Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

A Decision Planning Method for Unstructured Road Scenarios Based on Deep Reinforcement Learning

Liwei Jiang*, Manjiang Wang, Qian Qiu, Wenchao Xiao, Bo Zhou

Wuhan Bus Manufacturing Co. Ltd., Wuhan, China. 430200

*Corresponding to: Liwei Jiang

Abstract: The rapid development of autonomous driving has intensified research on decision-making in unstructured road scenarios. Conventional rule-based methods often suffer from poor adaptability, limited efficiency gains, and inadequate economic performance in such environments. This paper presents a Deep Q-Network (DQN)-based approach that defines observation states and decision actions, with a reward function incorporating efficiency, economy, safety, and comfort. Simulations in a mining road environment show that the method outperforms traditional approaches, enhancing decision-making capabilities in unstructured scenarios and offering new perspectives for autonomous driving in complex environments.

Keywords: Deep reinforcement learning; Unstructured road; Decision-making and planning; DQN; Mining area scenario

1 INTRODUCTION

In the core architecture of autonomous driving systems, the decision-planning module plays a crucial role. It is directly responsible for interpreting perception data, understanding the surrounding environment, and ultimately generating safe, efficient, and traffic-rule-compliant driving trajectories and behavior commands. The performance of this module fundamentally determines whether an autonomous vehicle can operate reliably in complex, dynamic, and uncertain real-world traffic environments [1-3].

For a long time, rule-based decision-planning methods [4-7] have been the mainstream technology in this field. The core paradigm of these methods is that domain experts manually design and code a comprehensive set of decision logic based on traffic rules, driving experience, and understanding of typical scenarios, such as state machines, decision trees, and if-then rule sets. During operation, the vehicle's behavior is driven by matching the current state to predefined rules. Talebpour et al. [8] proposed a lane-changing model based on game theory, and by designing a series of state logic, connection methods, and conditions, they achieved good lane-changing decision-making performance. Aksjonov et al. [9] proposed a rule-based autonomous driving decision-making scheme that addressed the challenges of complex intersections in mixed traffic environments and was applicable to other types of intersections with different traffic rules. Hwang et al. [10] proposed a reinforcement learning-based lane-changing policy network embedded in a finite state machine, which achieves high lane-changing performance without compromising safety. These methods perform well on structured roads; however, when facing unstructured road scenarios, especially in special environments such as mines, construction sites, and the wilderness, the decision-making performance of these methods significantly degrades or even fails.

In recent years, deep reinforcement learning (DRL) [8–10] has emerged as a promising technique for autonomous driving decision-making and planning. For instance, Wu et al. [11] proposed a general decision framework combining Monte Carlo Tree Search and DRL, later extending it to continuous state spaces without self-play for highway driving cases. Yuan et al. [12] introduced a game-theoretic DRL approach, enabling vehicles to use 2D LiDAR observations for decision-making at unsignalized intersections while modeling multiple interactive vehicles with conservative, aggressive, and adaptive behaviors. In [13], a hierarchical control framework was proposed for highway scenarios, leveraging a dueling deep Q-network to derive driving strategies. Shi [14] analyzed driving styles via surveys and developed a human-like decision model using DRL. Liao et al. [15] improved exploration in end-to-end training by introducing a random policy selection strategy, though their work mainly focused on policy randomness and experience replay rather than unstructured environments such as mining roads.

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

This study proposes a DQN-based decision-making framework tailored for mining road scenarios. The state and action spaces are carefully defined, and a reward function integrating efficiency, economy, safety, and comfort is designed. A comprehensive reinforcement learning environment is constructed to capture road complexity and dynamic interactions. Simulation results demonstrate significant improvements in driving efficiency, economy, safety, and comfort, thereby enhancing decision-making in unstructured mining environments. This study provides innovative ideas and methods for the application of autonomous driving technology in complex road environments such as mining areas, and demonstrates the strong adaptability and potential of reinforcement learning in practical traffic scenarios.

2 DQN-Based Decision Planning

2.1 Architecture of the DQN-Based Decision Model

Deep Q-Networks (DQN) combine deep learning with reinforcement learning and are primarily applied to problems with discrete action spaces. By interacting with the environment, the agent learns an optimal policy that maximizes cumulative rewards through appropriate action selection in given states.

In the proposed DQN-based decision system, the vehicle first acquires real-time state information—such as ego-pose and surrounding obstacles—from the perception module. After preprocessing, these inputs are encoded as a state vector and fed into the DQN. Based on the current state, the DQN evaluates candidate actions (e.g., turning left, turning right, cruising, following, or emergency braking with AEB) and outputs the optimal decision. The trajectory planning and control layer then generates a feasible trajectory and tracking commands, which are translated into low-level control signals such as steering angle, throttle, or braking force, and executed by the vehicle controller. After each action, the simulation environment updates the vehicle state and provides both new sensory information and a corresponding reward, which are used by the DQN to update its policy. This iterative process enables continuous interaction between agent and environment, progressively optimizing decision-making performance (see Figure 1).

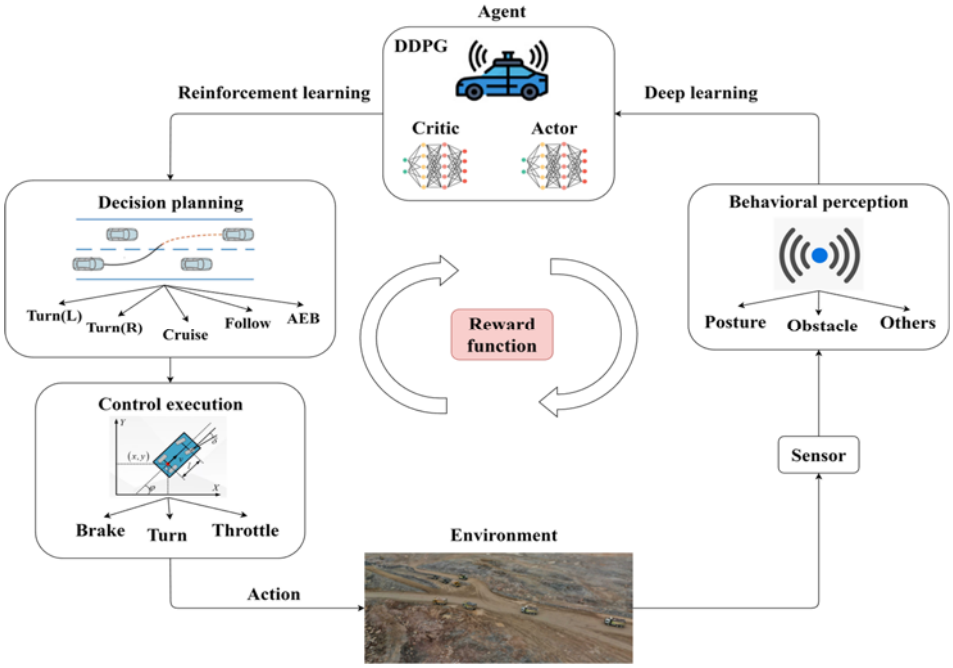

Figure 1 Overall System Architecture

2.2 Training State and Action Definitions

2.2.1 State Definition

In unstructured mining road scenarios, environmental uncertainty is significant. Therefore, the state space must capture not only the vehicle's kinematic state, but also road constraints and dynamic features of surrounding objects. In this study, the state space is partitioned into the following components:

Reference Line Constraint

The reference line provides the desired trajectory reference for vehicle movement. The lateral offset and heading angle deviation between the vehicle and the reference line are defined as follows:

$$s_{ref} = [d_{lat}, \Delta\phi_{ref}] \quad (1)$$

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

Where d_{lat} represents the lateral distance between the vehicle's center and the reference line, and $\Delta\phi_{ref}$ represents the difference between the vehicle's heading angle and the tangent angle of the reference line.

Road boundary constraints

Due to the presence of irregular slopes or temporary obstacles on mine site roads, vehicles need to consider the distance from the road boundary, defined as follows:

$$s_{road} = [d_{left}, d_{right}] \tag{2}$$

Where d_{left}, d_{right} represent the minimum distances from the vehicle to the left and right boundaries, respectively.

Obstacle information

Dynamic or static obstacles in the mining environment (such as large mining vehicles, pedestrians, fixed equipment, etc.) must be included in the state representation. Select the N nearest obstacles to the vehicle, and represent their relative position, relative speed, and relative heading as follows:

$$s_{obj}^{i} = [\Delta x^{i}, \Delta y^{i}, \Delta v^{i}, \Delta\phi^{i}], \quad i = 1, \ldots, N \tag{3}$$

Vehicle State

The kinematic state of the vehicle is defined as follows:

$$s_e = [x, y, v, \phi] \tag{4}$$

Here, (x, y) represents the vehicle's position in the global coordinate system, v represents the vehicle speed, and ϕ represents the heading angle.

Global information

Mining operations tasks typically rely on planned routes; therefore, the state information should also include global reference trajectory points or target task points, defined as follows:

$$s_{global} = [x^{g}, y^{g}, v^{g}, \phi^{g}] \tag{5}$$

Where $(x^{g}, y^{g}, v^{g}, \phi^{g})$ represents the position, velocity, and direction of the reference path or target point in the global coordinate system.

Therefore, the complete state set S is defined as follows:

$$S = (s_e, s_{ref}, s_{road}, s_{global}, ObjList) \tag{6}$$

Where, $ObjList = \{s_{Obj}^{1}, s_{Obj}^{2}, \ldots, s_{Obj}^{N}\}$.

2.2.2 Action Definition

In unstructured mining road scenarios, the vehicle must adapt its behavior flexibly according to road geometry, surrounding traffic participants, and unexpected events. In this study, the action space is discretized into a set of behaviors to ensure interpretability and practical feasibility in complex environments. The action set consists of five maneuvers: turning left, turning right, cruising, following, and emergency braking (AEB):

$$A = \{a_{turn}, a_{cruise}, a_{follow}, a_{\mathrm{AEB}}\} \tag{7}$$

2.3 Reward Function Design

The core mechanism of reinforcement learning lies in guiding the agent to learn an optimal policy by maximizing cumulative long-term rewards. This makes it essential that the reward function closely reflects the driver's decision logic and desired behavior. To achieve a comprehensive evaluation, the reward function is designed from four aspects: operational efficiency, economy, safety, and comfort.

2.3.1 Operational Efficiency and Economic Reward

Operational efficiency is critical for autonomous vehicles, particularly in logistics and transportation scenarios such as mining areas. Encouraging the vehicle to maintain higher speeds reduces travel time and enhances transportation efficiency

$$R_{\mathrm{eff}} = 1 - \frac{|v - v_{\mathrm{ref}}(k)|}{v_{\max}} \tag{8}$$

Here, v、v_{ref} represent the current vehicle speed and the reference speed, respectively. R_{eff} represents the reward term.

Evaluating economic performance can be quite complex; this article uses speed tracking performance as a simplified evaluation method. Since the target speed is designed based on optimal fuel efficiency, better speed tracking performance indicates better overall economic performance. Therefore, economic performance and operational efficiency can be considered equivalent and combined for evaluation.

2.3.2 Safety Reward

The safety reward is composed of two components: obstacle collision reward R_{safe1} and boundary reward R_{safe2}.

Collision reward R_{safe1}: Collisions are critical events and must be strictly avoided. A severe penalty is assigned if a collision occurs; otherwise, the reward is zero:

$$R_{\mathrm{safe1}} = \begin{cases} -1000, & \text{if collision} \\ 0, & \text{else} \end{cases} \tag{9}$$

Boundary reward R_{safe2}: Vehicle safety relative to road boundaries is evaluated based on the minimum lateral distance to the left and right boundaries and the minimum longitudinal distance to potential collision boundaries. Penalties are applied when the vehicle is too close to boundaries, while larger distances are rewarded to encourage safe lane-keeping:

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

$$R_{\text{safe2}} = \frac{d_{\text{lat}} - d_{\text{lat}}^{\text{th}}}{d_{\text{lat}}^{\max}} \tag{10}$$

Here, $d_{\text{lat}}^{\text{th}}$ represents the minimum allowable distance between the vehicle's outer boundary and the road boundary, and d_{lat} represents the actual distance.

Therefore, the final safety reward Rsafe is:

$$R_{\text{safe}} = R_{\text{safe1}} + R_{\text{safe2}} \tag{11}$$

2.3.3 Comfort Reward

The comfort reward evaluates the smoothness and naturalness of vehicle motion. It includes two aspects: (1) constraining abrupt speed changes by penalizing large differences from the previous time step, encouraging operation within reasonable acceleration limits to avoid sudden acceleration or braking; (2) limiting rapid changes in heading angle to prevent frequent left-right oscillations, thereby enhancing overall ride comfort.

$$R_{\text{conf}} = \delta_1\left(1 - \frac{|v - v_{\text{pre}}|}{v_{\max}}\right) + \delta_2\left(1 - \frac{|\phi - \phi_{\text{pre}}|}{\phi_{\max}}\right) \tag{12}$$

Here, v_{pre}、ϕ_{pre} represents the speed and heading angle at the previous time step.

2.3.4 Overall Reward

The total reward is obtained by summing the individual components, guiding the vehicle to consider operational efficiency, economy, safety, and comfort simultaneously, and thereby achieving optimal decision-making and planning.

$$R_{\text{t}} = \omega_1 R_{\text{eff}} + \omega_2 R_{\text{safe}} + \omega_3 R_{\text{conf}} \tag{13}$$

2.4 Environment Construction

The experimental environment in this study is built based on long-term operational data collected from 60 mining trucks in a mining area. Key features of the mining roads were extracted for analysis and integration.

2.4.1 Road Network Model

The mining road network can be represented as a directed graph, where a series of road segments connect critical nodes such as intersections and loading zones. Each road segment is characterized by three core parameters, which collectively influence the vehicle's dynamic behavior:

Slope angle: simulates uphill and downhill terrain

$$\theta_n \in [\theta_{\min}, \theta_{\max}] \tag{14}$$

Road curvature: simulating both curved and straight roads

$$\kappa_n = \frac{1}{R_n} \tag{15}$$

Where R_n represents the radius of curvature of the curve.

(c) Tire-road adhesion coefficient: simulates road conditions such as wet or muddy surfaces.

$$\mu_n \in [\mu_{low}, \mu_{high}] \tag{16}$$

These parameters are statistically derived from the historical operational data of 60 mining trucks and follow a specific empirical distribution $P(\theta, \kappa, \mu)$, enabling the simulation to randomly generate diverse road segment combinations that approximate the uncertainty of real-world environments.

2.4.2 Obstacle Generation Model

The generation of dynamic obstacles is a core component of the simulation environment. The physical attributes of these obstacles are determined by measuring the dimensions of vehicles operating in the mining area, and their size parameters (length L, width W, height H) are randomly generated within the measured range.

$$\begin{aligned} L_j &= L_{base} + \Delta L \cdot \alpha_j, \quad \alpha_j \in (0,1) \\ W_j &= W_{base} + \Delta W \cdot \beta_j, \quad \beta_j \in (0,1) \\ H_j &= H_{base} + \Delta H \cdot \gamma_j, \quad \gamma_j \in (0,1) \end{aligned} \tag{17}$$

Among them, $L_{base}, W_{base}, H_{base}$ is the reference size and $\Delta L, \Delta W, \Delta H$ is the possible change.

To realistically replicate mining traffic flow, the trajectories of dynamic obstacles are generated based on long-term operational data from 60 mining trucks. Typical behavior patterns, such as fully loaded uphill, empty downhill, and intersection merging, are obtained through clustering. The motion state of each obstacle is fully described by a state vector:

$$X_j(t) = [x_j(t), y_j(t), v_j(t), \phi_j(t)]^{\text{T}} \tag{18}$$

Here, $(x_j(t), y_j(t))$ represents the sequence of path points extracted from historical data. The trajectory of a dynamic obstacle is determined based on its location on different road segments; $v_j^{base}, \phi_j^{base}$ assigns different paths and base speeds to it, and a random perturbation $\varepsilon(t)$ is introduced to simulate the uncertainty in driving behavior.

$$v_j(t) = v_j^{base} + \varepsilon_v(t), \quad \varepsilon_v \sim \vartheta(0, \sigma_v^2) \tag{19}$$

$$\phi_j(t) = \phi_j^{base} + \varepsilon_\phi(t), \quad \varepsilon_\phi \sim \vartheta(0, \sigma_\phi^2) \tag{20}$$

Where noise variance $\sigma_v^2, \sigma_\phi^2$ is obtained from historical data.

2.5 DQN Network Design

The DQN employs a deep neural network as a function approximator to estimate the Q-value function Q(s,a), representing the expected cumulative reward for taking action aaa in state sss. A critic network predicts the current Q-values,

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

while experience replay stabilizes training by breaking correlations between consecutive samples and improving convergence. A target network is used to generate target Q-values, with parameters periodically copied from the critic to reduce bias and variance. The procedure is as follows:

(1) Initialize: the current network.

(2) Action selection: via ε-greedy strategy: initially, actions are chosen randomly with high probability (ε) to explore the environment; ε gradually decreases during training, favoring actions with the highest Q-value to balance exploration and exploitation.

(3) Execute action: a, observe reward r and next state s, and store (s, a, r, s') in the replay buffer.

(4) Sample mini-batches: from the buffer to compute target Q-values. The network is trained by minimizing the mean squared error (MSE) between predicted and target Q-values:

$$L = \frac{1}{N}\sum_{i=1}^{N}(Q(s,a;\theta) - y_i)^2 \quad (21)$$

The loss function is minimized through the mini-batch gradient descent algorithm to update the current network $Q(s,a;\theta)$.

(5) Every N steps, synchronize the current network $Q(s,a;\theta)$ with the target network.

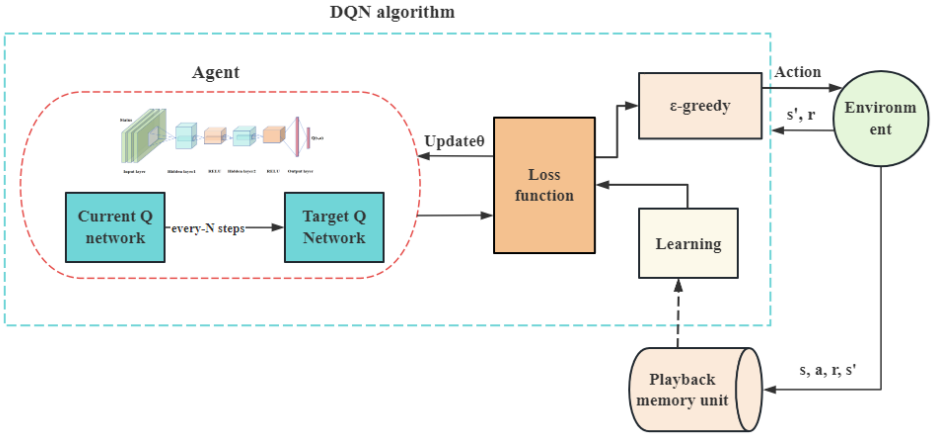

FIGURE 2 FLOWCHART OF THE DQN ALGORITHM

3 ALGORITHM TRAINING

The simulation platform used in this study is Prescan + Matlab. Prescan provides essential elements for autonomous driving, including driving environments, perception devices, and vehicle dynamics, and was used to construct a 3D mining scenario (Fig. 3). Matlab was employed for algorithm simulation, particularly reinforcement learning. The proposed method was compared with a rule-based baseline under identical conditions.

FIGURE 3 PRESCAN MINING SITE SCENE

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

3.1 TRAINING ENVIRONMENT INTEGRATION

The designed DQN agent was integrated into the training environment, which consists of global path planning, local path planning, speed planning, control units, and the Prescan simulation scenario. Obstacle and ego-vehicle states from Prescan were used as inputs to the agent, along with corresponding rewards. The agent's decisions were then passed to the local planning and control modules. The overall framework is shown in Fig. 4.

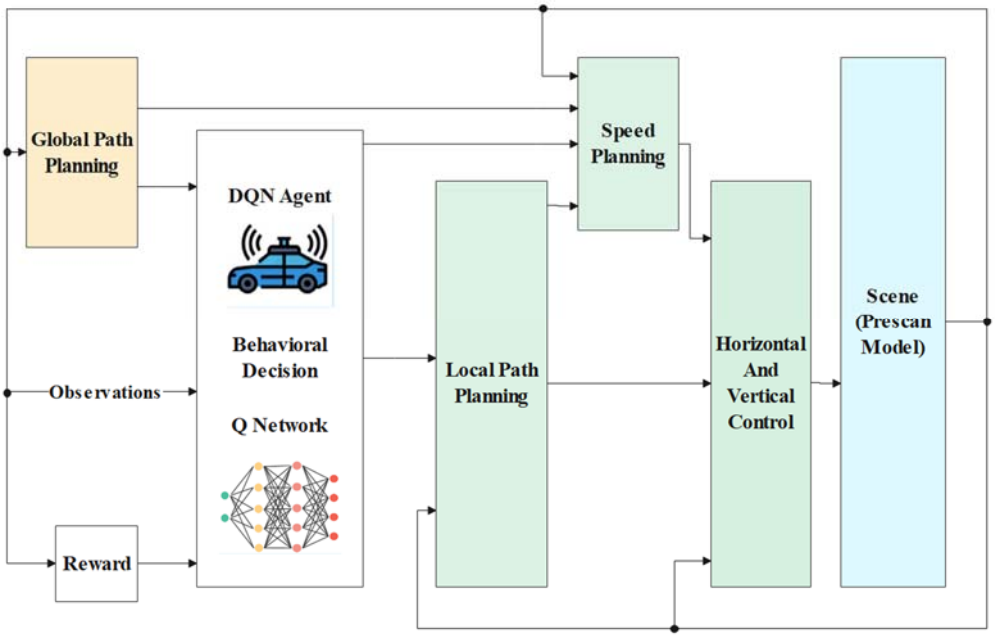

FIGURE 4 SYSTEM SOFTWARE ARCHITECTURE

3.2 MODEL TRAINING

Each training episode was limited to 1500 steps, terminating either at the maximum step count or upon reaching the destination. A total of 20,000 episodes were conducted. The training process is illustrated in Fig. 5, with key parameters listed in Table.1.

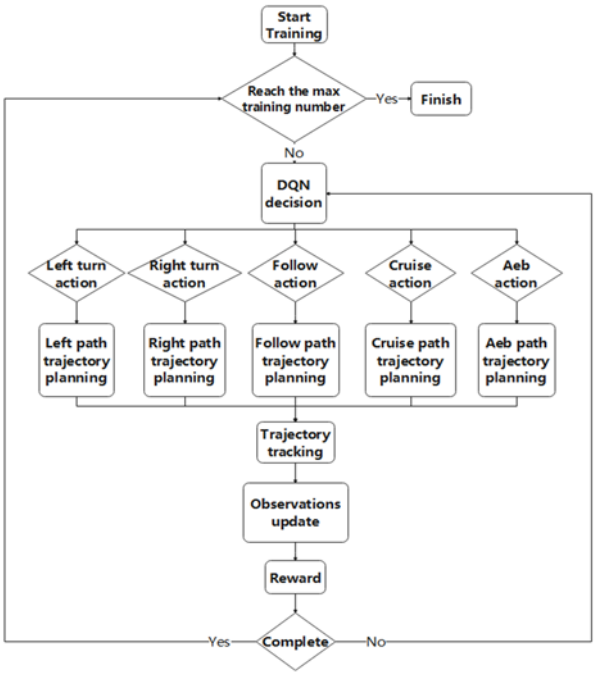

FIGURE 5 REINFORCEMENT LEARNING TRAINING PROCESS

TABLE 1 MODEL TRAINING PARAMETERS

Parameter	Description	Value
T_x	Simulation time step (s)	0.1
T_f	Total simulation time (s)	150
s_{len}	Total path length (m)	300
$road_{width}$	Total road width (m)	20

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

Parameter	Description	Value
V_{ref}	Maximum vehicle reference speed (m/s)	5
BatchSize	Number of samples selected per training iteration	128
γ	Discount factor for cumulative reward	0.9
a_{critic}	Critic network learning rate	0.001

Parameter	Description	Value
ϵ	Greedy policy parameter	0.0005
Episodes	Total number of simulation episodes	20000
$step_{max}$	Maximum number of iterations	1500

As shown in Fig. 7, after 20,000 episodes in fixed scenarios, the model converged, with rewards stabilizing as the agent learned the expected policy. The average driving speed increased from 1 m/s to 3.5 m/s, approaching the global target speed, while the average travel distance improved from 90 m to 330 m, aligning with the desired trajectory length.

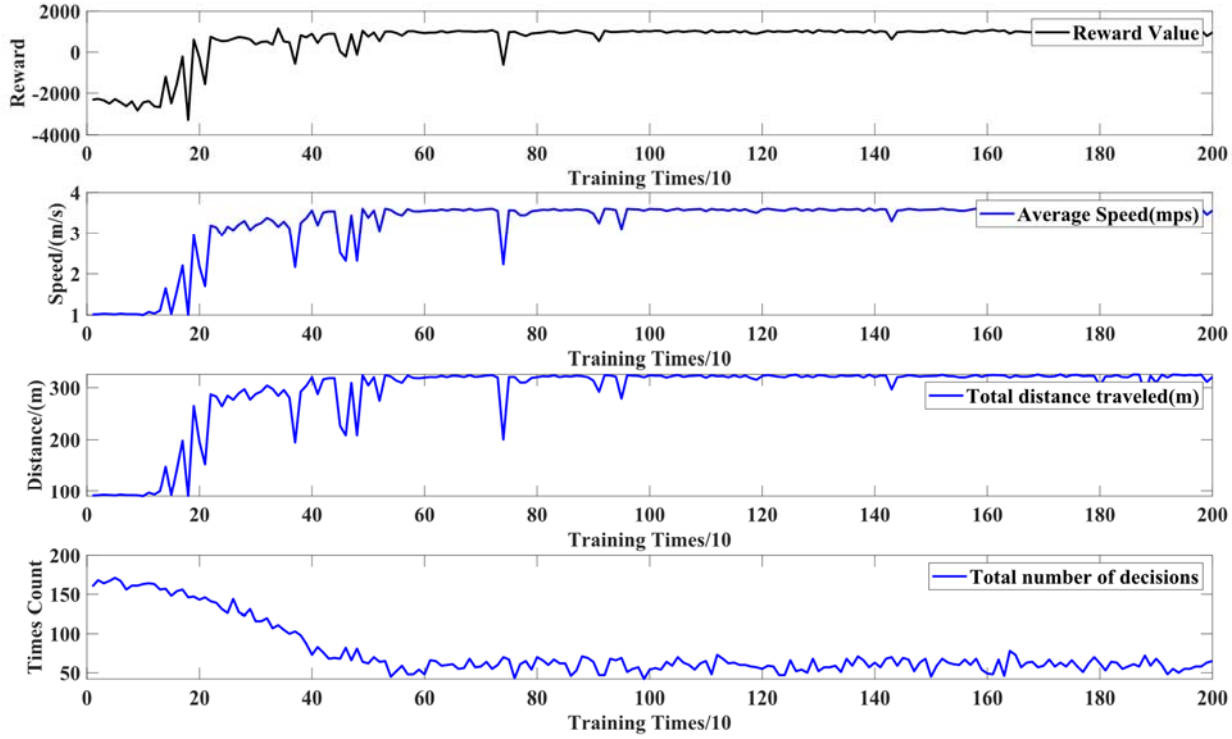

FIGURE 7 REWARD TRAINING RESULTS

4 EXPERIMENTAL TESTING AND ANALYSIS

Following practical operation workflows, both Model-in-the-Loop (MIL) and Hardware-in-the-Loop (HIL) tests were conducted under identical conditions to compare rule-based and DQN-based decision-making methods.

4.1 MIL TESTING AND ANALYSIS

As shown in Fig. 8, in the simulated mining road scenario, the rule-based method behaves conservatively in complex traffic. When encountering a slower lead vehicle, it tends to follow for extended periods without overtaking, resulting in lower average

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

speed and prolonged task completion. In contrast, the DQN-based method autonomously decides whether to overtake or cruise, significantly improving traffic efficiency and task completion time while ensuring safety.

Table 2 indicates that both methods perform comparably in terms of comfort. However, the DQN approach achieves higher efficiency by completing tasks faster. Although the frequency of acceleration, deceleration, and yaw rate exceeding thresholds is similar, the DQN method dynamically balances efficiency and comfort through real-time perception and learning, demonstrating superior intelligence and adaptability.

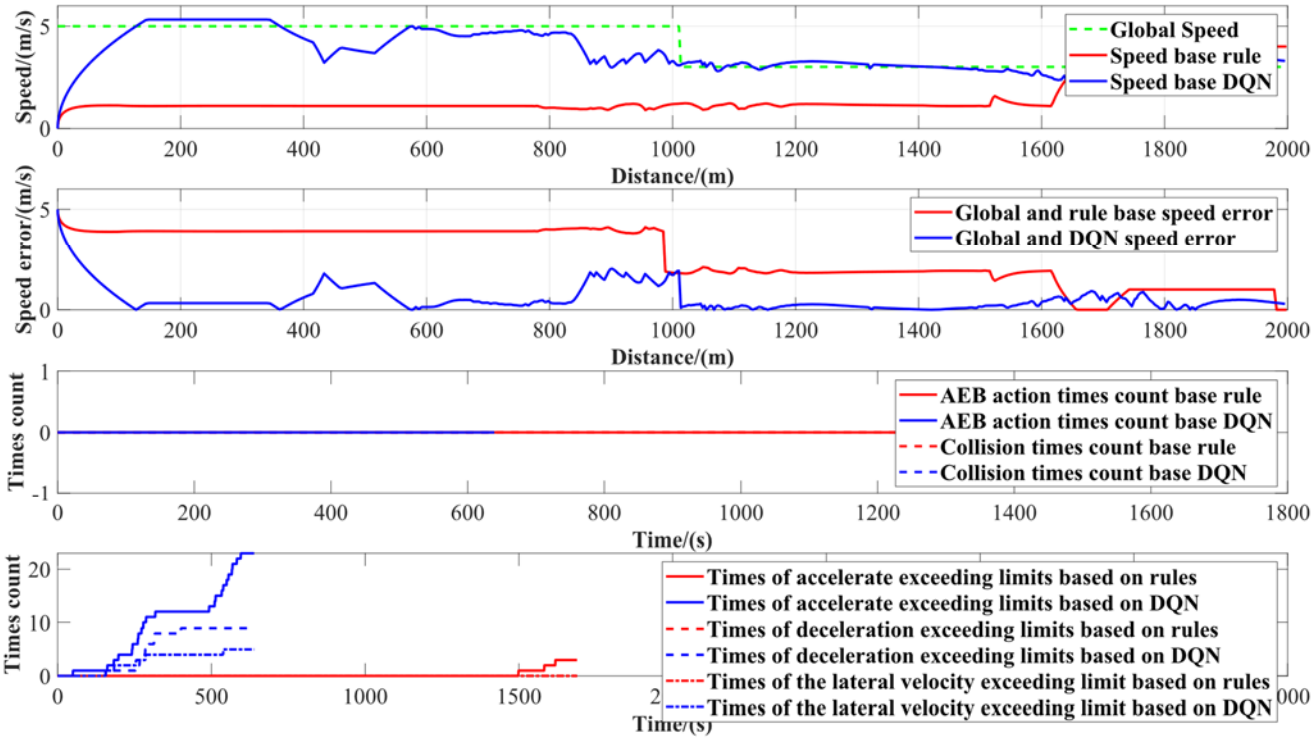

FIGURE 8 VEHICLE DRIVING DATA COMPARISON

TABLE 2 COMPARISON OF EVALUATION METRICS

Metric Category	Metric Name	Rule-based Decision-making Method	DDPG-based Decision-making Method
Driving efficiency	Time to complete the task (s)	245	122
	Average speed (m/s)	1.46	3.47
Comfort level	Number of times acceleration exceeded the threshold	28	32
	Number of times deceleration exceeded the threshold	25	23

4.2 HIL TESTING AND ANALYSIS

To further validate the feasibility and effectiveness of the proposed DQN-based decision-making method, Hardware-in-the-Loop (HIL) tests were conducted. The setup included a central domain controller, real-time simulator, sensor suite, and mining scenario simulation platform (Fig. 9).

The domain controller provides 254 TOPS + 230 DMIPS AI computing power, supports redundant multi-sensor fusion (8 cameras, 3 LiDARs, and 10 radar/ultrasonic sensors), and integrates 5G-BOX, RTK, and IMU modules. It is designed for harsh mining environments with ASIL-D safety, IP67 protection, wide voltage (9–36V), and operating range from −40°C to 85°C, supporting L2–L4 autonomous functions.

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

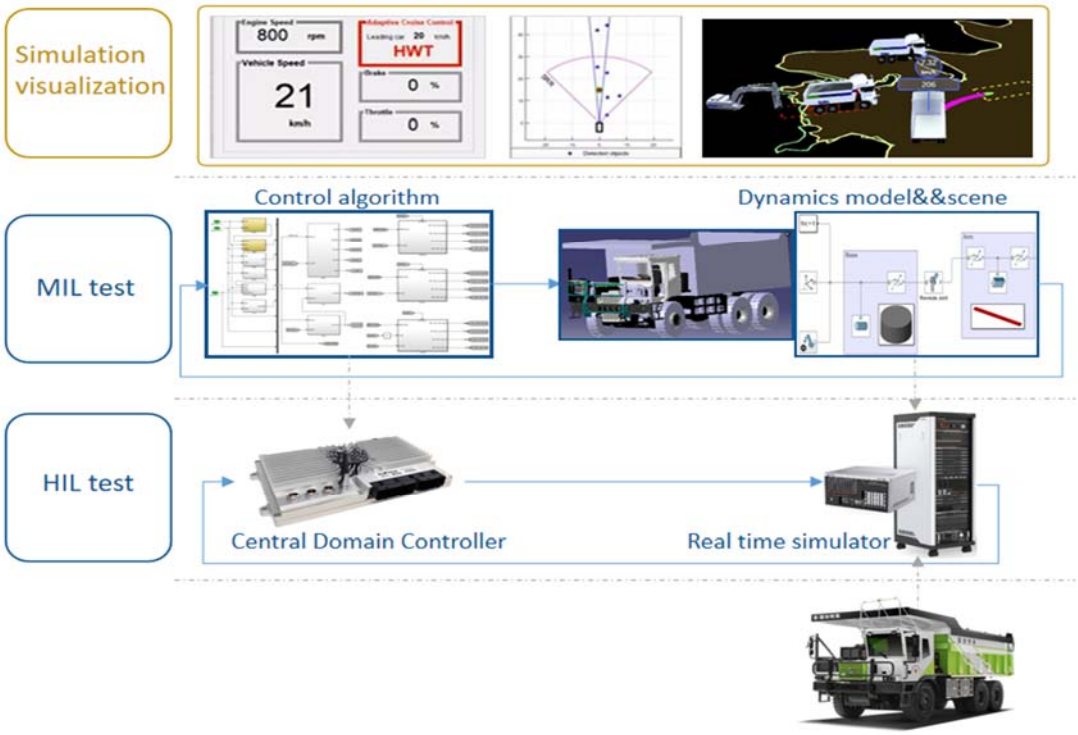

FIGURE 9 HIL TEST BENCH ARCHITECTURE

4.3 HIL TESTING PROCEDURE AND RESULTS

In the HIL tests, scenarios defined in MIL were migrated to the real-time simulator, covering diverse mining conditions such as slopes, intersections, and multi-vehicle traffic.

Curved road scenarios (Fig. 10): Rule-based methods tended to follow conservatively when facing multiple obstacles. In contrast, the DQN method, leveraging curvature, slope, and obstacle distribution, learned optimal strategies to execute safe overtaking.

Muddy slope scenarios (Fig. 11): Rule-based methods struggled to define safe overtaking boundaries, often resorting to low-speed following or stopping. The DQN method, trained on extensive experience including failed overtakes, was able to identify safe margins and complete overtaking effectively.

Overall, rule-based methods showed rigidity and delays in dynamic environments. The DQN policy, however, maintained safety and ride comfort while enabling adaptive and efficient decision-making, meeting practical mining operation requirements.

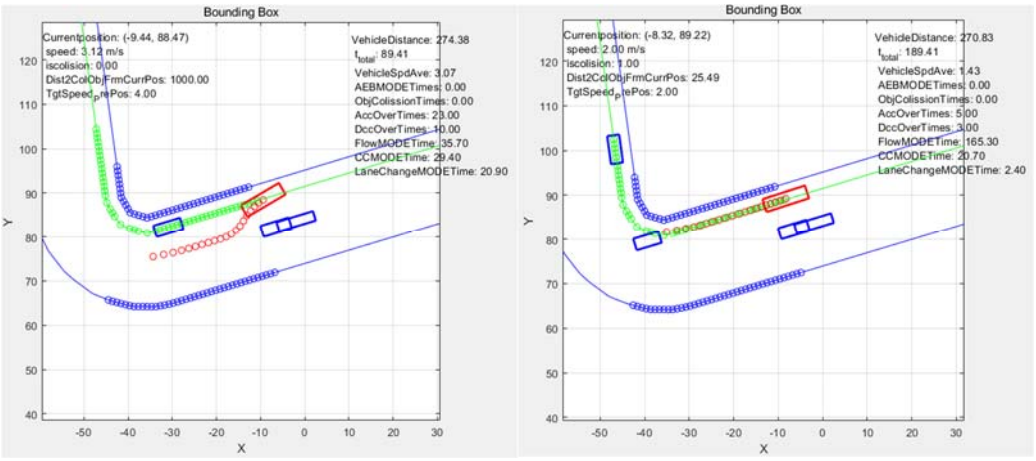

FIGURE 10 OBSTACLE AVOIDANCE SCENARIO ON A CURVED ROAD

Journal of http://www.satursonpublishing.com/
Research in Multidisciplinary Methods and Applications
ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

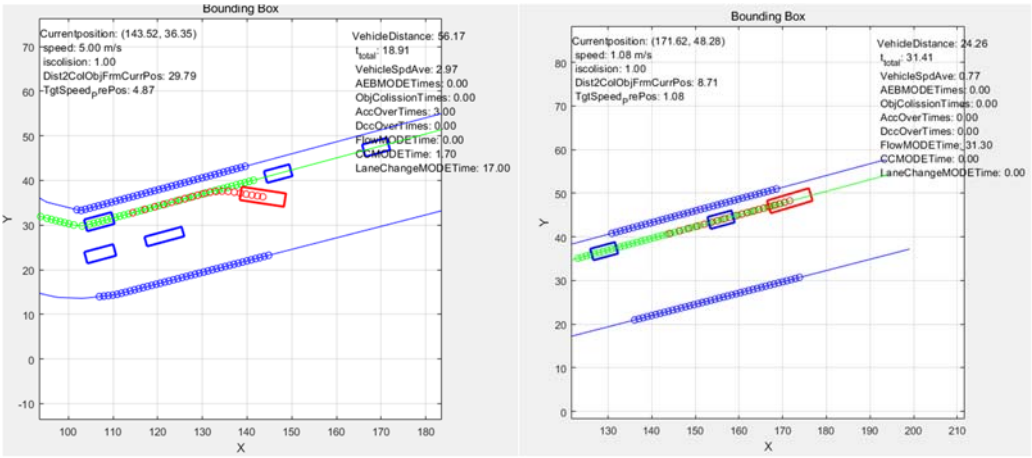

FIGURE 11 STRAIGHT, GENTLY SLOPING, MUDDY TERRAIN SCENE

5 CONCLUSION

This paper proposed a Deep Q-Network (DQN)-based decision-making and planning method to address complex driving challenges in unstructured road environments. By carefully defining the state-action space and designing a reward function that balances efficiency, economy, safety, and comfort, comprehensive experiments were conducted in simulated mining scenarios. Results from both MIL and HIL tests consistently demonstrated that the proposed method outperforms rule-based approaches in improving operational efficiency, reducing fuel consumption, enhancing safety, and ensuring ride comfort.

Looking ahead, with the growing demand for autonomous driving in mining and other special scenarios, the proposed approach shows strong potential for real-world deployment. Future work will further explore broader applications of deep reinforcement learning in decision-making and planning, contributing to improved performance and reliability of autonomous vehicles.

REFERENCES

[1] Xiong Lu, Kang Yuchen, Zhang Peizhi, et al. Research on the Behavior Decision-making System of Unmanned Vehicles [J]. Automotive Technology, 2018(8): 9..

[2] Hubmann C, Becker M, Althoff D, et al. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles[C]//2017 IEEE intelligent vehicles symposium (IV). IEEE, 2017: 1671-1678.

[3] Yurtsever E, Lambert J, Carballo A, et al. A survey of autonomous driving: Common practices and emerging technologies[J]. IEEE access, 2020, 8: 58443-58469.

[4] Wang Xiaoyuan, Yang Xinyue. Research on Driving Behavior Decision Mechanism Based on Decision Tree [J]. Journal of System Simulation, 2008, 20(2): 6.

[5] Aksjonov A, Kyrki V. Rule-based decision-making system for autonomous vehicles at intersections with mixed traffic environment[C]//2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021: 660-666.

[6] Bouchard F, Sedwards S, Czarnecki K. A rule-based behaviour planner for autonomous driving[C]//International Joint Conference on Rules and Reasoning. Cham: Springer International Publishing, 2022: 263-279.

[7] Xu C, Zhao W, Liu J, et al. An integrated decision-making framework for highway autonomous driving using combined learning and rule-based algorithm[J]. IEEE Transactions on Vehicular Technology, 2022, 71(4): 3621-3632.

[8] Talebpour A, Mahmassani H S, Hamdar S H. Modeling lane-changing behavior in a connected environment: A game theory approach[J]. Transportation Research Procedia, 2015, 7: 420-440.

[9] Aksjonov A, Kyrki V. Rule-based decision-making system for autonomous vehicles at intersections with mixed traffic environment[C]//2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021: 660-666.

[10] Hwang S, Lee K, Jeon H, et al. Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(10): 17594-17606.

[11] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey[J]. IEEE transactions on intelligent transportation systems, 2021, 23(6): 4909-4926.

[12] Xia Wei, Li Huiyun. Autonomous Driving Strategy Learning Method Based on Deep Reinforcement Learning [J]. Ensemble Technology, 2017, 6(3): 12.

[13] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: A brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

[14] Wu J, Yang H, Yang L, et al. Human-guided deep reinforcement learning for optimal decision making of autonomous vehicles[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2024.

[15] Yuan M, Shan J, Mi K. Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles[J]. IEEE Robotics and Automation Letters, 2021, 7(2): 818-825.

[16] Liao J, Liu T, Tang X, et al. Decision-making strategy on highway for autonomous vehicles using deep reinforcement learning[J]. IEEE Access, 2020, 8: 177804-177814.

[17] Shi Bowen. Research on humanized autonomous driving behavior decision-making based on deep reinforcement learning[D]. Jilin University, 2022.

ISSN 2957-3920 (Online) ISSN 3007-7060 (Print)
Volume 4, Issue 9, September 2025

[18] Wang Tinghan, Luo Yugong, Liu Jinxin, et al. End-to-end autonomous driving strategy based on deep deterministic policy gradient algorithm considering state distribution[J]. Journal of Tsinghua University: Science and Technology, 2021.