Advances in Computer, Signals and Systems (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2025.090117
ISSN 2371-8838 Vol. 9 Num. 1

Map-less Navigation Algorithm for Autonomous Vehicles Based on Deep Reinforcement Learning

Dayu Guo[1,a], Yuan Zhu[1,b], Ke Lu[1,c,*]

[1]*School of Automotive Studies, Tongji University, No. 4800 Caoan Road, Shanghai, China*
[a] *wong-tony@foxmail.com,* [b]*yuan.zhu@tongji.edu.cn,* [c]*luke@tongji.edu.cn*
**Corresponding author*

Keywords: Map-less Navigation, Deep Reinforcement Learning

Abstract: This paper focuses on the map-less navigation problem of autonomous vehicles based on deep reinforcement learning, and proposes a map-less navigation method for autonomous vehicles based on an improved Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. Aiming at the problems of navigation success rate, exploration performance, and training time of existing map-less navigation algorithms based on deep reinforcement learning, the following innovations are used to optimize the above performance: ① Optimize the neural network structure of the TD3 algorithm to enhance the exploration ability of autonomous vehicles in complex environments. ② Construct a composite reward function to integrate dense rewards and sparse rewards, which significantly speeds up the training speed of the algorithm. Finally, the algorithm in this paper only needs 12% of the training amount of the comparison algorithm to achieve the same success rate. A comprehensive test environment and a special test environment were built in a simulation environment for comparative experiments. The results show that the navigation success rate of the algorithm in this paper is increased by 11.80% in the comprehensive test environment; the obstacle avoidance success rate is increased by 40% and 70% in the special test environment, and the exploration success rate is increased by 100%. In the test of real complex environment, the navigation algorithm is not adjusted, and it can effectively drive the autonomous vehicle to perform map-less navigation. The navigation effect and portability of the algorithm are verified.

1. Introduction

The emergence of Deep Reinforcement Learning (DRL) has catalyzed innovative paradigms for autonomous navigation [1-4]. DRL-based systems endow agents with autonomous decision-making faculties through continuous environmental interaction coupled with deep network policy approximation, offering three strategic benefits: 1) Obviates the need for annotated training datasets 2) Facilitates comprehensive edge case discovery through self-supervised exploration surpassing conventional rule-based systems 3) Exhibits superior temporal action sequence learning capabilities compared to traditional machine learning approaches.

Notable DRL implementations include the BADGR framework [5, 6] leveraging monocular visual inputs for autonomous wayfinding, Huang et al.'s Goal-oriented Transformer Architecture (GTA) [7]

enhancing map-less navigation through heterogeneous sensor fusion, and Cimurs' Geometric Deep Autoencoder (GDAE) [8,9] achieving efficient environment modeling via LiDAR point cloud processing with enhanced computational economy and cross-domain adaptability.

The GDAE algorithm can achieve map-less navigation in unknown environments with less input information and a lightweight neural network. However, there is still much room for improvement in the algorithm's navigation success rate and algorithm training speed. Based on this, this paper studies how to use lidar to perceive the surrounding environment information in real time in an unfamiliar environment with complex static and dynamic obstacles, realize end-to-end obstacle avoidance decision control, and improve the navigation success rate of autonomous vehicles and speed up training. The main contributions of this paper are:

(1) In order to solve the problem that autonomous vehicles cannot find suitable driving routes in complex environments due to insufficient exploration, this paper optimizes the neural network architecture of the deep reinforcement learning algorithm, and improves the navigation success rate in complex environments by improving the algorithm's exploration.

(2) Constructing a composite reward function effectively improves the algorithm's training speed. This algorithm only requires 12% of the training volume of the GDAE algorithm to achieve the same navigation success rate.

2. Autonomous Vehicle Navigation Algorithm

The autonomous vehicle map-less navigation method proposed in this paper consists of the following three parts:

(1) Simulation Environment. The simulation environment is constructed in Gazebo, a autonomous vehicle simulator integrated with the Autonomous vehicle Operating System (ROS), which provides high-fidelity physics engine and sensor modeling capabilities. This platform replicates real-world complex unknown scenarios and serves as both a data source and validation benchmark for navigation algorithms.

(2) Reinforcement Learning Components in Navigation. For differential-drive autonomous vehicles, the action space is designed based on their kinematic model, comprising linear and angular velocities. The state space integrates environment states and vehicle states. A hybrid reward function combining sparse and dense rewards is implemented to accelerate training convergence, enhance obstacle avoidance capabilities, and improve trajectory smoothness.

(3) Enhanced TD3 Algorithm. The Twin Delayed Deep Deterministic Policy Gradient (TD3) [10], an actor-critic framework-based DRL algorithm for high-dimensional continuous action spaces, is optimized to address limitations observed in GDAE navigation. Key improvements include: Network Architecture Refinement: Enhanced neural network structures for improved policy stability and information utilization efficiency. Adaptive Noise Attenuation Strategy: Redesigned exploration noise decay mechanism that balances exploration-exploitation trade-offs during training phases.

2.1. Architectural Optimization of TD3 Algorithm

The proposed methodology introduces architectural refinements to the Critic Network within the TD3 framework. The network's input layer configuration processes both environmental state representations (44-dimensional vector) and actuator command signals (2-dimensional vector). While conventional TD3 implementations employ linear concatenation of these heterogeneous feature streams, our analysis reveals significant dimensional disparity between sensory inputs (44D) and control outputs (2D). This dimensional imbalance induces feature dominance phenomena where high-dimensional state representations disproportionately influence network activations, effectively suppressing the discriminative power of lower-dimensional action parameters during feature fusion.

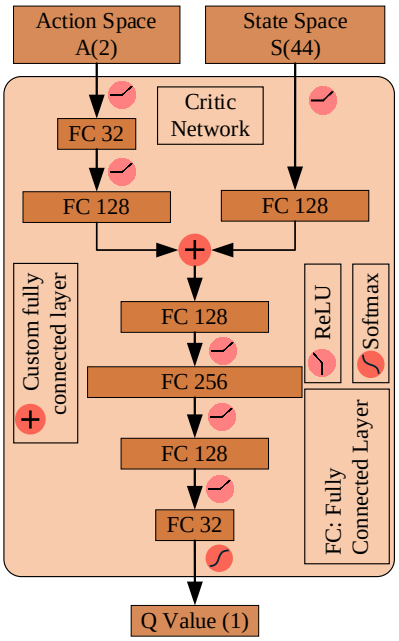


Figure 1: Improved Critic Network.

The proposed architecture addresses this dimensional disparity through optimized feature fusion strategy, as illustrated in Figure 1. Rather than direct vector concatenation, we implement dual-stream dimensional homogenization where both state (44D) and action (2D) vectors undergo feature projection via independent dense layers before integration. We devise a soft feature preservation mechanism within the Critic Network architecture: The state processing stream generates transformed representation $h_s = W_s s$, while the action processing stream produces residual-enhanced encoding $h_a = W_a a + b_a$.

$$h = \sigma(W_s s + W_a a + b_a) \tag{1}$$

The fusion mechanism employs weighted summation of both vectors while preserving only the action branch's bias term b_a. This configuration bears conceptual similarity to residual network principles, where the action branch's bias functions analogously to a residual component. Such design prevents complete overshadowing of action features by high-dimensional state representations during fusion. By maintaining localized enhancement of action features through bias retention, the architecture effectively mitigates information submergence issues where low-dimensional action vectors might otherwise be overwhelmed by state vectors. This strategic partial reinforcement enables enhanced network adaptability for discovering latent environmental-action correlations while maintaining dimensional compatibility between heterogeneous feature spaces.

2.2. State Space Design

The autonomous vehicle generates the action of the next moment based on the state space S, which contains the information of the autonomous vehicle in the environment at the current moment. According to the operating environment and algorithm requirements of the autonomous vehicle, the state space designed in this paper consists of the environment state S_{env} and the autonomous vehicle state $S_{vehiclel}$. Among them, S_{env} is composed of the environment state array. The sensor of the autonomous vehicle is a 360-degree LiDAR. If all the point cloud data are directly flattened as the environment state array, the dimension of the environment state array will be too high, which will

drown out the information of other states. The algorithm in this paper down samples the point cloud, divides the area around the vehicle into 40 intervals, and uses the nearest point cloud distance in each interval to form the environment state S_{env}. Using a 40-dimensional array to represent the environment can take into account the comprehensive expression of environmental information and the lightweight of the environment state array. $S_{vehiclel}$ contains the autonomous vehicle's own information, including the Cartesian distance dis_{goal} between the autonomous vehicle and the target , the relative angle θ_{goal} between the autonomous vehicle and the target , and the linear velocity v_x and angular velocity of the autonomous vehicle ω_z at the current moment .

$$S_{vehicle} = [dis_{goal}, \theta_{goal}, v_x, \omega_z] \tag{2}$$

Finally, the state space S is represented by a 44-dimensional vector concatenated from the environment state vector and the autonomous vehicle state vector.

$$S = [S_{env}, S_{vehicle}] \tag{3}$$

2.3. Action Space Design

The carrier of the algorithm in this paper is the Bingda-RK3566 autonomous vehicle in the simulation. This is an autonomous vehicle with differential steering. The algorithm controls the linear velocity and angular velocity of the autonomous vehicle. Therefore, the action space A is composed of the linear velocity v_x and angular velocity ω_z:

$$A = [v_x, \omega_z] \tag{4}$$

2.4. Reward Function Design

2.4.1. Sparse Reward Function

The episodic incentive mechanism operates exclusively at terminal states, assigning performance evaluations to the autonomous system. Termination conditions comprise three operational scenarios: successful navigation to target coordinates, physical collision detection, and exhaustion of predefined episode duration without task completion. This structural configuration facilitates exploratory behavior modulation while emphasizing temporal abstraction and strategic generalization in policy development. In our framework, distinct incentive values are assigned according to these terminal states through predefined weighting coefficients.

$$r_{sparse} = \begin{cases} 100, \text{ Reach Goal} \\ \text{-}100, \text{ Collide} \\ \text{-}100, \text{ Exhaust the Timesteps} \end{cases} \tag{5}$$

2.4.2. Dense Reward Function

The heading alignment reward mechanism evaluates the angular relationship between the robot and target at every time step. This calculation compares the current heading angle $\theta_{current}$ with the previous angle θ_{pre}. When both angles share directional consistency (identical sign) with reduced angular magnitude ($|\theta_{current}| < |\theta_{pre}|$), the system issues orientation improvement rewards. Conversely, directional inconsistency or angular magnitude increase triggers penalty deductions. This differential

reward structure r_θ serves dual objectives: 1) Promoting rapid directional convergence through immediate angular optimization incentives. 2) Implementing asymmetrical reward magnitudes where positive rewards exceed negative penalties, thereby preventing rotational oscillations while maintaining progressive heading adjustments.

$$r_\theta = \begin{cases} 1.5, \text{ if } \theta_{current} * \theta_{pre} \geq 0 \ \& \ |\theta_{current}| < |\theta_{pre}| \\ -2, \text{ else} \end{cases} \tag{6}$$

The Euclidean distance metric between the robotic agent and its target undergoes evaluation at every temporal iteration. This comparative analysis involves measuring the instantaneous separation distance $d_{current}$ against the preceding measurement d_{pre}. When the current proximity measurement demonstrates spatial advancement ($d_{current} < d_{pre}$), the system assigns progressive proximity incentives. Conversely, any measured regression in navigational progress ($d_{current} \geq d_{pre}$) triggers corresponding penalty deductions. To facilitate accelerated path optimization toward the objective, a target-oriented distance reward component r_{dist} is explicitly formulated through this conditional reward allocation mechanism.

$$r_{dist} = \begin{cases} 1.5, \text{ if } d_{current} \leq d_{pre} \\ -2, \text{ else} \end{cases} \tag{7}$$

$$r_{dense} = r_{dist} + r_\theta \tag{8}$$

The reward function of the algorithm is designed as:

$$\mathrm{R} = r_{sprase} + r_{dense} \tag{9}$$

3. Experiments

3.1. Special Scenario Navigation Experiment

In order to quantitatively analyse the obstacle avoidance and exploration performance of the algorithm, this paper designed three special scenarios. As shown in Figure 2, the left part is three simulation scenarios, and the right part is the corresponding real vehicle experimental scenario.

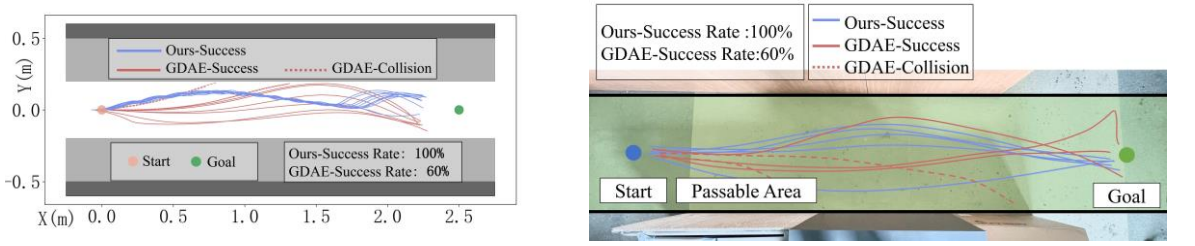
(a) A narrow path with a width of 1 meter. In this scenario, the autonomous vehicle does not need to make complex decisions. It only needs to drive forward and avoid obstacles on both sides in time when the vehicle body is close to the obstacles on both sides, and keep driving in the center. Therefore, scenario (a) avoids the influence of decision-making and only tests the obstacle avoidance performance of the algorithm.

(b) A narrow gate with a width of 1 meter. The fault tolerance space of the vehicle body when passing through the narrow gate is 20 cm. Unlike scenario (a), scenario (b) has no walls on both sides as constraints before passing through the narrow gate. It is necessary to adjust the direction in time in front of the door to pass through, which is more difficult. In scenario (b), the autonomous vehicle can always perceive the passable area in front, so this environment excludes the influence of road decisions and only tests the obstacle avoidance performance of the algorithm.

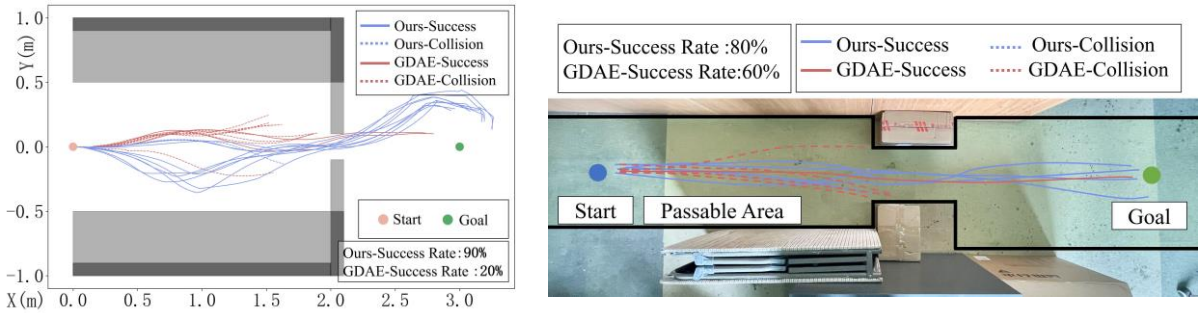
(c) A scenario of bypassing obstacles to test the exploration of the algorithm. . The obstacle in front of the autonomous vehicle is 4 meters long. At the starting point, there are no obvious obstacles farther away in the environment state array, that is, the autonomous vehicle cannot sense that both sides are passable at the starting point. In this scenario, there are no obstacles very close to the

autonomous vehicle, eliminating the influence of obstacle avoidance and only testing the exploratory nature of the algorithm.

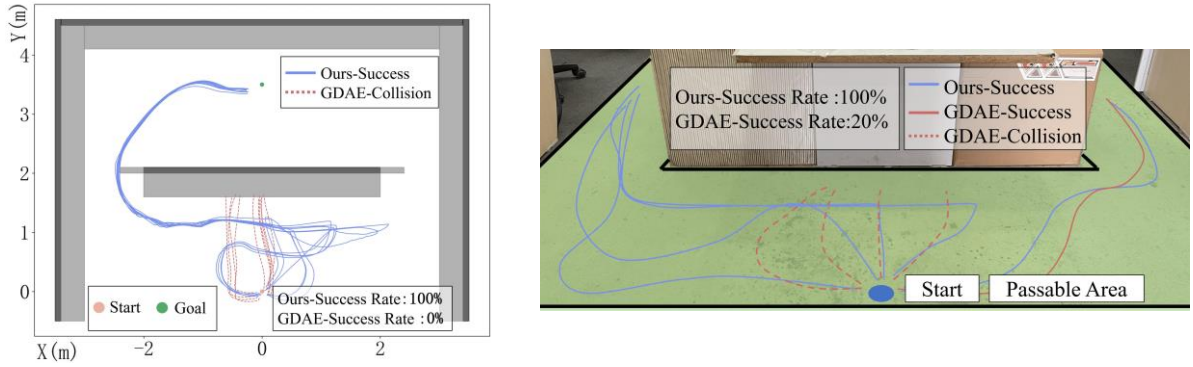
The proposed algorithm and GDAE algorithm were simulated and tested in the above three scenarios, and each scenario was simulated for 10 rounds. In the experiment of scenario (a), the autonomous vehicle using the proposed algorithm will turn earlier when approaching the wall, keep driving in the middle, and no collision occurs. However, the GDAE algorithm does not avoid in time or even does not avoid in time. In the experiment of scenario (b), although the first half of the trajectory of the autonomous vehicle using the proposed algorithm is not facing the narrow gate, it can turn in time before passing through the narrow gate, so that the vehicle body is in the middle of the channel and passes smoothly. Most of the trajectories of the GDAE algorithm did not turn in time, and only 2 trajectories passed through the narrow gate. In scenario (c), no matter where the autonomous vehicle using the proposed algorithm is facing in the initial direction, it will first drive to the front of the wall, then drive along the wall at a slower speed, and after finding a passable gap, it will pass through the gap with the optimal path and drive to the end. The algorithm always walks along the left side of the wall before passing through the obstacle, which may be because more similar paths were collected and learned during training. The GDAE algorithm can only give a local optimal strategy in this scenario. The autonomous vehicle first drives to the wall, then does not move or sways in place, and does not conduct effective exploration. In the process, it runs out of time or collides, resulting in navigation failure.



Scenario (a) The narrow path.



Scenario (b) The narrow gate.



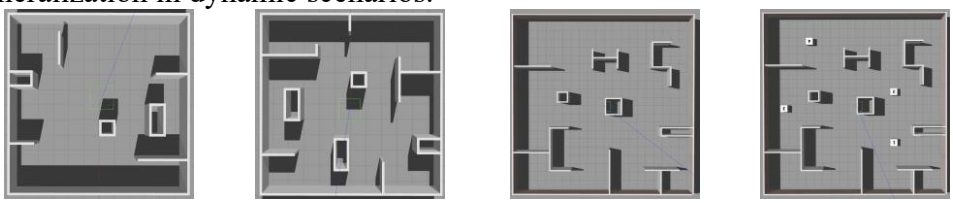
Scenario (c) The detour scenario.

Figure 2: Special scenario simulation and experimental environment.

Experiments in special scenarios have proved that the algorithm in this paper effectively improves the obstacle avoidance performance of the algorithm through the composite reward function, and effectively improves the exploration performance of the algorithm by optimizing the neural network structure.

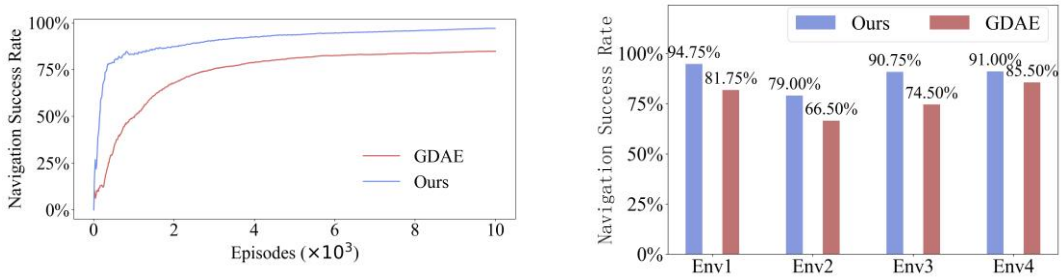
3.2. Special Scenario Navigation Experiment

In order to test the algorithm's training speed and navigation performance in complex environments, four different simulation environments were designed as shown in Figure 3: (a) ENV1: 10m×10m area containing static obstacles (21% coverage) for algorithm training. (b) ENV2: Enhanced version of ENV1 with increased static obstacles (27% coverage) for higher scene complexity. (c) ENV3: Expanded 20m×20m workspace with reconfigured static obstacles (23% coverage). (d) ENV4: ENV3 baseline with 4 random dynamic obstacles (24% coverage) to evaluate algorithm generalization in dynamic scenarios.



(a) ENV1. (b) ENV2. (c) ENV3. (d) ENV4.

Figure 3: Four simulation environments with complex obstacles.



(a) Navigation success rate in training. (b) Navigation success rate in testing.

Figure 4: Navigation success rate in training and testing.

As shown in Figure 4(a), after 10,000 rounds of training, the navigation success rate of the proposed algorithm is 97.12%, while the navigation success rate of the GDAE algorithm is 84.78%. The proposed algorithm can achieve a success rate of 84.58% after 1,200 rounds. The main reasons for the fast algorithm training speed and high success rate are: the reward function provides correct guidance for the navigation strategy of the autonomous vehicle. The optimization of the neural network structure effectively improves the exploration performance.

The proposed algorithm and the GDAE algorithm were tested in the four environments in Figure 3. 400 consecutive navigation experiments were performed in each environment, and the navigation success rate was recorded in Figure 4(b). The navigation success rate is an important criterion for evaluating navigation strategies and is a comprehensive reflection of the obstacle avoidance and exploratory nature of the algorithm. As shown in Figure 4(b), the navigation success rate of the proposed algorithm has been significantly improved in all four environments. In the experiment, if the starting point and the end point are both in a relatively open area, both algorithms can successfully navigate. If the starting point or the end point is in an obstacle-dense area at the edge, the success rate of the GDAE algorithm will drop significantly. Therefore, the proposed algorithm effectively improves the success rate in the comprehensive environment by improving the navigation success

rate in the obstacle-dense area.

4. Conclusions

In this paper, a map-less navigation method for autonomous vehicles based on deep reinforcement learning is proposed to solve the problems of low navigation success rate, insufficient exploration performance and slow training speed of map-less navigation algorithms in complex environments. Through theoretical analysis and experimental verification, the following conclusions are drawn:

(1) This paper constructs a composite reward function, combines the advantages of dense rewards and sparse rewards, and guides the convergence of acceleration strategies through multi-dimensional guidance. Experimental results show that the algorithm in this paper only needs 12% of the training amount of the comparison algorithm to achieve the same performance. And by optimizing the network structure of the TD3 algorithm, the exploration ability of the algorithm in complex environments is improved.

(2) The navigation performance of the algorithm in comprehensive navigation environment and special environment is compared and tested in simulation, and the transferability and robustness of the algorithm are verified through real vehicle experiments. Without adjusting the navigation algorithm, the real vehicle achieves efficient navigation in obstacle avoidance test environment and exploratory test environment.

The current algorithm still has the problem of large fluctuations in the trajectory of the autonomous vehicle and a large room for improvement in trajectory smoothness. Future research can further explore multi-modal sensor information fusion strategies, such as acceleration, vision and other information, to further improve the trajectory smoothness of the navigation algorithm.

References

[1] CHAI R, NIU H, CARRASCO J, et al. Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(4): 5778-5792.

[2] CAO X, REN L, SUN C. Research on obstacle detection and avoidance of autonomous underwater vehicle based on forward-looking sonar[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(11): 9198-9208.

[3] WANG H, HAO J, WU W, et al. A New AGV Path Planning Method Based On PPO Algorithm[C]. 2023 42nd Chinese Control Conference (CCC), 2023: 3760-3765.

[4] WANG R, XU L. Application of Deep Reinforcement Learning in UAVs: A Review[C]. 2022 34th Chinese Control and Decision Conference (CCDC), 2022: 4096-4103.

[5] KAHN G, ABBEEL P, LEVINE S. Badgr: An autonomous self-supervised learning-based navigation system[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 1312-1319.

[6] SHAH D, SRIDHAR A, BHORKAR A, et al. Gnm: A general navigation model to drive any robot[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 7226-7233.

[7] HUANG W, ZHOU Y, HE X, et al. Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 25(2): 1832-1845.

[8] CIMURS R, SUH I H, LEE J H. Goal-driven autonomous exploration through deep reinforcement learning[J]. IEEE Robotics and Automation Letters, 2021, 7(2): 730-737.

[9] CIMURS R, LEE J H, SUH I H. Goal-oriented obstacle avoidance with deep reinforcement learning in continuous action space[J]. Electronics, 2020, 9(3): 411.

[10] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]// International conference on machine learning. PMLR, 2018: 1587-1596.