Journal of Computers Vol. 36 No. 3, June 2025, pp. 191-207
https://doi.org/10.63367/19911599202506360301 3

Behavior-Aware Hierarchical Decision Strategy for High-Speed Autonomous Driving

Tao Ma[1,2*], Qian-Han Zhang[3], Bing-Yan Wei[3], and Shou-Xin Li[1]

[1] Hebei Institute of Mechanical and Electrical Technology, Automotive Engineering Department, Xingtai City 054000, Hebei Province, China

{taomas0315, qianhan19881314, wby83765, shouxin88986}@163.com

[2] Xingtai Technology Innovation Centre for New Energy Vehicle Lightweight, Xingtai City 054000, Hebei Province, China

[3] Hebei Institute of Mechanical and Electrical Technology, Electrical Engineering Department, Xingtai City 054000, Hebei Province, China

Received 21 May 2025; Revised 4 June 2025; Accepted 9 June 2025

Abstract. This paper introduces a method for driving style modeling and classification using a GMM model to capture the relationships among various features derived from driving data collection. Subsequently, principal component analysis is applied to reduce the dimensionality of driving features, a Bayesian approach is taken to develop a statistical probability model for driver characteristics, and an analysis of the model's statistical features is conducted. The Gibbs sampling algorithm is employed to estimate the parameters of the Gaussian model, uncovering similar characteristics within the data and categorizing driver intentions. Using the driving model, high-speed following is simulated. By optimizing the decision-making and control layers, the algorithm's efficiency is enhanced and the dimensionality of its parameters is reduced. The feasibility of the intelligent driving decision strategy developed in this study was validated using the Highway-env simulation platform.

Keywords: autonomous driving, reinforcement, model lightweighting, driving model

1 Introduction

With the maturity of 5G network and Internet of Things technologies, autonomous driving has emerged as a research hotspot in the current automotive technology scene. Autonomous vehicles can navigate to destinations independently according to passenger needs. This technology's superiority is evident in its ability to significantly reduce operational costs for vehicles in service. Additionally, since on-board control centers execute intelligent algorithms, autonomous vehicles do not engage in dangerous behaviors such as fatigue driving, thereby enhancing driving safety and reducing the incidence of traffic accidents. As the heart of autonomous driving, the decision-making and planning system is responsible for interactions with other traffic participants, selecting the optimal course of action to ensure driving tasks are completed safely [1].

The primary methods for enhancing autonomous driving technology typically involve enhancing the accuracy of driver modeling and optimizing deep reinforcement learning models. Regarding driver modeling, Soichiro Matsumoto of Hiroshima City University suggested incorporating the deviation of the front-wheel steering angle at the vehicle's center of gravity into existing models and adaptively estimating this deviation to refine the model [2].

A study conducted by the University of Washington also focused on the yielding behavior of cars in front of pedestrian crossings, inferring driving behavior by assessing drivers' deceleration intentions and using rectangular rapid flashing beacons (RRFBs) [3]. Yixin Zhu from Tongji University classified distracted car following behavior into five categories through the collection of driver behavior feature samples, and established a distracted car following behavior model for drivers during the driving process [4]. Haobin Jiang from Jiangsu University focuses on the turning behavior of curves in driving models and proposes a humanoid driver model that includes a visual perception module and a turning decision module. Analyzing existing driving behavior models, each

* Corresponding Author

model has its own unique application scenario, but driving is an advanced animal behavior action. Therefore, the establishment of driving models should be closer to real human behavior, and driving behavior should be more diverse [5].

In vehicle intelligent decision-making, there are currently two primary categories of decision-making methods: rule-based and learning-based. Rule-based decision-making methods are highly logical, with outcomes derived from a library of driver behavior rules compiled from both driving experience and traffic regulations [6]. During vehicle operation, logical assessments are made to select specific actions based on the surrounding environment. Typical rule-based models include finite state machines and decision trees. The finite state machine consists of various driving states and the transitions between them. Changes in the vehicle and environment trigger state transitions, resulting in the immediate generation of driving actions. Early autonomous vehicles, such as Stanford University's Junior autonomous car [7], Talos autonomous vehicle, etc [8]. In China, Nanjing University of Aeronautics and Astronautics employs decision tree classification algorithms to categorize human driving behaviors, subsequently utilizing a Bayesian network model to determine optimal driving actions based on the environment [9].

After the rapid advancement of artificial intelligence technology, learning based driving decision-making techniques have received increasing attention. The decision-making process originates from learning and selecting human driving data, forming a driving rule library of "scene features driving actions" through driving experience and traffic rules. Based on the actions in the current environment matching library, similar rules are matched according to the current scene to obtain driving actions [10].

In the learning based driving decision framework, deep reinforcement learning strategies combine the advantages of perception and decision-making. Mnih et al. proposed an Asynchronous Advantage Actor Critic (A3C) based on DQN in 2015, which enables multiple agents to explore asynchronously [11]. Yu et al. used a heuristic oriented reinforcement learning (RL) method to handle highway overtaking scenarios [12] and generate real-world decision strategies. They combined Dyna-Q with variable heuristic strategies to propose a special Dyna-H algorithm for highway overtaking decision-making problems. Three heuristic programming functions were constructed in the traditional RL method for autonomous vehicles to accelerate the learning process [13].

In China, Yinxiao Zhan proposed a behavior decision model that combines deep reinforcement learning and risk correction methods to improve attention to potentially dangerous vehicles and the ability to avoid dangerous driving decisions [14]. Tian Feng from Nanjing University of Science and Technology proposed a deep reinforcement learning autonomous driving decision-making method based on heterogeneous fusion features. The method effectively guides learning through a reward function tailored for autonomous driving that integrates information such as speed, steering wheel angle, vehicle position, and collision. It combines experience pool replay technology and target network technology to improve training convergence speed [15].

Based on the summary provided, this paper initially develops a sophisticated driving model by acquiring extensive data on driving styles via a dedicated driving experiment platform. When classifying driving styles, most existing studies rely on statistical features from natural driving data, which merely quantify the magnitude of individual driving behaviors without capturing their interrelationships. This paper employs a GMM model for driving style modeling and classification. Subsequently, principal component analysis is applied to reduce the dimensionality of driving features, and Bayesian methods are utilized to construct a statistical probability model for driver characteristics. The model's statistical features are then analyzed. The Gibbs sampling algorithm is employed to estimate parameters for the Gaussian model, uncovering similar characteristics within the data. Lastly, the K-means algorithm is applied to categorize driving intentions.

Building an accurate driving behavior model and improving the accuracy of driving decision models are the focus of this study. However, the current research scenarios for driving behavior models are either single driving scenarios or based on the driver's own driving actions as the research object. However, driving is an advanced animal behavior action that is influenced by road conditions, vehicle conditions, and one's own behavioral habits. Therefore, the establishment of driving models should comprehensively consider driving factors, approach real human behavior, and classify driving behavior in more detail. In terms of driving decision models, the useless search and disaster dimensions of algorithms, as well as the number of parameters in algorithms, result in high computational and resource requirements. Therefore, it is necessary for algorithms to lightweight decision-making in intelligent driving scenarios.

Therefore, the main work of this article is as follows:

1) Firstly, build a simulated experimental platform to obtain sufficient driving style data from drivers, and then perform statistical analysis on the driving data;

2) Due to the interaction between driving environment, driving conditions, and driver behavior, a driving style modeling and classification method based on GMM model is proposed to improve the accuracy of driver feature

classification;

3) After dimensionality reduction, the driving feature model is sampled and classified using Gibbs sampling algorithm to solve the parameters of the Gaussian model, mine the similar characteristics in the data, and use K-means algorithm to complete the classification of driving intentions;

4) For the lightweight processing of driving decision models, a two-layer reinforcement learning framework is used to replace the decision layer and control layer in autonomous driving tasks, in order to reduce the complexity of reinforcement learning in solving problems.

Section 2 focuses on the establishment of the driver model, where a high-fidelity simulated driving platform is built using Unreal Engine 5, CarSim, and real-time control hardware to collect behavioral data. The platform integrates DQN and SAC algorithms, as well as long short-term memory (LSTM) and self-attention mechanisms, aiming to improve decision-making capabilities under high-speed following conditions. A two-layer decision-control architecture is adopted to separate macro-behavioral decisions from fine-grained control tasks. Section 4 details the experimental platform and analysis, which uses the Highway-env simulator to create a controlled high-speed driving environment and test the proposed model. Comparative experiments show that the safety rate, average speed, and number of unnecessary lane changes are significantly improved after applying the improved decision-making algorithm. Section 5 summarizes and looks forward to future work, highlighting the success of the integrated GMM-reinforcement learning framework while pointing out its limitations, such as the need for multi-vehicle coordination and practical deployment challenges. It outlines future research directions, including multi-agent learning, transfer learning, and unified algorithm integration with dynamic input weighting to improve performance in complex and unstructured environments.

2 Establishment of Driver Model

Table 1 provides the full configuration of the simulation system used in the driving model, or both software and hardware used to create a high-fidelity virtual driving environment. The system contains an Intel Core i7 14700K CPU and 32 GB of RAM, both able to provide robust computational capabilities for the real-time data processing and rendering. It utilizes a NVIDIA RX4060 TI graphics card to handle the demanding graphical renderings for the scene renderings in high fidelity. In terms of driver input, the configuration includes a Logitech G923 force-feedback steering wheel and a Playseat driving simulator seat which can add to the realistic input and feedback from any driving input. The simulation utilizes the Windows 11 operating system for stability and compatibility with a wide array of software. The software that supports simulated driving sequences includes several other software products (3Ds Max 2022, Autodesk Maya 2021, and CarSim 2021) for vehicle dynamics modeling, visual rendering of the driving scenes (Unreal Engine 5), along with a VC++ 10.0 environment for development and integration of the simulation. The Logitech Games API supports the real-time processing of the driver control signals and RAW driver input interactions with the hardware based simulation. Overall the simulated driving environment supports similar levels of real-driving realism, as well as erring on the side of efficiency and collecting and format.

Table 1. List of driving model software and hardware

Name	Parameter information
CPU	Intel Core i7 processor 14700K
RAM	32G
Graphics card	NVIDIA RX4060 Ti
Simulated steering wheel	Logitech G923
Simulated driving seat for automobiles	Playseat
Operating system	Windows 11
3D Modeling software	3Ds Max 2022\Autodesk Maya2021\ Carsim2021
Scene rendering software	Unreal Engine 5
Software Development Interface	VC++ 10.0\ Logitech Games API

The simulated driving scenario established is shown in Fig. 1.

Fig. 1. Simulated driving scenario

Set vehicle driving conditions: acceleration condition, the preceding vehicle accelerates at 0 m/s^2-20 m/s^2-40 m/s^2-60 m/s^2-80 m/s^2-100 m/s^2, maintaining a constant acceleration of 1 m/s^2; Deceleration condition: The preceding vehicle decelerates at a speed of 100 m/s^2-80 m/s^2-60 m/s^2-40 m/s^2-20 m/s^2-0 m/s^2. The experimental process is similar to the acceleration condition; Uniform speed condition, 80 m/s^2. The process of collecting simulated driving scenarios is shown in Fig. 2.

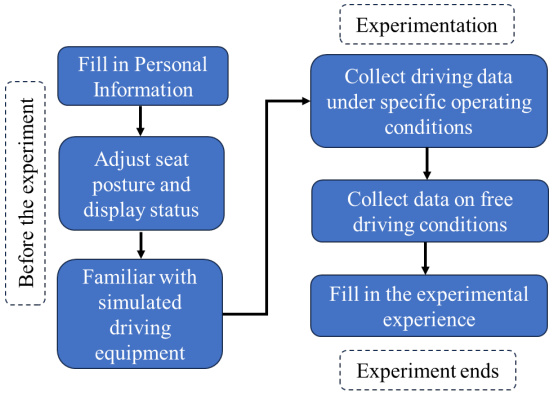

Fig. 2. Simulated driving scenario

2.1 Pre Processing of Driving Data

During vehicle operation, due to sensor acquisition errors and minor driver inputs, there may be some low-value speed signals, typically below 30 km/h. Such anomalous data, being both low in speed and brief in duration, needs to be eliminated. This paper employs the Pareto criterion for detecting and pruning outliers. Assuming the collected sample data is the mean $\overline{x_i}$ of X_n (n = 1, 2, 3 ..., i) and the residual error is $C_n = x_n - \overline{x_i}$, the standard deviation σ_n is calculated, and values exceeding $3\sigma_n$ are considered outliers and discarded [16]. The filtering method is described as follows:

$$\left| x_n - \overline{x_i} \right| > 3\sigma \tag{1}$$

The data results after outlier processing are shown in Fig. 3.

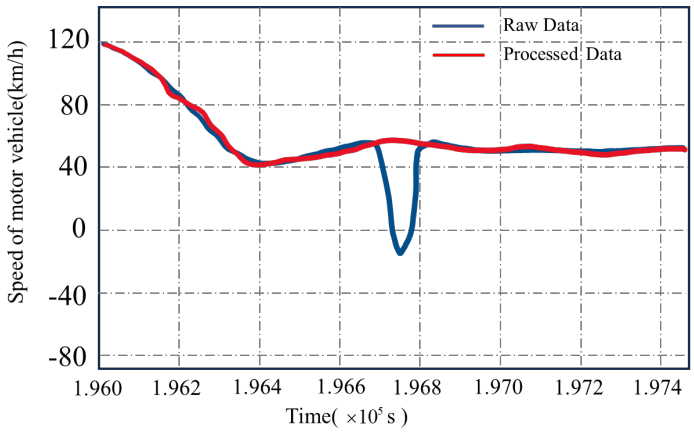

Fig. 3. Processed curve

2.2 Driving Feature Modeling and Classification

Driving feature classification based on data statistics reflects features solely through data volume, yet it is equally important to express the correlations among driving features. This section initially establishes a Gaussian distribution function for high-speed driving conditions and subsequently employs the Gibbs method [17] for sampling to extract the condition's features.

During high-speed driving, maintaining a following distance from the preceding vehicle is a typical condition. The driving condition of a car involves various parameters to sustain this condition, including throttle opening, rate of throttle change, and brake pressure, all of which indicate the driver's current intent. Additionally, the main vehicle's speed and following distance reflect the driver's style [18]. Thus, the parameters for the entire high-speed following process include:

1) The relative distance between the main vehicle and the preceding vehicle;
2) The relative speed between the main vehicle and the preceding vehicle;
3) Main vehicle speed;
4) Main vehicle acceleration;
5) Rate of throttle change;
6) Brake oil pressure.

Hence, the eigenvector representation is as follows:

$$f = \left[\Delta x, \Delta v, v_{main}, a_{main}, o_{\text{throttle}}, p_{oil}\right] \tag{2}$$

In the formula, Δx represents the relative distance between the main vehicle and the preceding vehicle, Δv represents the relative velocity between the main vehicle and the preceding vehicle, v_{main} represents the speed of the main vehicle, a_{main} represents the acceleration of the main vehicle, $o_{throttle}$ represents the rate of change in the throttle opening of the main vehicle, and p_{oil} represents the brake oil pressure. Before establishing a Gaussian distribution, calculate the correlation between various parameters. When the correlation between parameters is low, reduce the dimensionality of parameters to improve the speed of model solving. The formula for calculating the correlation coefficient is as follows:

$$c_{\alpha,\beta} = \frac{\operatorname{cov}(\alpha,\beta)}{\sigma_\alpha \sigma_\beta} = \frac{E\left[(\alpha-\mu_\alpha)(\beta-\mu_\beta)\right]}{\sigma_\alpha \sigma_\beta} \tag{3}$$

After assessing the correlation among the six parameters, which is not high, principal component analysis can be employed to reduce the dimensionality of multidimensional data, thus enhancing the model's generalization capability. According to the steps of principal component analysis, there are 6 parameter samples in dataset *f*, and 3 variables are observed for each parameter sample. The dataset matrix is as follows:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ \vdots & \vdots & \vdots \\ f_{61} & f_{62} & f_{63} \end{bmatrix} \quad (4)$$

Define $f_i = \left(f_{1i}, f_{2i}, \cdots, f_{6,i}\right)$, i = 1, 2, 3 and combine the original individual observed variables f_1, f_2 and f_3 to form three new variables, represented as follows:

$$\begin{cases} A_1 = \sum_{i=1}^{6} \omega_i \cdot f_{i1} \\ A_2 = \sum_{i=1}^{6} \omega_i \cdot f_{i2} \\ A_1 = \sum_{i=1}^{6} \omega_i \cdot f_{i3} \end{cases} \quad (5)$$

$$\omega_i = \sum_{j=1}^{3} \frac{f_{ij}}{\sum_{k=1}^{6}\sum_{l=1}^{3} f_{kl}}, i = 1, 2, \cdots, 6 \quad (6)$$

In the formula, ω_i represents the weight, and the variance contribution rate of principal components can be used to calculate the feature expression rate of $n(n \in [1,6])$ -dimensional features. The expression is as follows:

$$\beta_i = \frac{\lambda_i}{\sum \lambda_i}, i = 1, 2, 3 \quad (7)$$

λ_i is the variance, β_i is the variance contribution rate. Upon calculation, with a feature dimension of 3, 98.91% of the features can be represented, indicating an effective representation of actual driving data. Consequently, computations can be performed with the features reduced to 3 dimensions. Establish a statistical Gaussian distribution model based on the above six parameters.

$$GD_p\left(f|\alpha\right) = \sum_{i=1}^{m} \phi_i g\left(f|\alpha_i\right) \quad (8)$$

Among them, *m* represents the number of Gaussian distributions, ϕ_i represents the weight of Gaussian distributions [19], the sum of each weight is 1, $\alpha_i = \left(M_{u_i}, M_{\sigma_i^2}\right)$, where M_{u_i} is the mean matrix of Gaussian distribution, $M_{\sigma_i^2}$ is the covariance matrix, and $g\left(f|\alpha_i\right)$ is the *i*-th sub Gaussian model. Use Gibbs algorithm to sample Gaussian distribution model, and the sampling process is shown in Fig. 4:

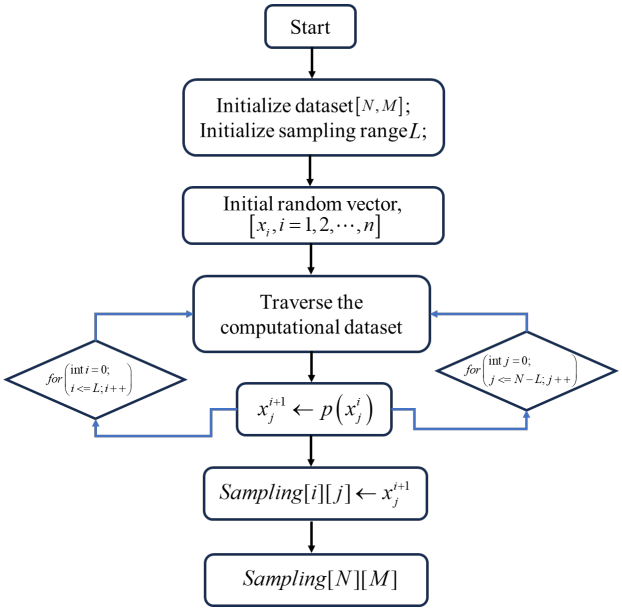

Fig. 4. Processed curve

After sampling, this article obtained the initialization values of the Gaussian mixture model with 5 Gaussian elements. In the context of probabilistic clustering, Gaussian mixture models (GMMs) are widely used due to their flexibility in modeling complex data distributions. GMMs assume that data is generated by a mixture of multiple Gaussian distributions, each of which is characterized by its own parameters: mean, covariance, and a mixing coefficient that represents its prior probability in the overall model. Proper initialization of these parameters is crucial for the convergence and performance of the expectation-maximization (EM) algorithm, which is often used to iteratively estimate the parameters. In this study, the authors initialized the GMM using five Gaussian components, as shown in Table 2. The table lists the initial values of the prior probability (mixing coefficient) assigned to each Gaussian component based on the sampling strategy applied to the dataset. These values are as follows: 0.0452 for component 1, 0.3569 for component 2, 0.2087 for component 3, 0.2104 for component 4, and 0.1788 for component 5. These values are normalized to sum to 1 to conform to the probabilistic nature of GMMs. It is noteworthy that the second Gaussian component is assigned the highest initial weight (0.3569), indicating that it may represent the most dominant cluster in the initial data distribution. In contrast, the first component has the lowest initial weight (0.0452), indicating that it may correspond to a smaller or less dense cluster. The weights of the remaining three components are relatively similar, ranging from 0.18 to 0.21, which means that these clusters are more balanced. These initial values are critical because they affect the early steps of the EM algorithm and may affect the final convergence and the quality of the clustering solution. Inappropriate initialization may lead to suboptimal convergence results or the algorithm being stuck in a local optimum. Therefore, as shown in Table 2, careful allocation of initial mixing coefficients is essential to obtain reliable and interpretable GMM results. The initialization strategy adopted in this paper can incorporate the structural information of the data and ensure that the prior probabilities can reflect meaningful preliminary groupings within the dataset. In summary, these initialization values are the basis for further optimization and improvement during iterative model training.

Table 2. Prior distribution parameters initialization for GMM

i	ϕ_i	μ_i	σ_i
1	0.0452	[18.7907 12.0891]	$\begin{bmatrix} 42.8728 & -6.0982 \\ -5.9569 & 26.0927 \end{bmatrix}$
2	0.3569	[−11.8978 −3.0673]	$\begin{bmatrix} 312.0091 & -5.4346 \\ -6.4929 & 2.3718 \end{bmatrix}$

3	0.2087	[22.0076 −1.9892]	$\begin{bmatrix} 439.6503 & -36.8027 \\ -45.5096 & 11.6737 \end{bmatrix}$
4	0.2104	[26.2104 1.9509]	$\begin{bmatrix} 90.1624 & -4.1284 \\ -4.1579 & 2.7627 \end{bmatrix}$
5	0.1788	[18.7763 1.9074]	$\begin{bmatrix} 242.0304 & 49.8902 \\ 47.1127 & 24.1667 \end{bmatrix}$

The application of the parameters obtained via Gibbs sampling into the GMM, allowed for an accurate reconstruction of the distribution of driver characteristic data. This is further shown in Fig. 5 comparing the prior and posterior distributions, where the magenta lines denote the prior densities and the cyan lines denote the posterior densities after a Bayesian update, and the underlying histogram (shown with blue bars) shows the normalized empirical data distribution. It is evident that the posterior curves align well with the histogram through the primary peaks and the overall shape of the recorded data. This demonstrates that the GMM, once it has been updated and discovered the sampled parameters, learned the complex, multimodal characteristics of the driver behavior dataset. The posterior distributions are smoother and show a better fit to areas of high density compared to the priors, particularly in the central modes of the distribution. Overall, this indicates that Gibbs sampling was successful in fine-tuning parameter estimates of a GMM and reflects the ability of probabilistic inference to better describe real-world behavioral characteristics - attributable to objective refinement of the GMM from Gibbs sampling inference.

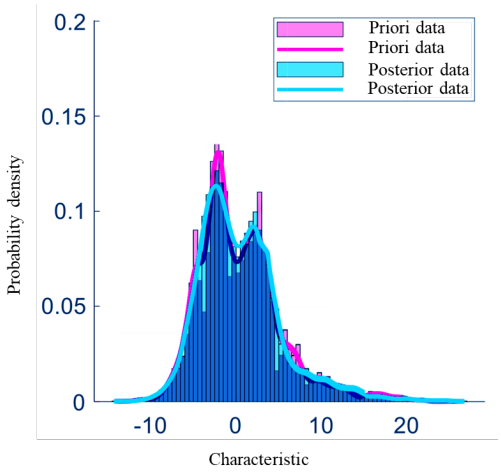

Fig. 5. Driver characteristic data distribution map

The posterior edge distribution of Gaussian mixture model can fit the original data sample well, qualitatively indicating that the established Gaussian mixture model can accurately describe the driver behavior characteristics data sample, thereby obtaining the driver's driving model and parameters.

3 Establishment of Driving Decision Strategy Model

Building on the driver feature model, this chapter finalizes the design of driving decision strategies. Utilizing driver behavior data, a model of driver behavior characteristics was developed to quantify driving behaviors, essentially analyzing the statistical properties of drivers' long-term habits. During the actual autonomous driving process, with its dynamic and real-time interactions, the data is vast and unpredictable. Consequently, this section develops a reinforcement learning framework, employing limited sample data to learn from driver decision-making behaviors, enabling vehicle driving decisions under high-speed following conditions. To enhance the decision-making efficiency of reinforcement learning models, the network incorporates long short-term memory units

and sub-attention mechanisms, and integrates rule-based decision strategies into the model. In order to improve the efficiency of decision making and facilitate better generalization of reinforcement learning models, the network structure incorporated long short-term memory units, sub-attention mechanisms, and rule-based decision strategies to improve modeling of human-like reasoning and adaptability in complex traffic environments.

3.1 Driving Environment Description

Based on the following driving conditions, ignoring the lateral motion of the vehicle, the longitudinal motion is approximated as an acceleration model, where the distance and relative velocity between the preceding vehicle and the main vehicle are represented as follows:

$$\begin{cases} R_s = s_{front} - s_{main} \\ R_v = v_{front} - v_{main} \end{cases} \tag{9}$$

In the formula, R_s represents the relative distance between the main vehicle and the preceding vehicle in the following behavior, R_v represents the relative velocity between the main vehicle and the preceding vehicle in the following behavior, s_{front} represents the preceding vehicle's travel, s_{main} represents the main vehicle's travel, v_{front} represents the preceding vehicle's speed, and v_{main} represents the main vehicle's speed. In the deep reinforcement learning framework, the main vehicle is the intelligent agent, and the environmental observation state of the intelligent agent is represented by the speed, relative distance, and relative velocity of the main vehicle [20]. The representation method is as follows:

$$S_{car} = \left[v_{main}, R_s, R_v \right] \tag{10}$$

S_{car} represents the status of the main vehicle, after observing and verifying the state, the intelligent agent selects the corresponding driving behavior, denoted as B, interacts with the environment, and finally returns a reward value R. In order to endow the model with the ability to extract complex environmental information similar to that of a human driver, the long short-term memory network and self-attention mechanism are fused with the DQN network [22], and the LSTM framework [23] and self-attention mechanism [21] are introduced to enable the intelligent agent to learn the long-term dependency relationship between environmental information and the ability to obtain key information in the environmental observation state more effectively. The improved model structure is shown in Fig. 6.

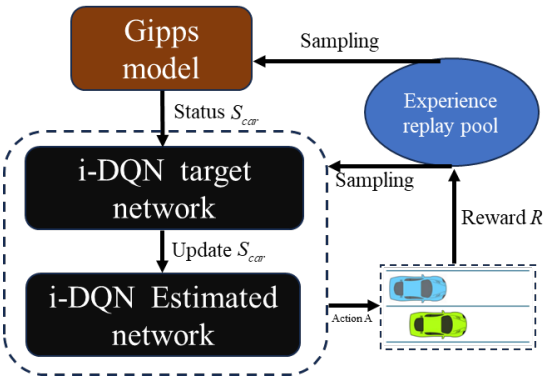

Fig. 6. Improved model structure

To enhance the efficiency of lane-changing tasks for highway vehicles, a structure composed of convolutional and fully connected layers is chosen as the model's foundational network. The input is transmitted through the convolutional layer to the self-attention layer, subsequently proceeding through the fully connected layer and

the long short-term memory network layer. The design of this integrated neural network structure aims to fully harness the capabilities of convolutional neural networks and long short-term memory networks, addressing the issue of gradient vanishing. The ReLU function [24] is employed as the activation function in the fully connected layer, enabling the network to more effectively capture nonlinear relationships during the learning process. The network structure is illustrated in Fig. 7.

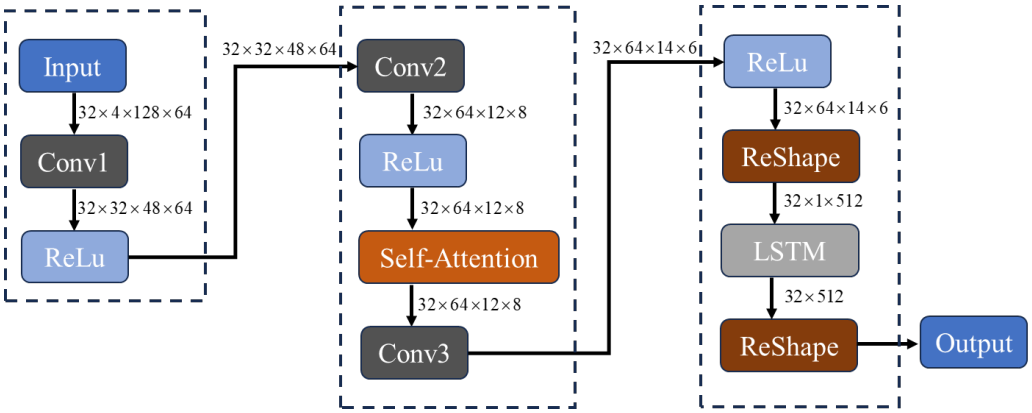

Fig. 7. The network structure of i-DQN

3.2 Design of Reward Function

The primary goal of applying reinforcement learning strategies is to maximize the reward function. In high-speed following scenarios, positive rewards arise from: collision avoidance, adherence to traffic rules, stability of following, legality of lane changes, and consistency of following distance [25]. These are represented sequentially:

1) No collision occurs

$$R_{non-collision} = \begin{cases} 1 & non-collision \\ 0 & otherwise \end{cases} \tag{11}$$

2) Comply with traffic rules

When driving at high speeds, it is most reasonable to drive in the right lane, and changing lanes to the left requires certain conditions. Therefore, in the process of high-speed autonomous driving, rewards are given for driving in the right lane and maintaining lane discipline, and no rewards are given for changing lanes to the left.

$$R_{rule} = \begin{cases} 1 & onright \\ 0 & otherwise \end{cases} \tag{12}$$

3) Stability of following the car during driving

$$R_{stable} = \begin{cases} -1 & v_{main} < 60km/h \\ v_{main}/60 - 1 & v_{main} \in [60,120km/h] \end{cases} \tag{13}$$

Among them, v_{main} represents the speed of the main vehicle.

4) Compliance of the lane change process.

When the intelligent agent needs to change lanes during driving, in order to encourage the agent to successfully change lanes without collision, the collision free successful lane change behavior is assigned a positive value. The reward function is defined as:

$$R_{change} = \begin{cases} 0.6 & no-collision \\ 0 & others \end{cases} \tag{14}$$

5) Stability of following distance

A stable following distance is essential for ensuring safe driving. Thus, it is imperative to maintain consistency in high-speed driving. Consequently, the reward function is as follows:

$$R_{con-dis} = \begin{cases} 1 & con-\text{distance} \\ -1 & others \end{cases} \tag{15}$$

After the above process, the overall reward function is established as follows:

$$R = \sum R_{non-collision} + R_{rule} + R_{stable} + R_{change} + R_{con-dis} \tag{16}$$

3.3 Construction of Deep Reinforcement Learning Models

Traditional reinforcement learning algorithms engage in much unnecessary exploration and fail to leverage prior knowledge, leading to poor learning efficiency and slow model convergence. This paper addresses the decision and control layers of reinforcement learning algorithms in autonomous driving processes, aiming to enhance the algorithm's efficiency and mitigate the curse of dimensionality. A two-layer reinforcement learning framework is employed to replace the decision and control layers in autonomous driving tasks. The decision layer employs reinforcement learning to learn macro behavioral decisions, while the control layer executes corresponding behaviors by altering the state space, thus simplifying the complexity of reinforcement learning problem-solving. The enhanced model framework is illustrated in Fig. 8.

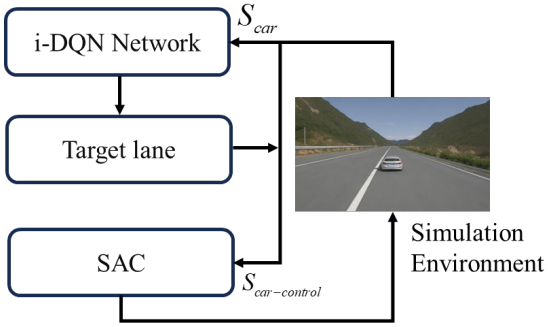

Fig. 8. Improved model framework

The decision-making layer is responsible for choosing whether to change lanes, using the DQN algorithm, which outputs three sub behaviors: lane keeping, left lane changing, and right lane changing. Therefore, its action space is defined as a three-dimensional hot encoding vector. The control layer is responsible for tracking the target trajectory and maintaining a safe distance from the preceding vehicle, using the Soft Action Critic Algorithm (SAC) algorithm [26]. Its output is a two-dimensional continuous variable $[a_1, sw_1] \in [-1,1]$ that controls the brake, throttle, and steering wheel, where a_1 is a positive value indicating the use of the accelerator pedal and a negative value indicating the use of the brake. A positive value of sw_1 indicates that the steering wheel rotates to the left, while a negative value indicates that the steering wheel rotates to the right.

The control layer and decision layer run at different time steps, and the execution frequency of the control layer is the same as the simulation frequency. After running a certain number of steps, the decision layer runs once. The decision-making layer outputs high-speed following instructions based on the environmental state S_{car}. When the decision-making layer does not execute or outputs lane keeping, the target lane remains unchanged,

and the decision-making layer outputs a left or right lane change to change the target lane. Then, combining the environmental state S_{car} with the trajectory point information on the target lane, the control layer state $S_{car-control}$ is generated, and the control layer tracks the target trajectory point to complete the high-speed following decision. The pseudocode of the algorithm is as follows:

SAC Algorithm
Input: Initialize strategy parameters, action value function, value function parameters, and experience replay set $[\theta, \phi_1, \psi, D]$;
$\psi_{t\arg} = \psi$;
for ($i = 0;\ i <= l;\ i++$)
End the loop after convergence;

4 Experimental Platform Construction and Experimental Analysis

Section IV outlines the construction and evaluation of the experimental platform used to validate the proposed autonomous driving decision-making algorithm. A simulated three-lane highway driving environment was created using the Highway-env platform, which contains detailed parameters such as vehicle size, sensor detection range, and control frequency. Each vehicle in the simulation is equipped with a lidar-based perception system and operates under the influence of a decision-making model that uses a proportional controller and a PD controller for longitudinal and lateral control, respectively. To replicate real driving behavior, this study integrates an intelligent driver model (IDM) and a MOBIL lane-changing model. This combination enables the vehicle to adjust speed and perform lane changes based on traffic dynamics and safety considerations. A large number of simulation runs were performed to ensure the robustness and reliability of the learning-based control strategy. The core performance indicator used is the safety rate, which evaluates the efficiency of the vehicle to maintain a safe distance and avoid collisions or emergency braking events. Comparative analysis shows that the combination of the IDM and MOBIL models significantly improves safety and stability compared to the baseline model. Further evaluation also shows that the improved algorithm achieves higher driving efficiency and stability by optimizing vehicle speed and reducing unnecessary lane changes. In summary, this section demonstrates that the proposed lightweight decision-making algorithm can improve the safety and practicality of autonomous driving in complex highway environments.

4.1 Simulation Environment Setup

This article uses the Highway-env [27] autonomous driving decision simulation environment to establish a three lane high-speed driving scenario, which includes the target vehicle using the autonomous decision-making system and the surrounding vehicles. Install a simulated LiDAR sensor in the target vehicle, which can perceive driving information around the target vehicle, including vehicles in the front left and right, front right, rear right, and rear left and right directions. The sensing range of the sensor is 150 meters. The simulation environment settings are shown in Table 3.

Table 3. Simulation environment settings

Parameter	Numerical value
Lane-width	3.75m
Vehicle length	4.9m
Vehicle width	1.82m
Maximum driving speed	120km/h
Sample time	1s
Sampling frequency	13Hz

4.2 Low-Level Controller Implementation

In the simulation environment, the five driving behaviors output by the behavior decision model are implemented by independent low-level continuous controllers. The longitudinal controller is a proportional controller, and its vehicle acceleration control quantity is shown in the following equation:

$$a_{main} = k_p \left(v_{main} - v_{t\arg et} \right) \tag{17}$$

In the formula, a_{main} is the vehicle acceleration controller quantity, v_{main} is the speed of the simulated experimental vehicle, v_{target} is the target speed of the vehicle, and k_p is the proportional gain of the controller. The lateral controller is a proportional derivative controller that combines nonlinear factors in the inverse kinematics model. Its position control calculation and heading control calculation are shown below.

$$\begin{cases} \Delta\lambda_h = \arcsin\left(\dfrac{v_h}{v_{main}} \right) \\ \lambda_h = d_{main} + \arcsin\left(\dfrac{v_h}{v_{main}} \right) \\ \dot{\lambda}_h = k_p \left(\lambda_h - \lambda \right) \\ \beta_{front} = \arcsin\left(\dfrac{s\dot{\lambda}_h}{2v_{main}} \right) \end{cases} \tag{18}$$

In the formula, $\Delta\lambda_h$ represents the heading change of the vehicle after applying the lateral speed command, v_h represents the target speed, v_{main} represents the main vehicle speed, d_{main} represents the heading of the lane line, λ_h represents the target heading of the vehicle following the lane line heading and position, $\dot{\lambda}_h$ represents the yaw rate command, and β_{front} represents the front wheel angle control quantity.

4.3 Application of IDM and MOBIL Models

To accurately model and simulate vehicle behaviors in high-speed traffic environments, this study adopts a hybrid approach that combines an intelligent driver model (IDM) and a minimized lane-change induced integral braking (MOBIL) model. IDM is used for longitudinal control, allowing the vehicle to adjust speed based on acceleration, deceleration, and headway; while MOBIL controls lateral control by making lane-changing decisions that balance individual interests and overall traffic flow stability. This integrated modeling approach is able to simulate more realistic and complex multi-vehicle interactions under high-speed conditions.

In the experimental setup, a virtual traffic scenario was constructed in which each autonomous vehicle operated under the joint influence of IDM and MOBIL. The simulation duration was fixed to 47 seconds to reflect a consistent observation window for all trials. To ensure sufficient model learning and stability, the simulation framework for each model configuration was executed 200,000 times to ensure that the vehicle control strategy can converge robustly under different traffic conditions and decision scenarios.

4.4 Safety Rate Evaluation and Model Comparison

The main performance metric to evaluate the effectiveness of the behavior modeling strategy is the safety rate of simulated vehicles. The safety rate is defined as the percentage of time steps in which all vehicles maintain a safe distance and avoid collision or emergency braking events. This metric directly reflects the success of the behavioral model in promoting safe and stable driving in complex environments.

Fig. 9 shows a comparative analysis of the vehicle safety rate before and after applying the IDM-MOBIL-based behavioral control. The solid blue line shows the trajectory of the safety rate under the baseline model (before), and the dashed blue line shows the performance after the integration of IDM and MOBIL (after). As shown in the figure, both curves show a downward trend in the first 20-30 iterations, indicating early learning

dynamics and model stability. However, the safety rate of the "after" curve is significantly improved and remains higher than the baseline for most of the simulation timeline.

Especially in the early stages of training (up to 30 iterations), the improved model has significantly reduced volatility and the safety rate remains above 70%, while the baseline model shows large volatility and average performance decreases. As training progresses, the safety rates of both models tend to stabilize, but the IDM-MOBIL integrated model continues to achieve better results, maintaining a margin of about 5-10 percentage points higher than the baseline model. This improvement shows that the use of IDM and MOBIL can enable vehicles to make smarter and more collaborative decisions in longitudinal and lateral movements, thereby improving overall safety in dense traffic simulations. In summary, the results in Fig. 9 validate the effectiveness of combining IDM and MOBIL for high-speed driving simulation. The integration of these models significantly improves the safety rate of simulated vehicles, confirming their suitability for realistic and risk-aware behavior planning in autonomous driving research.

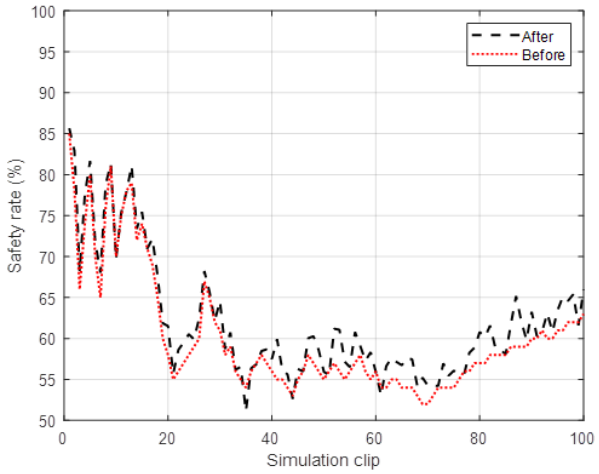

Fig. 9. Comparison of model security before and after improvement

4.5 Comparison of Algorithm Before and After Improvement

In order to evaluate the actual effect and efficiency of the proposed lightweight autonomous driving decision-making algorithm, a comprehensive comparison of the algorithms before and after the improvement was conducted. The evaluation focused on three key performance indicators: safety rate, average driving speed, and average number of lane changes. These indicators are crucial for evaluating the practical applicability of autonomous driving strategies, especially in high-speed traffic environments where safety and efficiency are crucial.

Fig. 10 shows the comparison results in the form of a bar chart. The safety rate (defined as the percentage of time steps in which all vehicles maintain a safe distance and avoid collision or emergency braking) is significantly improved from 92.67% of the original algorithm to 98.32% of the improved algorithm. The significant improvement of nearly 6 percentage points shows that the improved algorithm can effectively ensure safer driving behavior even in complex and dynamic traffic scenarios.

The improved algorithm also performs well in terms of average driving speed (a measure of the average speed of the autonomous vehicle throughout the simulation). The speed slightly increased from 30.29 m/s to 31.07 m/s, reflecting the system's ability to maintain a higher speed without compromising safety. This is particularly important in high-speed driving environments, where maintaining an optimal speed is critical to traffic flow and travel time efficiency.

The most significant improvement was seen in the improved method average number of lane changes, which dropped from 0.93 for the original algorithm to 0.37 for the improved algorithm. This reduction indicates that the enhanced algorithm significantly reduces unnecessary lane changes. Excessive lane changes not only disrupt traffic stability but also increase the risk of accidents. By minimizing such maneuvers, the improved model helps

provide a smoother, more stable, and safer driving experience. In addition, the reduction in the frequency of lane changes indicates that the vehicle is able to make more strategic and situationally aware decisions as it learns to weigh the benefits of lane changes against potential risks or minimal time gains.

Overall, these results indicate that the improved algorithm successfully balances the dual goals of safety and performance. It enables intelligent vehicles to make smarter and more effective decisions during the execution phase, thereby increasing driving speed and reducing disruptive or dangerous behaviors. Importantly, the algorithm does not sacrifice safety for speed or vice versa; instead, it optimizes both simultaneously. This fully demonstrates the advantages of the lightweight decision-making framework proposed in this study, as it maintains or even improves vehicle safety and traffic harmony while improving driving performance.

The results in Fig. 10 verify that the improved algorithm is very suitable for practical deployment in intelligent transportation systems and provides a powerful solution for autonomous driving in dense high-speed traffic environments.

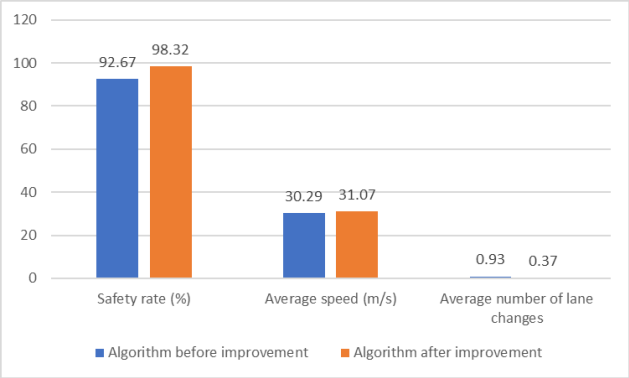

Fig. 10. Comparison before and after algorithm improvement

5 Conclusion

This paper initially establishes a data collection platform. Building on the collected data, the study introduces a method for driving style modeling and classification using the GMM model. Subsequently, principal component analysis is applied to reduce the dimensions of driving features, resulting in data with statistical characteristics only. To establish correlations among various driving models, the Gibbs sampling algorithm is employed to classify and solve the parameters of the Gaussian model, uncovering similar characteristics in the data and classifying drivers' intentions. For the driving decision model, the study enhances the DQN algorithm and applies the SAC algorithm to refine the existing reinforcement learning model. Finally, the feasibility of the intelligent driving decision strategy developed in this paper is validated using the Highway-env simulation platform.

Based on the above analysis, it can be seen that the optimization algorithm proposed in this paper can better meet the computational effects of end-to-end control strategies in the field of autonomous driving, especially in military material scheduling scenarios. However, there are also some areas worth further exploration in the research process:

1) The improved algorithm constructed in this study mainly targets a single vehicle control strategy, while in actual road operation, multiple vehicles need to be controlled simultaneously. If the effective combination of intelligent agent learning and autonomous driving scenarios can be achieved in complex residential environments, it is expected to continue to improve the development level of military automation technology.

2) The driving strategy based on deep reinforcement learning requires a lot of training time, and building real experimental scenarios is difficult and costly. Therefore, research in related fields is mostly focused on simulation. How to combine various learning methods such as transfer learning and imitation learning to smoothly deploy the algorithm to practical operating environments is an important problem that needs to be solved, and of course, it is also a field of great research value.

3) This article has made improvements to the model from three perspectives, and the simulation results are

significantly better than the original improved algorithm. If the three improvements can be combined to form a new algorithm, and the simulation comparison effect can also be significantly improved, it will be more conducive to the real-time processing of unstructured road information from end to end on the battlefield. At the same time, input data weight coefficients can be set to dynamically improve the performance

Adjust the impact of multiple inputs on model decision-making. In future work and study, we will continue to conduct more in-depth research and exploration on various aspects such as algorithm performance, model generalization ability, etc.

In future work, we will build on the model's applicability from single-vehicle to multi-vehicle cooperative decision-making in complex and dynamic environments. This includes the integration of multi-agent reinforcement learning, such as with applications in military material scheduling to improve coordination. We will also experiment with transfer learning and imitation learning for less training expense and to enable real world deployment. Another relevant direction is integrating the three improved modules into one single algorithm with dynamic input weighting to cater to ever-changing battlefield situations and unstructured environments. Additionally, we will continue to strive for enhanced model generalisation and real-time decision making under uncertainty.

6 Acknowledgement

Research on Intelligent New Energy Vehicle Charging Technology Based on Wind and Solar Energy Complementarity (ZX20240019).

References

[1] E. Schuetz, F.-B. Flohr, A Review of Trajectory Prediction Methods for the Vulnerable Road User, Robotics 13(1) (2024) 635-647.

[2] S. Matsumoto, M. Saito, Adaptive Identification Method for Vehicle Driving Model Capable of Driving with Large Acceleration Changes and Steering, Journal of Advanced Computational Intelligence and Intelligent Informatics 27(4) (2023) 609-615.

[3] H.-Z. Guo, L.-N. Boyle, Driving behavior at midblock crosswalks with Rectangular Rapid Flashing Beacons: Hidden Markov model approach using naturalistic data, Accident Analysis & Prevention 165(2022) 106406.1-106406.10.

[4] Y.-X. Zhu, D. Zhang, J.-H. Wang, J. Sun, Distracted Car-Following Behavior Analysis and Modeling Based on LargeScale Naturalistic Driving Experiment, Journal of Tongji University (Natural Science) 51(7)(2023) 1094-1104.

[5] H.-B. Jiang, Y. Yu, A.-X. Li, Human-like driver model in curve driving based on human visual behavior and ANFIS method, Journal of Jiangsu University (Natural Science Edition) 43(6)(2022) 621-627.

[6] M. Cina, A.-B. Rad, Categorized review of drive simulators and driver behavior analysis focusing on ACT-R architecture in autonomous vehicles, Sustainable Energy Technologies and Assessments 56(2023) 103044.1-103044.26.

[7] C. Badue, R. Guidolini, R.-V. Carneiro, P. Azevedo, V.-B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T.-M. Paixão, F. Mutz, L. Veronese, T. Oliveira-Santos, A. Ferreira De Souza, Self-driving cars: A survey, Expert Systems With Applications 165(2021) 113816.1-113816.27.

[8] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, J. Williams, The DARPA Urban Challenge-Autonomous Vehicles in City Traffic, third ed., Springer Berlin, Heidelberg, 2009 (Chapter 5).

[9] Y.-Z. Liu, Z.-Q. Huang, G.-H. Shen, J.-Y. Wang, H. Xu, Behavioral decision-making methods of autonomous vehicles based on decision tree and BN, Systems Engineering and Electronics 44(10)(2022) 3143-3154.

[10] W.-N. Caballero, D.-R. Insua, D. Banks, Decision support issues in automated driving systems, International Transactions in Operational Research 30(3)(2023) 1216-1244.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A.-A. Rusu, J. Veness, M.-G. Bellemare, A. Graves, M. Riedmiller, A.-K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518(2015) 529-533.

[12] L.-Y. Yu, X.-Y. Shao, X.-X. Yan, Autonomous Overtaking Decision Making of Driverless Bus Based on Deep Q-learning Method, in: Proc. On Robotics and Biomimetics, 2017.

[13] V. Mnih, A.-P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: Proc. IEEE International Conference on Robotics and Biomimetics, 2017.

[14] Y.-X. Zhan, X. Liu, J. Liang, Intelligent Vehicle Decision-Making Through Deep Reinforcement Learning with Risk Correction, Chinese Journal of Automotive Engineering 13(5)(2023) 656-667.

[15] T. Feng, C.-X. Shi, Y.-Q. Wang, Method of Deep Reinforcement Learning Autonomous Driving Strategy Based on Heterogeneous Fusion Features, Computer & Digital Engineering 50(9)(2022) 1929-1934.
[16] Y.-M. Gao, Q.-H. Lin, S.-P. Liu, C.-Y. Xu, Spatial Dimension Analysis and Judgement of Abnormal Rainfalls, Pearl River 43(12)(2022) 97-103+154.
[17] M.-M. Liu, Q. Dai, AlexNet vegetable recognition algorithm based on Gibbs sampling and residual connection, Computer Era (9)(2023) 43-47.
[18] Z.-K. Lin, X.-Z. Wu, Car-Following Model Considering Driver's Driving Style, Journal of Geo-information Science 25(9)(2023) 1798-1812.
[19] Z. Zhao, Z.-F. Zhu, W.-L. Wang, Gaussian Distribution Guided Position Relevance Weight for Sentiment Classification, Computer Systems & Applications 32(11)(2023) 232-239.
[20] X.-Y. Lin, Z.-M. Ye, B.-H. Zhou, DQN Reinforcement Learning-based Steering Control Strategy for Autonomous Driving, Journal of Mechanical Engineering 59(16)(2023) 315-324.
[21] B. Yang, X. Liang, H. Yin, Z. Jiang, X.-M. She, Self-attention mechanism-based CSI eigenvector feedback for massive MIMO, Telecommunications Science 39(11)(2023) 128-136.
[22] R. Raghavan, D.-C. Verma, T. Bansal, Using Double DQN and Sensor Information for Autonomous Driving, International Journal of Sensors, Wireless Communication and Control 12(1)(2022) 69-78.
[23] G.-K. Sahoo, S.-K. Das, P. Singh, Two layered gated recurrent stacked long short-term memory networks for driver's behavior analysis, Sādhanā: Academy Proceedings in Engineering Science 48(2023) 75.1-75.18.
[24] J. Bona-Pellissier, F. Bachoc, F. Malgouyres, Parameter identifiability of a deep feedforward ReLU neural network, Machine learning 112(2023) 4431-4493.
[25] W. Ran, H. Chen, J.-X. Yang, Y. Nishimura, C.-P. Guo, Y.-Y. Yin, Design Method of Motion Planning Reward Function Based on Utility Theory, Automotive Engineering 45(8)(2023) 1373-1382.
[26] M. Al-Sharman, R. Dempster, M.A. Daoud, M. Nasr, D. Rayside, W. Melek, Self-Learned Autonomous Driving at Unsignalized Intersections: A Hierarchical Reinforced Learning Approach for Feasible Decision-Making, IEEE Transactions on Intelligent Transportation Systems 24(11)(2023) 12345-12356.
[27] M.-S. Rais, R. Boudour, K. Zouaidia, L. Bougueroua, Decision making for autonomous vehicles in highway scenarios using Harmonic SK Deep SARSA, Applied Intelligence-The International Journal of Research on Intelligent Systems for Real Life Complex Problems 53(2023) 2488-2505.