *Front. Comput. Neurosci.* 13 (5), 5. doi:10.3389/fncom.2019.00005

Krauss, P., Schuster, M., Dietrich, V., Schilling, A., Schulze, H., and Metzner, C. (2019c). Weight statistics controls dynamics in recurrent neural networks. *PLoS One* 14 (4), e0214541. doi:10.1371/journal.pone.0214541

Kühn, R., and Bös, S. (1993). Statistical mechanics for neural networks with continuous-time dynamics. *J. Phys. A Math. General* 26 (4), 831–857. doi:10.1088/0305-4470/26/4/012

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521 (7553), 436–444. doi:10.1038/nature14539

Legenstein, R., and Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks* 20 (3), 323–334. doi:10.1016/j.neunet.2007.04.017

Lutz, M., Schuchhardt, J., and Schuster, H. G. (1992). Suppressing chaos in neural networks by noise. *Phys. Review Letters* 69 (26), 3717–3719. doi:10.1103/PhysRevLett.69.3717

Maheswaranathan, N., Williams, A. H., Golub, M. D., Ganguli, S., and Sussillo, D. (2019). Universality and individuality in neural dynamics across large populations of recurrent networks. *Adv. Neural Information Processing Systems* 2019, 15629–15641. doi:10.48550/arXiv.1907.08549

Mattia, M., and Del Giudice, P. (2002). Population dynamics of interacting spiking neurons. *Phys. Rev. E* 66 (5), 051917. doi:10.1103/PhysRevE.66.051917

Maximilian Schäfer, A., and Zimmermann, H. G. (2006). “Recurrent neural networks are universal approximators,” in *International conference on artificial neural networks* (Springer), 632–640.

Metzner, C., and Krauss, P. (2022). Dynamics and information import in recurrent neural networks. *Front. Comput. Neurosci.* 16, 876315. doi:10.3389/fncom.2022.876315

Metzner, C., Schilling, A., Maier, A., and Krauss, P. (2024). Recurrence resonance–noise-enhanced dynamics in recurrent neural networks. *Front. Complex Syst.* 2, 1479417. doi:10.3389/fcpxs.2024.1479417

Metzner, C., Schilling, A., Maier, A., and Krauss, P. (2025). Nonlinear neural dynamics and classification accuracy in reservoir computing. *Neural Comput.* 37 (8), 1469–1504. doi:10.1162/neco_a_01770

Meunier, D., Lambiotte, R., and Bullmore, E. T. (2010). Modular and hierarchically modular organization of brain networks. *Front. Neurosci.* 4, 200. doi:10.3389/fnins.2010.00200

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., et al. (2023). Recent advances in natural language processing via large pre-trained language models: a survey. *ACM Comput. Surv.* 56 (2), 1–40. doi:10.1145/3605943

Narang, S., Elsen, E., Diamos, G., and Sengupta, S. (2017). *Exploring sparsity in recurrent neural networks*. arXiv preprint arXiv:1704.05119.

Perin, R., Berger, T. K., and Markram, H. (2011). A synaptic organizing principle for cortical neuronal groups. *Proc. Natl. Acad. Sci.* 108 (13), 5419–5424. doi:10.1073/pnas.1016051108

Rajan, K., Abbott, L. F., and Sompolinsky, H. (2010). Stimulus-dependent suppression of chaos in recurrent neural networks. *Phys. Rev. E* 82 (1), 011903. doi:10.1103/PhysRevE.82.011903

Rodriguez, N., Izquierdo, E., and Ahn, Y.-Y. (2019). Optimal modularity and memory capacity of neural reservoirs. *Netw. Neurosci.* 3 (2), 551–566. doi:10.1162/netn_a_00080

Rosenbaum, C., Klinger, T., and Rish, I. (2018). “Routing networks: adaptive selection of non-linear functions for multi-task learning,” in *International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, April 30 - May 3, 2018.

Rosenbaum, C., Rish, I., and Klinger, T. (2019). *Routing networks and the challenges of modular and compositional computation*. arXiv preprint arXiv:1904.12574.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. *Adv. Neural Inf. Process. Syst.* 30. doi:10.48550/arXiv.1710.09829

Schilling, A., Tziridis, K., Schulze, H., and Krauss, P. (2021). The stochastic resonance model of auditory perception: a unified explanation of tinnitus development, zwicker tone illusion, and residual inhibition. *Prog. Brain Res.* 262, 139–157. doi:10.1016/bs.pbr.2021.01.025

Schilling, A., Gerum, R., Metzner, C., Maier, A., and Krauss, P. (2022). Intrinsic noise improves speech recognition in a computational model of the auditory pathway. *Front. Neurosci.* 16, 908330. doi:10.3389/fnins.2022.908330

Schilling, A., Sedley, W., Gerum, R., Metzner, C., Tziridis, K., Maier, A., et al. (2023). Predictive coding and stochastic resonance as fundamental principles of auditory phantom perception. *Brain* 146 (12), 4809–4825. doi:10.1093/brain/awad255

Schuecker, J., Goedeke, S., and Helias, M. (2018). Optimal sequence memory in driven random networks. *Phys. Rev. X* 8 (4), 041029. doi:10.1103/physrevx.8.041029

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., et al. (2017). *Outrageously large neural networks: The sparsely-gated mixture-of-experts layer*. arXiv preprint

Somogyi, P., Tamás, G., Luján, R., and Buhl, E. H. (1998). Salient features of synaptic organisation in the cerebral cortex. *Brain Res. Rev.* 26 (2-3), 113–135. doi:10.1016/S0165-0173(97)00061-1

Song, S., Jesper Sjöström, P., Reigl, M., Nelson, S., and Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology* 3 (3), e68. doi:10.1371/journal.pbio.0030068

Sporns, O., and Betzel, R. F. (2016). Modular brain networks. *Annu. Rev. Psychol.* 67, 613–640. doi:10.1146/annurev-psych-122414-033634

Strata, P., and Harvey, R. (1999). Dale’s principle. *Brain Res. Bull.* 50 (5-6), 349–350. doi:10.1016/s0361-9230(99)00100-8

Wallace, E., Maei, H. R., and Latham, P. E. (2013). Randomly connected networks have short temporal memory. *Neural Computation* 25 (6), 1408–1439. doi:10.1162/NECO_a_00449

Zador, A. M. (2019). A critique of pure learning and what artificial neural networks can learn from animal brains. *Nat. Commun.* 10 (1), 3770. doi:10.1038/s41467-019-11786-6