Research on Multi-mode Path Planning Algorithm for Nanorobots Based on Improved Deep Reinforcement Learning

Jitong Miao
School of Mechanical and Electrical Engineering, Hainan University, Haikou 570228, Hainan, China
Corresponding author: Jitong Miao (e-mail: 23220854060014@hainanu.edu.cn)

ABSTRACT To address strong dynamic uncertainty and limited adaptability of traditional algorithms in nanorobot path planning, especially in microscopic textile fiber and porous-material environments, this study proposes a multi-mode path planning algorithm based on an improved deep Q-network (NP-IDQN). The method is applicable to nanorobots driven by fluid, chemical, or electromagnetic micro-actuation fields, where stable navigation must satisfy both trajectory efficiency and physical feasibility. The algorithm is optimized at three levels. At the network-architecture level, residual connections, layer normalization, and Transformer mechanisms are introduced to enhance feature extraction and global correlation modeling for fluid–obstacle interactions. At the training-strategy level, curriculum learning with multigeometry and multi-difficulty scenarios is adopted, covering rectangular, spherical, and mixed obstacles to improve environmental generalization. At the decision-making level, a multi-objective reward function integrates distance optimization, obstacle avoidance, energy efficiency, and microscopic driving-force constraints. Simulation results show that NP-IDQN improves trajectory stability and reduces the risk of local optima compared with conventional reinforcement learning and classical planning methods. The study provides technical support for precise navigation of nanorobots in complex fiber networks, functional material synthesis, environmental purification, and electromagneticactuated micro/nanorobotic systems.

INDEX TERMS nanorobot, path planning, deep reinforcement learning, electromagnetic actuation, microscale navigation

I. INTRODUCTION

WITH the rapid development of nanotechnology, robotics, and artificial intelligence, the application prospects of nanorobots in medicine, electronics, and textile industrial engineering are becoming increasingly broad. These microscopic systems facilitate the precise manipulation of textile fiber structures and functional material surfaces, enabling advanced technical innovations in environmental safety and high-performance industrial manufacturing [1-3]. In the medical field, they can be used for drug transportation, virus detection, tissue repair, etc., and can perform highly precise targeted treatment within the human body. They can also be used for environmental cleaning and wastewater treatment, for example, extracting and collecting specific harmful substances, and achieving wastewater separation and purification [4]. Nanorobot path planning refers to the process where nanorobots autonomously plan navigation routes in a microenvironment and reach the target location along the routes. Unlike traditional robots in a macroscopic environment, the operating environment of nanorobots is more complex, including operations in cells, proteins, crystal structures, etc. Therefore, they are prone to significant influences from microscopic physical effects such as Brownian motion and fluid dynamics [5]. These factors pose significant challenges to the path planning of nanorobots within the complex morphological structures of textile fiber networks, requiring improved algorithms that can handle highly dynamic and uncertain textile industrial environments. Consequently, robust computational models are essential to meet the strict efficiency requirements for navigating microscopic industrial landscapes while maintaining high precision in material interaction [6].

Traditional path planning methods can be mainly classified into two categories: classical algorithms and biomimetic algorithms. Classical algorithms such as A* , artificial potential field (APF), and RRT * have achieved remarkable success in the macroscopic robotics field. Huang et al. proposed an improved A* algorithm to address the issues of low search efficiency and un-smooth paths in robot global path planning. They introduced a new neighborhood search method and designed an obstacle constraint method. Simulations in different scenarios and grid map

sizes demonstrated that compared with several classic variants of A* algorithm and recent improved versions, the improved A* algorithm not only generates the smoothest and shortest paths but also is more stable in terms of improving time efficiency [7]. Ding et al. found that the redundancy of robots in constrained environments makes it difficult to select appropriate inverse kinematics solutions, so they proposed a collision avoidance path planning method for continuum robots based on improved APF, using the beetle antenna search algorithm to solve the optimal problem of APF without performing velocity kinematics processing. Simulation and experimental results verified the effectiveness of the proposed path planning method [8]. Wu et al. observed the significant variance in RRT search time and its subpar performance in narrow channel scenarios. They responded with Fast-RRT, which improves search speed and enhances algorithm stability through a fast-sampling strategy confined to unexplored spaces. In the experiments, Fast-RRT outperformed RRT and RRT* algorithms in terms of speed and stability [9]. These classical algorithms are widely applied, but they have obvious limitations when ap-plied to nanorobots. These methods usually rely on precise environmental modeling and are difficult to adapt to the high uncertainty and dynamics of the microscopic environment. Another biomimetic algorithm, such as genetic algorithm (GA) [10] and ant colony optimization (ACO) [11], although having strong global search capabilities, have high computational complexity and slow convergence, and are not suitable for the real-time operation requirements of nanorobots. Che improved ACO to perform path planning for autonomous underwater vehicles (AUVs), and found that it still has problems such as local minima, poor quality, and low accuracy. After optimizing ACO with particle swarm optimization algorithm (PSO), the problems were somewhat solved [12]. In the field of nanorobots, Ahmad et al. recently explored the application of reinforcement learning (RL) in enhancing the navigation capabilities of micro/nanorobots in medical environments, and their research mainly discussed the potential of the proximal policy optimization (PPO) algorithm in improving the accuracy and reliability of nanorobot groups in advanced medical applications [13]. However, these methods are usually deterministic or rule-based algorithms that require precise modeling of the problem and then calculate the solution based on the model, and cannot learn implicit models and complex dynamics.

In recent years, advancements in machine learning and deep learning methods have pro-vided new ideas for path planning [14]. Particularly, deep reinforcement learning (DRL) com-bines deep learning with reinforcement learning, demonstrating significant advantages in handling high-dimensional state spaces and complex environments, enabling robots to learn optimal strategies through autonomous interaction with the environment [15-17]. Figure 1 illustrates the general framework of path planning for nanorobots based on DRL. The agent interacts with the nanorobot's operating environment by selecting actions based on the current policy, and receives feedback in the form of rewards and state updates. This interaction loop enables the robot to iteratively improve its strategy, thereby achieving optimal trajectory planning in complex three-dimensional environments. Currently, most DRL-based path planning methods are tailored for macroscopic environments, for example, Yang et al. used the deep Q-network (DQN) algorithm to address the two problems of slow convergence and excessive randomness in traditional robot path planning, thereby enhancing the efficiency of multi-robot path planning [18]. However, standard DRL algorithms still face some challenges in robot path planning. Firstly, they have low training efficiency, especially when a large number of interaction samples are required in complex environments. Secondly, the policy generalization ability is insufficient, with a sharp decline in performance when trained on unseen environment configurations; these limitations severely restrict the practical application of DRL in nanorobots. Therefore, scholars have also made targeted improvements. Gao et al. introduced a novel incremental training mode based on DRL, transferring the designed algorithm to a simple three-dimensional environment for re-training to obtain convergent network parameters, including the weights and biases of the deep neural network (DNN), enhancing the model's generalization ability across diverse scenarios [19]. Kumaar et al. observed that service robots often operate in vast, unknown environments for extended periods. Their proposed RL system utilizes the deep Q-Learning algorithm to learn the initial path using the topological map of the environment. When the environment changes, a transfer learning algorithm trains the agent in the new setting, and experimental results demonstrate that this RL system can learn all paths based on the initial topological map of different service environments, achieving an accuracy rate exceeding 98% [20]. Kumaar's research also suggests that DRL models can perform transfer learning to some extent, enabling them to adapt more swiftly to other environments.

While prior research has predominantly focused on macroscopic applications. Gu et al. have pioneered the expansion of nanorobot technology from liquid environments to dry solid surfaces. By harnessing machine vision and deep learning, they have achieved multi-degree-of-freedom movement and complex functionalities in nanorobots [21]. Yang et al. have employed DNNs to enable nanocolloid robots to navigate efficiently in unknown, complex environments. Robots trained with these models can adeptly avoid obstacles and minimize travel time using only local information inputs [22]. In addition, in a groundbreaking study published in Nature, Yang et al. identified a lack of intelligent behaviors in current micro-robots, particularly their inability to autonomously adjust movements in response to environmental changes. Therefore, a DRL framework for micro robots was proposed. The core of this framework is the DQN algorithm, and then the parameters are optimized through a magnetic nanoparticle swarm optimization strategy. Finally, the nanorobots can

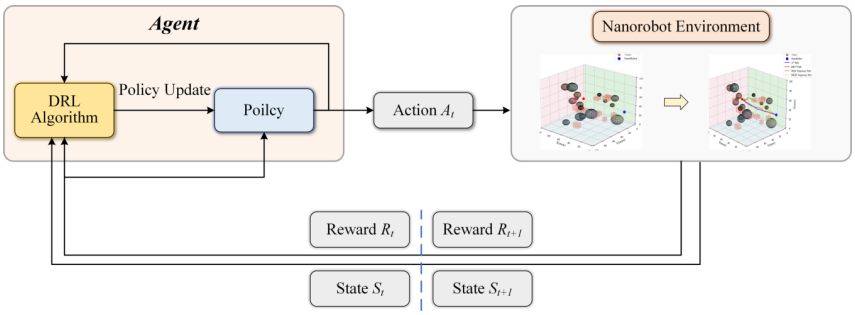


FIGURE 1. General framework of path planning for nanorobots based on DRL

autonomously make appropriate decisions for navigation in unstructured environments [23]. Therefore, the application potential of deep learning combined with optimization methods in nanorobots is considerable, but there are very few related studies at present. Moreover, the path planning of nanorobots also needs to consider the special physical characteristics of the microscopic environment. Factors such as fluid dynamics, Brownian motion, and non-uniform viscous resistance can significantly affect the movement trajectory of nanorobots [24]. Traditional planning methods often ignore these factors, resulting in impractical planning results in the actual environment. Therefore, an effective path planning algorithm must fully consider the microscopic physical constraints and generate physically executable trajectories [25].

In response to the aforementioned challenges, this study proposes a novel improved deep reinforcement learning framework specifically designed for path planning of nanorobots in diverse obstacle environments. Our main innovations include:

(1) A sophisticated network architecture was designed, integrating residual connections and layer normalization techniques to enhance gradient flow and training stability, addressing the performance degradation issue of traditional DQN in complex decision-making;

(2) A multi-environment training mechanism was introduced, enabling the agent to learn in rectangular obstacles, spherical obstacles, and mixed obstacles, significantly improving the generalization ability and adaptability of the strategy;

(3) A difficulty progressive curriculum was developed, ranging from simple loops to medium environments and then to complex loops, conforming to the natural laws of machine learning, which can significantly enhance the efficiency of machine learning.

II. MATERIALS AND METHODS

A. OVERVIEW OF THE NP-IDQN ALGORITHM DESIGN

Figure 2 shows the NP-IDQN network framework used for the path planning of nanorobots, aiming to address the challenges faced by nanorobot path planning in complex environments. In this framework, the Q target network is a specially designed IDQN algorithm, which will be elaborated in Section 2.2. This network enhances gradient flow and training stability by leveraging residual connections and layer normalization, thereby alleviating the performance decline problem of traditional DQN in complex decision-making scenarios. Additionally, our framework introduces a multi-environment training mechanism, enabling the nanorobot agent to learn in various environments including rectangular obstacles, spherical obstacles, and mixed obstacles, thereby improving its decision-making ability and path planning accuracy.

In Figure 2, environmental information such as obstacles, liquid conditions, and the state of the nanorobot is perceived and used to construct rewards and state reinforcement learning elements. During the training process, components such as the experience replay method can be utilized to train models with strong path planning capabilities. Compared to traditional methods, this framework not only replaces the traditional reinforcement learning method with an improved Q target network, but also overcomes the limitations of traditional DQN in complex environments through multi-environment training, making it highly suitable for the precise and adaptive path planning requirements of nanorobots.

B. IDQN

The detailed network structure of IDQN is shown in Figure 3. The input is the state "state", and the output is the action "Action". DQN is the traditional network. The traditional DQN adopts a simple layering of Linear + PReLU, directly mapping the states, lacking the modeling ability for deep features and global correlations. To address these limitations, we propose an improved DQN (IDQN) that integrates three key enhancements. First, LayerNorm and residual connections are adopted to stabilize training and strengthen gradient propagation. Then, a Transformer attention mechanism is introduced to model global environmental correlations. Together, these designs enable effective local feature extraction and global information modeling, making the network better adapted to complex micro–robot interaction scenarios.

Specifically, first of all, the state s is transformed linearly, then subjected to LayerNorm and ReLU operations, and the resulting feature is Z_{act}:

$$z_{\text{pre}} = w_{\text{pre}} \cdot s + b_{\text{pre}} \tag{1}$$

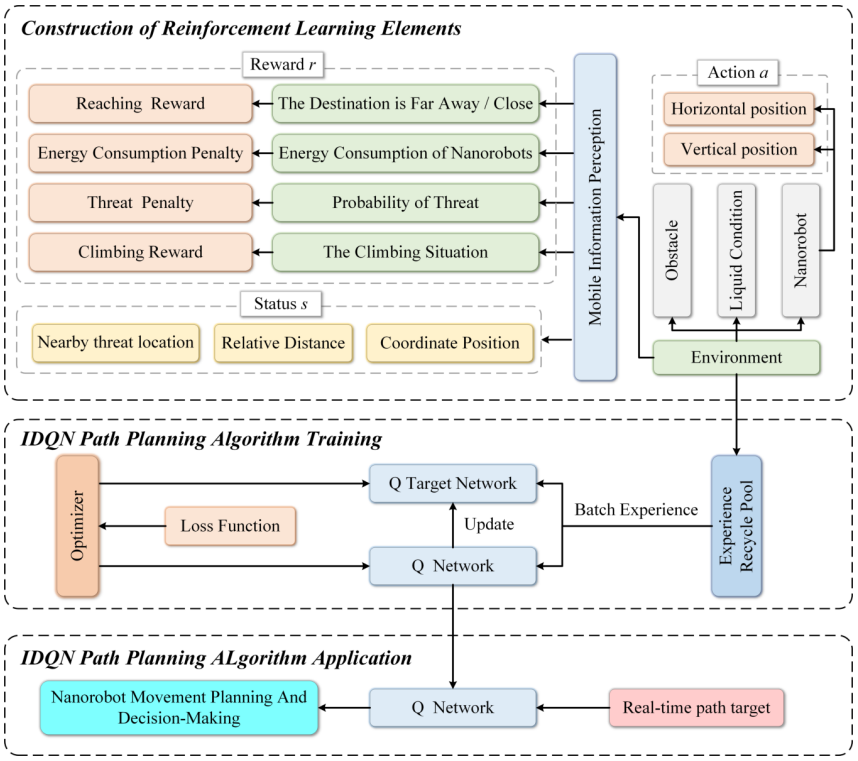


FIGURE 2. NP-IDQN Algorithm Design network framework

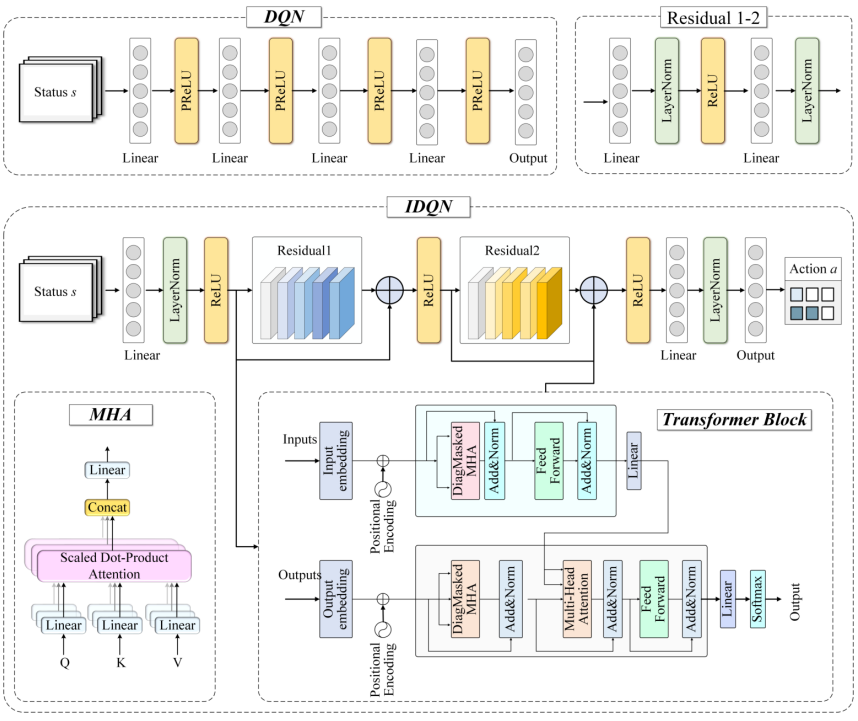


FIGURE 3. The IDQN framework proposed in this study is in line with the standard DQN framework.

$$z_{\ln} = \gamma \cdot \frac{z_{\text{pre}} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{2}$$

$$z_{\text{act}} = \text{ReLU}(z_{\ln}) \tag{3}$$

Among them, μ and σ^2 represent the characteristic mean and variance; γ and β are learnable parameters, w is the weight, and b is the bias.

Then, the pre-extracted feature Z_{act} is passed through the residual network. Specifically, the residual network under-

goes two linear transformations, LayerNorm, and residual connections:

$$z_{r1,1} = \mathrm{ReLU}(\mathrm{LayerNorm}(W_{r1,1}x + b_{r1,1})) \tag{4}$$

$$z_{r1,2} = \mathrm{LayerNorm}(W_{r1,2}z_{r1,1} + b_{r1,2}) \tag{5}$$

$$x_{\mathrm{res1}} = x + z_{r1,2} \tag{6}$$

$$x_{\mathrm{res1,act}} = \mathrm{ReLU}(x_{\mathrm{res1}}) \tag{7}$$

Among them, x refers to the input before the residual network, which is Z_{act}.

The input of the Transformer Block is the output x_{res} from the residual module. It is first modeled with MHA for global correlation, then processed by the feedforward network and residual normalization: Firstly, the MHA is used to obtain the attention result Z_{attn}:

$$Q = W_q x_{\mathrm{res}}, \quad K = W_k x_{\mathrm{res}}, \quad V = W_v x_{\mathrm{res}} \tag{8}$$

$$\mathrm{Attn}_i = \mathrm{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{9}$$

$$\mathrm{MHA}(x_{\mathrm{res}}) = W_o[\mathrm{Attn}_1; \mathrm{Attn}_2; \ldots; \mathrm{Attn}_h] \tag{10}$$

$$z_{\mathrm{attn}} = \mathrm{LayerNorm}(x_{\mathrm{res}} + \mathrm{MHA}(x_{\mathrm{res}})) \tag{11}$$

Among them, Attn represents the attention result, Q is the query matrix, K is the key matrix, V is the value matrix, h is the number of heads, and [] represents concatenation.

Then calculate the feedforward network and Add&Norm:

$$z_{\mathrm{ff}} = \mathrm{ReLU}(W_{\mathrm{ff1}} z_{\mathrm{attn}} + b_{\mathrm{ff1}}) \tag{12}$$

$$z_{\mathrm{ff}} = W_{\mathrm{ff2}} z_{\mathrm{ff}} + b_{\mathrm{ff2}} \tag{13}$$

$$z_{\mathrm{trans}} = \mathrm{LayerNorm}(z_{\mathrm{attn}} + z_{\mathrm{ff}}) \tag{14}$$

Finally, the output z_{final} from the Transformer / residual module is ultimately mapped to the action value $Q(s, a)$:

$$z_{\mathrm{out,act}} = \mathrm{ReLU}(z_{\mathrm{final}}) \tag{15}$$

$$\begin{aligned} z_{\mathrm{out,lin}} &= W_{\mathrm{out}} z_{\mathrm{out,act}} + b_{\mathrm{out}}, \\ W_{\mathrm{out}} &\in \mathbb{R}^{a \times a_{\mathrm{mid}}}, \quad b_{\mathrm{out}} \in \mathbb{R}^{a_o} \end{aligned} \tag{16}$$

$$Q(s, a) = \mathrm{LayerNorm}(z_{\mathrm{out,lin}}) \tag{17}$$

Among them, $d^{\,a}$ represents the action dimension.

In summary, IDQN adopts a multi-level feature extraction architecture, mapping the high-dimensional state space to a compact feature representation, significantly improving the learning efficiency and adapting to the complex state space of the multi-component and multiscale nanorobot. Through the Transformer attention mechanism, the nanorobot can adaptively focus on key environmental features, quickly identifying the optimal path and target area in complex environments, breaking through the limitations of traditional DQN in deep feature modeling and global correlation capture, providing more accurate value estimation capabilities for the reinforcement learning decision-making of nanorobots.

III. SIMULATION

A. SIMULATION ENVIRONMENT

The working environment of traditional robots is mainly on a plane, so the orientation control only needs to consider the XY axes. However, nanorobots operate in a three-dimensional space, and their environment model accordingly becomes a three-dimensional model, as shown in Figure 4. For path planning within this three-dimensional model, the origin of the nano-robot's operational space is positioned at the lower left corner, with the X and Y axes representing horizontal coordinates and the Z axis denoting the vertical coordinate. Moreover, the size and position of obstacles in the space are not fixed. In this study, the nanorobot is regarded as a point mass in the dynamical model, meaning that its rotational degrees of freedom are neglected and only the translational motion of its center of mass is considered. However, to account for the dominant microscopic physical effects in the fluid environment, equivalent fluid resistance and driving force constraints are incorporated as external forces acting on the point mass, following the typical modeling approach in micro-/nanorobot dynamics [24,25].

Specifically, three map environments of varying sizes, namely 50mm, 100mm, and 200mm, were designed.

Each environment contains three types of obstacles: spherical, rectangular, and a combination of both. These working environments are further categorized into simple (fewer than 12 obstacles), medium (12 or more but fewer than 18 obstacles), and difficult (more than 18 obstacles) modes based on obstacle quantity. Given the nanorobot's minute volume of a few tens of nm, it is regarded as a point mass within the millimeter-scale operational space, with its initial position and target positions randomly assigned. Spherical obstacles range from 8-12mm in size, while rectangular obstacles span 8-15mm.

B. STATE AND ACTION SPACE DEFINITION

In the DRL framework, a well-defined state space and physically feasible action space are critical to accurate and practical nanorobot path planning[26]. To meet these requirements, we design the state and action spaces based on three principles: dimensional necessity, performance sensitivity, and physical consistency. The state space must comprehensively depict the nanorobot's motion state, environmental interaction characteristics, and

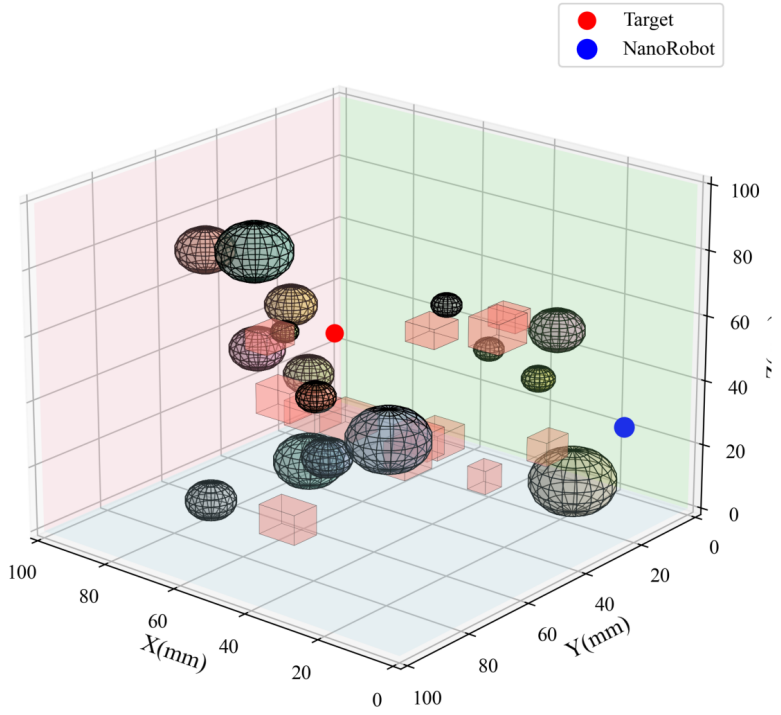


FIGURE 4. Three-dimensional model of the working environment for nanorobots

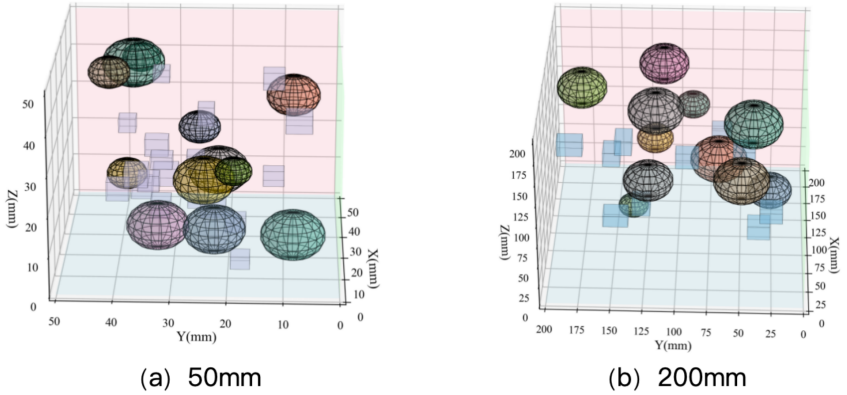


FIGURE 5. The working environments of the other two sizes of nanorobots: (a) A mixed obstacle difficult environment of 50×50 size; (b) A mixed obstacle difficult environment of 200×200 size

task progress. This paper constructs a 10-dimensional state vector $[x_t, y_t, z_t, v_{x,t}, v_{y,t}, v_{z,t}, \rho_t, d_{\text{obs},t}, step_t, b_t]$. Each dimension is verified for necessity through the control variable method (the training algorithm is IDQN, and the evaluation indicators include the target arrival success rate (TAR), the physical feasibility rate (PFR, the proportion of actions satisfying the dynamic constraints), the collision rate (CR, the number of contacts with obstacles / total number of motion steps), and the verification results are presented in Table 1.

Table 1 reveals that all 10 state dimensions are indispensable for effective path planning, with position and remaining energy being the most critical. Accurate position observation should be prioritized, while the influence of the already taken step length, though relatively weak, is retained to optimize path efficiency. The fluid resistance term P in the state space is not derived from a precise geometric drag model, but is instead calculated based on the nanorobot's instantaneous velocity and an equivalent resistance coefficient that captures the overall dissipative effect of the microscopic fluid environment. Specifically, $P = k_d \cdot \|v\|$, where k_d is an empirically determined coefficient that reflects the combined influence of viscous drag, Brownian motion-induced fluctuations, and non-uniform resistance in the microscopic environment. This simplified yet physically grounded representation enables the reinforcement learning agent to learn energy-efficient policies without overcomplicating the state space.

TABLE 1. Status and action verification results

State dimension	Physical meaning	Performance change after removal	Required basis
x_t, y_t, z_t	Three-dimensional position, unit: μm	TAR↓35.2%	Lack of position information will cause the robot to lose its spatial positioning reference and be unable to complete the goal-oriented path planning.
$v_{x,t}$, $v_{y,t}$, $v_{z,t}$	Three-dimensional velocity, unit: μm/s	PFR↓28.7%	Speed is directly related to the magnetic driving force constraint. Without speed information, it is easy to exceed the power limit.
ρ_t	Fluid resistance, unit: pN	TAR↓32.1%	The resistance determines the energy consumption for the robot's unit displacement, and directly affects whether the remaining energy can support the robot's journey to the destination.
$d_{\mathrm{obs},t}$	Obstacle distance	CR↑42.3%	Without distance information, it is impossible to prevent nanorobots from getting too close to obstacles.
$step_t$	Cumulative movement distance, unit: μm	TAR↓11.5%	A too-long step length indicates a serious detour in the path, which can easily lead to premature depletion of energy.
b_t	Current remaining energy	TAR↓36.2%	Exhaustion of energy will cause the robot to lose functionality. To prevent the interruption of the task, the intensity of the actions needs to be constrained based on the remaining energy.

In designing the action space and constraints, considering the impossibility of precise position control due to Brownian motion in microscopic environments, the action space is defined as a three-dimensional speed adjustment quantity rather than a position control quantity, that is, action $a_t = [\nabla v_{x,t}, \nabla v_{y,t}, \nabla v_{z,t}]$. Through speed closed-loop control, stable motion is achieved. To balance training efficiency and control accuracy, a discrete action space design is adopted: each speed component $[\nabla v_{x,t}, \nabla v_{y,t}, \nabla v_{z,t}]$ contains three adjustment options $[\nabla v_\delta, 0, -\nabla v_\delta]$, where $\Delta v = 0.2\,\mu$m/s is the speed adjustment step size (determined based on the robot's dynamic response characteristics; a large step size cause sudden speed changes, while a small step size increases training steps), resulting in a total of 27 actions. Prior to action execution, it is necessary to pass the constraint check, that is $|v_t| < v_{\max}$.

C. STATE AND ACTION SPACE DEFINITION

The reward function must guide the robot to simultaneously achieve obstacle avoidance, efficiency, and physical compliance [27]. This study designs a weighted multi-objective reward function to ensure balanced optimization of each goal. The reward function of this study mainly includes distance reward r_{dist}, obstacle avoidance reward r_{obs}, physical constraint reward r_{phys}, destination reward r_{goal}, and energy consumption reward r_{bt}. Therefore, the following formula holds:

$$r_t = w_1 r_{\mathrm{dist}} + w_2 r_{\mathrm{obs}} + w_3 r_{\mathrm{phys}} + w_4 r_{\mathrm{goal}} + w_5 r_{bt} \quad (18)$$

The distance-based rewards are implemented in an exponential decay form, such that the rewards decrease as the remaining distance increases, thereby motivating the robot to quickly approach the target.

$$r_{\mathrm{dist}} = \exp\left(-a \frac{\|(x_t, y_t, z_t) - (x_g, y_g, z_g)\|}{\|(x_0, y_0, z_0) - (x_g, y_g, z_g)\|}\right) - 0.5 \quad (19)$$

In the formula, a represents the attenuation coefficient.

The obstacle avoidance reward adopts a piecewise function. It severely penalizes the collision behavior and provides a gradient reward for movements within the safe distance:

$$r_{\mathrm{obs}} = \begin{cases} -2.0, & d_{\mathrm{obs},t} < d_{\mathrm{safe}}, \\ 0.3e^{-\beta(d_{\mathrm{safe}} - d_{\mathrm{obs},t})/d_{\mathrm{safe}}}, & d_{\mathrm{obs},t} \geq d_{\mathrm{safe}}. \end{cases} \quad (20)$$

In the formula, d_{safe} represents the safe distance, and β is the attenuation coefficient.

The physical constraints follow the microscopic physical laws and avoid planning physically impossible paths:

$$r_{\mathrm{obs}} = \begin{cases} -2.0, & d_{\mathrm{obs},t} < d_{\mathrm{safe}}, \\ 0.3e^{-\beta(d_{\mathrm{safe}} - d_{\mathrm{obs},t})/d_{\mathrm{safe}}}, & d_{\mathrm{obs},t} \geq d_{\mathrm{safe}}. \end{cases} \quad (21)$$

$$r_{\mathrm{phys}} = 0.2 \exp\left(-\lambda \frac{|\rho_t v_t - F_{\max}|}{F_{\max}}\right) \quad (22)$$

$$v_t = \sqrt{v_{x,t}^2 + v_{y,t}^2 + v_{z,t}^2} \quad (23)$$

In the formula, λ represents the attenuation coefficient, and $F_{\max}$ represents the maximum driving force of the nanorobot.

The terminal reward mainly enhances goal orientation. A high and sparse reward is given when reaching the target point, accelerating the convergence of the policy.

$$r_{\mathrm{goal}} = \begin{cases} 50, & \|(x_t, y_t, z_t) - (x_g, y_g, z_g)\| < 0.1, \\ 0, & \text{otherwise}. \end{cases} \quad (24)$$

The energy-saving rewards are mainly aimed at reducing the robot's loss of functionality and enabling it to reach the destination as quickly as possible within the limited energy supply.

$$r_{bt} = b_t \quad (25)$$

IV. RESULTS AND ANALYSIS

A. TEST ENVIRONMENT AND EVALUATION MECHANISM

The information of the algorithm software, hardware and simulation platform for this experiment is shown in Table 2. The specific settings are as follows: the learning rate is 0.0001, the Batch Size is 128, the capacity of the experience replay buffer is 0.00001, the discount factor is 0.95, the soft update coefficient of the target network is 0.01, the maximum greedy probability is 0.01, the maximum number of greedy actions is 10, the optimizer is Adam, and the number of attention heads is 8.

In addition, this study involves six evaluation indicators, including success rate (%), average path length (mm), average path planning time (s), loss value, Q value, and

TABLE 2. Experimental environment configuration

Category	Configuration Details
Hardware	Intel Xeon W-2245 CPU(3.9 GHz), NVIDIA RTX A5000GPU(24 GB VRAM), 64 GB DDR4 RAM
Software	Ubuntu 20.04 LTS, Python 3.8, PyTorch 1.12.0(CUDA11.6), NumPy 1.23.5, Matplotlib 3.6.2
Simulation platform	Custom 3D microscopic environment simulator, implementing physical calculations based on NumPy, and visualization using Matplotlib

reward value. The success rate is defined as the percentage of experimental times in which the algorithm-driven nanorobot successfully completes the target task out of the total experimental times. A higher success rate indicates that the algorithm has stronger adaptability to complex biological environments; the average path length and time are intended to determine which method can move to the destination more quickly and closer, thereby reducing the time spent moving within the biological body and energy consumption; the loss value is used to judge the fitting accuracy and convergence of the algorithm model. The Q value measures the rationality of the algorithm's decision-making; a high Q value for a certain state-action pair indicates that this decision is more likely to guide the robot towards task success; the feedback signal set according to the behavior effect of the nanorobot is the reward value. The higher the average reward value, the more the algorithm can continuously generate actions that approach the target and avoid risks, accelerate model convergence, and reflect the rationality of the algorithm's reward function design.

B. DIFFERENT PLANNING RESULTS FOR DIFFERENT LEVELS OF DIFFICULTY

In this study, we conducted a comparative analysis of four path planning algorithms (A*, RRT*, DQN, and IDQN) for nanorobot path planning in a rectangular obstacle environment. The experimental setup covered three different levels of complexity: simple, medium, and difficult, to simulate the diverse spatial constraints and obstacle distributions that nanorobots might encounter in biological environments. Three evaluation metrics, success rate, average path length (mm), and average path planning time (s), were used to conduct quantitative analysis of each algorithm. The results are shown in Table 3 (rectangular obstacle working environment), Table 4 (spherical obstacle working environment), and Table 5 (mixed obstacle working environment). Overall, the combined performance of the four algorithms in the three obstacle geometries showed a generally decreasing trend with increasing environmental complexity. However, there were significant differences in the decay rates of the algorithms, thus revealing the essential role of the path planning method in adapting to the three-dimensional nan oscale working space. From the quantitative results in Table 3, 4, and 5, different path planning algorithms ex hibited significant performance dif-ferences and patterns in the three core indicators: success rate, average path le ngth, and average path planning time.

A* and RRT* maintained a success rate of at least 89% in rectangular, spherical, and mixed scenarios, with the path length fluctuation range not exceeding 17 mm and the planning time always being less than 1.8 s, demonstrating their rapid response and low computational burden characteristics. However, as the obstacles transitioned from regular rectangles to random spheres and even mixed forms, the grid partition dimension of A* expanded sharply, resulting in a slight extension of the average path and an increase in time cost by approximately 60%, while RRT* experienced a 2 – 6 percentage point decrease in success rate due to the simultaneous increase in collision detection times and sampling rejection rates. This indicates that traditional path planning algorithms, with their mature search and sampling mechanisms, maintained a high success rate in various simple and medium complexity environments and had significantly lower average path planning times than reinforcement learning algorithms. However, the path length optimization ability of RRT* was limited as the complexity of the environment increased, while the path length of A* showed a steady growth trend, reflecting its inherent constraints in global optimality.

In contrast, the basic reinforcement learning algorithm DQN performed poorly. Its success rate decreased significantly as the environment evolved from simple to difficult and the obstacles transitioned from rectangular to spherical and then to mixed types. Specifically, the end-to-end policy of DQN achieved an initial success rate of >84% in simple scenarios with the help of deep network generalization ability. However, as the obstacle density and shape randomness increased, the exploration-exploitation trade-off of DQN quickly became unbalanced, resulting in a sudden drop in success rate to 50.8 – 59.0% in difficult scenarios. The average path length was significantly longer than that of other algorithms, and the planning time continuously increased and expanded with the iterative backtracking, reaching 5.5 s. This indicates its deficiency in easily falling into local optima and insufficient adaptability to the scene. Additionally, the extreme compression of the partially observable state space on the nanorobot's sensing bandwidth led to significant deviations in the experience samples, thereby inducing local convergence of the strategy.

After improving DQN, IDQN demonstrated excellent performance in path planning quality and robustness. Specifically, IDQN not only achieves success rates that are close to or superior to those of traditional algorithms (rectangular: 95.4%, spherical: 91.5%, mixed: 93.7%) across all obstacle types and complexity levels, but also has the shortest average

path length among all algorithms (simple rectangular: 78 mm, better than 80 mm of RRT*; medium spherical: 95 mm, similar to 94 mm of A*). This reduction in path length is particularly noticeable in spherical and mixed obstacles, where the robot significantly outperforms DQN by shortening the path by more than 20 mm and increasing the success rate by up to 42 percentage points. Moreover, it achieves a significant performance improvement while maintaining no significant difference in average path planning time compared to DQN, achieving a high reliability of 93.7% in difficult mixed scenarios with a time cost of only 4.7 seconds. This clearly validates the algorithm's ability to search for navigation paths close to the global optimum through continuous interaction with the environment. It also demonstrates that IDQN achieves a favorable balance between decision quality and computational cost, as well as strong adaptability and robustness for complex nanorobot path planning tasks Notably, IDQN achieves the highest success rate (95.4%) in Difficult mode, which may seem counter-intuitive but actually demonstrates the effectiveness of curriculum learning—the agent leverages progressively accumulated experience and the Transformer attention mechanism achieves sharper feature discrimination in dense obstacle fields. In summary, IDQN, by integrating deep representation and obstacle perception training, to some extent, overcomes the limitations of traditional methods in structured static micro-channels, providing empirical evidence for selecting path planners for nanorobots in complex, dynamic, and previously unknown biological environments.

Furthermore, we visualized the path planning results of the four algorithms under three levels of obstacle complexity. Spaces A, B, and C represent three different working environments. The path planning outcomes for single rectangular obstacles, single spherical obstacles, and mixed obstacles were also visualized, as depicted in Figures 6 (rectangular obstacles), Figure 7 (spherical), and Figure 8 (mixed obstacles). The red dots represent the starting points, the blue dots represent the target endpoints, the blue lines represent the path planning trajectory of A*, the red lines represent the path planning trajectory of RRT*, the green lines

represent the path planning trajectory of DQN, and the orange lines represent the path planning trajectory of IDQN. Overall, from Figure 6 to Figure 9, A* has the best smoothness. Although in simple environments, due to grid limitations or failure of the heuristic function, the paths become less natural, as the complexity of the environment increases, the smoothness and dynamic adaptability of the paths decrease, reflecting the limitations of traditional algorithms in complex constraint scenarios. RRT* has the roughest initial paths in all shape obstacles, but as the path is iteratively optimized, its smoothness improves, and the sampling advantage becomes apparent, enabling the nanorobot to bypass dense obstacles. The worst-performing algorithm is DQN. For example, in the simple environment of Figure 6, the B working space experiences collisions, and in the medium complexity obstacle environment of Figure 6, the A working space also encounters collisions. Beyond the initial paths, DQN's trajectories exhibit oscillation or backtracking, with the least smooth paths, displaying a repetitive and zigzag pattern, which also reflects its strategy exploration characteristic based on state-action, but it necessitates targeted optimization. After improvement, IDQN efficiently plans the optimal path without any detours. Its trajectory is smoother and shorter, and in all difficulty environments, IDQN's trajectory is closer to the perfect path than DQN's. While its overall performance is comparable to A*'s, combined with the quantitative analysis in Table 3-5, it can be concluded that IDQN is the optimal path planning algorithm in this study, to some extent, addressing the deficiencies of traditional DQN.

C. ANALYSIS OF RESULTS IN COMPLEX AND MIXED ENVIRONMENTS

Section 4.2 provided a comprehensive analysis of algorithm results from both quantitative and qualitative perspectives, considering varying levels of complexity in working environments. This section aims to analyze the detailed results of IDQN in a mixed complex obstacle environment.

The Loss curve and the average Q-value curve of IDQN are shown in Figure 9. The blue curve in Figure 9(a) represents the instantaneous loss for each training episode, and the red curve represents the moving average loss based on a window size of 50 (mean (window=50)). It can be seen that the initial loss value is relatively high (about 60,000), but as the training progresses to 16,000 episodes, the loss value gradually drops to 0. Overall, as the number of training episodes increases, the instantaneous loss shows a significant upward trend in the early stage, indicating that the network quickly captures the key mapping patterns between obstacles and targets by relying on the sufficient diversity of the experience replay pool in the initial stage. Subsequently, the loss curve enters a wide-amplitude oscillation range centered at 20,000, and the 50-round sliding average window shows a standard deviation of approximately $\pm$4,000, reflecting that the policy constantly encounters new local minima and collision boundaries in the highly random mixed geometric obstacles, resulting in frequent changes in the gradient direction. This phenomenon is highly consistent with the high uncertainty of a partially observable state space at the nanoscale. In the later stage, although the instantaneous loss still has certain fluctuations, the moving average loss curve gradually stabilizes, indicating that the IDQN algorithm has good convergence and optimization capabilities when dealing with complex environments, effectively reducing prediction errors and improving model performance. The continuous decline in loss values also reflects that the algorithm has achieved a good balance between exploration and exploitation, thereby achieving efficient learning in the mixed obstacle environment.

Figure 9(b) shows the average Q-value change curve of

TABLE 3. Evaluation results of various path planning algorithms in a rectangular obstacle environment

Algorithm	Disability level	Success rate	Average path length (mm)	Average path planning time (s)
A*	Simple	93.2%	85	1.0
A*	Medium	94.1%	88	1.3
A*	Difficult	91.8%	92	1.4
RRT*	Simple	92.3%	80	0.9
RRT*	Medium	91.2%	83	1.3
RRT*	Difficult	92.1%	81	1.5
DQN	Simple	88%	108	4.1
DQN	Medium	78%	125	4.3
DQN	Difficult	59%	135	4.9
IDQN	Simple	93.6%	78	4.0
IDQN	Medium	92.1%	85	4.3
IDQN	Difficult	95.4%	89	4.5

TABLE 4. Evaluation results of various path planning algorithms in spherical obstacle environments

Algorithm	Disability level	Success rate	Average path length (mm)	Average path planning time (s)
A*	Simple	91.8%	91	1.2
A*	Medium	90.5%	94	1.7
A*	Difficult	89.6%	98	1.8
RRT*	Simple	88.2%	86	1.1
RRT*	Medium	87.8%	89	1.7
RRT*	Difficult	86.1%	87	1.7
DQN	Simple	84.9%	116	4.5
DQN	Medium	70.5%	139	4.9
DQN	Difficult	55.2%	149	5.5
IDQN	Simple	97.4%	84	4.4
IDQN	Medium	93.4%	95	4.9
IDQN	Difficult	91.5%	99	4.9

TABLE 5. Evaluation results of various path planning algorithms in a mixed environment

Algorithm	Disability level	Success rate	Average path length (mm)	Average path planning time (s)
A*	Simple	93.4%	88	1.1
A*	Medium	92.1%	91	1.5
A*	Difficult	91.1%	95	1.6
RRT*	Simple	90.1%	83	1.0
RRT*	Medium	89.4%	86	1.5
RRT*	Difficult	91.5%	84	1.61
DQN	Simple	86.2%	112	4.3
DQN	Medium	74.5%	132	4.6
DQN	Difficult	50.8%	142	5.2
IDQN	Simple	95.6%	81	4.2
IDQN	Medium	91.6%	90	4.6
IDQN	Difficult	93.7%	94	4.7

IDQN during training in a mixed complex obstacle environment. The curve generally exhibits the typical characteristics of RL evolution, showing a stepwise ascent followed by plateau oscillation. Initially, the average Q-value rises rapidly, then stabilizes over time. In the early stage (0-6000), the steep slope of the Q mean with no significant regression indicates that the experience replay pool quickly accumulates high-reward samples. Network parameters are updated stably along the gradient direction, and the strategy initially acquires the ability to bypass irregular obstacles and move towards the target. The subsequent oscillation and flat stage (6000-16000) indicate that in the later stage of the training process, IDQN continuously improves its judgment ability on what actions to take in the mixed complex obstacle environment, gradually learning more effective path planning strategies and the learning effect becomes increasingly stable in the later stage. Therefore, it can be determined that the current strategy has fully exploited the potential

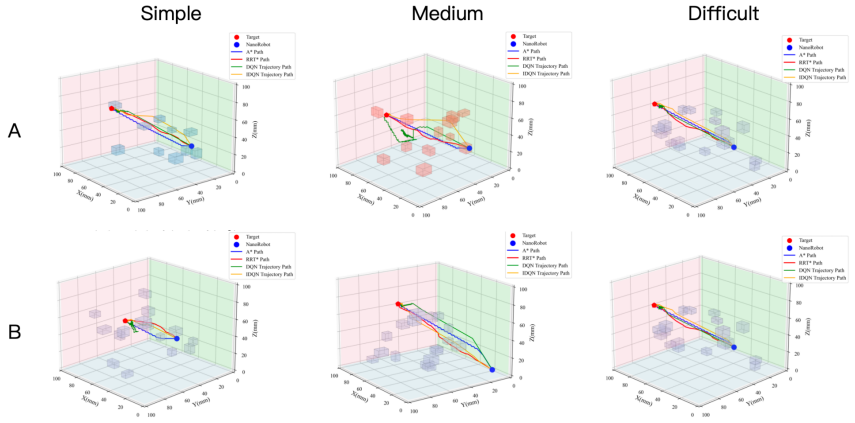


FIGURE 6. The path planning results in three complex working environments under rectangular obstacles

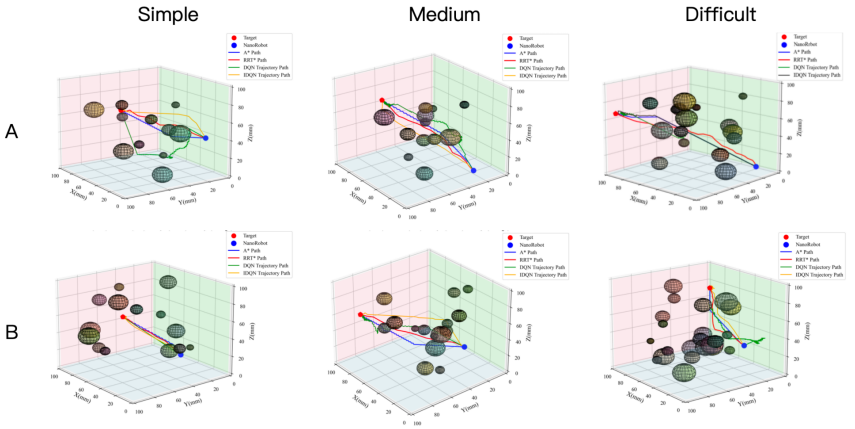


FIGURE 7. The results of path planning in three complex working environments under spherical obstacles

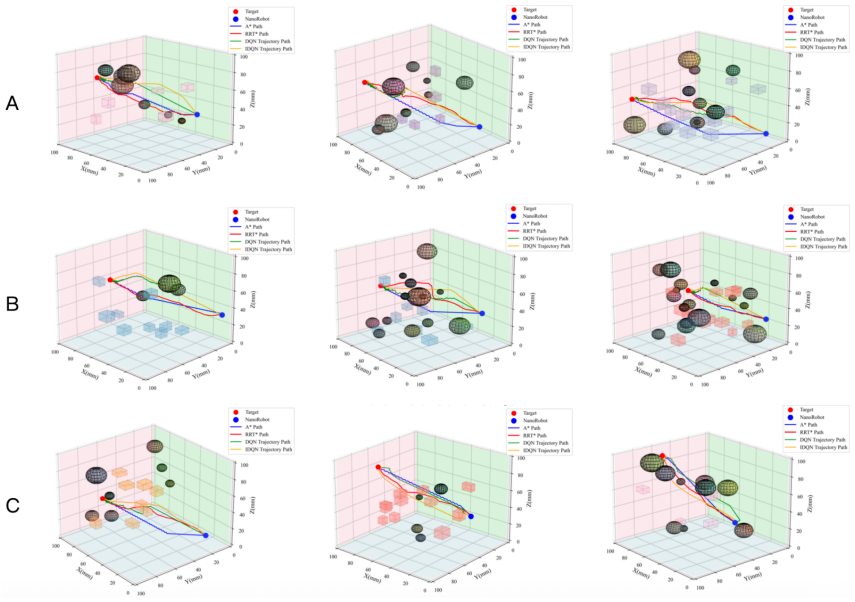
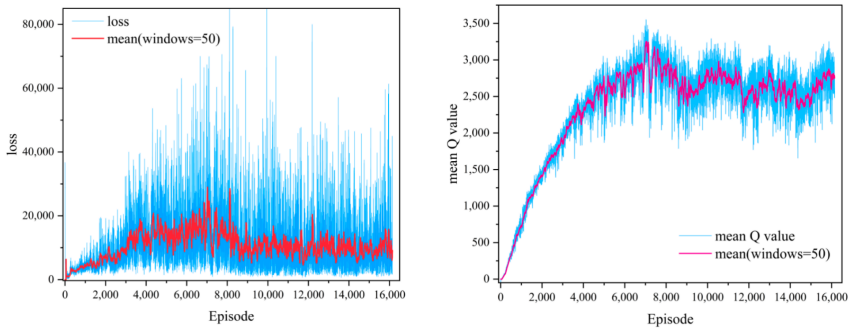


FIGURE 8. The planning results of three complex working environment paths under the condition of mixed disorders

optimal trajectory of the mixed obstacle scenario, providing a reliable value benchmark for the subsequent real-time navigation of nanorobots.

In addition, in Figure 10, the success rate curve and reward values of IDQN are also visualized. The reward values are compared with those of DQN, as shown in



(a) Loss (b) Average Q–value

FIGURE 9. Loss curve and Q-value curve in the IDQN challenging environment

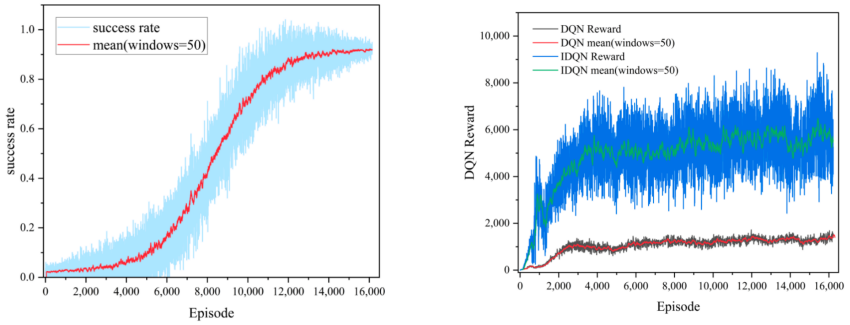
Figures 10(a) and 10(b).

From Figure 10(a), it can be seen that as the training rounds increase, the success rate continuously rises from a level close to 0, and approaches 1 in the later stages. The red moving average curve more smoothly reflects this upward trend. Specifically, the nanorobot is in the exploration stage in the early training period, so the success rate is low, indicating that the agent has not yet mastered the strategy for effective navigation and achieving the goal in complex environments. With the increase in the number of training rounds, the average improvement per 1,000 rounds is approximately 0.15, and a slight periodic decline is observed in the 50- round moving average window, reflecting that the strategy occasionally encounters new collision boundaries when attempting shorter paths, but thanks to the priority replay mechanism, the effective samples are reused, maintaining a stable upward trend overall. Subsequently, it gradually enters the saturation and theoretical upper limit stage, with the peak plateau value reaching 93.7% in the difficult mixed obstacle scenario. In Figure 10(b), the black line represents the instantaneous reward of DQN per episode, the red line represents the moving average reward of DQN with a 50-round window; the blue line represents the instantaneous reward of IDQN, and the green line represents the moving average reward of IDQN. It can be clearly seen that the reward of DQN is overall lower and remains at a low level after training; while the reward of IDQN is much higher, and it keeps increasing with the increase in training rounds, stabilizing in a higher range in the later stages. This indicates that the improved IDQN can more effectively obtain high rewards during training, and the learning effect is better than that of the basic DQN. Specifically, in the initial stage (0 – 2,000 rounds), both curves start from a low position close to 0, but IDQN, relying on importance sampling weighting and the reuse of obstacle perception features, has a significantly higher slope of the moving average than DQN, indicating that the ordinary experience replay has a significant sample efficiency bottleneck for the nanorobot navigation task with high variance and sparse rewards. In the middle learning period from 4,000 to 8,000 rounds, the DQN curve shows a slow growth phenomenon, with the mean only increasing from 2,000 to 4,000 points, and the standard deviation per 1,000 rounds is as high as $\pm$ 1,200 points, reflecting that the strategy repeatedly collides with dense obstacle boundaries, resulting in significant fluctuations in rewards. At this time, IDQN has steadily risen to the 6,000 – 7,000 point range, with the fluctuation range narrowing to $\pm$ 400 points, indicating that the priority replay pool effectively inhibits the accumulation of biased samples, and the value network's fitting accuracy for safe-short-range trajectories has significantly improved. After 8,000 rounds, the DQN curve gradually reaches its theoretical upper limit and shows almost no growth in the subsequent 8,000 rounds, revealing that the strategy distribution has fallen into a local optimum; while IDQN continues to increase monotonically, eventually converging to the 9,200 – 9,400 point plateau around 14,000 rounds, approximately 55% higher than DQN, with a fluctuation window of less than 3%, verifying that attention weighting and dynamic importance sampling can systematically mitigate sample sparsity and overestimation bias in partially observable, geometrically mixed nanorobot navigation scenarios, achieving superior cumulative returns and policy robustness.

D. PLANNING RESULTS FOR DIFFERENT MAP SIZES

Sections 4.3 and 4.4 primarily discuss the path planning results of different algorithms within a 100mm working environment, neglecting to cover outcomes in other map sizes. Therefore, this section conducts a comparative analysis in a mixed, challenging, and complex obstacle working environment, with results presented in Table 6.

In a mixed, difficult obstacle environment, varying map sizes exert differentiated impacts on the path planning performance of DQN and IDQN. This reflects that DQN is insufficiently adaptable to changes in map scale in complex environments, and the coverage efficiency of the experience replay pool for sparse rewards shows an exponential decay. The network parameter updates are overwhelmed by a large number of invalid collision samples, making it prone to getting stuck in local optima due to the expansion of the state space, thereby limiting the planning efficiency and success rate. In contrast, IDQN demonstrates superior performance in all map sizes, compressing the number of effective nodes to approximately 45% of DQN's, and signifi-



(a) Success rate curve (b) Reward value curve

FIGURE 10. Success rate curve in IDQN challenging environment and reward value curves before and after improvement

cantly reducing the average path planning time. For a 50mm map, the planning time is reduced by 44.7% compared to DQN. Even when the map size expands to 200mm, the time reduction rate remains at around 18.3%. This indicates that its attention mechanism can continuously focus on the high-value state subspaces as the map expands, avoiding the dilution of sampling density by geometric complexity. The above experimental results show that IDQN, through algorithm improvement, not only achieves efficient search and decision-making in small-scale maps, but also can optimize the planning time and node number through a better strategy learning mechanism in the face of the increased state space complexity brought by map scale expansion. It also demonstrates significant advantages in a mixed difficult obstacle multi-scale environment.

Furthermore, Figure 11 presents the path planning results of the four algorithms (A*, RRT*, DQN, IDQN) in a challenging environment with mixed obstacles, based on two different map sizes. By observing the results from the small-scale scenario (50mm) in the upper row, RRT* successfully forms a continuous and relatively smooth curve in the narrow passage by relying on dense sampling, although there are a few redundant nodes, it successfully avoids the irregular obstacles; the DQN trajectory is clearly jagged, showing frequent turns and local retractions, indicating that the strategy has not fully converged and the exploration noise still dominates. When the space expands to 200mm, the sampling advantage of RRT* is further amplified, and the node distribution adapts to the sparse distribution according to the obstacle density, showing an increasingly optimal characteristic; while the oscillation amplitude of the DQN trajectory significantly increases, long-distance detours and repeated crossing of areas occur, revealing its inherent defects after the state space increases dramatically, such as the explosion of value estimation variance and insufficient coverage of the replay pool. However, the proposed IDQN algorithm in this study can successfully avoid all obstacles, and the path is smooth and direct, showing extremely high decision-making efficiency without obvious ineffective wandering or falling into local optimum. The entire path tends to be globally optimal. This qualitative result is consistent with the quantitative data in Table 6, intuitively proving that the IDQN algorithm has outstanding environmental understanding ability and spatial search strategy, and its planning results have high safety, high efficiency, and excellent passability.

V. DISCUSSION

This study addresses the path planning problem of nanorobots in complex micro-environments and proposes an intelligent path planning algorithm based on the improved Deep Q Network (IDQN). This algorithm effectively solves the core problems faced by traditional algorithms in path planning in microscopic three-dimensional environments, such as dynamic uncertainty, complex physical constraints, and insufficient adaptability. At the same time, systematic comparative experiments were conducted with A* algorithm, traditional DQN algorithm, and the method based on artificial potential field in various complex simulation micro-environments. The performance of the proposed algorithm was comprehensively evaluated from multiple dimensions such as path optimality, computational efficiency, and environmental adaptability. The experimental results show that the IDQN algorithm maintains a high path planning success rate while significantly improving the convergence speed and path quality in various obstacle types and difficulty levels. This provides reliable technical support for the precise navigation of nanorobots in practical applications. From Sections 4.2 and 4.3, it can be observed that the geometric shape of obstacles has a significant impact on the performance of the path planning algorithm. It is found that traditional search algorithms such as A perform robustly in regular obstacles due to their structured heuristic search mechanism, but their path smoothness and dynamic adaptability significantly decline when facing spherical obstacles with continuous surfaces or randomly distributed mixed obstacles; RRT* methods, although having certain geometric independence, exhibit a sharp decrease in sampling efficiency in narrow channels and high-dimensional state spaces. In all obstacle conditions, the traditional DQN algorithm is prone to fall into local optimal solutions, showing repeated wandering in concave areas of obstacles and unable to find effective paths. The IDQN algo rithm, through its improved experience replay sampling strategy and adaptive

TABLE 6. Performance Comparison of Difficult-Degree DQN and IDQN in Mixed Disorders

Algorithm	Map Size	Success rate	Average path planning time	Number of Nodes	Time Reduction Rate	Node Reduction Rate
DQN	50-50mm	37.1%	3.8s	13	-	-
DQN	100-100mm	50.8%	5.2s	17	-	-
DQN	200-200mm	65.7%	7.1s	22	-	-
IDQN	50-50mm	91.4%	2.1s	7	44.7%	45.6%
IDQN	100-100mm	93.7%	4.6s	9	11.5%	45.2%
IDQN	200-200mm	94.3%	5.8s	12	18.3%	45.1%

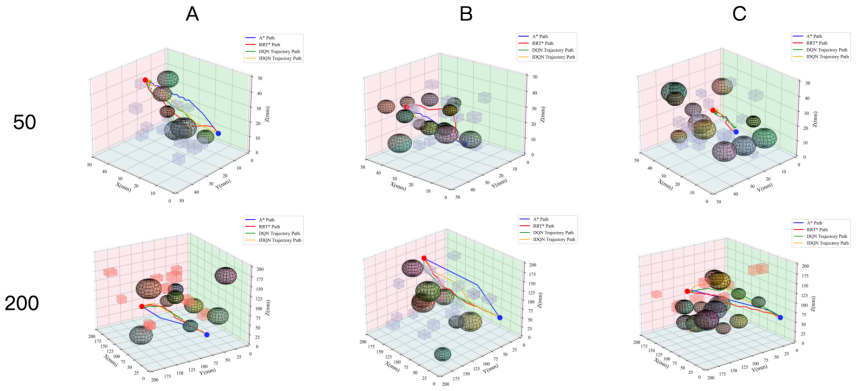


FIGURE 11. The planning results of two different-sized paths in a challenging mixed environment

exploration mechanism, and benefiting from the adaptive focusing ability of the Transformer attention mechanism on key environmental features, can effectively identify the spatial distribution patterns of different geometric shape obstacles, enabling nanorobots to exhibit stronger robustness in complex geometric environments and effectively avoid local optimal traps and find feasible paths within a reasonable time. This advantage is particularly evident in maze-like environments with multiple continuous bends.

The map dimension and size are important environmental parameters for nanorobot path planning, directly affecting the state space construction, search efficiency, and strategy generalization ability of the algorithm.

Their influence mechanism is reflected in two aspects: the increase in state complexity brought by the three-dimensional dimension and the expansion of the space exploration range caused by the size expansion [28]; Different from the working environment of traditional macro robots, which mainly relies on a two-dimensional plane, the three-dimensional working environment of nanorobots requires the path planning algorithm to construct a high-dimensional state vector including three-dimensional position and three-dimensional velocity (in this study, it is 10-dimensional). This puts forward higher requirements for the algorithm's feature extraction and dimension compression capabilities. The computational time of A* algorithm in three-dimen sional environments is nearly two orders of magnitude higher than that in two-dimensional environments, mainly due to the exponential growth of the number of nodes to be explored in high-dimensional space [29]. The methods based on reinforcement learning, due to their end-to-end learning characteristics, show better scalability when expanding dimensions, especially the IDQN algorithm, which can effectively handle high-dimensional state information and maintain high planning efficiency in three-dimensional complex environments [30]. In smaller-sized maps, the exhaustive search feature of traditional algorithms enables them to quickly find the optimal solution. However, in extremely large-sized maps, all algorithms face challenges such as memory limitations and computational efficiency. But the IDQN algorithm, through its experience replay mechanism and target network update strategy, significantly reduces memory consumption while maintaining path quality. This result indicates that IDQN has good scale invariance and generalization ability, which is suitable for the actual deployment requirements of nanorobots in variable-scale environments. It fully validates its efficient path planning ability in multi-scale three-dimensional environments. Although this study has achieved phased results in optimizing the path planning algorithm for nanorobots, there are still two main limitations: First, the geometric shapes of the obstacles set in the experiments are all fixed, and the dynamic obstacles that may exist in the micro-environment such as biological organisms, material crystals, and nano-pipes have not been considered, resulting in the lack of verification of the algorithm's adaptability in dynamic microscopic environments; Second, the research only improves the basic DQN algorithm and does not incorporate the advantageous characteristics of other classic algorithms into the fusion design. However, the experimental results show that the A* algorithm still has the advantages of smooth path and short planning time in regular obstacle environments, and its heuristic search idea has important reference value for improving the initial exploration efficiency of reinforcement

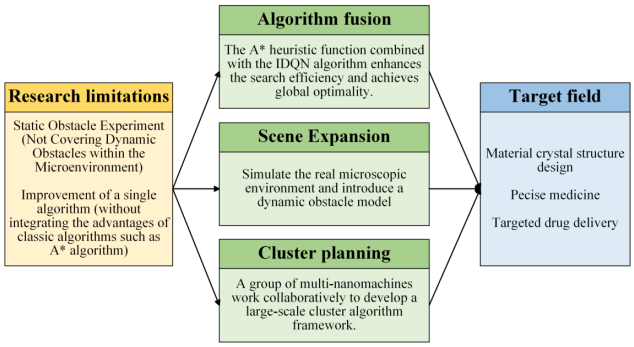


FIGURE 12. Current research limitations and the framework for the next step of the study

learning algorithms. Based on the current research results, future research work will focus on the following directions: (1) Integrate the exploration of the A* algorithm heuristic function with the IDQN algorithm, considering integrating the heuristic information of A* into the reward function design or state feature representation of IDQN to further enhance the search efficiency and global optimality of the algorithm in complex environments; (2) Subsequent research will first expand the experimental scenarios and introduce dynamic obstacle models to simulate more realistic biological microscopic environments; (3) Study the multi-nanorobot group collaborative path planning scenarios to meet more complex biomedical application requirements, and develop algorithm frameworks capable of handling large-scale cluster intelligence. These subsequent research works will further enhance the application potential of nanorobots in cutting-edge fields such as material crystal structure design, precise medicine, and targeted drug delivery. The limitation and outlook framework is shown in Figure 12.

VI. CONCLUSION

This study addresses the path planning problem of nanorobots in complex microscopic textile fiber environments and proposes an NP-IDQN framework based on improved deep reinforcement learning. By leveraging the IDQN algorithm at its core, this framework optimizes the navigation efficiency and adaptability of nanorobotic systems within dense textile industrial microstructures and technical material landscapes.

The experimental results show that NP-IDQN demonstrates superior path planning capabilities in rectangular, spherical, and mixed obstacle environments. It not only outperforms traditional A, RRT, and basic DQN algorithms in key indicators such as success rate, path length, and planning time, but also maintains high success rate and path quality in highly complex mixed obstacle environments. This verifies its good adaptability and robustness in microscopic complex environments, and proves that the proposed improvement effectively alleviates the problems of local optimum and sparse rewards. Specifically, in experiments with different obstacle types (rectangular / spherical / mixed), difficulty levels (easy / medium / difficult), and map sizes (50mm / 100mm / 200mm), IDQN maintains high success rate (up to 97.4%), short path length (more than 20mm shorter than DQN), and stable planning efficiency.

This algorithm is the first to combine the Transformer attention mechanism with deep reinforcement learning for nanorobot path planning. The 10-dimensional state space and multi-objective reward function constructed fully consider microscopic physical constraints such as fluid resistance and energy consumption, and the generated paths are physically feasible, providing key technical support for nanorobots to move from laboratory simulation to practical application.

AUTHOR CONTRIBUTIONS

Jitong Miao designed, collected and analyzed the data, and drafted the manuscript. Jitong Miao conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Jitong Miao participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] H. Liu, Z. Li, R. Zhang, Y. Liu, and Y. He, "A Novel Method for Technology Roadmapping: Nanorobots," Appl Sci, vol. 14, Art. no. 10606, 2024, doi: 10.3390/app142210606.

[2] J. Li, B. Esteban-Fernández de Ávila, W. Gao, et al., "Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification," Science robotics, vol. 2, no. 4, pp. eaam6431, 2017, doi: 10.1126/scirobotics.aam6431.

[3] H. Zhou, C. Mayorga-Martinez C, S. Pané, et al., "Magnetically driven micro and nanorobots," Chemical Reviews, vol. 121, no. 8, pp. 4999-5041, 2021, doi: 10.1021/acs.chemrev.0c01234.

[4] M. Urso, M. Ussia, and M. Pumera, "Smart micro-and nanorobots for water purification," Nature Reviews Bioengineering, vol. 1, no. 4, pp. 236-251, 2023, doi: 10.1038/s44222-023-00025-9.

[5] X. Kong, P. Gao, J. Wang, et al., "Advances of medical nanorobots for future cancer treatments," Journal of Hematology & Oncology, vol. 16, no. 1, pp. 74, 2023, doi: 10.1186/s13045-023-01463-z.

[6] Z. Wu, Y. Chen, D. Mukasa, et al., "Medical micro/nanorobots in complex media," Chemical Society Reviews, vol. 49, no. 22, pp. 8088-8112, 2020, doi: 10.1039/D0CS00309C.

[7] J. Huang, C. Chen, J. Shen, et al., "A self-adaptive neighborhood search A-star algorithm for mobile robots global path planning," Computers and Electrical Engineering, vol. 123, Art. no. 110018, 2025.

[8] M. Ding, X. Zheng, L. Liu, et al., "Collision-free path planning for cable-driven continuum robot based on improved artificial potential field," Robotica, vol. 42, no. 5, pp. 1350-1367, 2024, doi: 10.1017/S026357472400016X.

[9] Z. Wu, Z. Meng, W. Zhao, and Z. Wu, "Fast-RRT: A RRT-Based Optimal Path Finding Method," Appl Sci, vol. 11, Art. no. 11777, 2021, doi: 10.3390/app112411777.
[10] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," Procedia Computer Science, vol. 127, pp. 180-189, 2018.
[11] M. Morin, I. Abi-Zeid, and G. Quimper C, "Ant colony optimization for path planning in search and rescue operations," European Journal of Operational Research, vol. 305, no. 1, pp. 53-63, 2023.
[12] G. Che, L. Liu, and Z. Yu, "An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 8, pp. 3349-3354, 2020, doi: 10.1007/s12652-019-01531-8.
[13] A. Ahmad H, A. Hussain, and N. Akhtar M, "Enhancing Micro/Nanorobot Navigation for Medical Applications Using Reinforcement Learning," 2024 19th International Conference on Emerging Technologies (ICET). IEEE, pp. 1-5. https:// ieeexplore.ieee.org/abstract/document/10935174, 2024.
[14] M. Xi, J. Yang, J. Wen, et al., "An information-assisted deep reinforcement learning path planning scheme for dynamic and unknown underwater environment," IEEE Transactions on Neural Networks and Learning Systems, 2023, doi: 10.1109/tnnls.2023.3332172.
[15] Z. Li, N. Shi, L. Zhao, et al., "Deep reinforcement learning path planning and task allocation for multi-robot collaboration," Alexandria Engineering Journal, vol. 109, pp. 408-423, 2024.
[16] J. Xin, H. Zhao, D. Liu, et al., "Application of deep reinforcement learning in mobile robot path planning," 2017 Chinese Automation Congress (CAC). IEEE, pp. 7112-7116, 2017, doi: 10.1109/CAC.2017.8244061.
[17] L. Yang, J. Jiang, F. Ji, et al., "Machine learning for micro-and nanorobots," Nature Machine Intelligence, vol. 6, no. 6, pp. 605-618, 2024, doi: 10.1038/s42256-024-00859-x.
[18] Y. Yang, L. Juntao, and P. Lingling, "Multi-robot path planning based on a deep reinforcement learning DQN algorithm," CAAI Transactions on Intelligence Technology, vol. 5, no. 3, pp. 177-183, 2020, doi: 10.1049/trit.2020.0024.
[19] J. Gao, W. Ye, J. Guo, and Z. Li, "Deep Reinforcement Learning for Indoor Mobile Robot Path Planning," Sensors, vol. 20, pp. 5493, 2020, doi: 10.3390/s20195493.
[20] N. Kumaar A A and S. Kochuvila, "Mobile service robot path planning using deep reinforcement learning," IEEE Access, vol. 11, pp. 100083-100096, 2023, doi: 10.1109/ACCESS.2023.3311519.
[21] Z. Gu, R. Zhu, T. Shen, et al., "Autonomous nanorobots with powerful thrust under dry solid-contact conditions by photothermal shock," Nature Communications, vol. 14, no. 1, pp. 7663, 2023, doi: 10.1038/s41467-023-43433-6.
[22] Y. Yang, A. Bevan M, and B. Li, "Efficient navigation of colloidal robots in an unknown environment via deep reinforcement learning," Advanced Intelligent Systems, vol. 2, no. 1, Art. no. 1900106, 2020, doi: 10.1002/aisy.201900106.
[23] L. Yang, J. Jiang, X. Gao, et al., "Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time distribution planning," Nature Machine Intelligence, vol. 4, no. 5, pp. 480-493, 2022, doi: 10.1038/s42256-022-00482-8.
[24] Z. Li, K. Wang, C. Hou, C. Li, F. Zhang, et al., "Self-Sensing Intelligent Microrobots for Noninvasive and Wireless Monitoring Systems," Microsyst Nanoeng, vol. 9, no. 1, pp. 102, 2023, doi: 10.1038/s41378-023-00574-4.
[25] Q. Wang, Q. Wang, Z. Ning, F. Chan K, J. Jiang, et al., "Tracking and Navigation of a Microswarm under Laser Speckle Contrast Imaging for Targeted Delivery," Sci Robot, vol. 9, no. 87, 2024, doi: 10.1126/scirobotics.adh1978.
[26] G. Li, C. Shao, Z. Wang, Y. Lu, K. Deng, and D. Gao, "A Dual-Robot Digital Radiographic Inspection System for Rocket Tank Welds," Appl Syst Innov, vol. 8, pp. 151, 2025, doi: 10.3390/asi8050151.
[27] X. Li, T. Li, Y. Zhang, Y. Zhang, Z. Li, L. Ban, and K. Sun, "A-TEB: An Improved A Algorithm Based on the TEB Strategy for Multi-Robot Motion Planning," Sensors, vol. 25, pp. 6117, 2025, doi: 10.3390/s25196117.
[28] S. Swinton, E. McGookin, and D. Thomson, "Improving Rover Path Planning in Challenging Terrains: A Comparative Study of RRT-Based Algorithms," Robotics, vol. 14, pp. 135, 2025, doi: 10.3390/robotics14100135.
[29] P. Zhang, J. Liu, Y. Fu, and J. Sun, "A Planning Framework Based on Semantic Segmentation and Flipper Motions for Articulated Tracked Robot in Obstacle-Crossing Terrain," Biomimetics, vol. 10, pp. 627, 2025, doi: 10.3390/biomimetics10090627.
[30] S. Lin, A. Liu, J. Wang, and X. Kong, "A Review of Path-Planning Approaches for Multiple Mobile Robots," Machines, vol. 10, pp. 773, 2022, doi: 10.3390/machines10090773.