Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD38752

DDPG-Based Path Planning for Medical Microrobot Vascular Navigation

Mingcheung Wang

American International School, Hong Kong, China
simonwmz0511@gmail.com

Abstract. This research focuses on the critical challenge of autonomous path planning for medical microrobots in complex and constrained environments, such as targeted drug delivery and minimally invasive surgery. Traditional planning algorithms often struggle with the dynamic nature and stringent safety requirements of medical applications. To address this, we propose a framework based on deep reinforcement learning (DRL) to train an agent capable of navigating from a start to a target point while adhering to strict safety boundaries. The problem is formulated as a Markov Decision Process (MDP), defining the robot's state by its position and orientation. We systematically investigate and compare two prominent DRL algorithms: Deep Q-Networks (DQN) for discrete action spaces and Deep Deterministic Policy Gradient (DDPG) for continuous control, allowing for finer and smoother movements. A key innovation of this work is the design of a composite reward function tailored for medical scenarios. This function incorporates incentives for reaching the target, penalties for boundary violations and deviations from a predefined safety corridor, and considerations for path efficiency and smoothness. This meticulously crafted reward mechanism ensures that the agent prioritizes safety and precision above all else. The models, built upon Multi-Layer Perceptrons, are trained to optimize the long-term cumulative reward, effectively learning a robust policy for safe and efficient navigation in simulated medical environments. This study demonstrates the significant potential of DRL in developing intelligent and reliable guidance systems for next-generation medical robotics.

Keywords: Medical Microrobots, Path Planning, Deep Reinforcement Learning

1. Introduction

With the rapid development of micro-nano technology and biomedical engineering, medical microrobots (MMRs) have demonstrated revolutionary potential in fields such as targeted drug delivery, minimally invasive surgery, and in vivo diagnostics [1]. According to industry reports, over twenty million minimally invasive surgeries are performed worldwide each year. However, traditional robotic systems are limited by the size of their mechanical arms and rigid structures, making it difficult to safely operate within capillaries (diameters less than 5–10 μm) or fragile tissues (such as the retina or brain nerves) [1]. By contrast, microrobots—with their microscale dimensions and controllable mobility—can overcome these limitations and enable precise medical

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD38752

interventions. Nevertheless, current medical microrobots remain primarily at the laboratory validation stage, and the technology has yet to reach a level of maturity for large-scale clinical applications. One of the main challenges facing microrobotic systems is path planning, which can be categorized into three core aspects: (1) the dynamic variability of the environment, (2) the computational difficulty of real-time decision-making, and (3) the challenge of quantitatively defining complex safety constraints [2]. Deep Reinforcement Learning (DRL) provides a powerful, data-driven toolkit that can address these three issues, enabling the solution of complex modeling and computational problems in medical microrobot path planning that traditional methods struggle to handle [3].

Medical microrobots differ fundamentally from traditional industrial robots, primarily due to biomechanical constraints and real-time decision-making demands. From a biomechanical perspective, internal tissue deformation causes real-time variations in obstacle boundaries (e.g., stomach wall contraction leads to path deviations). Non-Newtonian fluids (such as blood or mucus) introduce nonlinear resistance to motion [1]. In terms of decision-making, traditional offline planning algorithms (such as A*) rely on static map updates and cannot respond dynamically to real-world changes [4]. Compared with such methods, deep reinforcement learning has several key advantages: adaptive environmental transformation, end-to-end decision-making, and non-reliance on precise environment modeling [3]. Traditional offline planning methods depend heavily on predefined data models and large-scale computation. For instance, the A* algorithm must traverse a vast number of nodes to ensure accuracy [4], which is time-consuming in complex maps such as vascular networks. Furthermore, when the real-world environment deviates from the static model used in computation, the algorithm must be reconstructed to ensure reliable results [2]. DRL models, however, adapt through continuous learning and environmental feedback, enabling flexible and robust decision-making. Before the advent of end-to-end approaches, traditional decision systems resembled assembly-line pipelines, where each step depended on the accuracy of the previous one. Consequently, any error would propagate and amplify in subsequent steps [5]. Moreover, such systems required independent design and tuning for every module. By contrast, end-to-end learning employs a single unified neural network to replace the entire pipeline. It accepts raw input and directly maps it to the final output through learning. The advantage of this approach lies in the model's ability to automatically discover task-specific features that often outperform human-designed ones [6].

This study aims to design a lightweight DRL framework for path planning in dynamic physiological environments, with the specific goal of validating algorithmic effectiveness through simulation experiments [2]. Traditional planning methods often suffer from poor adaptability, high computational latency, and difficulties in ensuring safety—challenges this work seeks to address. Chapter 1 introduces the medical applications of microrobots in targeted therapy and minimally invasive surgery, identifies the challenges of path planning under dynamic and complex conditions, explains the advantages of DRL, and outlines the research objectives and structure of the thesis. Chapter 2 reviews the development of path planning for medical microrobots, analyzing traditional algorithms, learning-based approaches, and DRL applications, summarizing their limitations and identifying future research directions. Chapter 3 presents the fundamentals of reinforcement learning and the necessity of integrating deep neural networks, proposes an MDP-based modeling approach for medical path planning, and adopts Deep Deterministic Policy Gradient (DDPG) as the core algorithm with a CNN–MLP hybrid structure. Chapter 4 describes the experimental environment, evaluation metrics, and key hyperparameters, and demonstrates the superior performance of the proposed method in terms of success rate, efficiency, and safety. Finally, Chapter 5 concludes the

DOI: 10.54254/2755-2721/2026.LD38752

study, highlighting the effectiveness of DDPG for medical path planning and the value of the lightweight framework, while discussing limitations and future research directions in transfer learning, imitation learning, and multi-objective path planning.

2. Related work

As a key technology for achieving precision medicine, medical microrobots have become an increasingly prominent research topic in recent years. In terms of path planning, existing methods can generally be divided into two main categories. The first category comprises traditional model-based methods, such as potential field methods, genetic algorithms, and bio-inspired algorithms [7]. These approaches typically rely on precise environmental modeling and offer interpretability and controllability. However, their performance in dynamic and complex environments is limited due to their dependence on fixed models. The second category includes early learning-based methods, such as Q-learning and SARSA [8]. These methods iteratively update their policies through continuous interaction with the environment, thereby overcoming some modeling constraints. However, they still face difficulties in handling high-dimensional state spaces and achieving fast convergence. Consequently, their applicability in complex medical scenarios remains unsatisfactory. Therefore, developing a new path planning approach that can address these limitations has become essential.

With the integration of deep learning and reinforcement learning, Deep Reinforcement Learning (DRL) has demonstrated tremendous potential in path planning tasks. Value-based methods, such as DQN and its improved variants (Double DQN, Dueling DQN) [9], can effectively approximate optimal policies in high-dimensional state spaces. Meanwhile, policy-gradient-based methods, such as REINFORCE, A2C, and PPO [6], have shown greater stability and efficiency in continuous action spaces and complex environments. In recent years, several studies have introduced DRL into medical contexts. For example, DRL has achieved promising results in vascular microrobot navigation and tumor-targeted path planning [1]. These studies confirm DRL's advantages in addressing nonlinearity, dynamic constraints, and high uncertainty, proving its potential for handling complex real-world medical environments.

Despite these advances, DRL-based medical microrobot path planning still faces several limitations. First, deep models typically involve high computational complexity, making them difficult to deploy on resource-constrained microrobotic platforms [2]. Second, a significant simulation-to-reality gap exists between virtual environments and actual medical conditions, often resulting in decreased model performance in real applications [10]. Third, sparse reward problems persist, leading to slow convergence and limited generalization capabilities in complex tasks [11]. Thus, the key challenge is to enhance model robustness and transferability while maintaining computational efficiency.

In summary, research on medical microrobot path planning has evolved from traditional algorithmic methods to learning-based approaches and, more recently, to DRL-based techniques. While DRL demonstrates unique strengths in managing complex and dynamic constraints, current research still requires breakthroughs in algorithm lightweighting, simulation-to-reality transfer learning, and reward function design [5]. Addressing these challenges will guide future research directions and lay the groundwork for the practical implementation of medical microrobots.

DOI: 10.54254/2755-2721/2026.LD38752

3. Fundamentals of reinforcement learning models and system design

3.1. Core concepts of reinforcement learning

The core principle of Reinforcement Learning (RL) is that an agent learns an optimal policy through continuous interaction with its environment to maximize long-term cumulative rewards [12]. The environment represents the external world in which the agent operates; in medical robot path planning, this corresponds to vascular networks, biological tissues, or surrounding obstacles. A state (S) refers to an abstract description of the environment at a given time, such as the robot's current position, target coordinates, and obstacle distribution. An action (A) is an operation the agent can take in the current state—such as changes in movement direction, speed, or rotation angle. A reward (R) is the feedback signal from the environment to the agent's action, reflecting its contribution toward the task goal—for example, receiving a positive reward when reaching the target point and a negative reward when colliding with obstacles.

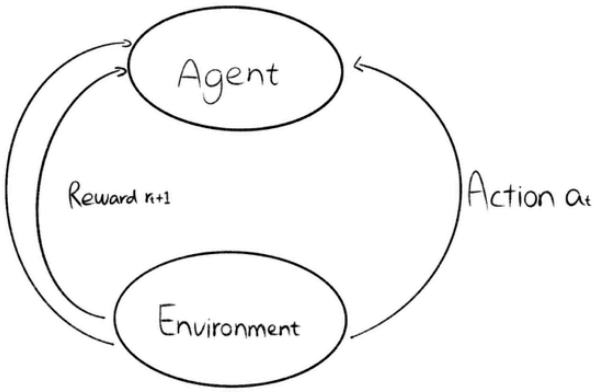

Figure 1. Basic framework of reinforcement learning

As illustrated in Figure 1, the agent selects an action a_t based on the current environmental state s_t. The environment then provides the next state s_{t+1} and an immediate reward r_{t+1}, forming a closed-loop learning system. The Markov Decision Process (MDP) provides a mathematical framework for this process [13]. An MDP consists of a set of states, a set of actions, a state transition probability function, and a reward function, collectively describing the decision-making process. Figure 1 shows a typical Markov state transition example, where the probability values along arrows represent the likelihood of transitions between states, embodying the Markov property.

Based on this, the Markov Reward Process (MRP) further introduces reward signals into state transitions. Through the value function, the long-term expected return of a given policy can be computed. Another key concept in reinforcement learning is the trade-off between exploration and exploitation [14]. Exploration encourages the agent to try new actions to discover potentially better policies, whereas exploitation selects the current best-known action based on experience. A proper balance between the two prevents the algorithm from becoming trapped in local optima, which is crucial for achieving efficient convergence in complex environments.

DOI: 10.54254/2755-2721/2026.LD38752

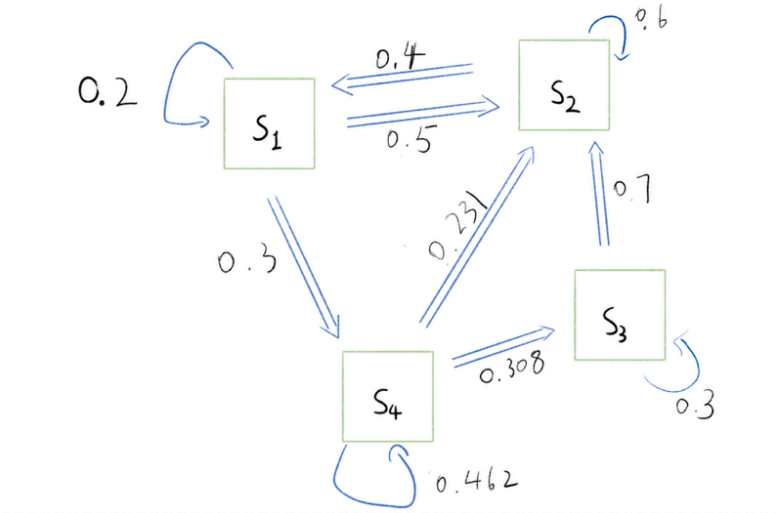

Figure 2. Markov state transition diagram

As shown in Figure 2, the model establishes a Markov chain containing four states (S1–S4). All state transition probabilities are normalized to ensure that the total probability of transitions from any state equals one. State S1 has a 20% self-loop probability, a 50% probability of transitioning to S2, and a 30% probability of transitioning to S4. State S2 transitions to S1 with a probability of 40% and maintains a 60% self-loop probability. State S3 transitions to S2 with a probability of 70% and has a 30% self-loop probability. State S4's total transition probability is originally 1.3; after normalization, it transitions to S2, S3, and itself with approximately 23.1%, 30.8%, and 46.2% probabilities, respectively. This normalization ensures compliance with Markov chain mathematical properties while preserving the original relative transition ratios.

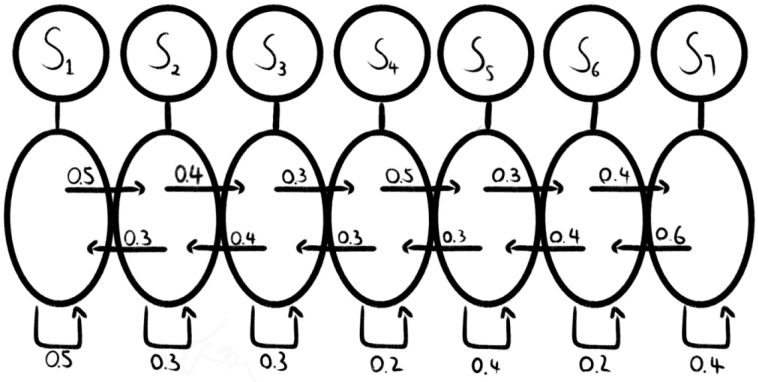

Figure 3. Extended Markov chain

Figure 3 extends the previous model to include seven states (S1–S7), introducing additional transition pathways to better simulate the dynamic evolution of complex environments. Bidirectional transitions between S1 and S2 each occur with a 0.5 probability, while states such as S3–S5 adopt different transition ratios to represent the microrobot's diverse motion behaviors in vascular environments. This expanded model captures more complex path-planning dynamics and serves as a higher-dimensional foundation for subsequent training and experimentation [15].

DOI: 10.54254/2755-2721/2026.LD38752

The Markov Reward Process (MRP) incorporates reward signals into state transitions, enabling the computation of the long-term expected return under a given policy through the value function, thus providing a more systematic mathematical framework for reinforcement learning [16].

3.2. The role of deep neural networks in RL

In traditional Q-learning frameworks, the state–action value function is typically represented as a Q-table. However, in high-dimensional state spaces, this representation leads to the curse of dimensionality, rendering training infeasible. To address this, deep neural networks (DNNs) are introduced into reinforcement learning to approximate Q-functions or policy functions, thereby effectively mitigating the explosion of state space [9]. Among such methods, Actor–Critic frameworks have emerged as the dominant solution for complex tasks. The Actor network generates specific actions based on the current state, while the Critic network evaluates these actions' long-term values. To enhance stability, modern DRL models commonly employ experience replay and target networks during training. The Deep Deterministic Policy Gradient (DDPG) algorithm is a representative example that combines value functions with policy gradients to achieve efficient learning in continuous action spaces [17]. The Actor network outputs deterministic actions directly, avoiding the inefficiency of extensive sampling, while the Critic network evaluates these actions' long-term returns to guide the Actor's updates. Experience replay breaks temporal correlations between data, enhancing generalization, while soft target network updates ensure training stability and prevent numerical oscillation.

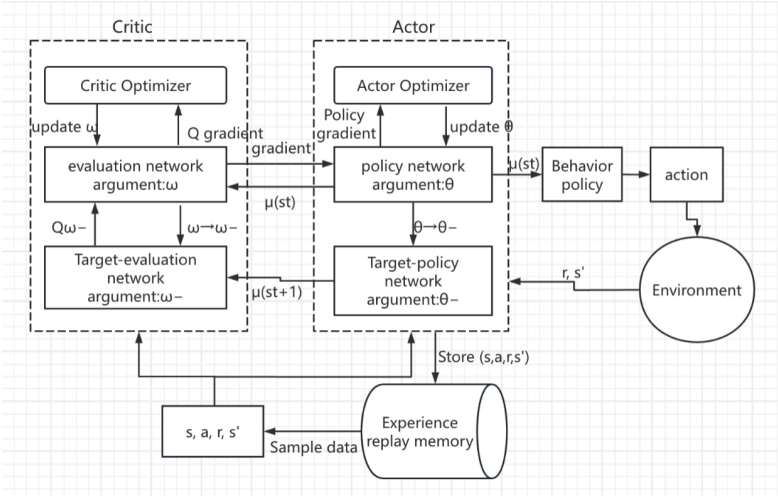

Figure 4. Structure of the DDPG algorithm

As shown in Figure 4, the DDPG algorithm consists of two main components: the Actor network and the Critic network. The Actor module contains a policy network parameterized by θ, which receives the environmental state s_t and outputs a deterministic action $\mu(s_t)$, Using this behavior policy, the agent executes actions in the environment to obtain feedback, including reward r and next state s'. The Critic module includes an evaluation network, parameterized by ω, that computes the action-value function $Q(s,a)$. A target evaluation network (ω^-) is used to stabilize learning by comparing target and current estimates, updating parameters to improve prediction accuracy. Similarly, the Actor module includes a target policy network (θ^-),that outputs the corresponding next-state action $\mu(s_{t+1})$ and works with the target Critic to ensure stable training. Interaction data tuples (s,a,r,s′) are stored in an experience replay buffer, which samples mini-batches for training to

DOI: 10.54254/2755-2721/2026.LD38752

break correlations and enhance learning robustness [18]. During training, the Critic network is updated via Q-value gradients, while the Actor network is updated using policy gradients obtained from the Critic's evaluations—forming a joint optimization loop. Through this process, DDPG achieves efficient policy learning in continuous action spaces with guaranteed stability and convergence.

3.3. Loss functions

In medical robot path planning, the problem can naturally be modeled as an MDP. The state space includes the robot's current position, target location, and obstacle information, typically represented through sensors or grid maps. The action space may be discrete (e.g., up, down, left, right, stop) or continuous (e.g., velocity or angular control). The reward function should balance effectiveness and safety—for instance, high rewards for reaching the target, positive feedback for avoiding obstacles, and negative rewards for collisions or deviations. A well-designed reward function can effectively guide the agent toward optimal path-planning strategies in complex environments [19]. Training DDPG involves updating both the Critic and Actor networks.

3.3.1. Critic loss function

The Critic is updated similarly to DQN, with the objective of minimizing mean squared error (MSE) loss.

The target Q-value y_t is computed using the target Actor and Critic networks:

$$y_t = R_{t+1} + \gamma Q_{target}(s_{t+1}, \mu_{target}(s_{t+1}; \theta_\theta^{target}); \theta_Q^{target}) \tag{1}$$

μ_{target} and Q_{target} refer to the Target Actor Network and the Target Critic Network, respectively.

The Critic loss function L(θ^Q) is defined as:

$$L(\theta^Q) = \frac{1}{N} \sum_{i=1}^{N} (y_i - Q(s_i, a_i; \theta^Q))^2 \tag{2}$$

where (s_i, a_i, R_{i+1}, s_{i+1}) are samples drawn from the replay buffer.

3.3.2. Actor loss function

The Actor aims to learn a policy that maximizes the Q-value estimated by the Critic network.

This objective can be expressed as minimizing the following loss function:

$$L(\theta^\mu) = -\frac{1}{N} \sum_{i=1}^{N} Q(s_i, \mu(s_i^\theta); \theta^Q) \tag{3}$$

The gradient of this loss, representing the policy gradient, can be written as:

$$\nabla_\theta J_\theta = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta Q(s, a; \theta^\theta) \Big|_{s=s_i, a=\mu(s_i)} \nabla_\theta \mu(s; \theta^\mu) \Big|_{s=s_i} \tag{4}$$

This formulation shows that the update direction of Actor parameters depends on both the Critic's gradient evaluation of actions(∇_a Q) and the product of the gradient of the action itself with respect to the parameters($\nabla_{\theta^\mu} \mu$).

DOI: 10.54254/2755-2721/2026.LD38752

3.4. Network architecture design

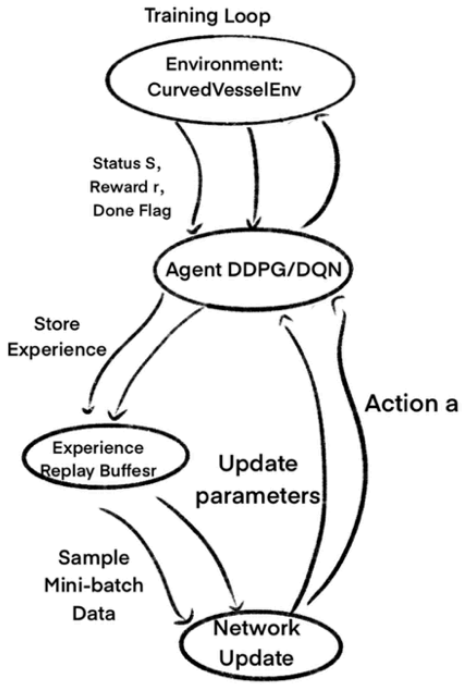

Figure 5. Actor-critic framework for path planning

The proposed architecture integrates visual perception and geometric information in a lightweight, multi-modal Actor–Critic framework. As shown in Figure 5, the overall pipeline includes environment modeling, experience storage, parameter updates, and performance evaluation. The input comprises an 84×84 image (capturing vessel morphology and obstacles) and coordinate vectors (e.g., robot pose and target). A Convolutional Neural Network (CNN) extracts spatial features from images, while a Multilayer Perceptron (MLP) processes numerical states. The Actor outputs continuous control commands, and the Critic estimates the corresponding state–action value. This multi-modal fusion enhances perception and decision robustness while maintaining a lightweight profile conducive to deployment in constrained robotic platforms [20].

4. Experimental design and result analysis

4.1. Experimental setup and environment configuration

All experiments in this study were conducted on an Apple MacBook Pro equipped with an M2 Pro chip, 16 GB of unified memory, and the macOS operating system, ensuring computational efficiency and stability for small- to medium-scale deep reinforcement learning tasks [21]. The experimental environment was built upon a custom CurvedVesselEnv simulation, which emulates the movement of medical microrobots within complex vascular channels and integrates factors such as obstacle distribution, vessel curvature variation, and target-point configuration to better approximate real-world medical scenarios. The environment outputs the robot's current position, relative distance to the target, and sensed obstacle information, constituting the state space; the action space permits continuous velocity-control operations for flexible path selection. The reward function jointly considers goal-reaching efficiency, path length, and collision penalties, achieving a balance between

DOI: 10.54254/2755-2721/2026.LD38752

safety and effectiveness [22]. The entire framework—including environment modeling, agent interaction, experience replay, and policy updating—provides a reproducible platform for subsequent analyses. During training, the learning rate, discount factor γ, and PPO clip range were adjusted as key hyperparameters [23]. The learning rate controls the magnitude of parameter updates—overly large values cause divergence, while excessively small ones slow convergence. The discount factor balances immediate versus long-term rewards: higher γ values emphasize global optimality, whereas lower ones bias toward local decisions. The clip range, specific to PPO, constrains policy updates to prevent instability.

4.2. Performance comparison and analysis

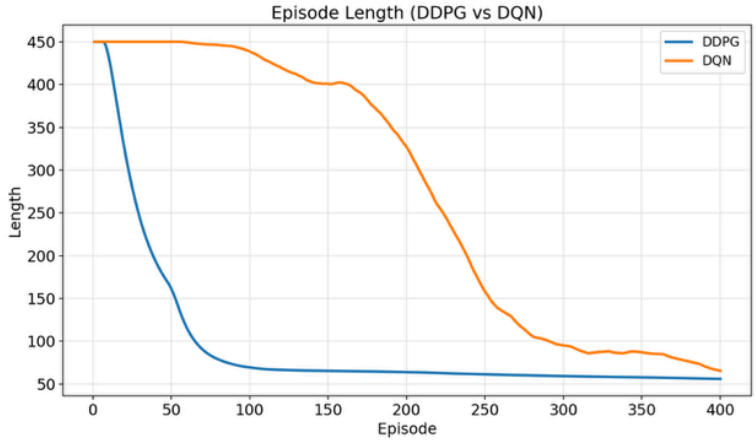

Figure 6. Path length comparison between DDPG and DQN

Figure 6 shows how path length changes with training iterations. Initially, both DDPG and DQN record approximately 450 steps. However, DDPG rapidly drops below 100 steps within 50 episodes and stabilizes after 150 episodes, while DQN requires 300 episodes to reach a comparable level. These results demonstrate DDPG's faster convergence and superior global-search capability, enabling efficient path optimization early in training.

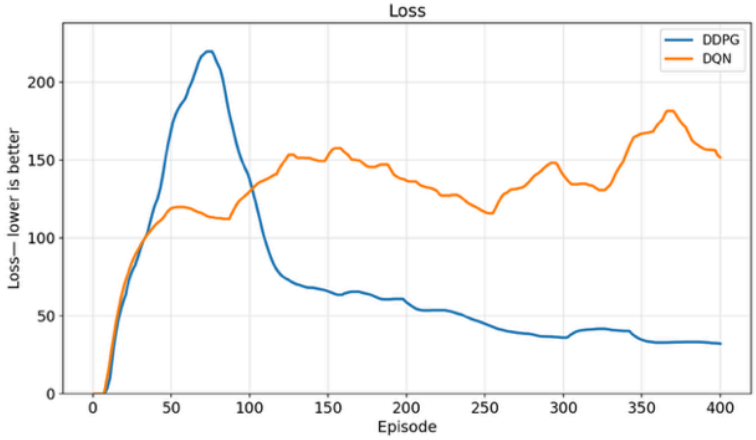

Figure 7. Training loss curve comparison

DOI: 10.54254/2755-2721/2026.LD38752

Figure7 presents the training-loss convergence of both algorithms. The DDPG loss declines sharply in the initial stage and stabilizes after 150 episodes with minor fluctuations, indicating smooth network updates and numerical stability. By contrast, DQN exhibits large oscillations and irregular spikes, revealing overfitting and gradient-instability issues. The results confirm that DDPG's target-network and replay-buffer mechanisms effectively improve learning stability in continuous-action domains.

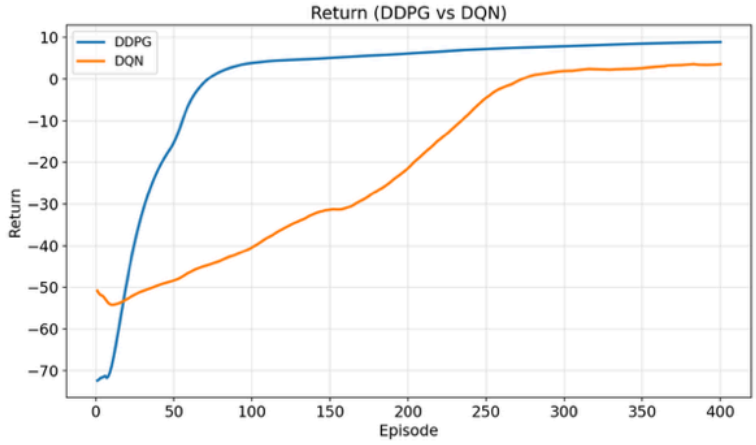

Figure 8. Average return (reward) per episode

Figure 8 depicts the evolution of average episodic return during training. DDPG achieves a steep performance increase after 100 episodes, rapidly approaching its maximum and maintaining stability. DQN shows delayed improvement, only stabilizing after 200 episodes. The smoother DDPG curve indicates higher sample efficiency and the ability to discover high-reward trajectories with fewer random explorations.

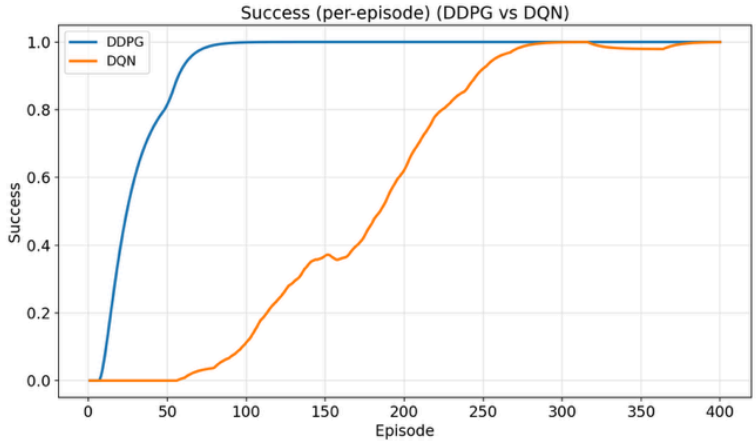

Figure 9. Success rate comparison between DDPG and DQN

Figure 9 compares task-completion rates. DDPG exceeds 90 % success after 120 episodes and reaches nearly 100 % by 200 epochs, whereas DQN achieves similar rates only after 300 episodes and displays greater early-stage fluctuation. Thus, DDPG demonstrates higher stability, faster convergence, and stronger generalization in complex environments.

DOI: 10.54254/2755-2721/2026.LD38752

4.3. Path planning and motion control evaluation

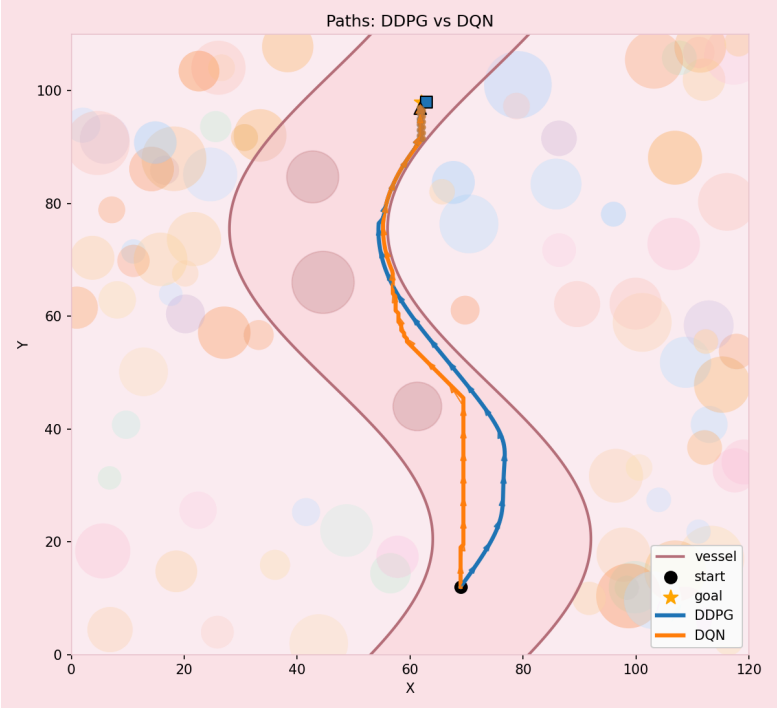

Figure 10. Path visualization of DDPG and DQN in curved vessel environment

The visualization shows that after 200 training episodes, DDPG converges with an average path length of 80–100 units—about 75 % shorter than at initialization. After 100 episodes, its success rate exceeds 95 %, and the average collision count drops below one per episode. Paths are smooth and minimally curved, indicating accurate motion control and effective obstacle avoidance. These outcomes highlight DDPG's stability and efficiency in continuous-control tasks, providing a safe and feasible solution for microrobotic navigation.

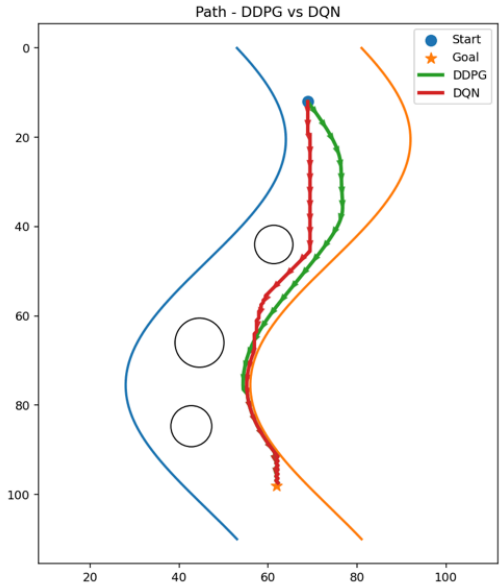

Figure 11. Comparative path analysis of multiple algorithms

DOI: 10.54254/2755-2721/2026.LD38752

When applying DQN to the same vascular environment, convergence occurs only after 400 episodes, with mean path lengths of 150–200 units—roughly 60–80 % longer than DDPG—and a success rate below 80 %. Average collision counts remain around 3–4 per episode, reflecting unstable action selection in continuous spaces. Trajectories show excessive detours and local oscillations, evidencing weak adaptation to vascular geometry. In comparison, DDPG outperforms DQN in convergence speed, path optimality, and stability.

5. Conclusion

This study focuses on the path planning challenges faced by medical microrobots operating in complex vascular environments. Using deep reinforcement learning (DRL) methods, it constructs and validates a lightweight learning framework tailored to these conditions. Earlier chapters provided a theoretical foundation by introducing the core principles of reinforcement learning and explaining the role of deep neural networks in policy learning. Building on this foundation, the study modeled the medical path planning task as a Markov Decision Process (MDP) and implemented experiments using the Deep Deterministic Policy Gradient (DDPG) algorithm as a representative approach. In the experimental stage, the performances of DDPG and Deep Q-Network (DQN) were compared under identical simulated vascular conditions. The results demonstrated that DDPG achieved faster convergence, smoother trajectories, and greater safety in continuous action spaces. These findings not only validate the potential of deep reinforcement learning in the medical microrobotics field but also provide an empirical basis for future improvements.

From the perspective of research contribution, this study demonstrates several areas of innovation and scholarly value. First, at the modeling level, it tightly integrates medical microrobot path planning with a reinforcement learning framework by carefully designing the state space, action space, and reward function. This approach establishes a simulation environment that more closely reflects real-world medical scenarios. Unlike traditional static maps, the environment dynamically responds to changes in vascular structures and obstacle distributions, better capturing the inherent uncertainty of physiological conditions. Second, in terms of algorithmic design, the study focuses on DDPG—a representative method for continuous control problems—and conducts a systematic comparison with DQN. The results highlight the advantages of deterministic policy gradient methods for medical path planning tasks. Third, the evaluation metrics extend beyond simple success rates in reaching targets, incorporating multiple dimensions such as path length, planning time, and collision frequency. This multi-dimensional evaluation offers a more comprehensive assessment of algorithmic performance.

Nevertheless, despite the promising results, several limitations of this research must be acknowledged. First, the experiments were conducted entirely within a simulated environment. Although the CurvedVesselEnv model accounts for vascular curvature and obstacle distribution, it still diverges from real physiological conditions. For instance, the dynamic effects of blood as a non-Newtonian fluid are not fully captured, and the elasticity and contractility of vessel walls are not modeled. Consequently, there may be significant performance degradation when transferring the model from simulation to real-world robotic operations. Second, although DDPG performs effectively in continuous action spaces, its training process remains computationally expensive and sensitive to hyperparameter tuning. Inappropriate parameter selection can lead to slower convergence or entrapment in local optima. Third, regarding safety, while DDPG significantly reduces collision frequency, it does not completely eliminate risk. In clinical contexts, even minor collisions could cause irreversible tissue damage, imposing much stricter safety requirements for real-world applications.

DOI: 10.54254/2755-2721/2026.LD38752

To address these limitations, several directions for future research are proposed. First, at the environmental modeling level, future studies should integrate fluid dynamics simulations to capture the effects of blood flow and vascular elasticity, thereby creating more realistic training scenarios. Combining DRL environments with medical imaging data could enable personalized modeling of specific organs and vascular structures, improving the precision of simulation-based training. Second, algorithmic optimization could incorporate principles of imitation learning and transfer learning, using expert demonstrations or historical data to pre-train agents. This would reduce exploration time and enhance the quality of initial policies. Additionally, Hierarchical Reinforcement Learning (HRL) represents a promising approach by decomposing complex tasks into sub-tasks, thereby improving training efficiency and stability. Third, in terms of model deployment, future research should explore lightweight and efficient neural network architectures capable of real-time decision-making on microrobot hardware platforms with limited computational resources. Achieving this will require not only algorithmic refinement but also hardware-software co-optimization aligned with advances in embedded systems and microelectronics.

Beyond these technical enhancements, future research may also expand in several complementary directions. One is the issue of multi-robot collaboration. Current experiments focus solely on a single robot operating within a vascular environment, whereas real medical applications often involve multiple microrobots cooperating to perform complex tasks—such as joint drug delivery or synchronized lesion localization. Addressing this requires reinforcement learning algorithms capable of handling multi-agent interactions, such as Multi-Agent Reinforcement Learning (MARL). Another promising direction is multi-objective optimization. In real clinical scenarios, microrobots must balance competing objectives such as minimizing path length, ensuring safety, optimizing drug release efficiency, and conserving energy. Developing strategies to balance these objectives effectively remains an open research challenge. Lastly, the issue of clinical interpretability is critical. Although deep reinforcement learning offers strong performance, its "black-box" nature remains a barrier to adoption in the medical domain. Future work should therefore prioritize the development of explainable decision-making mechanisms, enabling clinicians to understand and trust the microrobot's in-body navigation strategies—an essential step toward clinical acceptance.

In summary, this study explores the potential of deep reinforcement learning in medical microrobot path planning and experimentally demonstrates the advantages of the DDPG algorithm in continuous control tasks. The research not only provides a methodological foundation for autonomous decision-making in complex and dynamic environments but also outlines pathways for future innovation. Building on these findings, subsequent work should expand across environmental modeling, algorithmic optimization, model deployment, and clinical application. With the ongoing convergence of artificial intelligence and biomedical science, deep reinforcement learning is poised to play an increasingly vital role in intelligent healthcare. As a critical instrument of precision medicine, medical microrobots will continue to reveal their transformative potential in both research and clinical practice.

References

[1] Xu, T., et al. (2010). A swimming microrobot powered by oscillating magnetic fields. *Applied Physics Letters, 96*(15), 153701. https://doi.org/10.1063/1.3405719

[2] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine, 34*(6), 26–38. https://doi.org/10.1109/MSP.2017.2743240

[3] Mnih, V., et al. (2013). Playing Atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602

DOI: 10.54254/2755-2721/2026.LD38752

[4] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research, 5*(1), 90–98. https://doi.org/10.1177/027836498600500106
[5] Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research, 32*(11), 1238–1274. https://doi.org/10.1177/0278364913495721
[6] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*(3–4), 229–256. https://doi.org/10.1007/BF00992696
[7] Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. MA, USA: Addison-Wesley.
[8] Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*(3–4), 279–292. https://doi.org/10.1007/BF00992698
[9] Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://doi.org/10.1038/nature14236
[10] Zhang, J., et al. (2018). Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5628–5635). https://doi.org/10.1109/ICRA.2018.8460653
[11] Andrychowicz, M., et al. (2017). Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS), 30*, 5048–5058.
[12] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). Cambridge, MA, USA: MIT Press.
[13] Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York, NY, USA: John Wiley & Sons.
[14] Norris, J. R. (1997). *Markov chains*. Cambridge, UK: Cambridge University Press.
[15] Howard, R. A. (1960). *Dynamic programming and Markov processes*. Cambridge, MA, USA: MIT Press.
[16] Bertsekas, D. P. (2019). *Reinforcement learning and optimal control*. Belmont, MA, USA: Athena Scientific.
[17] Lillicrap, T. P., et al. (2016). Continuous control with deep reinforcement learning. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. arXiv:1509.02971
[18] Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)* (pp. 1587–1596).
[19] Silver, D., et al. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning (ICML)* (pp. 387–395).
[20] Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research, 17*(39), 1–40.
[21] Apple Inc. (2023). MacBook Pro with M2 Pro and M2 Max - Technical specifications. *Apple*. https://www.apple.com/macbook-pro-14-and-16/specs/
[22] Brockman, G., et al. (2016). OpenAI Gym. *arXiv preprint arXiv:1606.01540*.
[23] Schulman, J., Levine, S., Moritz, P., Jordan, M., & Abbeel, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (pp. 1889–1897).