frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 08 October 2025
DOI 10.3389/fnbot.2025.1646336

OPEN ACCESS

EDITED BY
Zhong Chen,
Southern Illinois University Carbondale,
United States

REVIEWED BY
Anh-Tu Nguyen,
Hanoi University of Industry, Vietnam
Yang Yan,
Southern Illinois University Carbondale,
United States
Yuchen Liu,
Xi'an Jiaotong-Liverpool University, China

*CORRESPONDENCE
Yuwen Zhou
✉ 13645694933@163.com

RECEIVED 13 June 2025
ACCEPTED 29 August 2025
PUBLISHED 08 October 2025

CITATION
Zhou Y and Zhang W (2025) End-to-end robot intelligent obstacle avoidance method based on deep reinforcement learning with spatiotemporal transformer architecture. *Front. Neurorobot.* 19:1646336. doi: 10.3389/fnbot.2025.1646336

End-to-end robot intelligent obstacle avoidance method based on deep reinforcement learning with spatiotemporal transformer architecture

Yuwen Zhou[1]* and Weizhong Zhang[2]

[1]School of Engineering Mathematics and Technology, University of Bristol, Bristol, United Kingdom, [2]Anhui Xinhua University, Hefei, China

To enhance the obstacle avoidance performance and autonomous decision-making capabilities of robots in complex dynamic environments, this paper proposes an end-to-end intelligent obstacle avoidance method that integrates deep reinforcement learning, spatiotemporal attention mechanisms, and a Transformer-based architecture. Current mainstream robot obstacle avoidance methods often rely on system architectures with separated perception and decision-making modules, which suffer from issues such as fragmented feature transmission, insufficient environmental modeling, and weak policy generalization. To address these problems, this paper adopts Deep Q-Network (DQN) as the core of reinforcement learning, guiding the robot to autonomously learn optimal obstacle avoidance strategies through interaction with the environment, effectively handling continuous decision-making problems in dynamic and uncertain scenarios. To overcome the limitations of traditional perception mechanisms in modeling the temporal evolution of obstacles, a spatiotemporal attention mechanism is introduced, jointly modeling spatial positional relationships and historical motion trajectories to enhance the model's perception of critical obstacle areas and potential collision risks. Furthermore, an end-to-end Transformer-based perception-decision architecture is designed, utilizing multi-head self-attention to perform high-dimensional feature modeling on multi-modal input information (such as LiDAR and depth images), and generating action policies through a decoding module. This completely eliminates the need for manual feature engineering and intermediate state modeling, constructing an integrated learning process of perception and decision-making. Experiments conducted in several typical obstacle avoidance simulation environments demonstrate that the proposed method outperforms existing mainstream deep reinforcement learning approaches in terms of obstacle avoidance success rate, path optimization, and policy convergence speed. It exhibits good stability and generalization capabilities, showing broad application prospects for deployment in real-world complex environments.

KEYWORDS

deep reinforcement learning, spatiotemporal attention mechanism, transformer architecture, end-to-end obstacle avoidance, autonomous robot navigation

1 Introduction

With continuous breakthroughs in artificial intelligence (Chen et al., 2025), automatic control, and environmental perception technologies (Ni et al., 2025), the autonomous navigation and obstacle avoidance capabilities of robots have become key factors in enabling intelligent mobile robotic systems (Ullah et al., 2024). Especially in various typical application scenarios such as warehouse logistics, urban delivery, agricultural operations, and public services, robots are often required to operate continuously in dynamic, complex, and even unknown environments, placing higher demands on their path planning and obstacle avoidance decision-making capabilities (Katona et al., 2024). Meanwhile, the uncertainty, diversity, and real-time nature of these environments pose significant challenges to the perception, reaction speed, and robustness of robotic systems (Wang et al., 2021).

Against this background, intelligent obstacle avoidance technologies for robot groups have rapidly developed and become a hot research topic in the robotics field. Specifically, path planning technologies are responsible for identifying a feasible route from a starting point to a target point within a given map or environment, while obstacle avoidance technologies require robots to dynamically perceive and avoid both static and dynamic obstacles during movement, achieving local or global path adjustment and optimization (Almazrouei et al., 2023). Traditional methods mostly rely on a sequential "mapping-then-planning" framework, which, while effective in structured environments, often faces issues such as high computational complexity, delayed responses, and strong dependence on environmental stability when dealing with scenarios featuring randomly appearing obstacles or frequent environmental changes (Guo et al., 2022).

In complex scenarios, robots must simultaneously handle various types of static and dynamic obstacles, such as stacked objects, pedestrians, other mobile devices, and obstacles resulting from unexpected events (Zheng et al., 2022). Traditional path planning methods often struggle to cope with such situations. For instance, heuristic algorithms like the widely used A $\times$ algorithm (Liu et al., 2022) demonstrate strong feasibility and controllability in static environments but suffer from inefficiencies and poor adaptability in dynamic or time-sensitive applications. Additionally, intelligent optimization methods such as Ant Colony Optimization (ACO; Miao et al., 2021) and Particle Swarm Optimization (PSO; Shami et al., 2022) have improved path quality and global search capability to some extent, but still face bottlenecks such as unsmooth paths, susceptibility to local optima, and slow convergence rates, making them inadequate for modern robots operating in highly dynamic environments.

On the other hand, traditional obstacle avoidance frameworks typically adopt modular processing pipelines, where perception, mapping, path planning, and control are handled in separate stages (Qin et al., 2023). While this design offers clear structure and relatively low engineering implementation costs, in practical deployments, accumulated errors between modules, delays in information transmission, and a lack of global modeling capability often degrade obstacle avoidance performance. Particularly in unstructured or semi-structured environments such as farmlands, mountainous areas, narrow corridors, or temporary warehouse setups, it is difficult to completely model feasible paths in advance, limiting the adaptability of traditional methods to diverse and uncertain scenarios. As application scenarios become increasingly complex and expectations for robot intelligence continue to rise, building intelligent obstacle avoidance systems with enhanced environmental awareness, dynamic adaptability, path quality assurance, and decision-making optimization has become a key research direction for mobile robotics (Lahn et al., 2023).

In recent years, Deep Reinforcement Learning (DRL) has emerged as a prominent approach in robot obstacle avoidance research due to its strong autonomous learning capabilities and adaptability (Dai et al., 2024). Especially DRL models based on algorithms like Deep Q-Network (DQN) can continuously interact with the environment to learn optimal obstacle avoidance strategies in complex scenarios, eliminating the need for prior modeling and handcrafted rules (Issa et al., 2021). However, when handling high-dimensional sensory data and dynamic obstacle distributions, relying solely on perception modules based on traditional convolutional neural networks often fails to capture temporal dependencies and global environmental features, limiting the generalization and robustness of learned policies.

To address these challenges, this paper proposes an end-to-end intelligent robot obstacle avoidance method that integrates deep reinforcement learning with a spatiotemporal Transformer architecture. This method adopts DQN as the core for policy learning, introduces the powerful global modeling capability of the self-attention mechanism in Transformers, and incorporates a spatiotemporal attention structure to jointly perceive and model the spatial distribution and temporal evolution of obstacles, thereby improving the adaptability of decision-making strategies to dynamic environmental changes (Phatak et al., 2023). The proposed architecture establishes an integrated end-to-end learning pipeline from perception to decision-making, effectively overcoming the limitations of traditional methods in multi-stage processing and local feature modeling.

Before detailing our contributions, we justify our choice of DQN over other DRL frameworks. While PPO and A3C offer superior sample efficiency and stability in continuous action spaces, and SAC excels in stochastic environments, DQN provides several advantages for our specific application: (1) discrete action spaces are more suitable for robot navigation commands (move forward, turn left/right, stop), (2) the Q-value function naturally aligns with our spatiotemporal feature representation from the Transformer, (3) the experience replay mechanism enables efficient utilization of our diverse simulation data, and (4) the deterministic policy output ensures consistent obstacle avoidance behaviors essential for safety-critical applications.

The contributions of this paper are as follows:

1. Deep Q-Network (DQN) is introduced as the core framework for robot obstacle avoidance strategy learning, constructing an adaptive learning path from perceived states to action decisions. Traditional obstacle avoidance methods rely on handcrafted rules or supervised signals and struggle with continuous decision-making in dynamic, complex, or unknown environments. In contrast, by combining DQN with environmental state learning to approximate Q-value functions, robots can autonomously optimize obstacle avoidance strategies

through interaction with the environment without requiring explicit models. Compared with static strategies or conventional Q-learning, DQN utilizes deep neural networks to achieve high-dimensional state space mapping, significantly improving the generalization and decision-making efficiency of obstacle avoidance behaviors. Furthermore, experience replay and target network update mechanisms stabilize the learning process and prevent policy oscillations.

2. A spatiotemporal attention mechanism is incorporated as a crucial component for perception modeling and policy optimization in obstacle avoidance tasks, aiming to simultaneously capture key spatial features and temporal evolution patterns in the environment. Traditional perception modules tend to focus only on local spatial information at the current moment, lacking dynamic modeling capabilities for historical states, resulting in instability when dealing with moving obstacles or complex scenarios. To address this, this paper proposes a fused attention structure combining temporal and spatial information. By embedding a spatiotemporal joint attention module during feature encoding, the model can dynamically focus on critical obstacles, path boundaries, or motion trends across different time points and spatial regions. This mechanism constructs dependencies between local and global information, enabling the model to more accurately identify potential collision risks and safe passages without losing important contextual information.
3. An end-to-end Transformer architecture is designed as the core modeling framework for the robot obstacle avoidance system, aiming to overcome the information fragmentation and feature loss issues caused by traditional separated perception-decision structures. Compared with previous approaches relying on handcrafted feature extraction or stacked convolutional-recurrent networks, the proposed Transformer-based framework offers powerful global modeling capabilities and parallel processing efficiency. It can capture long-range dependencies of critical obstacles and potential paths in multi-scale spatial environments. Specifically, the Transformer encoder performs sequential modeling of spatiotemporal information, while the decoder directly outputs action policies or value functions, providing high-quality semantic representations for deep reinforcement learning. This Transformer-based architecture not only improves the generalization and response speed of obstacle avoidance strategies in dynamic and complex scenarios but also significantly reduces reliance on intermediate modules or manually designed features, demonstrating excellent end-to-end intelligent obstacle avoidance performance.

The structure of this paper is organized as follows:

Section 2 reviews related work and discusses prior research, summarizing their strengths and limitations. Section 3 details the proposed method, including the DQN framework, spatiotemporal attention mechanism, and Transformer architecture, along with explanations of the algorithmic workflow. Section 4 presents the experimental setup, comparative evaluations, ablation studies, and visualized results. Finally, Section 5 discusses the conclusions, limitations of this study, and outlines directions for future research.

2 Related work

We organize our analysis of existing obstacle avoidance methods along four critical dimensions: (1) perception mechanism and feature extraction capabilities, (2) reinforcement learning framework and policy optimization strategy, (3) architectural integration level and information flow design, and (4) temporal modeling and sequential dependency handling. This systematic framework allows us to identify specific gaps that our proposed method addresses.

In dynamic, variable, and unstructured environments, the primary challenges faced by robots in obstacle avoidance include perception uncertainty, the complexity of environment modeling, the real-time nature of path planning, and the robustness of decision-making behavior (Wijayathunga et al., 2023). Traditional methods, such as path planning algorithms based on A$\times$, Dijkstra, and Ant Colony Optimization, have demonstrated certain effectiveness in structured environments. However, these approaches typically require static modeling of the environment and rely on accurate maps, lacking the adaptability to handle unexpected obstacles and dynamic changes. Moreover, these algorithms commonly suffer from limitations such as tortuous paths, slow convergence, and sensitivity to local optima, making them inadequate for complex, real-world scenarios.

To address the limitations of traditional methods, the research community has introduced Simultaneous Localization and Mapping (SLAM) techniques (Alsadik and Karam, 2021), enabling robots to achieve self-localization and environment modeling in unknown environments. Significant progress has been made in SLAM, from classical Extended Kalman Filter (EKF)-based methods to enhanced algorithms like Particle Filter (PF) and Rao-Blackwellized Particle Filter (RBPF; Zhang, 2022). The Gmapping algorithm proposed by Grisetti et al. (2007), a representative of RBPF-SLAM, achieved high-precision 2D map construction. With advances in visual perception technologies, Visual SLAM (VSLAM) has emerged as a research hotspot, integrating robust image feature detection algorithms such as SIFT and SURF, and has been widely applied in object recognition and 3D mapping (Ansari, 2019; Ni et al., 2024). However, SLAM still faces challenges in practical applications, including slow processing speeds and sensitivity to feature occlusion and lighting variations, which limit its obstacle avoidance responsiveness in highly dynamic environments.

In recent years, the rise of deep learning has brought new paradigms to robotic obstacle avoidance. As research in this field has progressed, increasing efforts have been dedicated to enhancing the real-time performance, accuracy, and generalization capability of obstacle avoidance systems to meet the demands of complex and variable environments (Raghvendra et al., 2024). Notably, the emergence of transformer-based reinforcement learning approaches, including Decision Transformers (Chen et al., 2021) and Trajectory Transformers (Janner et al., 2021), has introduced sequence modeling paradigms to RL that show promise for handling temporal dependencies in robotic decision-making. These attention-based architectures have demonstrated effectiveness in offline RL settings and long-horizon planning tasks, though their application to real-time obstacle avoidance remains limited due to computational constraints and the need for

specialized spatial reasoning mechanisms. From the evolution of recent work, it is evident that research paradigms have gradually expanded from rule-driven approaches to data-driven, structure-integrated, and task-specialized directions, with transformer-based methods representing the latest frontier in end-to-end learning approaches.

Firstly, explorations into lightweight visual perception are of significant practical value. For instance, the method proposed in Misir and Celik (2025), based on an improved MobileNetV2 for visual obstacle avoidance, designed a neural network with low parameter volume, combined with a custom dataset integrating color intensity and ultrasonic data, achieving high recognition accuracy and successful deployment on a mobile robot platform (Zhang et al., 2022). This demonstrates considerable practicality and engineering feasibility. However, the closed nature of its experimental scenes and data construction limits the model's generalization ability in complex and dynamic environments. Furthermore, it lacks quantitative analysis regarding algorithmic latency and computational load, posing reliability challenges for real-world applications. Meanwhile, task-specialized pathways for robotic obstacle avoidance have also demonstrated notable advantages. For example, Jian et al. (2025) focused on surgical robotics and proposed the STV-PDNN structure, integrating velocity control optimization and null-space redundancy control, achieving high-precision and stable trajectory control during surgical procedures with excellent policy robustness and trajectory continuity. However, its validation scope is narrow, confined to specific procedures and structures, lacking comprehensive evaluations in diversified anatomical scenarios and effective strategies for handling dynamic obstacles and sensor interference. From the perspective of control and trajectory optimization integration, Li et al. (2025b) and Wang et al. (2025b) proposed trajectory tracking methods combining Model Predictive Control (MPC) with Sliding Mode Control, and an integrated planning-tracking obstacle avoidance framework, respectively. Both approaches improved system performance through soft and hard constraint modeling, optimized planning, and controller design. Notably, Wang et al. (2025b) enhanced system adaptability in unstructured environments by introducing dynamic obstacle intention modeling into optimization objectives, breaking the conventional hierarchical design. Nevertheless, both studies suffer from limited application scenarios, simplified dynamic obstacle modeling, and incomplete complexity analysis, which restrict their scalability to real-world multi-target, multi-obstacle interactive environments. In the field of coordinated obstacle avoidance for multiple manipulators in open environments, Wu et al. (2025) proposed a dual-arm collaborative obstacle avoidance method, incorporating obstacle classification mechanisms and avoidance direction adjustment strategies. It exhibited strong flexibility and collision avoidance capabilities during practical task execution, especially for fine-grained obstacle avoidance control in complex operation scenarios. However, this method has yet to adequately address the complexity and dynamic diversity of obstacles in extreme environments. It also lacks in-depth investigation into system robustness under sensor errors and has not conducted quantitative comparisons with mainstream methods, affecting the comprehensiveness of its practical value verification.

To systematically position our contribution, we identify four key limitations in existing approaches: (1) Perception Limitations: Most methods rely on CNN-based local feature extraction or simple attention mechanisms that fail to capture long-range spatiotemporal dependencies crucial for dynamic obstacle prediction, (2) Integration Gaps: Traditional approaches maintain separation between perception and decision modules, leading to information loss and suboptimal policies, (3) Temporal Modeling Deficiencies: Existing methods either ignore temporal dynamics entirely or use limited memory mechanisms (LSTM) that suffer from gradient vanishing in long sequences, and (4) Scalability Issues: Many approaches are tested only in simplified scenarios and lack comprehensive evaluation across diverse environments. Our method addresses these limitations through: unified spatiotemporal modeling via Transformer architecture, complete end-to-end integration eliminating information bottlenecks, comprehensive temporal dependency modeling without gradient issues, and extensive validation across multiple challenging environments.

Based on our systematic analysis, while existing works demonstrate progress in individual components, they exhibit fundamental limitations in achieving true end-to-end spatiotemporal modeling for dynamic obstacle avoidance in the practical deployment of obstacle avoidance algorithms, control precision, and task adaptability, revealing a multi-level integration trend from visual perception and motion control to policy optimization. However, several unresolved issues remain: most methods have been validated in limited environments, lacking systematic evaluations for dynamic, unstructured, and multi-obstacle interactive scenarios; obstacle avoidance systems often decouple perception and decision-making, leading to response delays and increased system complexity; and some lightweight or end-to-end solutions lack coordinated optimization of hardware resources and real-time performance. Against this backdrop, this paper aims to develop an end-to-end deep reinforcement learning obstacle avoidance framework integrating a spatiotemporal attention mechanism. Leveraging the advantages of the Transformer architecture in sequential modeling and global perception, combined with the adaptive learning capability of policy optimization, this study addresses the decision-making challenges of obstacle avoidance in densely dynamic environments, filling the current research gap between sequential modeling capacity, dynamic adaptability, and deployment-friendliness.

Although considerable achievements have been made in SLAM and DRL, VSLAM and image processing, as well as path planning and control strategies, there is still a lack of a method capable of integrating environment perception, spatiotemporal sequence modeling, and autonomous policy learning into a unified, end-to-end obstacle avoidance framework without relying on explicit mapping. Existing methods often adopt decoupled designs for perception, mapping, and control, resulting in complex systems, difficult deployments, and delayed responses. To address this research gap, this paper proposes an end-to-end robot intelligent obstacle avoidance method based on deep reinforcement learning and a spatiotemporal Transformer architecture. This approach employs Transformer as the backbone model, incorporating spatial attention mechanisms to enhance environmental obstacle modeling and integrating DRL for self-optimized policy learning,

aiming to achieve an efficient, robust, and generalizable robotic obstacle avoidance system in dynamic environments.

3 Method

In the data collection phase, NH-ORCA (Non-Holonomic Optimal Reciprocal Collision Avoidance) algorithm is employed for multi-robot collision avoidance to generate training trajectories that account for the kinematic constraints of wheeled mobile robots. The overall architecture is shown in Figure 1.

Figure 1 illustrates the complete workflow of our proposed end-to-end obstacle avoidance system, which consists of three interconnected phases: dataset collection, training, and inference. In the dataset collection phase (left panel), we generate diverse training scenarios through two pathways: single robot exploration using frontier exploration techniques to create basic navigation data $\{g_i, \tau_i\}$, and multi-robot collision avoidance scenarios using NH-ORCA (Non-Holonomic Optimal Reciprocal Collision Avoidance) algorithm to produce complex interaction data $\{g_i^{\text{ca}}, \tau_i^{\text{ca}}\}$. Additionally, we collect RGB image pairs with and without dynamic obstacles $\{o_i^s, o_i^d\}$ to form our comprehensive dataset $\mathcal{D}_{\text{ssl}}$. The training phase (top right) integrates our NavFormer architecture with both offline reinforcement learning and self-supervised learning mechanisms, where the Causal Transformer processes the collected trajectories and visual encoders handle the multimodal sensory inputs. The inference phase (bottom right) deploys the trained NavFormer model in real-time, taking current observations (o, r) and generating appropriate actions (a) for multi-robot cluttered dynamic environments. The bidirectional arrows indicate the iterative feedback between training and data collection, enabling continuous improvement of the obstacle avoidance policy.

3.1 Deep Q-network

The end-to-end architecture integrates data collection, feature extraction, and decision-making in a unified learning framework, eliminating the need for separate perception and planning modules typical in conventional robotic systems.

Our implementation incorporates several key extensions to the standard DQN framework to adapt it to spatio-temporal obstacle avoidance tasks. When integrated with Transformer-based feature extraction, the Q network processes high-dimensional spatio-temporal features rather than raw observation data. The modified experience replay algorithm maintains the sequence dependencies required for dynamic obstacle tracking by preserving temporal consistency. Therefore, by considering obstacle distance-adaptive greedy exploration for safer exploration and multi-scale reward shaping, we simultaneously balance immediate collision avoidance and long-term navigation efficiency.

To achieve efficient and intelligent obstacle avoidance for mobile robots in complex and dynamic environments, we adopt the Deep Q-Network (DQN) as the core policy learning algorithm. On this basis, a spatiotemporal perception mechanism is integrated to construct an intelligent obstacle avoidance system that unifies perception and decision-making. The DQN algorithm approximates the state-action value function through a deep

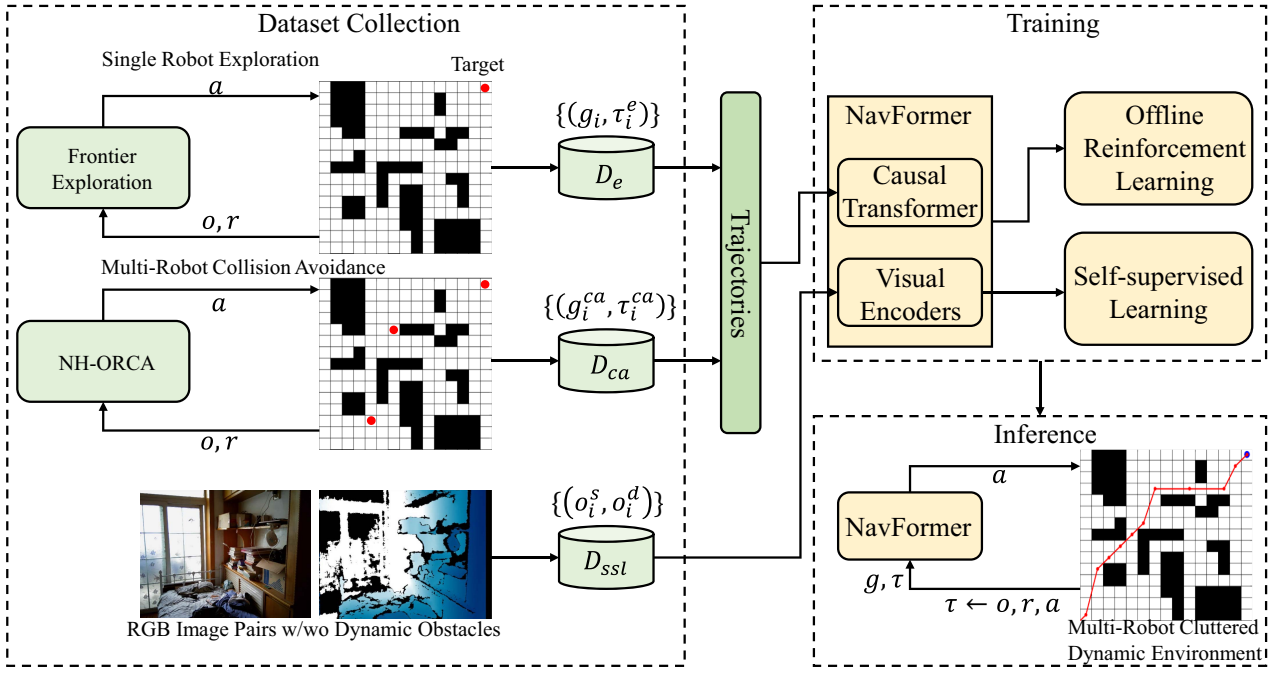

FIGURE 1
Overall algorithm architecture. NH-ORCA (Non-Holonomic Optimal Reciprocal Collision Avoidance) generates training data for multi-robot collision avoidance scenarios.

neural network, effectively addressing the limitations of traditional Q-learning in high-dimensional state spaces while offering strong policy learning capabilities and environmental adaptability. Our enhanced DQN architecture integrating spatiotemporal Transformer features is detailed in Figure 2, where the Transformer module processes multimodal observations into high-dimensional representations that enable the Q-network to make globally-informed decisions.

In our approach, the observation space $\mathcal{S}$ encompasses multi-modal sensory inputs including RGB images, depth maps, and robot kinematic states. Specifically, each state $s_t \in \mathcal{S}$ is defined as $s_t = \{I_t^{\text{rgb}}, I_t^{\text{depth}}, p_t, v_t, \theta_t\}$, where $I_t^{\text{rgb}} \in \mathbb{R}^{H\times W\times 3}$ represents the RGB observation, $I_t^{\text{depth}} \in \mathbb{R}^{H\times W\times 1}$ denotes the depth information, and $\{p_t, v_t, \theta_t\}$ capture the robot's position, velocity, and orientation respectively.

As illustrated in Figure 2, our architecture processes multimodal inputs through the Transformer module to generate embedded patches, which are then fed into the modified Q-network. The patch + position embedding corresponds to our spatiotemporal feature integration, while the multi-head attention mechanism enables selective focus on critical obstacles. The final $Q(s, a)$ output directly maps to discrete navigation actions.

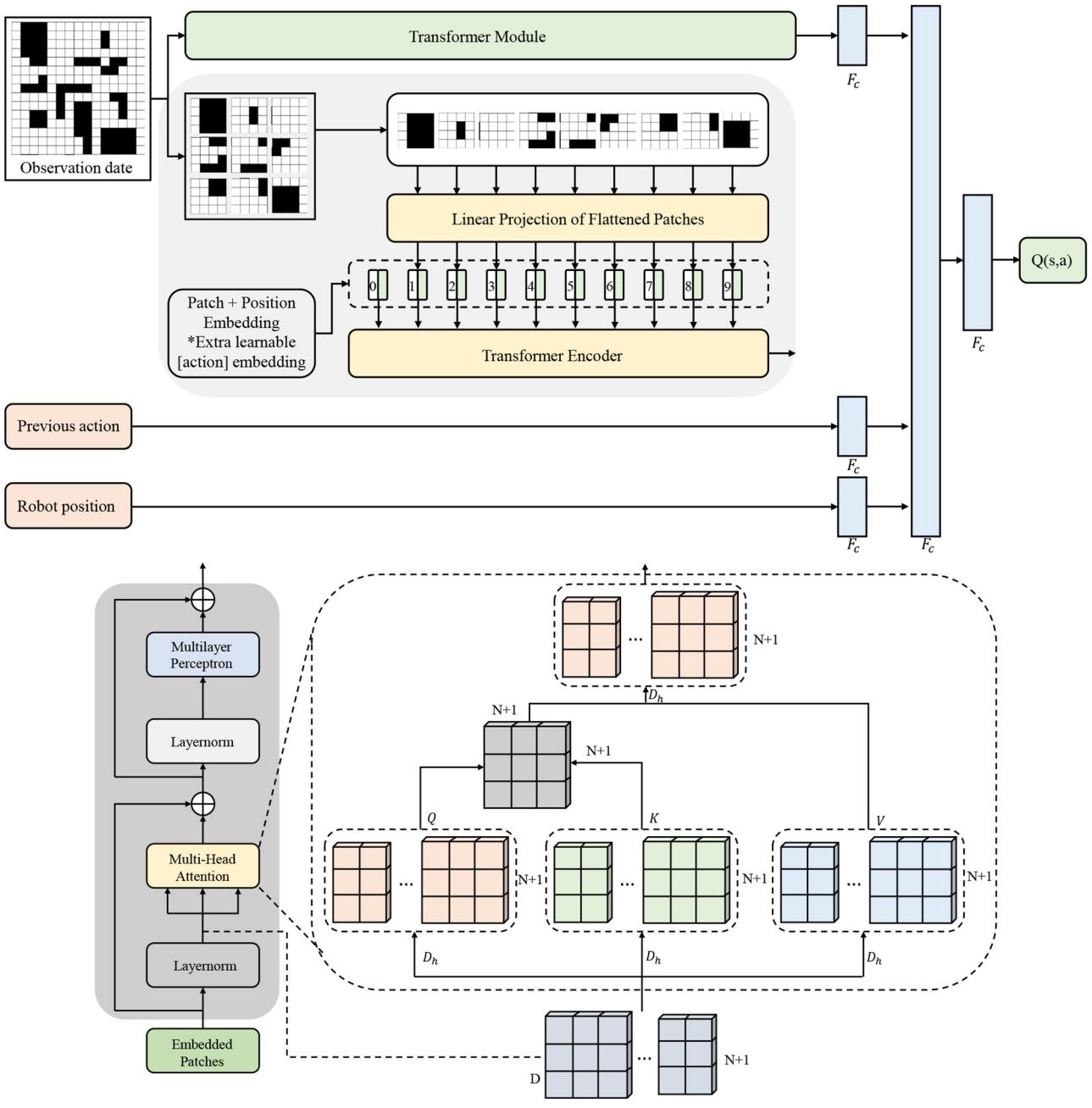

FIGURE 2
Transformer architecture diagram.

Within the reinforcement learning framework, the robot obstacle avoidance task can be modeled as a Markov Decision Process (MDP), denoted as a quintuple:

$$M = \langle S, A, P, R, \gamma \rangle \tag{1}$$

where $\mathcal{S}$ represents our enhanced state space that integrates spatiotemporal features extracted by the Transformer encoder, specifically $\mathcal{S} = \{Z'', p_t, v_t, \theta_t\}$ where Z'' denotes the Transformer output features and $\{p_t, v_t, \theta_t\}$ represent robot kinematic states; $\mathcal{A}$ denotes our discrete action space $\mathcal{A} = \{\text{forward}, \text{left}, \text{right}, \text{stop}\}$ designed for safe navigation in dynamic environments. $P(s'|s, a)$ is the state transition probability, describing the probability of transitioning to the next state s' after executing action a in state s; $R(s, a)$ is the reward function, defining the feedback obtained after performing an action; $\gamma \in [0, 1]$ is the discount factor, used to balance long-term returns and immediate rewards.

To handle the high-dimensional observation space effectively, we employ a hierarchical feature extraction strategy where convolutional layers process visual inputs while fully connected layers handle kinematic states. The spatiotemporal attention mechanism (detailed in Section 3.2) then selects the most relevant features across both spatial and temporal dimensions for decision-making.

In practice, our adapted DQN processes the Transformer output features Z (from Equation 24) as state representations, enabling the Q-network to make decisions based on globally-aware spatiotemporal features rather than local observations. The training process incorporates curriculum learning where obstacle complexity gradually increases, and the replay buffer maintains temporal windows to preserve sequential dependencies essential for dynamic obstacle prediction

In DQN, a deep neural network $Q_\theta(s, a)$ is employed to approximate the Q-function, where θ denotes the network parameters. The network takes the current state (e.g., image) as input and outputs Q-value estimates for each possible action. Training is performed by minimizing the mean squared error loss function:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(y_t - Q_\theta(s, a)\right)^2\right] \tag{2}$$

where the target Q-value y_t is defined as:

$$y_t = r + \gamma \max_{a'} Q_{\theta^-}(s', a') \tag{3}$$

Here, θ^- represents the parameters of a target network, which are periodically copied from the current network parameters θ to stabilize training.

To further improve the stability of value estimation and the convergence speed of the policy, we adopt a decoupled action selection and evaluation process. The target value is modified as follows:

$$y_t = r + \gamma Q_{\theta^-}(s', \arg\max_{a'} Q_\theta(s', a')) \tag{4}$$

Our reward function $R(s, a)$ is designed to encourage safe navigation while maintaining efficiency:

$$R(s, a) = R_{\text{goal}} + R_{\text{collision}} + R_{\text{progress}} + R_{\text{smoothness}} \tag{5}$$

where R_{goal} provides positive reward for reaching the target, $R_{\text{collision}}$ heavily penalizes collisions (-100), R_{progress} rewards forward movement toward the goal, and $R_{\text{smoothness}}$ encourages stable control actions to avoid oscillatory behavior.

Additionally, considering that different actions do not always have significant differences in certain states, we decompose the Q-value function into a state-value function $V(s)$ and an advantage function $A(s, a)$, formulated as:

$$\begin{aligned} Q(s, a; \theta, \alpha, \beta) &= V(s; \theta, \beta) \\ &+ \left(A(s, a; \theta, \alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s, a'; \theta, \alpha)\right) \end{aligned} \tag{6}$$

In this formulation, θ represents the shared parameters of the convolutional layers; α corresponds to the parameters of the advantage function branch; β corresponds to the parameters of the state-value function branch.

The DQN framework follows standard implementation with experience replay and target networks. We adopt Double DQN and Dueling DQN variants to improve learning stability and value estimation accuracy.

3.2 Spatiotemporal attention mechanism

In the end-to-end intelligent obstacle avoidance method proposed in this paper, a spatiotemporal attention mechanism is introduced as a key module to enhance the robot's perception of the temporal evolution and spatial distribution of obstacles in dynamic environments. Unlike traditional static image-based perception, the information received by a robot during real-world obstacle avoidance is characterized by significant temporal dependencies and spatial locality. Changes in obstacle positions, motion trends, and occlusion relationships often require joint modeling of both temporal and spatial dimensions. To address this, we design a spatiotemporal fusion model based on the Self-Attention mechanism, which extracts critical dynamic behavioral features from sequential historical states and models both local structures and global dependencies in the spatial dimension, thereby comprehensively improving the robot's understanding of complex scenarios and the robustness of its decision-making policy. The architecture is shown in Figure 3.

Figure 3 illustrates the detailed architecture of our Spatiotemporal Attention (SPA) mechanism, which consists of three main components: Co-attention Gate, Motion Guided Attention Module, and Decoder Block. The Co-attention Gate takes feature maps F^a and F^m as inputs, where F^a represents appearance features and F^m denotes motion features extracted from consecutive frames. The gate mechanism uses average pooling and 1×1 convolutions followed by sigmoid activation to generate attention weights g^a and g^m, which modulate the

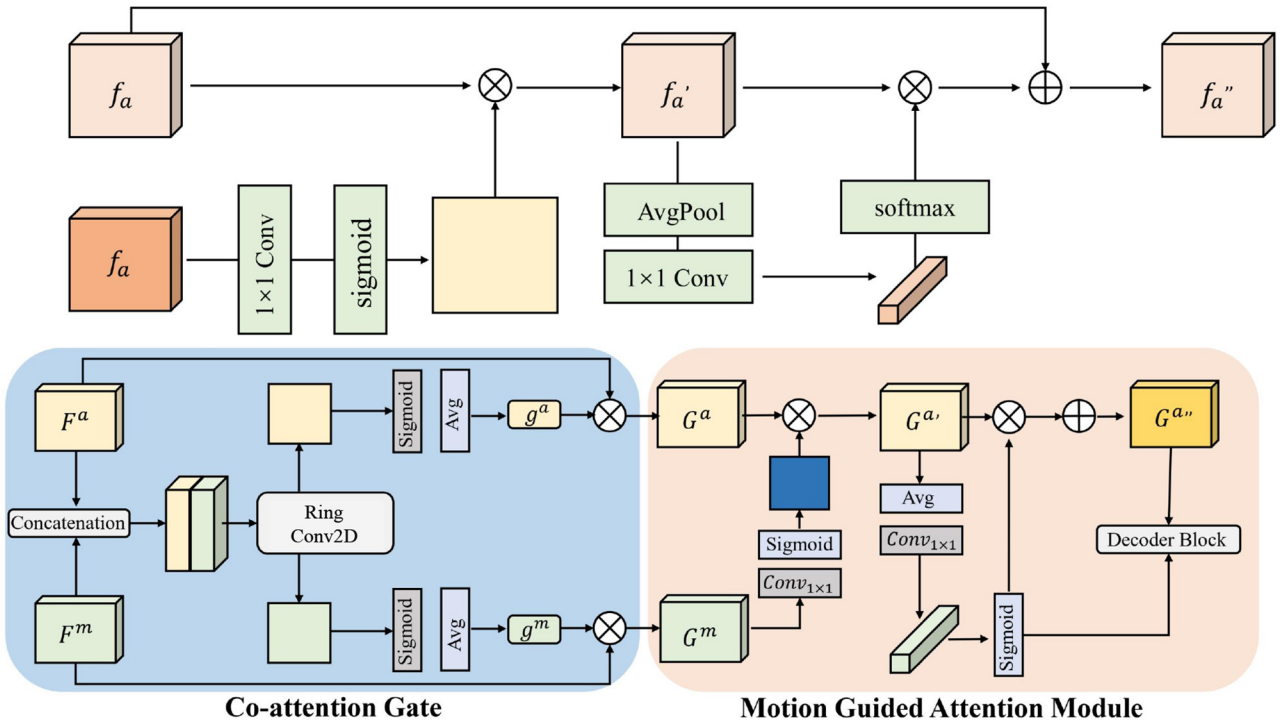

FIGURE 3
Architecture diagram of SPA.

importance of appearance and motion information respectively. The Motion Guided Attention Module further refines these features through ring convolution operations and sigmoid gating to produce enhanced spatial-temporal representations G^a and G^m. Finally, the Decoder Block integrates these multi-scale features and outputs the final attended representation f^{a*} that captures both spatial dependencies and temporal dynamics essential for obstacle avoidance decision-making.

At each time step t, the robot obtains a sequence of continuous observation image frames of length T through its sensors, denoted as:

$$X = \{x_{t-T+1}, x_{t-T+2}, \ldots, x_t\},\ x_i \in \mathbb{R}^{H\times W\times C} \tag{7}$$

where H, W, and C represent the height, width, and number of channels of each image, respectively. For unified processing, each image frame is first embedded into a fixed-dimensional feature vector through convolutional layers or an encoder:

$$F = \{f^a, f^m\} \tag{8}$$

where f^a represents appearance features and f^m denotes motion features as shown in Figure 3, with each feature vector $f_i \in \mathbb{R}^d$

The feature sequence is then input into the spatiotemporal attention module for processing. To jointly model temporal dependencies and spatial perception, a dual-attention mechanism is introduced, consisting of a Temporal Attention module and a Spatial Attention module, defined as follows.

Temporal attention aims to capture key temporal features from the historical state sequence that influence current decision-making. For each time step $i \in [t-T+1, t]$, query, key, and value vectors are obtained through linear transformations:

Following the Co-attention Gate module in Figure 3, we compute gated features:

$$g^a = \sigma(\mathrm{Conv}_{1\times1}(\mathrm{AvgPool}(f^a))) \odot f^a \tag{9}$$
$$g^m = \sigma(\mathrm{Conv}_{1\times1}(f^m)) \odot f^m \tag{10}$$

All time steps are then organized into matrix form:

$$Q_T = [q_{t-T+1}, \ldots, q_t]^T, \quad K_T = [k_{t-T+1}, \ldots, k_t]^T, \quad V_T = [v_{t-T+1}, \ldots, v_t]^T \tag{11}$$

The temporal attention output is computed as:

$$\mathrm{Attn}_{time}(F) = \mathrm{softmax}\left(\frac{Q_T K_T^T}{\sqrt{d_k}}\right) V_T \tag{12}$$

In our formulation, f^a represents appearance features, f^m denotes motion features, G^a and G^m are the corresponding gated attention maps, and the operations $\odot$ and $\oplus$ represent element-wise multiplication and addition respectively. This process achieves weighted aggregation of key historical moments, strengthening the robot's understanding of obstacle motion trends and behavior patterns.

For spatial attention modeling, at each time step i, based on the image feature map $f_i \in \mathbb{R}^{H\times W\times d}$, positional embeddings and local window partitioning are applied to extract spatial information. Denoting the embedding at each spatial position as $f_i^{(h,w)} \in \mathbb{R}^d$, the attention computation is as follows:

$$q_{h,w} = W_Q^{(s)} f_i^{(h,w)}, \quad k_{h',w'} = W_K^{(s)} f_i^{(h',w')}, \quad v_{h',w'} = W_V^{(s)} f_i^{(h',w')} \tag{13}$$

The corresponding spatial attention output is:

$$\mathrm{Attn}_{space}(f_i)_{h,w} = \sum_{(h',w')\in\mathcal{N}(h,w)} \alpha_{(h,w),(h',w')} \cdot v_{h',w'} \tag{14}$$

with the attention weight computed by:

$$\alpha_{(h,w),(h',w')} = \frac{\exp\left(q_{h,w}\cdot k_{h',w'}/\sqrt{d_k}\right)}{\sum_{(h'',w'')\in\mathcal{N}(h,w)} \exp\left(q_{h,w}\cdot k_{h'',w''}/\sqrt{d_k}\right)} \tag{15}$$

where $\mathcal{N}(h,w)$ denotes the neighborhood region centered at (h,w).

After obtaining the temporal attention output $\hat{F}_i = \mathrm{Attn}_{time}(F)$ and the spatial attention output $\hat{f}_i^{space}$ for each frame, a spatiotemporal fusion module is introduced to aggregate the spatial features of key frames in the time series:

$$\widetilde{f}_t = \sum_{i=t-T+1}^{t} \lambda_i \cdot \mathrm{Flatten}\left(\hat{f}_i^{space}\right), \quad \lambda_i = \mathrm{softmax}(\phi(\hat{f}_i)) \tag{16}$$

where $\phi(\cdot)$ is an attention scoring function used to compute fusion weights. The final fused feature $\widetilde{f}_t$ is then fed into the policy network or reinforcement learning decision module.

In summary, the spatiotemporal attention mechanism introduced in this paper provides essential perception and modeling support for intelligent obstacle avoidance by mobile robots in complex dynamic environments. By jointly modeling the behavioral evolution features in the temporal dimension and obstacle distribution information in the spatial dimension, it addresses the limitations of traditional perception methods in terms of locality, sequential dependency modeling, and global consistency representation. Temporal attention dynamically captures critical historical states, revealing obstacle movement trends and potential risk changes, while spatial attention precisely focuses on key local regions in the current observation frame, enabling fine-grained recognition of multiple obstacles and multi-scale targets. The combination of the two not only enhances the robot's overall environmental understanding but also provides higher-quality, decision-relevant semantic feature inputs for downstream policy networks, thereby significantly improving the robustness and generalization capability of obstacle avoidance strategies in complex, changing scenarios. Theoretically, this mechanism exhibits excellent scalability and modularity, allowing it to naturally integrate with deep reinforcement learning frameworks and build a unified end-to-end perception-decision system. It thus lays a solid foundation for the development of high-performance, adaptive robotic navigation systems.

3.3 Transformer architecture

The key distinction of our "end-to-end" transformer lies in its architectural adaptations for direct sensor-to-action mapping: (1) Input Integration: Unlike conventional transformers that process single-modality sequential data, our architecture simultaneously handles RGB images, depth information, and kinematic states through unified patch embeddings, (2) Attention Mechanism: We modify the self-attention to incorporate spatial proximity bias for obstacle-aware feature selection, (3) Output Decoder: The final layer directly outputs Q-values for discrete actions rather than requiring separate decision-making modules, and (4) Training Objective: The entire pipeline is optimized end-to-end using reinforcement learning signals rather than supervised learning on intermediate representations. As the core module for perception and modeling, the Transformer architecture is employed to uniformly process sequential input data from multimodal sensors. With its powerful global modeling capability and multi-head self-attention mechanism, it effectively extracts key spatiotemporal features that influence robot obstacle avoidance behavior. Compared to traditional Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), the Transformer architecture can model long-term dependencies in parallel and maintains strong feature alignment capabilities in both spatial and temporal dimensions, making it particularly well-suited for obstacle behavior modeling and multisource information fusion in dynamic environments. The architecture is shown in Figure 4.

Within a temporal window of length T, the robot collects a sequence of multimodal perception frames:

$$X = \{x_1, x_2, \ldots, x_T\},\ x_t \in \mathbb{R}^{H\times W\times C} \tag{17}$$

where each x_t represents the image/depth frame obtained at time t, and H, W, and C denote the height, width, and number of channels of the frame, respectively. Each frame is first subjected to feature extraction and flattening, followed by a linear transformation to obtain a unified-dimensional embedding representation:

$$z_t = \mathrm{Flatten}(x_t)\cdot W_E + b_E,\ z_t \in \mathbb{R}^d \tag{18}$$

where $W_E \in \mathbb{R}^{(H\times W\times C)\times d}$ is the embedding matrix and $b_E \in \mathbb{R}^d$ is the bias vector. To clarify the architectural integration: the ResNet-based feature extractor (Figure 4) serves as the visual encoder that processes raw sensor inputs into compact spatial representations $z_t \in \mathbb{R}^{512}$. These features are then temporally organized and fed into our Transformer encoder (described below) to capture spatiotemporal dependencies. The Transformer output Z is subsequently processed by the Dueling DQN network (also shown in Figure 4) to generate action decisions. To preserve temporal information, a positional encoding $p_t \in \mathbb{R}^d$ is introduced, and the final input sequence to the Transformer is:

$$Z = \{z_1 + p_1, z_2 + p_2, \ldots, z_T + p_T\} \tag{19}$$

The Transformer encoder consists of multiple stacked layers, each composed of a Multi-Head Self-Attention mechanism and a

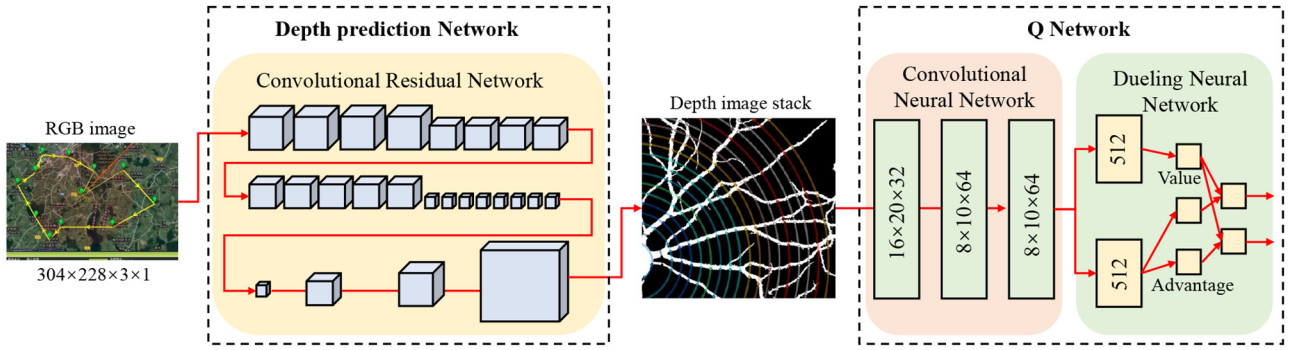

FIGURE 4
Architecture diagram of DQN.

Feed Forward Network (FFN). The core computation within each layer involves three main steps:

For each layer input $Z = \{z_1, z_2, \ldots, z_T\}$, query, key, and value vectors are generated through linear projections:

$$Q = ZW^Q, \quad K = ZW^K, \quad V = ZW^V, \; W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k} \tag{20}$$

The attention matrix is then computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{21}$$

To enhance the model's representation capability, h attention heads are introduced, with each head corresponding to different linear projections:

$$\text{MultiHead}(Z) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{22}$$

where:

$$\text{head}_i = \text{Attention}(ZW_i^Q, ZW_i^K, ZW_i^V), \; W^O \in \mathbb{R}^{hd_v \times d} \tag{23}$$

After each sublayer, residual connections and Layer Normalization are applied:

$$Z' = \text{LayerNorm}(Z + \text{MultiHead}(Z)) \tag{24}$$

$$Z'' = \text{LayerNorm}(Z' + \text{FFN}(Z')) \tag{25}$$

The Feed Forward Network (FFN) is defined as:

$$\text{FFN}(x) = \sigma(xW_1 + b_1)W_2 + b_2 \tag{26}$$

where $W_1 \in \mathbb{R}^{d \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d}$.

To further enhance the Transformer's ability to model temporal continuity and spatial distribution characteristics, this paper introduces a position-sensitive spatiotemporal weighting mask mechanism within the multi-head attention module. During the attention weight computation, positions with larger spatial or temporal distances are assigned lower weights:

$$\alpha_{ij} = \frac{\exp\left(\frac{q_i \cdot k_j}{\sqrt{d_k}} - \lambda \cdot \delta_{ij}\right)}{\sum_{j'} \exp\left(\frac{q_i \cdot k_{j'}}{\sqrt{d_k}} - \lambda \cdot \delta_{ij'}\right)} \tag{27}$$

where δ_{ij} represents the spatiotemporal distance between positions i and j, and λ is a decay factor used to suppress interference from distant positions. In the final encoded output Z'', each vector carries rich semantic information from multiple time steps and spatial locations, serving as the state input to the reinforcement learning policy network to guide the robot's obstacle avoidance decision-making.

This contrasts sharply with traditional robotic systems that employ separate modules for perception (CNN-based feature extraction), mapping (SLAM), planning (A* or RRT), and control (PID controllers), each requiring manual tuning and intermediate data conversion. Our end-to-end approach eliminates these boundaries by learning a unified representation that directly maps sensory observations to optimal actions through the transformer's global attention mechanism.

In summary, as a sequence modeling method based on the self-attention mechanism, the Transformer architecture offers an efficient, unified perception and decision-making modeling paradigm for robot obstacle avoidance tasks through its exceptional global dependency modeling capability and parallel computing advantages. By dynamically assigning weights to relationships between time steps and spatial positions in the input sequence, the Transformer adaptively focuses on the most critical spatiotemporal feature information for obstacle avoidance decision-making, significantly enhancing the robot's understanding of dynamic environments and response speed. By integrating the Transformer architecture into the reinforcement learning framework, this study constructs an end-to-end unified learning pipeline from raw perception to behavioral decision-making. This approach also provides a theoretically rigorous and practically efficient solution for multimodal information fusion, multi-object dependency modeling, and decision policy optimization in complex environments. Therefore, the Transformer is not

only a perception enhancement module but also a fundamental component for improving the synergy, generalization, and robustness of autonomous robot obstacle avoidance systems.

4 Experiment

4.1 Experimental setup and dataset

To verify the effectiveness and generalization capability of the proposed method, all experiments were conducted on a unified software and hardware platform to ensure the comparability and reproducibility of different approaches under the same environment. All models were implemented based on PyTorch and trained on a high-performance workstation running Ubuntu 20.04, equipped with an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM) and an Intel Core i9-13900K processor. For model training, the Adam optimizer was employed to update the parameters of the policy network, with an initial learning rate set to 1e-4 and a batch size of 64. The target network was synchronized every 1,000 steps. The training process consisted of 1,000 epochs, and a dropout rate of 0.1 was applied to prevent overfitting.

This work systematically conducted experiments and performance evaluations of the proposed end-to-end robot intelligent obstacle avoidance method using three interactive simulation environments (RoboTHOR, CARLA, TurtleBot3) for data generation and one real-world dataset (EuRoC MAV) for validation. These cover a wide range of application scenarios, from static indoor environments to dynamic outdoor scenes, and from ground robots to aerial platforms, thereby verifying the model's generalization ability and robustness under multi-modal input and complex spatiotemporal conditions. To clarify our experimental methodology: RoboTHOR, CARLA, and TurtleBot3 serve as simulation environments where we generate synthetic training and testing data by running our proposed algorithm in various scenarios. The EuRoC MAV represents a real-world dataset containing pre-recorded multimodal sensor data from actual aerial vehicle flights. This hybrid approach allows us to evaluate our method's performance across both controlled simulation conditions and real-world data complexities. The four datasets used are introduced as follows:

- RoboTHOR

  Developed by the Allen Institute for AI, RoboTHOR (Deitke et al., 2020) is an interactive visual navigation and obstacle avoidance simulation platform that provides highly realistic 3D indoor home environments. It supports robots performing navigation and obstacle avoidance tasks in multi-room structures. The platform integrates various sensory inputs (such as RGB images, depth maps, and semantic labels) and control outputs, making it suitable for training and evaluating end-to-end vision-control integrated models. In this study, a multi-scene indoor obstacle avoidance test set was constructed using RoboTHOR to evaluate the path planning accuracy and local obstacle avoidance capability of the proposed method in static and mildly dynamic indoor environments.
- CARLA

  CARLA (Gannamaneni et al., 2021) is an open-source autonomous driving simulator built on Unreal Engine, widely used for research on navigation and obstacle avoidance in urban road environments. It provides a realistic simulation of dynamic elements such as traffic flow, traffic lights, pedestrians, and vehicles, and supports multi-sensor simulation (RGB, LiDAR, IMU, GPS, etc.). In this study, CARLA was employed to build highly dynamic complex obstacle scenarios to examine the model's spatiotemporal modeling ability and obstacle avoidance robustness in urban environments with dynamic obstacles and sudden disturbances.
- TurtleBot3

  TurtleBot3 (Kashyap and Konathalapalli, 2025) is a compact mobile robot platform officially supported by ROS, whose Gazebo simulation model is commonly used for teaching and research on tasks such as path planning, navigation, and obstacle avoidance. Its clean and controllable simulation environment is ideal for training and debugging reinforcement learning models. In this work, the proposed method was trained and evaluated across multiple TurtleBot3 indoor maps. Taking advantage of its precise control model and high reproducibility, the algorithm's path efficiency and control stability in typical indoor environments were verified.
- EuRoC MAV Dataset

  The EuRoC MAV Dataset (Burghoffer et al., 2023) is a multi-modal navigation dataset for Micro Aerial Vehicles (MAVs), provided by the European Robotics Research Center. It includes real aerial photography data from multiple industrial and indoor scenarios, combining high-frequency image frames, IMU sensor data, and ground-truth pose labels. Widely used in tasks such as VIO, SLAM, and obstacle avoidance, this study selected several challenging scenes from this dataset to test the obstacle avoidance decision performance of the proposed method on aerial platforms, with particular focus on strategy adaptability and sequential modeling ability under highly dynamic, high-degree-of-freedom movements.

4.2 Evaluation metrics

To comprehensively assess the performance of the proposed deep reinforcement learning-based intelligent obstacle avoidance method incorporating a spatiotemporal Transformer architecture, four key evaluation metrics were selected from multiple dimensions, including obstacle avoidance capability, path efficiency, and task response time. These metrics effectively reflect the safety, intelligence, and practicality of the model in different environments and are suitable for obstacle avoidance tasks in dynamic and complex scenarios.

Our comprehensive hyperparameter configuration includes DQN parameters with learning rate $\alpha = 1e-4$ (selected from tested range $1e-5$ to $1e-3$), discount factor $\gamma = 0.99$, experience replay buffer size of 100,000, target network updates every $1,000$ steps, and ε-greedy exploration decaying from 1.0 to 0.01 over

50,000 steps; Transformer architecture with hidden dimension $d = 512$, 8 attention heads, 6 encoder layers, feedforward dimension $d_{\text{ff}} = 2048$, dropout rate of 0.1, and maximum positional encoding length of 1,000; spatiotemporal attention mechanism using temporal window $T = 10$ frames, 7×7 spatial neighborhood, attention temperature $\tau = 0.1$, and fusion weight decay $\lambda = 0.001$; and training configuration employing batch size 64, Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), weight decay $1e-4$, gradient clipping threshold 1.0, and exponential learning rate decay of 0.95 every 100 epochs. Systematic sensitivity analysis across all parameters revealed robust performance within $\pm 20\%$ hyperparameter variations, with learning rate and attention head number identified as the most critical factors affecting convergence speed and final performance respectively.

- Obstacle Avoidance Success Rate (OASR)

  This metric measures whether the robot successfully avoids all obstacles and reaches the target point during the task. It reflects the overall obstacle avoidance capability and strategy effectiveness of the system.

$$Success\ Rate(OASR) = \frac{N_{success}}{N_{total}} \times 100\% \tag{28}$$

  where $N_{success}$ is the number of successful, collision-free task completions, and N_{total} is the total number of experiments. In this work, OASR is used to compare the safety navigation capability of different methods in complex dynamic environments and serves as a core metric for strategy reliability evaluation.

- Collision Rate (CR)

  The Collision Rate measures the frequency of collisions occurring during the navigation process, serving as a negative indicator for obstacle avoidance robustness and safety. It is defined as:

$$Collision\ Rate(CR) = \frac{N_{collision}}{N_{total}} \times 100\% \tag{29}$$

  where $N_{collision}$ is the number of tasks in which at least one collision occurred. The proposed strategy enhances the perception of dynamic obstacles through the spatiotemporal modeling capability of the Transformer architecture, thereby reducing collision rates at the control level and improving system safety.

- Average Path Length

  This metric reflects the spatial efficiency of the path chosen by the robot from the starting point to the target point. Shorter paths generally indicate more optimized strategies.

$$Average\ Path\ Length = \frac{1}{N_{success}} \sum_{i=1}^{N_{success}} L_i \tag{30}$$

  where L_i represents the total path length in the i^{th} successful navigation, and $N_{success}$ is the number of successful task completions. In this work, this metric is used to compare the global efficiency of different strategies in path planning, with particular attention to path convergence performance under dynamic obstacle interference.

- Average Navigation Time

  The Average Navigation Time assesses the robot's response speed and execution efficiency in completing the task, which is closely related to control decision latency and the stability of obstacle avoidance behavior.

$$Average\ Navigation\ Time = \frac{1}{N_{success}} \sum_{i=1}^{N_{success}} T_i \tag{31}$$

  where T_i represents the time taken to successfully complete the i^{th} task, and $N_{success}$ is the number of successful task completions. The proposed end-to-end architecture reduces intermediate processing time from perception to control and improves the strategy's foresight through the global modeling capability of the Transformer, effectively shortening the task response time.

4.3 Experimental results analysis

We compare the proposed algorithm with representative baseline methods that encompass different paradigms in the field of obstacle avoidance based on deep reinforcement learning. A bio-inspired multi-underwater spherical robot control system employs a mobile obstacle avoidance strategy using a collaborative formation mode. This method combines a perception module based on convolutional neural networks (CNN) with standard deep reinforcement learning (DQN) for multi-robot coordination, representing a traditional vision-based reinforcement learning approach. A fast finite-time binary formation control method with obstacle avoidance functionality, suitable for multi-agent systems with delays. This method employs a distributed consensus algorithm combined with A3C reinforcement learning, focusing on multi-agent coordination in delayed environments. Baseline methods are implemented using their original hyperparameters (if available) or carefully tuned to ensure optimal performance in the experimental environment.

To comprehensively verify the effectiveness of the proposed end-to-end robot intelligent obstacle avoidance method based on deep reinforcement learning and a spatiotemporal Transformer architecture, this section systematically analyzes and compares the experimental results on four different types of datasets (RoboTHOR, CARLA, TurtleBot3, EuRoC MAV). We evaluate the model's performance under multi-scene and multi-task conditions from four key metrics: obstacle avoidance success rate (OASR), collision rate (CR), average path length (APL), and average navigation time (ANT). Particular emphasis is placed on analyzing the model's obstacle avoidance robustness, path planning efficiency, and real-time control capability in complex dynamic environments. Additionally, comparisons with various representative baseline models are conducted to highlight the advantages of the proposed method in global modeling, spatiotemporal feature extraction, and policy optimization.

As shown in Table 1, in the RoboTHOR environment, the proposed method achieves an OASR of 91.21%, significantly higher

TABLE 1 Comparison of indicators of various models on RoboTHOR and CARLA dataset.

Method	RoboTHOR				CARLA			
	OASR(%)	CR(%)	APL(m)	ANT(s)	OASR(%)	CR(%)	APL(m)	ANT(s)
MOAC-MURS (Li et al., 2025a)	88.15	10.12	22.34	52.36	80.56	17.23	36.8	63.65
DWA-DRL (Wang et al., 2025a)	77.23	20.25	38.4	65.93	83.89	5.57	24.55	54.56
FTBF-OA (Chang et al., 2025)	76.12	14.98	45.2	68.74	81.67	15.05	33.65	66.87
CBF-MPC (Liu et al., 2025)	79.45	11.34	32.15	62.41	75.01	22.75	41.3	67.38
PMO-DQN (Hu et al., 2025)	75.08	21.53	49.75	69.99	74.96	24.03	47.98	70.08
GP-RL-UAV (Enriquez et al., 2025)	82.34	12.56	27.91	57.15	82.78	13.78	29.1	58.1
Ours	91.21	3.15	16.85	46.82	92.35	4.2	18.2	48.13

TABLE 2 Comparison of indicators of various models on urtleBot3 and EuRoC MAV dataset.

Method	TurtleBot3				EuRoC MAV			
	OASR(%)	CR(%)	APL(m)	ANT(s)	OASR(%)	CR(%)	APL(m)	ANT(s)
MOAC-MURS (Li et al., 2025a)	88.34	18.45	31.46	56	76.66	13.08	42.11	67.15
DWA-DRL (Wang et al., 2025a)	86.12	19.6	34.92	64.93	82.67	10.01	28.25	59.25
FTBF-OA (Chang et al., 2025)	77.23	25.25	43.69	66.64	73.78	27.73	50.46	68.93
CBF-MPC (Liu et al., 2025)	90.43	7.89	21.71	51.79	88.05	9.98	23.82	53.87
PMO-DQN (Hu et al., 2025)	89.45	16.17	26.07	61.45	80.19	11.17	30.71	60.76
GP-RL-UAV (Enriquez et al., 2025)	75.56	26.5	48.05	69.08	79.34	11.22	37.53	65.51
Ours	94.55	6.75	15.52	45.46	95.61	6.05	19.86	49.92

than the best-performing comparison method (MOAC-MURS, 88.15%) and over 16 percentage points better than the lowest value (Hu et al., 75.08%). Meanwhile, the CR of the proposed method is only 3.15%, much lower than other methods (e.g., DWA-DRL. 20.25%, PMO-DQN. 21.53%), demonstrating strong policy robustness and safety. In terms of path efficiency, the APL is only 16.85 meters, superior to all comparison models, indicating that the model can seek more optimal strategies while ensuring safety. The proposed method also excels in ANT, requiring only 46.82 s to complete a task, significantly faster than FTBF-OA. (68.74 s) and Hu et al. (69.99 s), demonstrating quicker response capability and control efficiency. In the more challenging CARLA urban driving environment, the proposed method continues to maintain its leading performance, achieving an OASR of 92.35%, the only model exceeding 90%, far surpassing comparison methods such as PMO-DQN. (74.96%) and CBF-MPC. (75.01%). In terms of CR, it remains the lowest at 4.2%, highlighting excellent dynamic obstacle avoidance capability. For path planning and time efficiency, the proposed method requires only 18.2 meters and 48.13 s, outperforming other methods like PMO-DQN. (47.98 meters, 70.08 s) by a wide margin.

In summary, in Table 2 and Figure 5, experimental results on TurtleBot3 and EuRoC MAV further confirm the universality and adaptability of the proposed method in static + dynamic, ground + aerial, structured + unstructured environments. Whether in OASR, path efficiency, or task response time, the proposed method consistently demonstrates significant advantages, showcasing the natural strength of the Transformer in spatiotemporal feature modeling and the potential of deep reinforcement learning for policy adaptive optimization. It also indicates that the proposed perception-decision integrated architecture has broad practical deployment value.

As shown in Table 3, the resource consumption and efficiency comparison experiments reveal that the proposed method not only outperforms other methods in performance metrics but also exhibits significant advantages in computational efficiency, model size, and inference speed, fully demonstrating the model's lightweight design and deployment friendliness. On RoboTHOR, the proposed method achieves the shortest training time (61.34 s), over 38% faster than the slowest (Enriquez et al., 99.81 s), indicating higher efficiency in parameter updates. The inference time per step averages only 92.37 milliseconds, notably lower than Wang et al. (199.83 ms) and Li et al. (178.46 ms), favoring real-time robot responses in practical applications. In terms of computational complexity, the proposed method requires only 15.34 GFLOPs, the lowest among all methods (compared to Enriquez et al. 34.17 G, Li et al. 30.48 G), confirming its deployability on resource-constrained platforms. Furthermore, the model's parameter count is only 142.83 M, the smallest among all methods, reflecting highly optimized network design. On the more complex CARLA dataset, the proposed method maintains its lead, with a training time of 63.27 s, over 15% shorter than most other comparison methods. The inference time averages just 95.12 milliseconds, much faster than Wang et al. (148.73 ms), Li et al. (183.27 ms), and Liu et al. (194.59 ms). In terms of computational complexity, the model requires only 16.78 GFLOPs, significantly lower than Li et al. (33.99

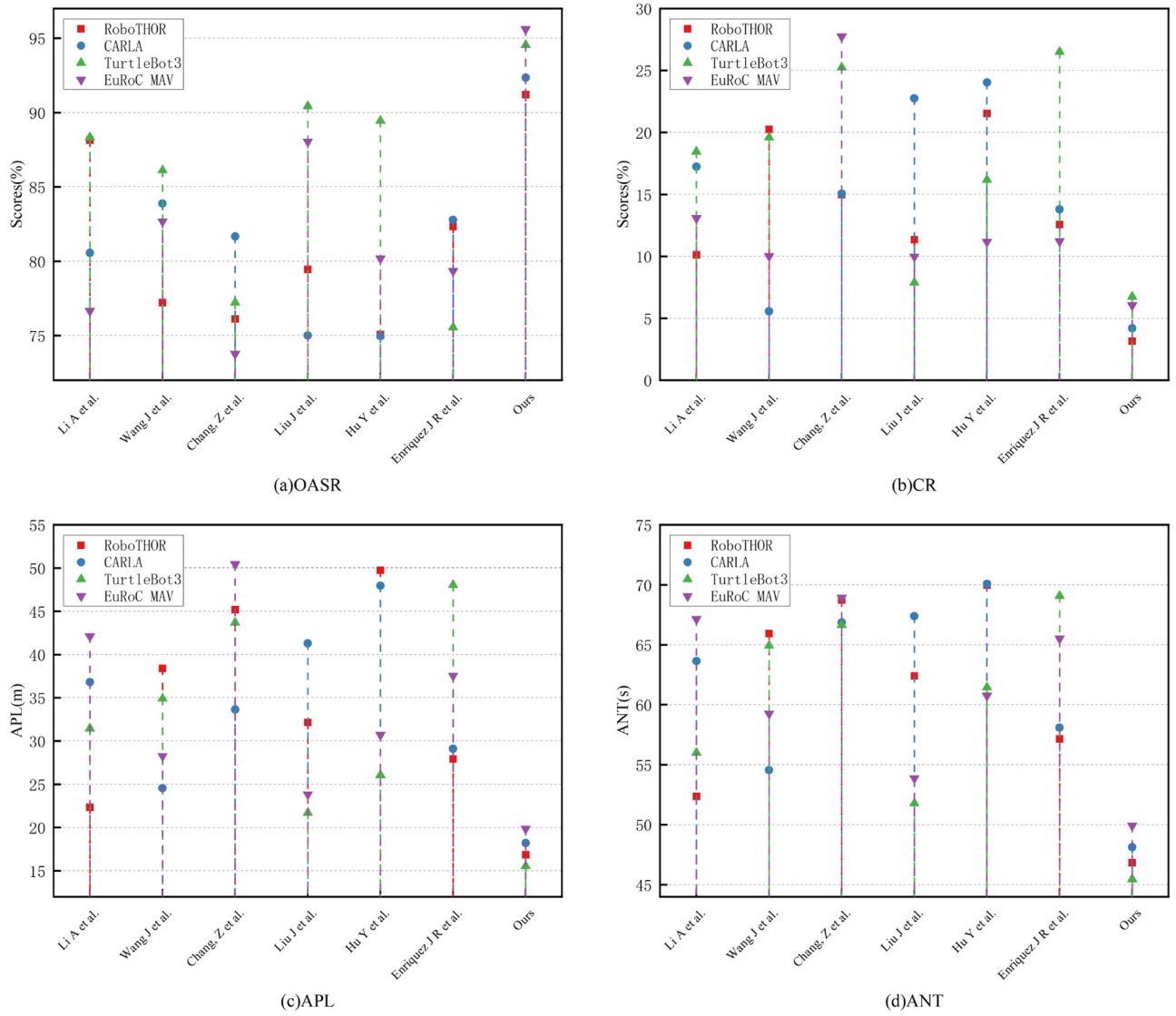

FIGURE 5
Comparative visualization of each model indicator on four dataset.

TABLE 3 Comparison of training indicators on RoboTHOR and CARLA dataset.

Method	RoboTHOR				CARLA			
	Training time(s)	Inference time(ms)	Flops(G)	Para.(M)	Training time(s)	Inference time(ms)	Flops(G)	Para.(M)
MOAC-MURS (Li et al., 2025a)	87.69	178.46	30.48	317.52	97.52	183.27	33.99	368.42
DWA-DRL (Wang et al., 2025a)	68.97	199.83	26.93	264.16	84.19	148.73	27.84	227.91
FTBF-OA (Chang et al., 2025)	93.28	128.53	21.69	235.49	70.58	132.05	22.05	293.05
CBF-MPC (Liu et al., 2025)	75.42	143.29	23.57	392.71	95.73	194.59	31.12	209.68
PMO-DQN (Hu et al., 2025)	82.13	108.64	17.82	188.27	78.36	165.94	19.21	325.78
GP-RL-UAV (Enriquez et al., 2025)	99.81	156.81	34.17	359.03	89.47	117.88	24.36	399.91
Ours	61.34	92.37	15.34	142.83	63.27	95.12	16.78	156.34

G) and Liu et al. (31.12 G). The parameter count remains the lowest at 156.34 M, verifying the model's balanced design in ensuring performance while minimizing computational overhead.

As shown in Table 4, the proposed method consistently outperforms all comparison methods across all metrics, reflecting a comprehensively optimized approach. On the TurtleBot3 dataset,

TABLE 4 Comparison of training indicators on urtleBot3 and EuRoC MAV dataset.

Method	TurtleBot3				EuRoC MAV			
	Training time(s)	Inference time(ms)	Flops(G)	Para.(M)	Training time(s)	Inference time(ms)	Flops(G)	Para.(M)
MOAC-MURS (Li et al., 2025a)	98.06	169.08	29.73	333.67	100.08	188.42	32.67	347.98
DWA-DRL (Wang et al., 2025a)	86.92	200.05	25.27	241.73	73.41	192.63	24.98	253.89
FTBF-OA (Chang et al., 2025)	80.29	137.62	28.56	385.24	88.24	145.9	26.11	315.55
CBF-MPC (Liu et al., 2025)	72.76	152.34	21.14	195.12	69.83	114.21	19.97	287.31
PMO-DQN (Hu et al., 2025)	91.17	122.79	35.62	278.49	79.62	126.55	23.42	214.76
GP-RL-UAV (Enriquez et al., 2025)	76.84	176.15	20.06	302.1	94.37	158.77	34.55	384.12
Ours	65.53	101.46	18.03	170.59	67.89	99.83	15.89	131.02

the training time (65.53 s) is about 10% shorter than the next-best method (Liu et al., 72.76 s), inference time (101.46 milliseconds) is 17% lower than the best competing method (Hu et al., 122.79 milliseconds), and both GFLOPs (18.03G) and parameter count (170.59 M) are the lowest, demonstrating higher computational efficiency and a higher degree of lightweight design. On the EuRoC MAV dataset, the proposed method again maintains its lead, with the shortest training time (67.89 s), inference time (99.83 milliseconds), and the lowest GFLOPs (15.89 G) and parameter count (131.02 M), reducing overhead by 20–50% and 20–60%, respectively, compared to other methods. Notably, certain methods like Enriquez et al. have parameter counts as high as 384.12 M, yet their inference time (158.77 ms) remains significantly inferior to the proposed method, highlighting the efficiency of the model's structural optimization. Figure 6 presents a comparative visualization of the training metrics for each model in the four datasets. Calculating efficiency relationships and resource utilization patterns is important for practical deployment considerations. We considered the trade-off between efficiency and performance, achieving better results while reducing computational overhead. In terms of scalability across datasets, even with increased environmental complexity, consistent efficiency improvements can be maintained. We highlighted the relationship between model parameters, floating point operations (FLOPs), and actual performance improvements.

Overall, comprehensive analysis indicates that the proposed method leads not only in obstacle avoidance accuracy and path efficiency but also in model size, computational complexity, training convergence speed, and inference response time. It fully embodies the innovative concept of lightweight, efficient, and deployable model design. Especially in practical robotic applications, the proposed model's high efficiency, low latency, and low resource consumption characteristics provide a solid foundation for deployment on embedded platforms or resource-constrained systems, offering considerable practical value and broad promotion potential.

As shown in Table 5, ablation study results demonstrate that the full model proposed in this paper outperforms all variant models (with any key module removed) on all core metrics across the four datasets, verifying the indispensable roles of the deep reinforcement learning module (DR), spatiotemporal attention mechanism (SPA), and Transformer architecture in the system's overall performance. The full model shows significant advantages in OASR, CR, APL, and ANT, fully reflecting the rationality and synergistic effect of the proposed structural design.

On RoboTHOR, the full model achieves an OASR of 91.21%, while removing the DR module drops it to 73.59%, and removing SPA further drops it to 61.34%, demonstrating the crucial role of reinforcement learning for long-term decision-making in dynamic environments and the SPA for feature representation and obstacle perception. Removing the Transformer causes OASR to decline to 67.28% and CR to surge to 35.62%, while APL increases dramatically to 51.49 meters, confirming the importance of Transformer in capturing global temporal dependencies. The full model's APL of 16.85 meters and ANT of 46.82 s remain optimal.

In the CARLA urban scene, the full model maintains its lead with a 92.35% OASR and 4.2% CR. Removing the Transformer drops OASR to 63.17%, increases CR to 26.05%, and significantly raises both APL (58.36 meters) and ANT (81.37 s), showing the critical importance of the Transformer in complex, dynamic, multi-object environments. Removing DR or SPA also results in noticeable performance declines, proving their indispensable roles in policy optimization and spatiotemporal feature extraction.

Similar trends appear on TurtleBot3, where the full model leads with a 94.55% OASR and the lowest CR of 6.75%. Removing SPA causes the OASR to plummet to 64.96%, while removing the Transformer keeps OASR at 74.31% but CR surges to 38.71%, confirming the SPA's role in integrating local and global obstacle information for safe avoidance.

In the EuRoC MAV dataset, the full model achieves a 95.61% OASR and 6.05% CR, proving strong robustness and precise path planning capability in high-DOF, temporally sensitive UAV obstacle avoidance tasks. Removing the Transformer drops OASR to 75.82%, raises CR to 22.54%, and nearly doubles ANT to 99.67 s, demonstrating the Transformer's essential role in modeling high-dimensional state transitions and sequential behaviors. Figure 7 illustrates the results of the ablation experiments. We provide insights into component interdependencies and synergistic effects. The display module provides a hierarchy of the most significant performance improvements contributed by components, while demonstrating how the combination of DR, SPA, and Transformer produces synergistic improvements that exceed their individual contributions. In fault mode analysis, we

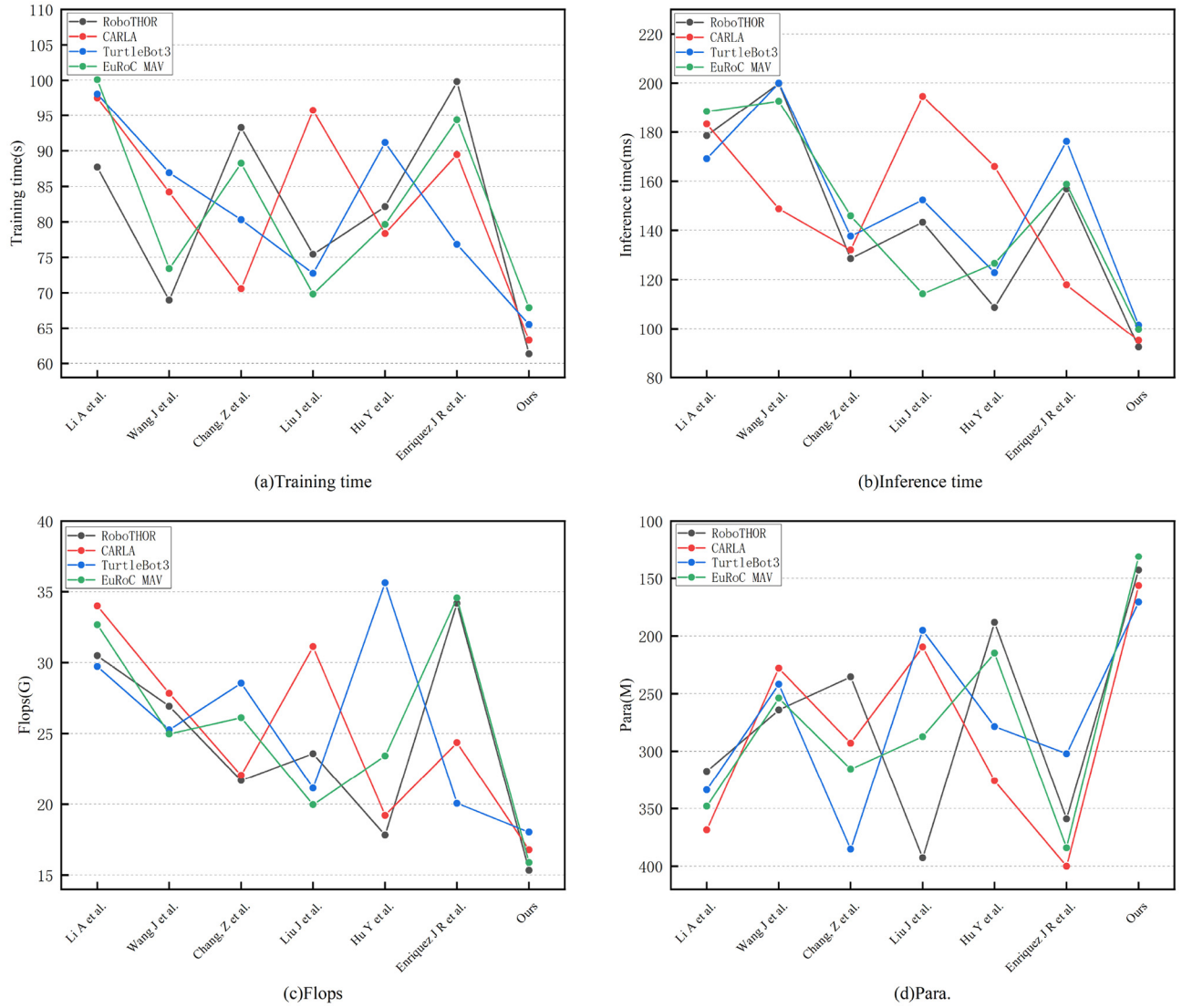

FIGURE 6
Comparative visualization of training metrics on four datasets.

TABLE 5 Ablation experiments of this model on four datasets.

Module	RoboTHOR				CARLA			
	OASR(%)	CR(%)	APL(m)	ANT(s)	OASR(%)	CR(%)	APL(m)	ANT(s)
w/o DR	73.59	11.34	38.62	76.89	69.42	31.49	41.78	72.16
w/o SPA	61.34	23.78	27.83	67.23	71.83	15.97	32.14	96.08
w/o Trans.	67.28	35.62	51.49	92.54	63.17	26.05	58.36	81.37
Ours	91.21	3.15	16.85	46.82	92.35	4.2	18.2	48.13
Module	**TurtleBot3**				**EuRoC MAV**			
	OASR(%)	CR(%)	APL(m)	ANT(s)	OASR(%)	CR(%)	APL(m)	ANT(s)
w/o DR	66.05	18.23	36.29	79.02	62.78	40.27	34.91	84.65
w/o SPA	64.96	29.16	63.12	69.58	68.53	12.89	44.03	74.93
w/o Trans.	74.31	38.71	29.67	88.41	75.82	22.54	65.59	99.67
Ours	94.55	6.75	15.52	45.46	95.61	6.05	19.86	49.92

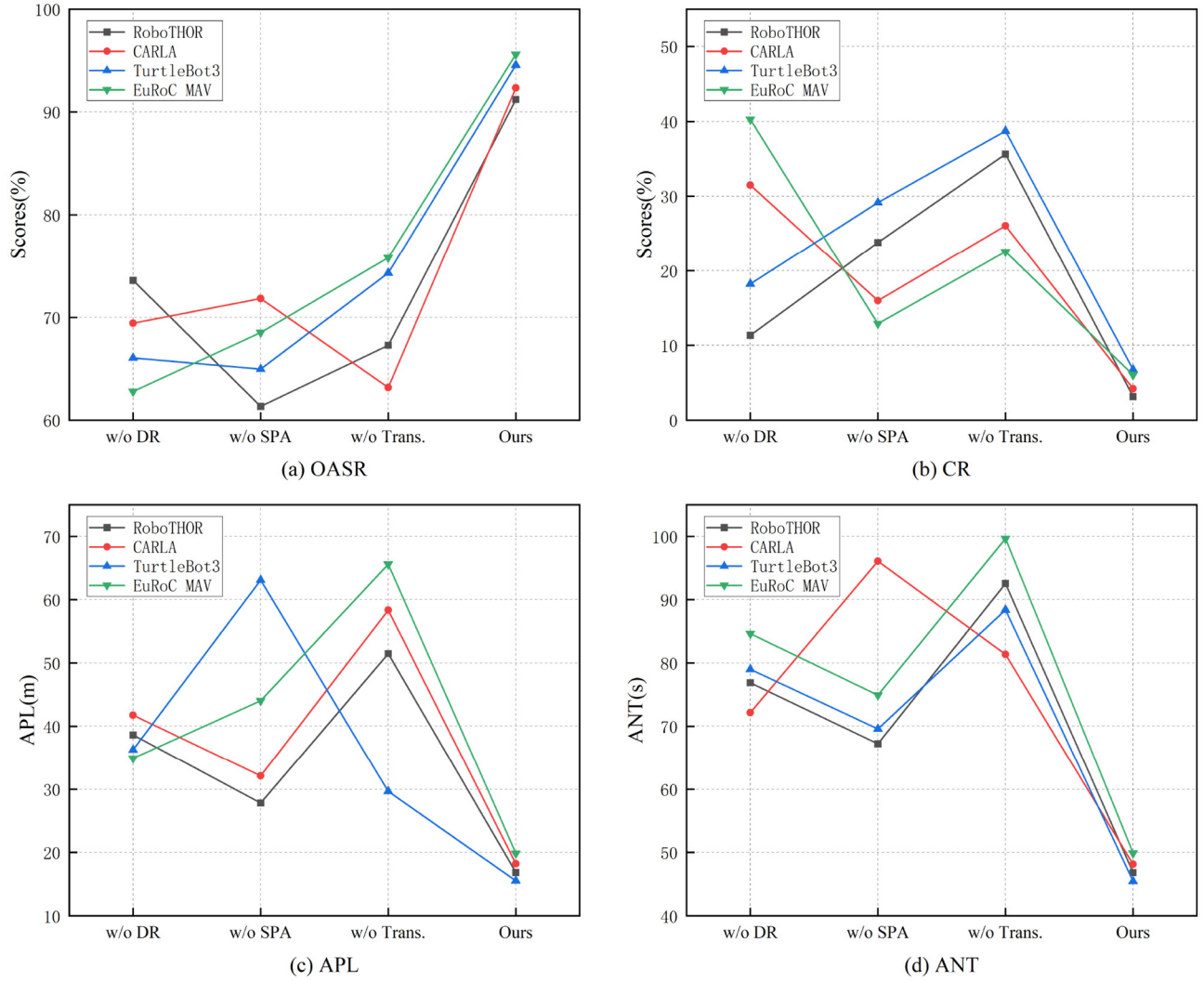

FIGURE 7
Comparative visualization of ablation experiments on four dataset.

identify which performance aspects are most affected when specific components are removed.

In summary, Table 5 clearly verifies that the deep reinforcement learning module ensures global policy optimality, the spatiotemporal attention mechanism strengthens dynamic change and key obstacle perception, and the Transformer provides powerful long-term sequence modeling and global dependency awareness. The collaboration among these components enables the proposed method to consistently achieve optimal performance and efficiency across various complex environments, demonstrating excellent scalability and generality, and forming a vital foundation for building high-performance intelligent obstacle avoidance systems.

5 Discussion and conclusion

The end-to-end robot intelligent obstacle avoidance method proposed in this paper, based on deep reinforcement learning and a spatiotemporal Transformer architecture, demonstrates excellent performance across multiple simulation environments and when evaluated on real-world sensor data. It consistently outperforms existing mainstream methods in simulation-based evaluations in terms of obstacle avoidance success rate, path efficiency, decision response speed, and model lightweight design. Notably, while our method shows promising results in simulation environments and when tested on real-world sensor data, comprehensive validation in actual physical deployment scenarios remains to be conducted. However, two critical limitations specific to our framework require immediate attention. First, our spatiotemporal attention mechanism exhibits domain sensitivity during sim-to-real transfer, where attention weights learned on simulated obstacle patterns may not effectively transfer to real-world sensor noise and lighting variations. This limitation is particularly pronounced in our Transformer encoder's spatial attention computation (Equations 14–16), where synthetic depth data characteristics differ significantly from real sensor outputs. Second, while our integrated architecture achieves superior performance, the computational

overhead of multi-head attention combined with experience replay creates latency bottlenecks that limit real-time deployment on resource-constrained robotic platforms, particularly affecting the 92.37 ms inference time requirement for safe obstacle avoidance.

Based on these specific limitations, we identify two concrete research directions for immediate improvement. First, developing domain-adaptive spatiotemporal attention mechanisms that can automatically adjust attention weights during deployment through meta-learning approaches. This involves creating attention regularization techniques that maintain spatial-temporal modeling capabilities while adapting to real-world sensor characteristics, potentially through adversarial training between simulated and real sensor data. The specific implementation would focus on modifying our attention computation (Equation 16) to include domain-invariant features. Second, architectural optimization for embedded deployment through selective attention pruning and quantization techniques specifically designed for our Transformer-DQN integration. This involves developing attention head importance scoring to identify which spatial-temporal attention patterns are most critical for obstacle avoidance, enabling targeted compression without performance degradation. These improvements directly address our framework's deployment challenges while maintaining its core spatiotemporal modeling advantages.

In summary, although our spatio-temporal Transformer-DQN framework has made progress in simulated environments, addressing specific challenges such as domain transfer sensitivity and computational efficiency is crucial for achieving robust real-world deployment. These targeted improvements represent the most impactful next steps in translating our theoretical findings into practical robot navigation systems.

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.

Author contributions

YZ: Writing – original draft, Writing – review & editing, Project administration. WZ: Writing – original draft.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

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.

Generative AI statement

The author(s) declare that no Gen AI was 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

References

Almazrouei, K., Kamel, I., and Rabie, T. (2023). Dynamic obstacle avoidance and path planning through reinforcement learning. *Appl. Sci.* 13:8174. doi: 10.3390/app13148174

Alsadik, B., and Karam, S. (2021). The simultaneous localization and mapping (SLAM): an overview. *Surv. Geospat. Eng. J.* 1, 1–12. doi: 10.38094/sgej1027

Ansari, S. (2019). "A review on sift and surf for underwater image feature detection and matching," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (Coimbatore: IEEE), 1–4. doi: 10.1109/ICECCT.2019.8869489

Burghoffer, A., Seyssaud, J., and Magnier, B. (2023). "Ov 2 slam on euroc mav datasets: a study of corner detector performance," in *2023 IEEE International Conference on Imaging Systems and Techniques (IST)* (Copenhague: IEEE), 1–5. doi: 10.1109/IST59124.2023.10355706

Chang, Z., Zong, G., Song, S., and Zhao, X. (2025). Fast finite-time bipartite formation with obstacle avoidance for time-delay multiagent systems: application in mobile robot swarm. *IEEE Trans. Ind. Inform.* 21, 4586–4594. doi: 10.1109/TII.2025.3545050

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., et al. (2021). Decision transformer: reinforcement learning via sequence modeling. *Adv. Neural Inf. Process. Syst.* 34, 15084–15097.

Chen, Y., Li, Z., You, S., Chen, Z., Chang, J., Zhang, Y., et al. (2025). Attributive reasoning for hallucination diagnosis of large language models. *Proc. AAAI Conf. Artif. Intell.* 39, 23660–23668. doi: 10.1609/aaai.v39i22.34536

Dai, W., Jiang, Y., Liu, Y., Chen, J., Sun, X., and Tao, J. (2024). "Cab-kws: contrastive augmentation: an unsupervised learning approach for keyword spotting in speech technology," in *International Conference on Pattern Recognition* (Springer: New York), 98–112. doi: 10.1007/978-3-031-78122-3_7

Deitke, M., Han, W., Herrasti, A., Kembhavi, A., Kolve, E., Mottaghi, R., et al. (2020). "Robothor: an open simulation-to-real embodied ai platform," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 3164–3174. doi: 10.1109/CVPR42600.2020.00323

Enriquez, J. R., Castillo-Toledo, B., Gennaro, S. D., and García-Delgado, L. A. (2025). An algorithm for dynamic obstacle avoidance applied to uavs. *Rob. Auton. Syst.* 186:104907. doi: 10.1016/j.robot.2024.104907

Gannamaneni, S., Houben, S., and Akila, M. (2021). "Semantic concept testing in autonomous driving by extraction of object-level annotations from carla," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 1006–1014. doi: 10.1109/ICCVW54120.2021.00117

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* 23, 34–46. doi: 10.1109/TRO.2006.889486

Guo, B., Guo, N., and Cen, Z. (2022). Obstacle avoidance with dynamic avoidance risk region for mobile robots in dynamic environments. *IEEE Robot. Autom. Lett.* 7, 5850–5857. doi: 10.1109/LRA.2022.3161710

Hu, Y., Zhang, Y., Song, Y., Deng, Y., Yu, F., Zhang, L., et al. (2025). Seeing through pixel motion: learning obstacle avoidance from optical flow with one camera. *IEEE Robot. Autom. Lett.* 10, 5871–5878. doi: 10.1109/LRA.2025.3560842

Issa, R. B., Das, M., Rahman, M. S., Barua, M., Rhaman, M. K., Ripon, K. S. N., et al. (2021). Double deep q-learning and faster r-cnn-based autonomous vehicle navigation and obstacle avoidance in dynamic environment. *Sensors* 21:1468. doi: 10.3390/s21041468

Janner, M., Li, Q., and Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Adv. Neural Inf. Process. Syst.* 34, 1273–1286.

Jian, X., Wu, B., Song, Y., Liu, D., Wang, Y., Wang, W., et al. (2025). Performing task automation for surgical robot: a spatial-temporal varying primal-dual neural network with guided obstacle avoidance and null space optimization. *Expert Syst. Appl.* 273:126780. doi: 10.1016/j.eswa.2025.126780

Kashyap, A. K., and Konathalapalli, K. (2025). Autonomous navigation of ROS2 based turtlebot3 in static and dynamic environments using intelligent approach. *Int. J. Inf. Technol.* 1–23. doi: 10.1007/s41870-025-02500-5. [Epub ahead of print].

Katona, K., Neamah, H. A., and Korondi, P. (2024). Obstacle avoidance and path planning methods for autonomous navigation of mobile robot. *Sensors* 24:3573. doi: 10.3390/s24113573

Lahn, N., Raghvendra, S., and Zhang, K. (2023). A combinatorial algorithm for approximating the optimal transport in the parallel and MPC settings. *Adv. Neural Inf. Process. Syst.* 36, 21675–21686.

Li, A., Guo, S., Li, C., Yin, H., Liu, M., and Shi, L. (2025a). A moving obstacle avoidance strategy-based collaborative formation for the bionic multi-underwater spherical robot control system. *Ocean Eng.* 329:121120. doi: 10.1016/j.oceaneng.2025.121120

Li, B., Ji, Z., Zhao, Z., and Yang, C. (2025b). Model predictive optimization and terminal sliding mode motion control for mobile robot with obstacle avoidance. *IEEE Trans. Ind. Electron.* 72, 9293–9303. doi: 10.1109/TIE.2025.3536628

Liu, J., Yang, J., Mao, J., Zhu, T., Xie, Q., Li, Y., et al. (2025). Flexible active safety motion control for robotic obstacle avoidance: a CBF-guided mpc approach. *IEEE Robot. Autom. Lett.* 10, 2686–2693. doi: 10.1109/LRA.2025.3534519

Liu, L., Wang, B., and Xu, H. (2022). Research on path-planning algorithm integrating optimization a-star algorithm and artificial potential field method. *Electronics* 11:3660. doi: 10.3390/electronics11223660

Miao, C., Chen, G., Yan, C., and Wu, Y. (2021). Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Comput. Ind. Eng.* 156:107230. doi: 10.1016/j.cie.2021.107230

Misir, O., and Celik, M. (2025). Visual-based obstacle avoidance method using advanced cnn for mobile robots. *Internet Things* 31:101538. doi: 10.1016/j.iot.2025.101538

Ni, X., Li, J., Xu, J., Shen, Y., and Liu, X. (2024). Grey relation analysis and multiple criteria decision analysis method model for suitability evaluation of underground space development. *Eng. Geol.* 338:107608. doi: 10.1016/j.enggeo.2024.107608

Ni, X., Liu, X., Pang, S., Dong, Y., Guo, B., Zhang, Y., et al. (2025). Global marine methane seepage: spatiotemporal patterns and ocean current control. *Mar. Geol.* 487:107589. doi: 10.1016/j.margeo.2025.107589

Phatak, A., Raghvendra, S., Tripathy, C., and Zhang, K. (2023). "Computing all optimal partial transports," in *International Conference on Learning Representations* (Kigali: OpenReview).

Qin, H., Shao, S., Wang, T., Yu, X., Jiang, Y., and Cao, Z. (2023). Review of autonomous path planning algorithms for mobile robots. *Drones* 7:211. doi: 10.3390/drones7030211

Raghvendra, S., Shirzadian, P., and Zhang, K. (2024). A new robust partial *p*-wasserstein-based metric for comparing distributions. *arXiv preprint arXiv:2405.03664*. doi: 10.48550/arXiv.2405.03664

Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., and Mirjalili, S. (2022). Particle swarm optimization: a comprehensive survey. *IEEE Access* 10, 10031–10061. doi: 10.1109/ACCESS.2022.3142859

Ullah, I., Adhikari, D., Khan, H., Anwar, M. S., Ahmad, S., and Bai, X. (2024). Mobile robot localization: current challenges and future prospective. *Comput. Sci. Rev.* 53:100651. doi: 10.1016/j.cosrev.2024.100651

Wang, J., Yang, L., Cen, H., He, Y., and Liu, Y. (2025a). Dynamic obstacle avoidance control based on a novel dynamic window approach for agricultural robots. *Comput. Ind.* 167:104272. doi: 10.1016/j.compind.2025.104272

Wang, Y., Li, X., Zhang, J., Li, S., Xu, Z., and Zhou, X. (2021). Review of wheeled mobile robot collision avoidance under unknown environment. *Sci. Prog.* 104:00368504211037771. doi: 10.1177/00368504211037771

Wang, Z., Wang, S., Zheng, S., Xie, S. Q., Xie, Y., and Wu, H. (2025b). Simultaneous planning and tracking framework for obstacle avoidance of autonomous mobile robots in dynamic scenarios. *IEEE Trans. Ind. Electron.* 72, 8219–8229. doi: 10.1109/TIE.2024.3522489

Wijayathunga, L., Rassau, A., and Chai, D. (2023). Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: a review. *Appl. Sci.* 13:9877. doi: 10.3390/app13179877

Wu, Y., Jia, X., Li, T., and Liu, J. (2025). A real-time collision avoidance method for redundant dual-arm robots in an open operational environment. *Robot. Comput. Integr. Manuf.* 92:102894. doi: 10.1016/j.rcim.2024.102894

Zhang, M., Xie, K., Zhang, Y.-H., Wen, C., and He, J.-B. (2022). Fine segmentation on faces with masks based on a multistep iterative segmentation algorithm. *IEEE Access* 10, 75742–75753. doi: 10.1109/ACCESS.2022.3192026

Zhang, Z. (2022). "Overview of filtering algorithms for autonomous mobile robots," in *Proceedings of the 3rd International Symposium on Artificial Intelligence for Medicine Sciences* (Amsterdam: Association for Computing Machinery), 568–572. doi: 10.1145/3570773.3570871

Zheng, J., Li, W., Hong, J., Petersson, L., and Barnes, N. (2022). "Towards open-set object detection and discovery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA: IEEE), 3961–3970. doi: 10.1109/CVPRW56347.2022.00441