ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Real-Time Path Planning for Unmanned Vehicles in Dynamic Environments via Improved Deep Reinforcement Learning

Zhen Long[1,*] *and Shihao Lei*[1]

[1] School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, 212013, Jiangsu, China

SUMMARY: *Aiming at the problems faced by unmanned vehicles in dynamic environment, such as the uncertainty of obstacle movement, the limited time delay of online re planning, and the difficulty of taking into account the safety and smoothness of trajectory, this paper proposes a real-time path planning method based on improved deep reinforcement learning. Based on the local dynamic occupation representation, the method integrates risk information such as relative distance, relative speed, collision time and obstacle density, and introduces attention weighting to enhance the perception of key targets; At the decision-making level, improved SAC and priority experience playback are used to improve the stability of continuous motion learning; In the execution layer, the safety action correction and MPC smooth optimization are combined to improve the trajectory executability. Four kinds of dynamic scenes were constructed based on carla-ros joint environment, and compared with A*, DWA, PPO and original SAC. The results show that the proposed method has better comprehensive performance in terms of success rate, collision rate, path length, minimum safe distance and online planning delay, while maintaining good robustness and generalization ability in the absence of speed, higher obstacle density and noisy conditions.*

KEYWORDS: *Deep reinforcement learning; autonomous vehicle; dynamic environment; real-time path planning; safety constraint*

1 Introduction

The path planning of unmanned vehicle is changing from the accessibility calculation in the static map to the online decision-making and real-time control in the dynamic environment. For application scenarios such as park inspection, warehousing and logistics, low-speed distribution and open road testing, the environment of vehicles is not stable, and pedestrian crossing, temporary parked vehicles, interactive traffic flow and local occlusion will continue to change the passable space. Therefore, the planning module not only needs to give a path to reach the target point, but also needs to maintain a fast response speed, a large enough safety margin and a stable trajectory shape that can be tracked by the underlying controller under the condition of frequent perceptual updates and continuous changes in obstacle status. Previous studies have shown that deep reinforcement learning can learn the strategic relationship between obstacle avoidance and reaching goals through environmental interaction, and show strong adaptability in dynamic environmental path planning [1]; At the same time, reward shaping has a positive effect on improving the convergence speed and behavior quality of strategies, which also shows that path planning in dynamic scenes is not only a search problem, but also involves the

*2232307007@stmail.ujs.edu.cn
https://doi.org/10.65102/is2026812
Published: 30 April, 2026

coupling between state expression, reward design and control constraints [2].

The key to the difficulty of real-time path planning in dynamic environment is that three kinds of constraints exist at the same time. At the environmental level, the position, speed and direction of movement of obstacles change continuously, and the vehicle must adjust the local trajectory in time according to the short-term prediction results. At the computational level, the planning delay will directly compress the security decision window, making the problem of "correct planning but lagging implementation" more prominent in high-speed update scenarios. At the control level, safety, smoothness and efficiency tend to restrain each other. Excessive conservatism will lengthen the travel time, excessive aggressiveness will increase the collision risk, and excessive changes in path curvature and heading will weaken the actual traceability. In recent years, the research on automatic driving has been expanded from single vehicle steady-state control to lane changing, Lane merging and intensive interaction scenarios, which shows that path planning is no longer faced with simple geometric obstacle avoidance, but a continuous decision-making problem involving risk judgment and behavior coordination [3]. When the observation is affected by noise, occlusion and local distortion, the insufficient robustness of the strategy will further amplify the misjudgment and hysteresis, which is also one of the important sources of planning system failure in dynamic scenes [4].

From the existing methods, traditional global planning algorithms such as A*, Dijkstra and RRT * have mature accessibility and search performance in static environment, but they usually rely on relatively stable environment expression. Once dynamic obstacles frequently disturb the original path, they need to search repeatedly, and the overhead of line re planning is large. Local planning methods such as DWA and APF have certain real-time performance, but they are easy to produce oscillation, excessive detour or local optimization in high-density obstacles, narrow channels and complex traffic situations. In contrast, deep reinforcement learning can directly learn strategy mapping between perceptual input and control output, and has shown potential in tasks such as lane keeping, overtaking and collision avoidance [5]. The problem is that if the pure DRL method is directly used for real-time path planning in dynamic environment, it is often subject to the defects of unstable training, sensitive reward, difficult explicit expression of security constraints, and not smooth output action. In the absence of risk information modeling and trajectory feasibility correction, although the strategy may learn to reach the goal in simulation, it may not be able to maintain security and stability in complex dynamic scenes.

Recent research progress also shows that simply comparing the benefits of different reinforcement learning algorithms is not enough to support the in-depth discussion of engineering problems. Collision prediction and risk assessment are gradually incorporated into the control and decision-making process, indicating that path planning needs to identify potential conflicts earlier, rather than passively avoid when the danger is approaching [6]. The introduction of model information and additional priors for continuous action decision-making can improve the efficiency of policy update and the quality of action, which shows that real-time planning tasks put forward higher requirements for the algorithm structure itself [7]. In mixed traffic flow, there is an obvious coupling between the vertical and horizontal behavior of vehicles. The speed change, yield relationship and local passable area jointly affect the optimal trajectory. It is difficult to adapt to complex interactive scenes by relying solely on the current position and target direction for local optimization [8]. The relevant research review also pointed out that the main bottleneck of reinforcement learning in autonomous driving behavior planning has changed from "whether we can learn strategies" to "whether the state contains key risk information, whether the rewards reflect multi-objective conflict, and whether the constraints are sufficient to inhibit dangerous actions" [9]. In the dense parallel and strong interactive environment, the reason why safety reinforcement learning attracts more attention

is precisely because the task evaluation standard has expanded from reaching the goal to completing the task with acceptable risk [10].

Based on the above issues, this article proposes a planning framework that integrates dynamic obstacle prediction, risk perception state modeling, and improved deep reinforcement learning for real-time path planning in unmanned vehicle dynamic environments. This method incorporates the relative position, relative velocity, heading change trend, and collision risk indicators of obstacles into the state representation. By improving the strategy learning, the online decision-making ability in the continuous action space is enhanced, and constraint correction and trajectory smoothing mechanisms are added after the strategy output to reduce the impact of dangerous actions and trajectory jitter on operational stability. The focus of this article is reflected in three aspects: constructing a risk perception state modeling method for dynamic environments, enabling strategies to identify potential conflict areas in advance; Design a composite reward function that includes safe distance, heading deviation, path smoothing, time cost, and target advancement terms, and combine constraint correction mechanisms to enhance the safety and feasibility of the strategy; Conduct comparative experiments with traditional planning methods and standard DRL methods in multiple dynamic scenarios, and verify the effectiveness of the proposed method from dimensions such as success rate, collision rate, planning time, and trajectory quality.

2 Methods

2.1 Problem Formulation and System Architecture

This article defines real-time path planning for unmanned vehicles as a continuous decision-making problem in a two-dimensional dynamic environment.The pose and velocity of the unmanned vehicle at time t in a plane coordinate system are $q_t = [x_t, y_t, \psi_t, v_t]^\top$, where x_t and y_t represent position, ψ_t represents heading angle, and v_t represents longitudinal velocity. There are N_t dynamic obstacles in the environment, whose states are denoted as $o_t^i = [x_t^i, y_t^i, v_{x,t}^i, v_{y,t}^i]^\top$. Considering that occlusion, intersection, and temporary intrusion targets can alter local passable areas, planners cannot solely generate paths based on static maps, but also need to jointly characterize the relative distance, relative velocity, and potential conflict time of obstacles. This type of processing method is consistent with the planning approach of autonomous driving under random safety constraints, which maintains an acceptable safety margin through risk boundaries even when locally observable information is incomplete [11]. To this end, this article constructs an occupancy grid Gt within the local map window and extracts risk features such as relative displacement of target points, nearest obstacle distance, and minimum collision time TTC to characterize the spatial state and interaction strength faced by the current decision.

Vehicle kinematics is described using a simplified bicycle model, with its discrete form written as

$$x_{t+1} = x_t + v_t \cos\psi_t \, \Delta t \tag{1a}$$

$$y_{t+1} = y_t + v_t \sin\psi_t \, \Delta t \tag{1b}$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{L} \tan\delta_t \, \Delta t \tag{1c}$$

$$v_{t+1} = v_t + a_t \Delta t \tag{1d}$$

where L is the wheelbase, ψ_t is the front wheel steering angle, and a_t is the longitudinal acceleration. Based on this model, this article defines the reinforcement learning state space as

$$s_t = [x_t, y_t, v_t, \psi_t, \Delta x_t^g, \Delta y_t^g, G_t, d_t, v_t^r, \tau_t^{\min}] \tag{2}$$

where Δx_t^g,Δy_t^g are the relative positions of the target point, d_t is the set of relative distances to obstacles, v_t^r is the set of relative velocities, and $\tau_t^{\min}$ represents the latest collision time. The action space adopts a continuous control form

$$a_t = [\Delta v_t, \Delta \delta_t] \tag{3}$$

Representing speed increment and steering angle increment respectively. The reason for adopting this definition is that continuous actions are more suitable for describing the process of fine-tuning the trajectory of unmanned vehicles, and are also easier to integrate with the strategy update mechanism based on Soft Actor Critic, thereby maintaining decision continuity and local adjustability [12].

The reward function is constructed around three dimensions: approaching the goal, avoiding risks, and maintaining smoothness. Assuming the total return is

$$r_t = w_1 r_t^{prog} + w_2 r_t^{col} + w_3 r_t^{safe} + w_4 r_t^{head} + w_5 r_t^{smooth} + w_6 r_t^{time} \tag{4}$$

where r_t^{prog} represents the goal advancement reward, which is used to encourage vehicles to shorten their distance from the target point; r_t^{col} is a collision penalty, which gives a larger negative value when a collision or boundary crossing occurs; r_t^{safe} is used to constrain the minimum safe distance and continuously increase the penalty when the vehicle approaches the risk boundary; r_t^{head} represents heading deviation penalty, used to suppress ineffective oscillation; r_t^{smooth} is given by the change in actions between adjacent moments to reduce abrupt control; r_t^{time} is a time cost used to avoid policy stagnation in local areas. Considering that autonomous vehicles need to have passability and maintain the rationality of longitudinal speed changes in dynamic scenarios, this reward decomposition method can better balance operational efficiency and driving comfort, and is also in line with the human driving style modeling approach for longitudinal control of autonomous vehicles [14].

The system structure is shown in Figure 1. In Figure 1, the two-dimensional dynamic environment and sensor perception layer include the vehicle state x_t,y_t,ψ_t,v_t, target relative displacement Δx_t^g,Δy_t^g, local occupancy grid G_t, and risk features d_t,v_t^r,$\tau_t^{\min}$, which together form the state vector s_t. The state encoding and improved DRL decision layer inputs the vehicle state, target relative position, risk features, and local environment representation into the policy network, which outputs the original action $a_t = [\Delta v_t, \Delta \delta_t]$. After feasibility screening, safety constraint correction, and MPC smoothing correction, the control instruction is issued. This link corresponds to a closed-loop process of "perception encoding decision correction execution", in which the policy network generates the expected speed and steering adjustment, the safety module constrains and corrects the action based on the minimum safe distance, steering amplitude, and drivable area, and then uses a local smoother or MPC to generate a short-term trackable trajectory, which is then output as control instructions by the underlying controller. Considering the existence of external disturbances and model mismatches in actual operation, it is necessary to retain disturbance suppression capability in the control output stage to reduce the deviation between planning results and execution behavior. This is consistent with the relevant research conclusions of deep reinforcement learning assisted vehicle control [13].

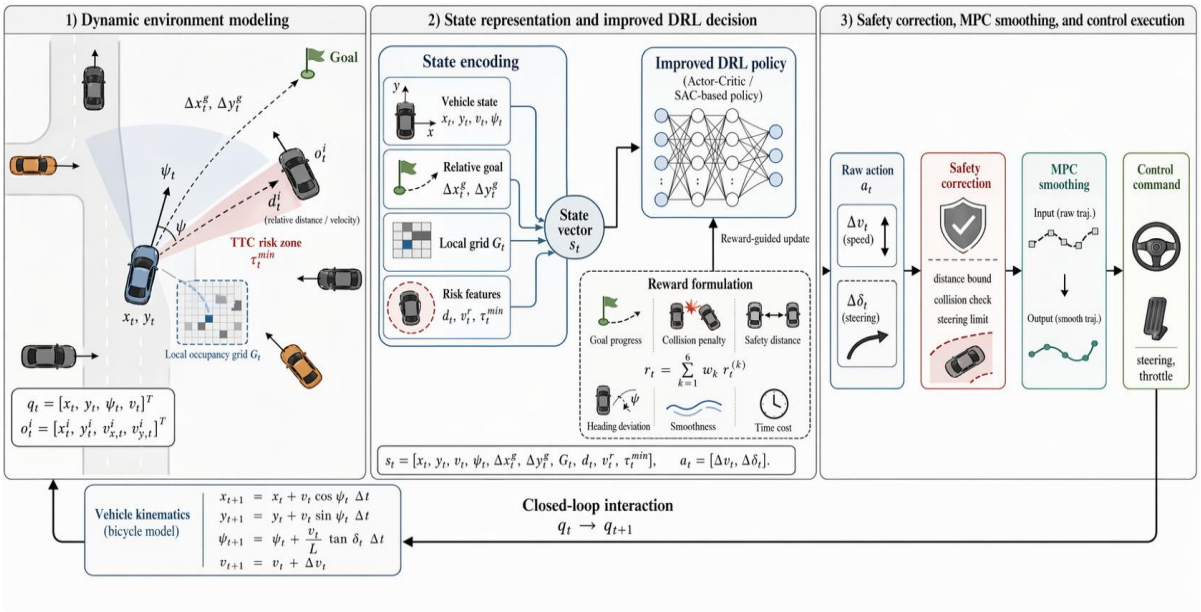

Figure 1: Problem formulation and system architecture for real-time path planning of unmanned vehicles in dynamic environments.

Because the unmanned vehicle working environment has observation noise, shelter mistakes, and abnormal interferences, only depending on original perception to do direct decision making can easily cause strategies to have too much sensitivity to local wrong information. For the alleviation of this problem, this present paper brings in local grid compression expression, the smallest TTC danger item, and obstacle movement tendency information into the construction of state, and restrains the spread of unsafe behaviors through constraint rectification in following decision-making. Therefore, the system can keep a comparatively stable reply even when perception deviation exists. This method that gives first priority to robustness in the state design and decision chain therefore helps to promote the reliability of the strategy in complex dynamic environments [15]. Hence, the problem pattern and system frame which are defined in this part give a united base for the later promotion of deep reinforcement learning algorithms, constraint correcting mechanisms, and experiment analysis..

2.2 VR Scenario Design and Immersive Instructional Intervention

To improve the planning quality of autonomous vehicles in dynamic environments, this paper improves the basic reinforcement learning framework from four aspects: state representation, strategy learning, safety correction, and trajectory smoothing. The overall method chain is shown in Figure 2, and the functional division of each improved module in the system is shown in Table 1. This method first extracts risk related features in the local environment, then generates continuous actions through improved SAC, and then screens out high-risk control variables through safety constraints. Finally, MPC is used to smooth and optimize the short-term trajectory, thus balancing obstacle avoidance efficiency, action stability, and execution feasibility [16].

Table 1: Core modules of the improved DRL-based real-time path planning method

Module	Core design	Main effect
State encoding	Dynamic grid + risk features + attention	Higher scene awareness
SAC policy	Twin-critic continuous control	More stable action output
Replay strategy	Priority-based sampling	Faster policy update
Safety layer	Action correction under risk constraints	Lower collision rate
MPC layer	Local trajectory smoothing	Better continuity and execution quality

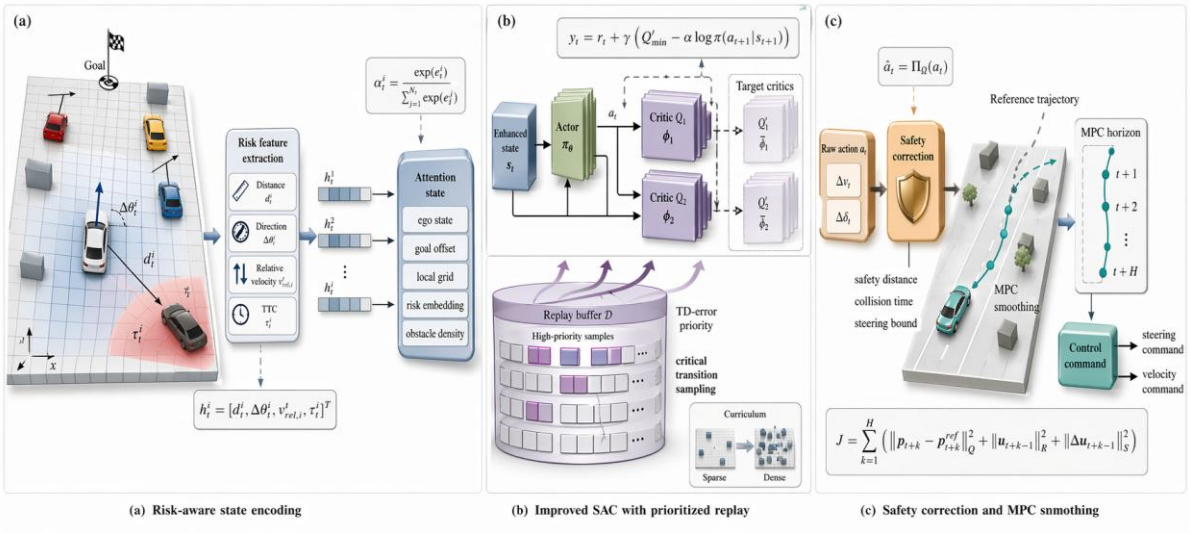

Figure 2: Improved DRL-based real-time path planning method with risk-aware state encoding, prioritized SAC learning, safety correction, and MPC smoothing.

In the state representation layer, this paper retains the local dynamic occupancy grid, and constructs a compact risk feature direction for each obstacle.

$$h_t^i = [d_t^i, \Delta\theta_t^i, v_t^{rel,i}, \tau_t^i]^\top \tag{5}$$

where d_t^i is the relative distance between the vehicle and the ith obstacle, $\Delta\theta_t^i$ is the relative azimuth, $v_t^{rel,i}$ is the relative velocity, and τ_t^i is the corresponding collision time estimation. This feature simultaneously depicts the degree of spatial proximity, movement trend and potential conflict intensity. After combining the semantic information of the local environment, the targets that really need priority attention in the scene can be identified more stably, which is consistent with the research conclusion of improving the quality of decision perception based on semantic segmentation [17]. Considering the continuous changes in the number and importance of obstacles in dynamic scenes, this paper introduces attention weighting to give higher weights to key obstacles:

$$\alpha_t^i = \frac{exp(e_t^i)}{\sum_{j=1}^{N_t} exp\,(e_t^j)} \tag{6}$$

where e_t^i is the correlation score of the ith obstacle, α_t^i is the normalized weight, and N_t is the number of obstacles currently participating in the coding. After weighted aggregation, the enhanced state includes the state of the vehicle itself, the relative position of the target, the local grid, the risk characteristics and the obstacle density. In Figure 2, subgraph (a) shows the process of risk feature extraction and attention weighting in a local dynamic environment.

The strategy learning part adopts the improved SAC. The continuous motion space is more suitable for describing the joint fine-tuning of speed and steering, so this paper retains its random strategy update framework, and uses the double critic structure to weaken the overvaluation. The target value is written as

$$y_t = r_t + \gamma(Q_{\min}, - \alpha \log \pi\,(a_{t+1} \mid s_{t+1})) \tag{7}$$

where r_t is the immediate reward, γ is the discount factor, $Q_{\min}$, is the smaller value of the network output of the two target critics, α is the entropy temperature coefficient, $\pi(.)$ is the strategy distribution. This setting helps to stabilize the policy update in high-dimensional continuous control and maintain a relatively stable convergence trend in high-frequency interactive scenes [18]. At the same time, experience playback introduces priority sampling, and high TD error samples are selected into training batches more frequently to improve the utilization of key conflict segments. This idea is consistent with the relevant research on improving experience playback to improve the efficiency of autonomous driving decision [19]. During the training process, the scene complexity gradually increases with the obstacle density and occlusion intensity, thus reducing the strategy shock in the early exploration stage [20]. As shown in Figure 2 (b), the enhanced state enters the improved SAC structure, and the strategy update is completed by combining the priority experience playback.

To reduce the risk of dangerous actions entering the execution layer directly, the original actions enter the safety correction module after output:

$$\hat{a}_t = \Pi_\Omega(a_t) \tag{8}$$

where, a_t is the original action output by the policy network, $\hat{a}_t$ is the modified security action, and $\Pi_\Omega(.)$ represents the operation projected to the security action set. The safety set is constrained by the minimum safety distance, collision time threshold and steering boundary, which can compress the output range of high-risk actions. This step explicitly connects the constraint information between the decision-making layer and the executive layer, which helps to reduce the impact of sudden cut in, sidetracking and jerk control on vehicle stability, which is consistent with the relevant research on reinforcement learning and path tracking collaborative design [21].

At the trajectory feasibility level, this paper converts the safety action into a short-term reference trajectory, and uses MPC for rolling optimization. Its cost function is written as

$$J = \sum_{k=1}^{H} \| p_{t+k} - p_{t+k}^{ref} \|_Q^2 + \| u_{t+k-1} \|_R^2 + \| \Delta u_{t+k-1} \|_S^2 \tag{9}$$

where H is the prediction time domain, p_{t+k} is the optimized future position, p_{t+k}^{ref} is the reference trajectory point, u_{t+k-1} is the control input, Δu_{t+k-1} is the control increment at the adjacent time, Q and R and S are the corresponding weight matrices. The objective function constrains trajectory deviation, control amplitude and control change rate at the same time, so that the generated trajectory has better ride comfort and tracking performance in addition to obstacle avoidance. In Figure 2 (c), the original action enters the MPC smoothing

module after safety correction, and finally forms an executable control command. The smoothed trajectory curve is more continuous than the original polyline path, which is consistent with the research results of road adaptive accurate tracking [22].

2.3 Experimental Settings and Evaluation Metrics

The experiment was completed in the joint environment of Carla 0.9.15 and ROS Noetic. The training, online decision-making and scene interaction were carried out on a unified platform. The statistical analysis and trajectory post-processing were assisted by Python and MATLAB. The vehicle adopts a small four-wheel unmanned vehicle model, and the body size and speed range are controlled within the common range of low-speed automatic driving experiment to simulate the tasks of Park distribution, patrol inspection and structured road traffic. The hardware composition of the platform is shown in Figure 3. The layout relationship of LiDAR, Camera, IMU, Controller, Drive wheel and Computing Unit is given in Figure 3 to illustrate the basic support conditions of the sensing, computing and control execution link.

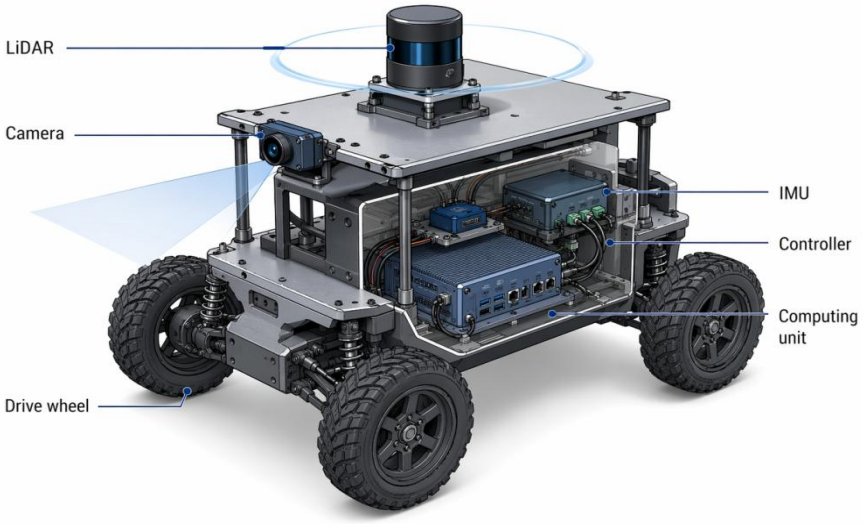

Figure 3: Hardware architecture of the unmanned vehicle platform.

The scene design focuses on the most common four types of interactive tasks in the dynamic environment, including single dynamic obstacle head-on crossing, multi obstacle crossing, sudden obstacle insertion and narrow passage meeting. Among them, the local plane area of 40m × 40m is used for the open scene, and the length of the road type scene is set to 160m to give consideration to local obstacle avoidance and medium and short distance online re planning. The number of obstacles is set to 3, 5, 7 and 10, corresponding to different interaction intensities from sparse to crowded; The obstacle speed and vehicle speed change together to test the response ability of the strategy under the conditions of speed difference and space compression. Compared with the experimental scheme using only a single obstacle density or a single interactive form, this setting is more suitable for observing the stability boundary of the method in the dynamic scene, and is more convenient for the comprehensive discussion of path quality, security and real-time in the following paper.

The training phase is organized by gradually increasing the complexity, with a total scale of 6×10^5 steps, corresponding to about 800 episodes. In the early stage, low obstacle density and weak interaction scenes were mainly used, and in the middle and late stage, cross crossing, sudden insertion and narrow meeting samples were gradually added to reduce the strategy oscillation during early exploration. In the test phase, 100 independent tests were conducted

under four scenarios, and five random seeds were used to repeat the experiment. The comparison methods include A*, DWA, RRT *, PPO, Vanilla SAC and the method in this paper. See Table 2 for the main experimental settings.

Table 2: Core experimental settings for dynamic path planning evaluation

Category	Item	Setting
Platform	Environment	CARLA 0.9.15+ROS Noetic
Vehicle	Size/wheelbase/speed range	1.20m×0.78m/0.86m/2-10m/s
Scenario	Scene types	Single crossing, multi-obstacle crossing, sudden insertion, narrow passage meeting
Scenario	Map scale/obstacle number	40m×40m or 160m road; 3/5/7/10 obstacles
Training	Training scale	6×10^5 steps, about 800 episodes, batch size 256
Testing	Evaluation and baselines	100 trials per scene, 5 random seeds; A*, DWA, RRT*, PPO, Vanilla SAC, Proposed

The evaluation index is developed from four dimensions: task completion effect, operation safety, path quality and real-time. The task completion effect was measured by success rate and collision rate; The average path length and maximum curvature are used for path geometry; The minimum safety distance shall be adopted for safety; Average jerk is used to control ride comfort; Average planning time is adopted for real-time performance; For the learning method, further statistics of cumulative rewards are used to analyze the strategy performance after training. The success rate is defined as the proportion of vehicles arriving at the target area without collision within the specified time window. The collision rate is defined as the proportion of physical contact in the test task. The average planning time is counted according to the time-consuming of a single online decision. Such a set of indicators can cover the core dimensions of "whether it can reach", "whether it is safe enough", "whether the trajectory is smooth" and "whether it can meet the requirements of real-time operation", so it is more suitable for comparing the comprehensive differences between traditional planning and reinforcement learning methods in a dynamic environment.

3 Results and Discussion

3.1 Convergence and Real-Time Performance Analysis

In order to verify the training stability and online real-time performance of the proposed method in dynamic environment, this paper makes a unified evaluation of PPO, original SAC, DWA and this method under four kinds of test scenarios, and uses the average results of five random seeds for statistics. The training phase focuses on observing the change trend of average return with the number of returns, while the testing phase records the time-consuming of single-step decision-making and the total delay of online re planning, which is used to evaluate the deployment feasibility of the method under the condition of high-frequency local updates.

Figure 4 shows the training convergence curves of different methods in dynamic planning tasks. In order to reduce the influence of single training fluctuation, the curve is drawn by moving average method, and the fluctuation bands under different random seeds are superimposed. It can be seen that the method in this paper is still in the stage of obvious exploration in the first 150 rounds, and the average return increases rapidly; After entering 200

rounds, the slope of the curve began to slow down, indicating that the strategy has gradually formed a stable behavior of obstacle avoidance and reaching the target. From 350 to 450 rounds, the average return of this method basically entered the stable stage, the sliding average reward increased from 0.74 to 0.81, and the standard deviation converged to about 0.06. In contrast, the inflection point of convergence of the original SAC appeared after about 430 rounds, while PPO still had obvious oscillation near 500 rounds. The verification set statistics show that the final average cumulative reward of this method is 12.8% higher than that of the original SAC and 21.4% higher than that of PPO, indicating that the risk perception state coding and security correction mechanism have a continuous gain on the quality of policy convergence.

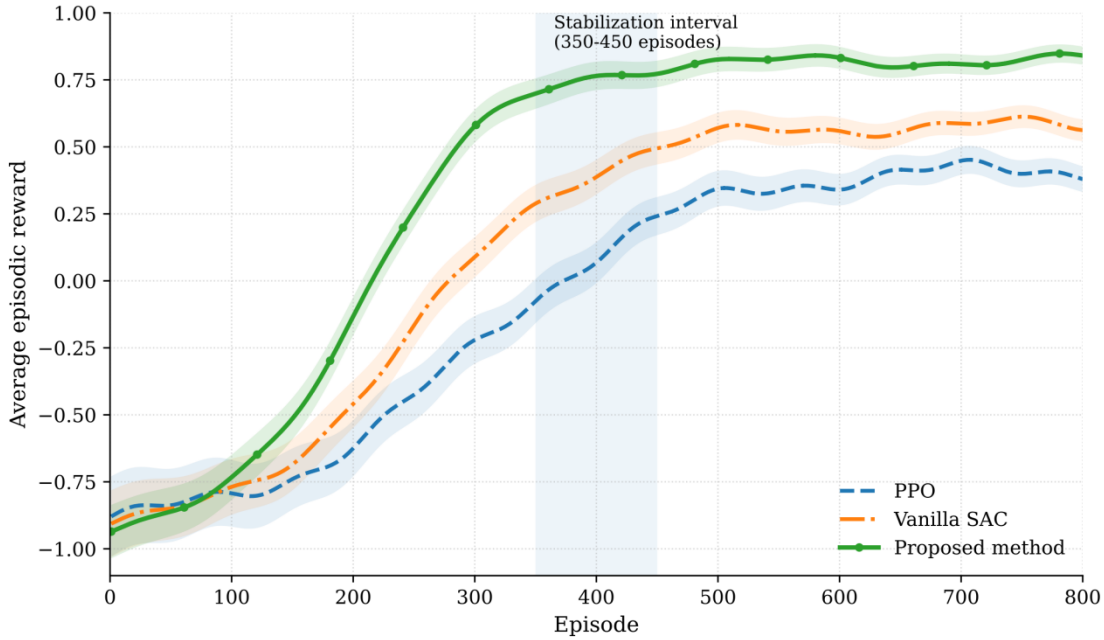

Figure 4: Training convergence curves under dynamic planning scenarios.

From the real-time results, the average decision-making time of this method in the test phase is 23.7ms, the original SAC is 31.4ms, the PPO is 34.8ms, and the DWA is 18.6ms. DWA still has some advantages in local decision-making speed, but its path quality and stability under complex interaction are insufficient, and it is more prone to bypass fluctuations and local failures in high-density dynamic scenes. After taking the time of perceptual refresh, state update, security correction and local smoothing into account, the total online re planning time of this method is 46.2ms, which is lower than the real-time operation threshold of 50ms, and can meet the online planning requirements under 20Hz control frequency. When subdivided by scene, the average re planning delay of the proposed method in the single dynamic obstacle crossing and narrow channel meeting scenes is 41.8ms and 48.7ms, respectively. Even when the number of obstacles increases to 10, the delay growth still remains within the acceptable range.

The curve change in Figure 4 is consistent with the above delay results. The fluctuation of this method is narrowed in the middle and late training, which shows that the strategy output is more stable, which directly reduces the additional burden in the safety correction and trajectory smoothing stages; The back segment of the original SAC and PPO curves still fluctuates, and its decision output is more prone to frequent correction and local recalculation in complex interaction scenarios. To sum up, the method in this paper not only maintains a fast convergence speed, but also controls the single-step decision-making time in the range of 20-35ms, and steadily compresses the total online re planning time within 50 ms, reflecting a good real-time execution potential.

3.2 Comparative Results of Path Quality and Safety

Figure 5 shows the comparison of path trajectories of different methods in four typical dynamic scenarios, including single obstacle crossing, multi obstacle crossing, sudden obstacle insertion and narrow channel meeting. Each sub map adopts a unified coordinate range and the same annotation method, which is convenient to directly compare the differences between A*, DWA, PPO, original SAC and the method in this paper in local obstacle avoidance form, steering continuity and target approach path. The figure also shows the trajectory of the dynamic obstacle and the boundary or static constraint area to illustrate the planning response of each method under the same environmental conditions.

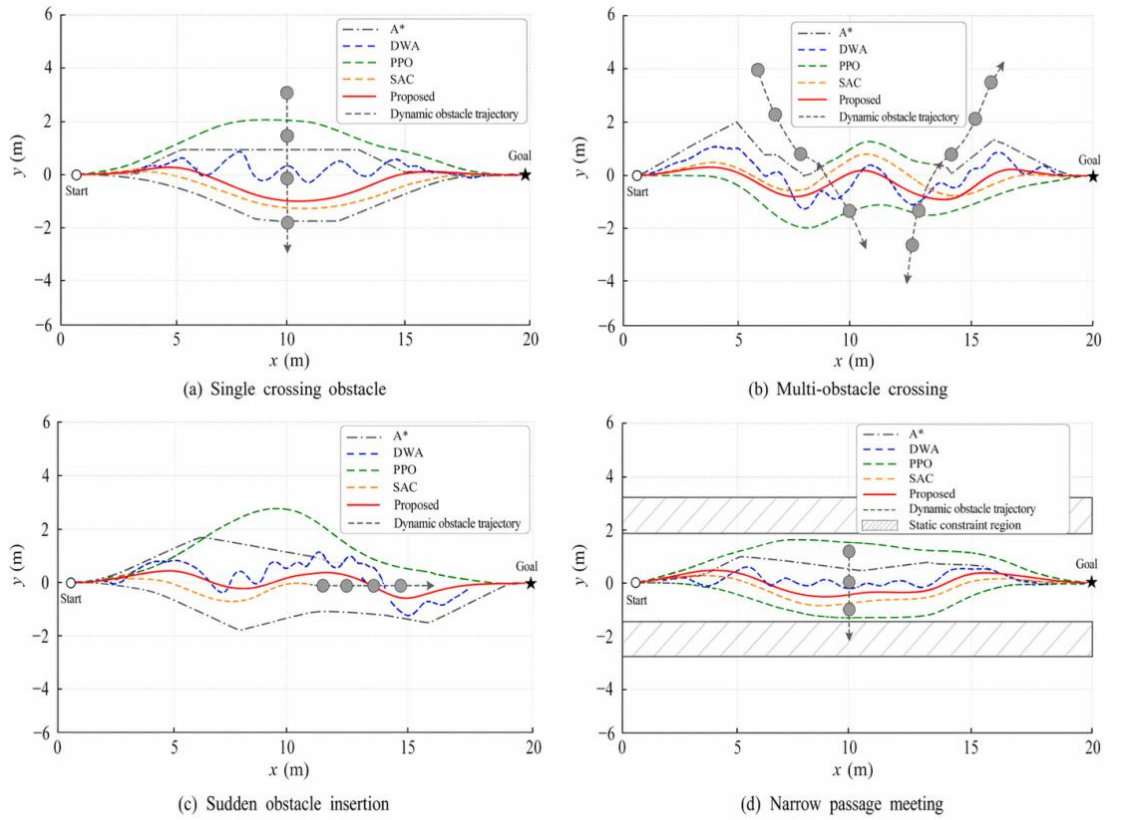

Figure 5: Trajectory comparison in representative dynamic scenarios.

As can be seen from Figure 5, A* can provide a shorter path in terms of static accessibility, but it has many turns in dynamic interference scenes, and the local obstacle avoidance action is hard, which is easy to form obvious turning points in narrow channels and sudden insertion conditions. The local reaction speed of DWA is fast, but there are obvious swings and repeated corrections in the trajectory, especially in the scene of multi obstacle intersection and car passing, which is more likely to produce detour redundancy. The overall feasibility of PPO is better than the traditional local method, but it still shows a large lateral offset and conservative path in the high interaction area. The original SAC can get a relatively smooth trajectory, but it is still close to the obstacle boundary in the multi obstacle intersection area. In contrast, the method in this paper maintains a more consistent steering transition in the four types of scenarios: faster return to the main heading after passing the obstacle in the single obstacle crossing scenario, smaller lateral offset in the burst insertion scenario, more uniform path curve in the narrow channel, and no obvious sharp turning angle. This shows that the improved strategy achieves a more stable balance between local obstacle avoidance and path efficiency.

To further quantify this difference, table 3 summarizes the success rate, collision rate, average path length, average planning time and minimum safety distance of each method on the unified test set. This table corresponds to the average results of four scenarios, five random seeds and 100 independent tests for each scenario, and can reflect the comprehensive performance of the method in terms of path quality and security.

Table 3: Core experimental settings for dynamic path planning evaluation

Method	Success rate (%)	Collision rate (%)	Average path length (m)	Average planning time (ms)	Minimum safety distance (m)
A*	78.4	13.2	49.3	41.6	0.31
DWA	84.7	9.6	47.8	18.6	0.42
PPO	86.9	8.1	46.9	34.8	0.48
Vanilla SAC	89.5	5.7	45.8	31.4	0.56
Proposed	93.8	3.4	44.1	23.7	0.68

Table 3 shows that this method has achieved the highest success rate of 93.8% and the lowest collision rate of 3.4%, which is 4.3% higher and 2.3% lower than the original SAC. In terms of path efficiency, the average path length of this method is 44.1m, which is shorter than PPO, DWA and A*, indicating that this method does not rely on overly conservative detour for security. At the same time, the minimum safe distance is 0.68m, which is higher than other comparison methods, indicating that the trajectory retains a more reasonable avoidance margin in the complex interaction area. Combined with the real-time results in Section 2.1, it can be seen that the method in this paper still maintains the single-step decision-making level of 23.7ms in the planning time, so the improvement of path quality is not at the cost of significantly sacrificing online efficiency.

The above results are consistent with the method design. The risk perception state representation improves the identification ability of the strategy for key obstacles and potential conflict areas, the priority experience playback strengthens the learning of high-risk interactive samples, the safety correction suppresses the high-risk action output, and MPC smoothing further improves the curvature continuity and local tracking feasibility. After multi-layer improvement, the trajectory can shorten unnecessary detour and reduce local oscillation while maintaining a high safety margin. Therefore, it shows better comprehensive quality in terms of success rate, collision rate, path length and trajectory smoothness.

3.3 Ablation, Robustness, and Generalization Discussion

In order to identify the actual contribution of each improved module to the performance, this paper conducted four groups of ablation experiments on the unified test set, respectively removing the attention mechanism, priority experience playback, security constraint module and MPC smoothing layer. The results are shown in Figure 6. All results were the average of 5 random seeds and 100 independent tests in each type of scenario. Figure 6 shows that the complete model maintains the most balanced performance among success rate, collision rate, planning time and trajectory smoothness. After removing the safety restraint module, the success rate decreased from 93.8% to 88.1%, and the collision rate increased from 3.4% to 8.7%, which was the most obvious degradation in the four groups of ablation, indicating that the module is the most critical to reduce the collision risk. After removing the attention mechanism, the success rate decreased to 91.6%, and the collision rate increased to 4.9%, indicating that the weighting of key obstacles is helpful to improve the risk identification ability in complex interaction scenes. After removing the priority experience playback, the decline of

each index is relatively small, but it still brings a certain loss of success rate, indicating that the reuse of high-risk samples improves the efficiency of strategy learning. After removing MPC, the average planning time decreased from 23.7ms to 18.9ms, which was the most significant module affecting the planning time; But at the same time, the average jerk increased from 0.61m/s³ to 1.08m/s³, and the trajectory continuity became significantly worse. It can be seen that the safety constraint layer mainly determines the collision rate, and the MPC layer mainly affects the planning time and trajectory smoothness. After the constraint layer is introduced, the radical actions in the strategy output will be screened out before execution, the minimum safe distance will be maintained, and the local steering jump will be less, so the trajectory shows a more stable shape.

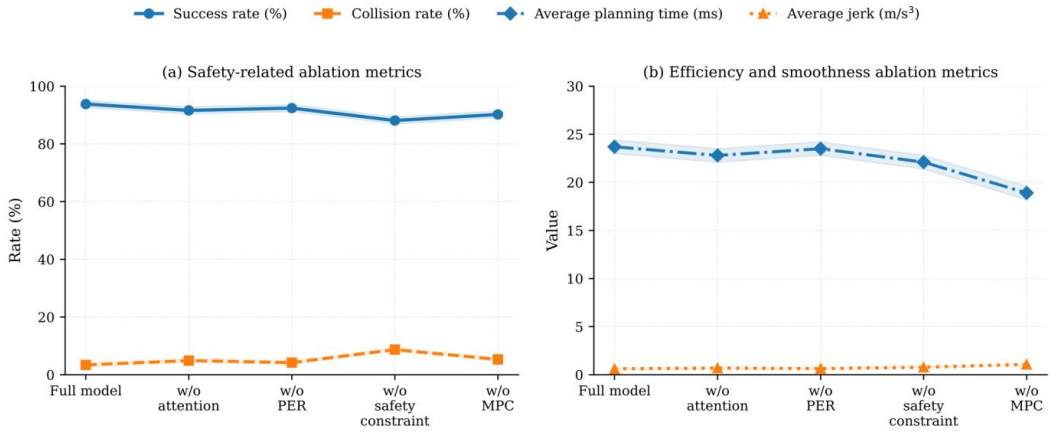

Figure 6: Ablation results of the proposed method.

At the level of robustness and generalization, this paper further investigated the test conditions of unprecedented obstacle speed, higher obstacle density and position disturbance, and the results are shown in Table 4. Compared with the test set in the domain, the overall performance of the model in the scenario outside the distribution has declined, but the decline remains within an acceptable range. When the obstacle speed expanded to 6-8m/s and 8-10m/s, the success rate remained at 91.5% and 88.9%, respectively; When the number of obstacles increased to 12 and 14, the success rate was 89.7% and 86.8%, respectively; When the standard deviation of position noise is 0.05m and 0.10m, the success rate is still 90.8% and 87.6%. This shows that the proposed method has good adaptability to the increase of scene complexity and the disturbance of perception error. It should be noted that higher obstacle density has more obvious impact on the minimum safe distance and planning time, indicating that intensive interaction is still the main source of degradation of generalization performance.

Table 4: Robustness and generalization results under shifted test conditions

Test condition	Success rate (%)	Collision rate (%)	Avg. planning time (ms)	Min. safety distance (m)
In-domain test set	93.8	3.4	23.7	0.68
Unseen speed: 6-8m/s	91.5	4.2	24.6	0.64
Unseen speed: 8-10m/s	88.9	5.1	25.9	0.6
Higher density: 12 obstacles	89.7	5.4	27.3	0.58
Higher density: 14 obstacles	86.8	6.7	29.5	0.54
Position noise: σ=0.05m	90.8	4.8	24.8	0.62
Position noise: σ=0.10m	87.6	6.1	26.1	0.57

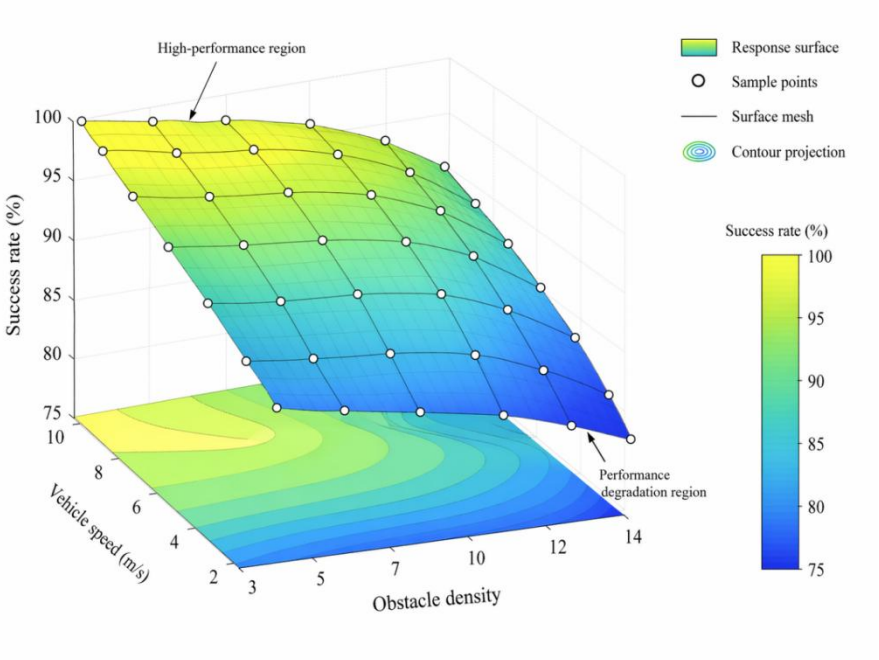

Figure 7: Success-rate surface under varying obstacle densities and vehicle speeds.

In order to further observe the impact of the change of task difficulty on the overall performance, Figure 7 shows the three-dimensional response surface of the success rate when the obstacle density and vehicle speed change together. It can be seen that as the obstacle density increases from 3 to 10 and the vehicle speed increases from 2m/s to 10m/s, the success rate gradually decreases from 98.2% in the high value area to 79.6%. The overall surface showed a smooth downward trend, indicating that the model did not appear abrupt instability in the scene of increasing difficulty, but showed continuous degradation characteristics. This result and the generalization test in Table 4 confirm each other, and it also shows that the method in this paper still has good predictability and stability under the conditions outside the distribution.

4 Conclusion

Focusing on the task of real-time path planning for unmanned vehicles in dynamic environment, this paper proposes a planning method based on improved deep reinforcement learning, and carries out joint design in four levels: state modeling, policy learning, security constraints and trajectory smoothing. Specifically, the method enhances the ability of the strategy to identify key conflict targets by introducing local dynamic occupation information, obstacle movement risk characteristics and attention weighting mechanism; The convergence quality of continuous action decision is improved by improving SAC and priority experience playback; The feasibility and stability of the trajectory are improved by safety action correction and MPC smooth optimization. The experimental results show that the method has achieved relatively stable comprehensive advantages in the success rate, collision rate, average path length, minimum safety distance and online planning delay, which shows that it can form a more reasonable balance between dynamic obstacle avoidance quality, path efficiency and control smoothness.

(1) The main value of this paper is to further expand "reachable" to "secure, real-time and executable". Compared with the scheme that only focuses on the success rate of single obstacle avoidance or the optimal static path, the method in this paper can better reflect the advantages

of continuous decision-making in dense interactive scenes. The training convergence analysis, path trajectory comparison and ablation experiments show that the risk perception state representation improves the identification ability of complex local environment, the safety constraint layer significantly inhibits the output of high-risk actions, and MPC smoothing reduces the mutation and jitter in the trajectory. The performance improvement thus obtained does not come from a single module, but from the collaborative optimization between the perception, decision and execution links.

(2)From the perspective of application, the results of this paper have clear engineering significance for real-time planning in dynamic environment. In park inspection, logistics distribution and complex road testing, unmanned vehicles often face problems such as uncertain obstacle movement, continuous change of local space and strict restriction of control cycle. Relying solely on traditional heuristic planning or unconstrained reinforcement learning strategy, it is difficult to consider both timeliness and safety at the same time. This method can maintain a high success rate and a low collision rate while maintaining a short one-step decision-making time, indicating that it has a certain potential for online deployment. This means that the framework is not only suitable for algorithm comparison in simulation scenarios, but also provides a scalable technical basis for subsequent migration to higher complexity unmanned vehicle platforms.

(3)There are still some directions to be further promoted in this paper. The current experiment is mainly based on the simulation environment, and the problems of perception bias, dynamic mismatch and control delay in the migration from simulation to real vehicle need to be further solved in the future; In the scenario of more complex traffic interaction, the multi vehicle collaborative decision-making and yield mechanism still need to be modeled separately; For the environment with stronger occlusion and incomplete observation, the ability of state estimation under some observable conditions should continue to be strengthened; At the same time, the robustness of multi-sensor fusion in complex dynamic scenes still has room for improvement. In addition, the generalization verification for higher obstacle density, cross scene distribution offset and stricter security constraints should also become an important focus of follow-up research. Only by further strengthening these aspects, the proposed method can more fully support the stable application in real dynamic traffic environment.

References

[1] Hu, Y., et al. (2023). Path planning for autonomous vehicles based on deep reinforcement learning in dynamic environment. Applied Sciences, 13(18), 10056.

[2] Alzubaidi, L., et al. (2023). Autonomous vehicle navigation in dynamic environments based on deep reinforcement learning and reward shaping. IEEE Access, 11, 27127-27137.

[3] Li, Y., et al. (2023). Lane change strategies for autonomous vehicles: A deep reinforcement learning approach based on transformer. IEEE Transactions on Intelligent Vehicles, 8(3), 2197-2211.

[4] He, X., et al. (2023). Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach. IEEE Transactions on Intelligent Vehicles, 8(1), 184-193.

[5] Ashwin, R., & Naveen Raj, V. R. (2023). Deep reinforcement learning for autonomous

vehicles: Lane keep and overtaking scenarios with collision avoidance. International Journal of Information Technology, 15(7), 3541-3553.

[6] Candela, V., et al. (2023). Risk-aware controller for autonomous vehicles using model-based collision prediction and reinforcement learning. Artificial Intelligence, 320, 103923.

[7] Wu, G., et al. (2023). Dyna-PPO reinforcement learning with Gaussian process for the continuous action decision-making in autonomous driving. Applied Intelligence, 53(13), 16893-16907.

[8] Wang, Z., et al. (2024). A deep reinforcement learning-based approach for autonomous lane-changing velocity control in mixed flow of vehicle group level. Expert Systems with Applications, 238, 122158.

[9] Wu, G., et al. (2024). Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey. Transportation Research Part C: Emerging Technologies, 164, 104654.

[10] Hou, M., et al. (2024). Merging planning in dense traffic scenarios using interactive safe reinforcement learning. Knowledge-Based Systems, 290, 111548.

[11] Aguilar-Marsillach, C., et al. (2024). Autonomous vehicle planning in occluded merges with stochastic safety constraints. IFAC-PapersOnLine, 58, 216-221.

[12] Elallid, B., et al. (2024). Enhancing autonomous driving navigation using soft actor-critic. Future Internet, 16(7), 238.

[13] Wang, H., et al. (2024). Enhancing active disturbance rejection design via deep reinforcement learning and its application to autonomous vehicle. Expert Systems with Applications, 239, 122433.

[14] Gao, Y., et al. (2024). Human-like mechanism deep learning model for longitudinal motion control of autonomous vehicles. Engineering Applications of Artificial Intelligence, 133, 108060.

[15] He, X., et al. (2024). Trustworthy autonomous driving via defense-aware robust reinforcement learning against worst-case observational perturbations. Transportation Research Part C: Emerging Technologies, 163, 104632.

[16] Zhang, H., et al. (2025). Decision-making of autonomous vehicles in interactions with jaywalkers: A risk-aware deep reinforcement learning approach. Accident Analysis & Prevention, 210, 107843.

[17] Gao, S., et al. (2025). Reinforcement learning decision-making for autonomous vehicles based on semantic segmentation. Applied Sciences, 15(3), 1323.

[18] Wang, Z., et al. (2025). Enhancing lane change safety and efficiency in autonomous driving through improved reinforcement learning for highway decision-making. Electronics, 14(5), 918.

[19] Wang, X., et al. (2025). Highway autonomous vehicle decision-making method based on

prior knowledge and improved experience replay reinforcement learning algorithm. Expert Systems with Applications, 284, 127927.

[20] Yang, B., et al. (2025). Decision making for highway autonomous driving using hybrid reinforcement learning. Journal of Control and Decision, 12(6), 1043-1051.

[21] Han, J., et al. (2025). Hybrid path tracking control for autonomous trucks: Integrating pure pursuit and deep reinforcement learning with adaptive look-ahead mechanism. IEEE Transactions on Intelligent Transportation Systems, 26(5), 7098-7112.

[22] Han, J., Sun, X., et al. (2025). Road-adaptive precise path tracking based on reinforcement learning method. Sensors, 25(15), 4533.

[23] Li, J., et al. (2025). Research on dynamic trajectory planning based on model predictive theory for complex driving scenarios. Sensors, 25(23), 7241.

[24] Elallid, B., et al. (2025). Secure and efficient vehicle control of autonomous vehicles using federated deep reinforcement learning. Applied Soft Computing, 185, 113924.

[25] Elallid, B., et al. (2025). Enhancing autonomous vehicle control in complex scenarios with deep transformer reinforcement learning. Engineering Applications of Artificial Intelligence, 158, 111483.