Xiong *et al.*
Chinese Journal of Mechanical Engineering (2025) 38:30
https://doi.org/10.1186/s10033-024-01160-z

Chinese Journal of Mechanical Engineering

ORIGINAL ARTICLE **Open Access**

Rule-Guidance Reinforcement Learning for Lane Change Decision-making: A Risk Assessment Approach

Lu Xiong[1], Zhuoren Li[1], Danyang Zhong[1], Puhang Xu[1] and Chen Tang[1*]

Abstract

To solve problems of poor security guarantee and insufficient training efficiency in the conventional reinforcement learning methods for decision-making, this study proposes a hybrid framework to combine deep reinforcement learning with rule-based decision-making methods. A risk assessment model for lane-change maneuvers considering uncertain predictions of surrounding vehicles is established as a safety filter to improve learning efficiency while correcting dangerous actions for safety enhancement. On this basis, a Risk-fused DDQN is constructed utilizing the model-based risk assessment and supervision mechanism. The proposed reinforcement learning algorithm sets up a separate experience buffer for dangerous trials and punishes such actions, which is shown to improve the sampling efficiency and training outcomes. Compared with conventional DDQN methods, the proposed algorithm improves the convergence value of cumulated reward by 7.6% and 2.2% in the two constructed scenarios in the simulation study and reduces the number of training episodes by 52.2% and 66.8% respectively. The success rate of lane change is improved by 57.3% while the time headway is increased at least by 16.5% in real vehicle tests, which confirms the higher training efficiency, scenario adaptability, and security of the proposed Risk-fused DDQN.

Keywords Autonomous driving, Reinforcement learning, Decision-making, Risk assessment, Safety filter

1 Introduction

Autonomous vehicle, which has great potential to reduce traffic accidents and jams, is a future trend in automobiles [1]. Decision-making is a central component of an autonomous driving system since the decision-making module determines the behavior of the vehicle. More specifically, it outputs specific target points, target poses, vehicle speed and other boundary constraints based on behavioral patterns, which are later utilized in the planning module to generate trajectories [2, 3]. The lane change maneuver is an important part of behavioral decision-making [4, 5]. The current widely adopted technical routes for lane change decision-making can be divided into rule-based approaches and learning-based approaches [6, 7].

1.1 Rule-Based Decision-making

The rule-based lane change decision-making system determines the behavior of vehicles based on an established rule base. The rule-based method has the following advantages: simple to apply, highly interpretable, safe and stable [8].

Nilsson et al. [9] made simple and clear logical rules to recognize the behavioral intentions of surrounding vehicles by observing the lateral and longitudinal movement law of traffic participants, and accordingly proposed an appropriate scheme to be applied to highway lane change decision-making [10]. Constantin et al. [11] constructed a decision tree by enumerating all the possible resulting navigation decisions associated with each obstacle. The vehicle will consider the optimal decision behavior for each lane from left to right each time it approaches an

*Correspondence:
Chen Tang
chen_tang@tongji.edu.cn
[1] School of Automotive Studies, Tongji University, Shanghai 201804, China

obstacle, but the decision tree-based method faces the problem of difficult state classification for complex working conditions.

Brechtel et al. [12] combined dynamic Bayesian network based continuous space prediction with discrete-space Markov Decision Process (discrete-space MDP), so that the decision-making system can cope with the uncertainty in the evolution of the lane change state. Ref. [13] estimated the distribution of potential driving intentions of surrounding vehicles based on their historical trajectories. They used the Partial Observable Markov Decision Process (POMDP) solution framework to take the coupling effect between multiple traffic participants into account, and verified the effectiveness of the decision-making algorithm in lane change and intersection scenarios. Bahram et al. [14] used game theory to find a sequence of actions in the planning time domain to balance environmental risk and vehicle intent, solving for an optimal strategy considering the interaction.

The risk assessment-based decision-making method can model and evaluate the risk degree of the driving process for autonomous vehicles. The concept of Artificial Potential Field (APF) in the field of robot path planning is a risk assessment method that has subsequently been widely used in the field of autonomous driving and assisted driving [15, 16]. Ref. [17] added the influence of driver behavior characteristics, traffic environment and motion information on potential energy distribution based on the theory of artificial potential energy field, and constructed a unified model of "driving risk field" for human-vehicle-road closed-loop system with "kinetic energy field", "potential energy field" and "behavior field". However, the model can only describe the risk distribution at fixed moments, lack judgement of the driving intentions of traffic vehicles on structured roads, and provide insufficient analysis on how to calculate the collision risk in the presence of dynamic obstacles.

In summary, the rule-based decision-making method has a large number of parameters that need to be transformed using expert experience. In addition, it is difficult to consider the dynamic interaction characteristics of traffic participants and is not well adapted to traffic environments with complex dynamic constraints.

1.2 Learning-based Decision-making

Learning-based decision-making methods, especially reinforcement learning (RL), offer better adaptability than rule-based decision-making methods. Value-based RL methods are widely used in decision-making training for autonomous vehicles, the simplest and easiest of which is the Q-learning method. Ref. [18] used Q-learning to implement safe lane change decision-making for intelligent vehicles in their own built traffic simulation environment. Based on the Q-learning algorithm, Ref. [19] used Deep Neural Network (DNN) instead of the previous Q-table, allowing the algorithm to handle tasks in continuous state space for training. In addition, the method of experience replay was proposed to break the correlation between data, solving the problem of instability and divergence in the combination of RL and neural networks. It has opened up the research boom of deep reinforcement learning (DRL). In follow-up studies, improved methods such as prioritized experience replay have been proposed to improve the training speed and effectiveness of DQN [20]. Wang et al. [21] used DQN, which is combined with the rule-based constraints, to investigate decision-making algorithms for intelligent vehicles in lane change scenarios. Shi et al. [22] proposed a hierarchical structure based on DQN by dividing the decision-making and control processes into two related processes.

Also, many improved versions have been derived based on the original DQN method, such as Double DQN (DDQN) [23] which distinguishes the network for selecting optimal actions from the network for target value prediction, as well as Dueling DQN [24], which divides the Q-value into two parts, the state value and the dominance function, for separate calculations, both achieving faster and more stable learning. To solve the problem of continuous motion control, the DDPG algorithm was proposed. Ref. [25] used DDPG to make the intelligent vehicle conduct trial-and-error training in the high-fidelity virtual simulation environment, and finally realized the safe lane change decision-making of intelligent vehicles in complex traffic flow on structured roads.

DRL algorithms interact with the environment through agents and optimize the expected long-term reward. This function reflects a high-level goal by giving positive rewards or punishment for the direct result of the action. Although the resulting decisions may obtain higher expected rewards, there is no guarantee of their safety. In addition, the reward function may be difficult to design in many application scenarios, resulting in a final behavior that is not consistent with the desired goal.

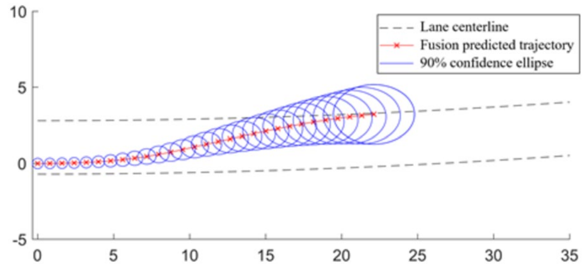


Figure 1 Uncertainty distribution of predicted trajectory

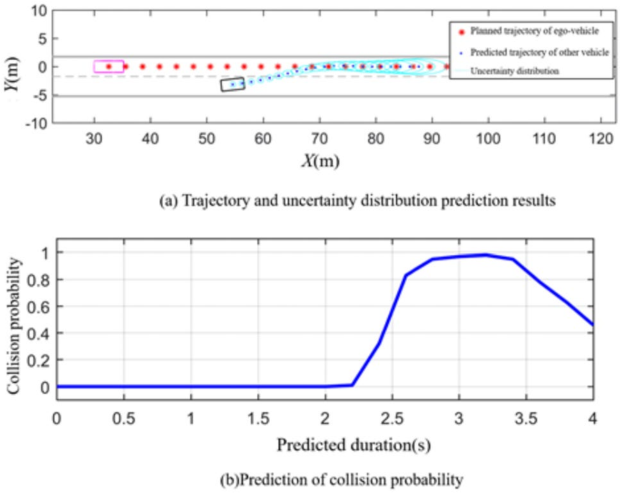
1.3 Summary

To sum up, rule-based decision-making methods have poor generalization capability due to the high dimensionality and strong uncertainty of the autonomous driving scenarios, but have the advantages of strong interpretability and easy traceability to generate stable, reliable and predictable results. On the other hand, learning based decision-making methods such as RL have strong scenario adaptability but have problems in terms of safety and reliability. They are also less efficient in recognizing dangerous scenes during the learning process. Which leads to prolonging the training time.

To solve the above problems, this study proposes a hybrid framework to combine rule-based and learning-driven approaches. A rule-based prediction uncertainty model of surrounding vehicles is constructed as a safety assessment mechanism. Decision outputs of the DRL are filtered for both network training and deployment. On this basis, this study further constructed a Risk-fused DDQN approach with a risk assessment mechanism. Simulation and real vehicle test results in different lane change scenarios showed that the proposed Risk-fused DDQN approach has higher convergence reward values and shows better safety.

The main contributions of this study are as follows:

1) A risk-fused DDQN framework is developed to combine the advantages of both rule-based and learning-driven approaches, which improves the safety performance of DRL.
2) Based on risk assessment with trajectory uncertainty, a safe filter mechanism is constructed to identify dangerous actions during the training process.
3) By replacing the dangerous actions with a rule-based safe action, the exploration of DRL agents is prolonged to obtain a better driving policy.



(a) Trajectory and uncertainty distribution prediction results

(b)Prediction of collision probability

Figure 2 Predicted trajectory and collision probabilities

The paper is constructed as follows: Section 2 introduces a safety judgement mechanism based on behavioral decision-making risk assessment and a fused DDQN training algorithm Risk-fused DDQN that incorporates this safety judgement mechanism. Section 3 designs a set of simulation training environments for autonomous driving decision-making systems on structured roads. Section 4 conducts comparison simulations and experimental tests. Section 5 summarizes the conclusion and outlook of the study.

2 Hybrid Training Framework Considering Behavioral Risk Assessment

2.1 Trajectory Prediction and Collision Evaluation

To enhance the ability of DNN to recognize dangerous actions, this study introduces a lane change trajectory risk assessment algorithm that considers the predicted trajectory of surrounding vehicles.

Firstly, the predicted lane change behavioral trajectory T_{LC} of surrounding vehicles can be obtained from Ref. [26], where the vehicle speed is assumed to be constant. In addition, considering that the vehicle has inertia and the speed as well as the yaw rate remain constant over a short period of time, the kinematic model can be expressed as

$$\begin{cases} v_x(t) = v_0 \cos(\omega_0 t + \varphi_0), \\ v_y(t) = v_0 \sin(\omega_0 t + \varphi_0), \end{cases} \tag{1}$$

where v_0, ϕ_0, ω_0 represent the initial vehicle speed, the initial heading angle and the initial yaw rate at the current moment, respectively. According to the initial position and the initial heading (x_0, y_0, ϕ_0) of the vehicle at the current moment, the predicted kinematic trajectory T_{kin} is then obtained according to Eq. (1). A cubic curve $w(t) = a_3t^3 + a_2t^2 + a_1t + a_0$ is adopted to fuse the above two trajectories to obtain the fusion predicted trajectory T_{fu} as expressed by Eq. (2).

$$T_{fu}(t) = w(t) \cdot T_{LC}(t) + (1 - w(t)) \cdot T_{kin}(t). \tag{2}$$

The noise of the position and heading information of other vehicles, i.e. x, y and ϕ satisfy the Gaussian distribution $\sum_{x,y,\varphi} \sim N(0, \sigma^2_{x,y,\varphi})$, which represents the covariance of this 3D Gaussian distribution [27]. On this basis, the spread of uncertainty along this trajectory is derived and the uncertainty distribution of the predicted trajectory with 90% confidence ellipse is shown in Figure 1.

Every traffic participant around is traversed. The fusion predicted trajectory T_{fu} with the uncertainty of the jth surrounding vehicle is taken with the planning trajectory T_p of the ego-vehicle for collision risk calculation. The collision probability at moment i represented by Eq. (3) is then calculated by sampling the trajectory with uncertainty of other vehicles N times with a Gaussian distribution.

$$P_{ij}(T_p, T_{fu}(t)) = \frac{1}{N}\sum_{n=1}^{N} I_c(S_{ego}, S_{other}), \tag{3}$$

where S_{ego} represents the pose rectangular box of the ego-vehicle planning trajectory at moment i, S_{other} represents the pose rectangular box of the surrounding vehicles obtained at the nth sampling, I_c represents the collision detection function. If there is an intersection between S_{ego} and S_{other}, then a single count is performed, as shown by the Eq. (4):

$$I_c(S_{ego}, S_{other}) = \begin{cases} 0\, S_{ego} \cap S_{other} = 0, \\ 1\, S_{ego} \cap S_{other} \neq 0. \end{cases} \tag{4}$$

The above collision probability calculation is looped for all discrete points along the predicted trajectory to predict collision probability. Figure 2(a) shows a top view of the actual traffic scenario, which contains the planned trajectory of the ego-vehicle, the predicted trajectory of the other-vehicle, and its uncertainty distribution. Figure 2(b) shows the collision probability at each moment in the prediction horizon.

2.2 Risk Assessment for DRL Decision-making

Current DRL methods face two major problems. Firstly, they are too greedy in pursuing higher self-rewards and thus could fall into local optimal policy. The trained agents are easy to lead to possible collisions and poor safety in lane change decision-making. Secondly, the training efficiency needs to be improved. Free exploration of action space during training process generates frequent collisions, which causes frequent resetting of the simulation environment and limits the exploration space of DRL agent.

Therefore, a safe assessment mechanism is established to judge the safety of behavioral decisions by considering collision probability of lane change trajectories. Assessment indices such as the number of high-risk trajectory points, the peak collision probability and the reciprocal of the peak time are considered. The proposed safe assessment mechanism is then used to judge the safety of the decisions made by the agent and correct dangerous actions during the training process to increase the average episode length.

A trajectory point may interfere with the predicted trajectories of multiple vehicles. After obtaining the collision probability p_{ij} between the ego-vehicle and the jth surrounding vehicle at the moment i, the collision probabilities at the moment i are sorted to obtain an ordered sequence $p_{ij}^{ordered}$ ranked from the largest to the smallest value since the high collision risk point with a higher collision probability was given priority. The collision probabilities on the $p_{ij}^{ordered}$ are weighted and summed according to their ranking. Then the integrated collision probability $P_i \in [0,1]$ at the moment i can be expressed as,

$$P_i = \sum_j \frac{1}{j} p_{ij}^{ordered}. \tag{5}$$

Based on this, the following indicators are extracted from the trajectories with collision probabilities at each moment for safety assessment:

1) The number of high-risk trajectory points C_{HR}: The collision risk value P_0 at each point of the ego-vehicle lane change decision-making trajectory is traversed. The index is increased if P_0 is greater than a certain threshold.
2) The peak collision probability C_p: C_p can quantify how dangerous the trajectory is and characterize its safety. C_p can be expressed as

$$C_P = \min\{\max(P_i), 1\}, i = 1 \sim I, \tag{6}$$

where I represents the number of trajectory points on the predicted trajectory, and P_i represents collision probability at each moment.

3) The reciprocal of the time-to-peak C_{TTP} index: The safety of trajectories with the same peak collision probability varies due to different time-to-peak (TTP) index. The peak collision probability characterizes the potential collision risk. The longer the time-to-peak index, the greater safety margin is reserved for the autonomous vehicle to re-plan its action to cope with dangerous scenarios. The indicator can be expressed as

$$C_{TTP} = \frac{1}{TTP}. \tag{7}$$

The three safety assessment indicators mentioned above are normalized. The number of high-risk trajectory points C_{HR}, the peak integrated collision probability C_p, and the reciprocal of the time-to-peak C_{TTP} are normalized to the range [0, 20], [0, 1], [0.05, 20] respectively. The normalized indicators are then weighted and summed to obtain the integrated lane change risk which can be expressed as

$$risk = w_{HR}\frac{C_{HR}}{20} + w_P C_P + w_{TTP}\frac{C_{TTP} - 0.05}{20 - 0.05}, \tag{8}$$

where, w_{HR}, w_P, w_{TTP} are the weight coefficients of the corresponding indicators.

2.3 Risk Assessment for DRL Decision-making

The trade-off between exploration and data utilization efficiency is the key feature of reinforcement learning. In order to get a higher reward, the agent must try behaviors that have not been tried before. However, free exploration can be quite costly, especially when learning on physical platforms, such as autonomous vehicles or other robotic platforms. Such problems also exist in simulation environments. For example, when training in a highway lane change scenario, using unguided exploration may often result in collisions or near-collision situations, which resets the simulation and thus slows down the learning. A detailed investigation of various security mechanisms adopted in the reinforcement learning process can be found in Ref. [28].

This study introduces a security screening mechanism that considers predicted trajectories of surrounding vehicles by integrating the previously proposed collision probability and security judgement mechanism for lane-change scenarios based on the DDQN. The security screening mechanism is applied in both the model training and validation stages to improve the safety of behavioral decision-making. When the risk factor as calculated by Eq. (8) exceeds the threshold, corresponding actions are considered dangerous, and are then corrected to avoid frequent collisions.

In this study, the action set of an autonomous vehicle is defined as: left lane change, right lane change, increase target speed, decrease target speed, and IDLE. A correction mechanism is constructed for dangerous actions in longitudinal and lateral directions with pseudo-code as shown in Algorithm 1. The input to the action correction function is decision 1 of action a which is judged to be dangerous by risk assessment. The function corrects the action according to the logic proposed by Algorithm 1 and returns the corrected $a_{corrected}$.

Algorithm 1 DDQN training algorithm based on risk assessment and behavioral decision correction

```
Input: Initialize main value network parameters $\theta$, copy parameters to generate target value network $\theta_{targ} \leftarrow \theta$, experience replay buffer $D$, current episode $e = 0$, maximum number of training episodes $E$, experience replay buffer capacity $N_r$, sampled trajectory length $N_t$, target network update cycle $N_{targ}$,
while $e < E$ do
    Initialize the traffic environment state $s$;
    Extract observations $o$;
    While not terminal do
        Randomly explore the main value network and $\varepsilon$-greedy for action $a$;
        if risk is low then
            $a_{coorected} = a$
        else
            $a_{coorected} = ActionCorrection(a)$
        end
        execute $a_{coorected}$ and get $s'$ and $r$;
        save ($s$, $a_{coorected}$, $r$, $s'$) into D;
        if $|D| \geq N_r$ then
            Delete oldest data;
        end
        if time to update the network then
            Randomly sample $N_t$ data from $D$ to generate $a$
            Define $a_{\max}(s'; \theta) = \mathrm{argmax} a'(s', a'; \theta)$
            Calculate the target value $y_j =$
            $\begin{cases} r & \text{,if } s' \text{ is terminal} \\ r + \gamma Q(s', a_{\max}(s'; \theta); \theta_{targ}) & \text{,otherwise} \end{cases}$
            Calculate the gradient of $\| y_j - Q(s, a; \theta) \|^2$ to update the main value network;
            $\theta_{targ} \leftarrow \theta$ per $N_{targ}$
        end
    end
end
```

The pseudo-code for DDQN training based on risk assessment and behavioral decision correction is shown in Algorithm 1. The risk assessment mechanism described in Sections 2.1 and 2.2 are adopted to determine the risk of action a, and the correction mechanism for dangerous actions is shown in Algorithm 1. Other main processes such as Experience Replay, Target Value Calculation and Network Update are consistent with the classical DDQN model.

Most of the unreasonable and dangerous actions made by the agent can be corrected during the training process

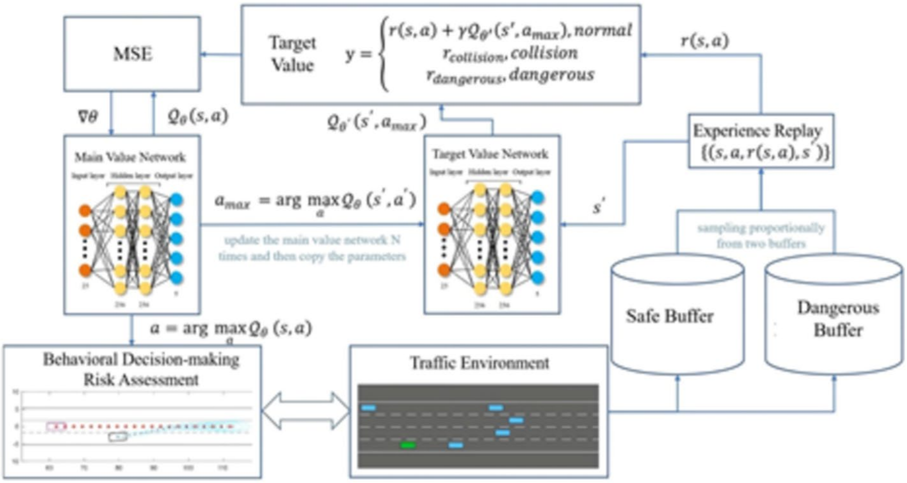

MSE
Target Value
$y = \begin{cases} r(s,a) + \gamma Q_{\theta'}(s', a_{max}), normal \\ r_{collision}, collision \\ r_{dangerous}, dangerous \end{cases}$
$r(s,a)$
$\nabla\theta$
$Q_\theta(s,a)$
$Q_{\theta'}(s', a_{max})$
Experience Replay
$\{(s, a, r(s,a), s')\}$
Main Value Network
Input layer Hidden layer Output layer
$a_{max} = \arg\max_a Q_\theta(s', a')$
update the main value network N times and then copy the parameters
Target Value Network
Input layer Hidden layer Output layer
s'
sampling proportionally from two buffers
Safe Buffer
Dangerous Buffer
$a = \arg\max_a Q_\theta(s,a)$
Behavioral Decision-making Risk Assessment
Traffic Environment

Figure 3 Fusion training based on risk assessment

based on the above safety evaluation and action correction mechanism. Therefore, the agent can obtain a larger average episode length during the training and thus obtain better training results.

Although the action correction based on risk assessment allows the agent to avoid most of the dangerous behavioral decisions, there is no guarantee that a safe action can be chosen in all working conditions. Meanwhile, the agent relies heavily on the protection of the risk assessment mechanism, increasing the computational burden on the system. In order to get better decision-making results and significantly reduce the reliance on the rule-based protection mechanism, it is necessary to set penalty feedback and experience sampling on the sequence of state actions corrected by the introduced rule, so that the policy network can achieve better results.

2.4 Fusion Training Based on Risk Assessment

This section proposes a rule-based fusion DRL model for lane change scenarios, namely Risk-fused DDQN.

Based on traditional DRL method for decision-making, trajectory prediction and uncertainty representation of surrounding traffic participants constructed in Section 2.1 are adopted in the risk assessment. The output actions of the model in training and validation process are corrected by the risk assessment mechanism as described in Section 2.3. Apart from that, training of the model is supervised by storing the sequence of dangerous state actions for fixed-proportion sampling and special punishment as detailed below.

As shown in Figure 3, the proposed Risk-fused DDQN is realized on the basis of DDQN algorithm and differs from DDQN mainly in four stages of reinforcement learning algorithm: action selection, experience storage, trajectory sampling, and target value calculation, which will be described as follows. The pseudo-code of the Risk-fused DDQN training is shown in Algorithm 2.

Firstly, in the action selection stage, the main value network of the agent obtains the environment state s at the current moment during the interaction with the environment and selects the decision of action a accordingly. Risk-fused DDQN then judges the safety of the action a through the security judgement mechanism for behavioral decision-making based on risk assessment. If the action is judged to be dangerous, it will be corrected to a safe action $a_{coorected}$ by the dangerous action correction mechanism proposed in Section 2.3, and then the safe action will be executed by the agent. If the action is judged to be safe, in contrast, no correction will be made and the agent can continue to execute the original action.

Algorithm 2 Risk-fused DDQN training algorithm based on risk-assessment supervision

```
Input: $\theta$, $\theta_{targ} \leftarrow \theta$, safe experience replay buffer $D_{safe}$, dangerous experience replay buffer $D_{dangerous}$, safe experience replay buffer capacity $N_{safe}$, safe experience replay buffer capacity $N_{dangerous}$, $e = 0$, $E$, $N_t$, $N_{targ}$
while $e < E$ do
    Initialize the traffic environment state $s$;
    While not terminal do
        Obtain action $a = \text{argmax}_a Q(s, a; \theta)$ from the main value network with a random exploration probability $\varepsilon$;
        if risk is high then
            is_safe = $False$;
            Save ($s$, $a$, $r_{\text{dangerous}}$, *) into $D_{dangerous}$;
            $a = ActionCorrection(a)$
        end
        Execute $a$ and get $s'$ and $r$;
        if is_safe == $Ture$ then
            save ($s$, $a$, $r$, $s'$) into $D_{safe}$;
        end
        if | $D_{safe}$ | $\geq N_{safe}$ then
            Delete oldest data;
        end
        if | $D_{safe}$ | $\geq N_{safe}$ then
            Delete oldest data;
        end
        if time to update the network then
            Sample a total of $N_t$ data from $D_{safe}$ and $D_{dangerous}$ in a fixed proportion $k$ to generate ($s$, $a$, $r$, $s'$);
            Define $a_{\max}(s'; \theta) = \text{argmax}a'(s', a'; \theta)$
            Calculate the target value $y_j$ =
            $\begin{cases} r_{collision} & \text{,if } s' \text{ is collision} \\ r_{dangerous} & \text{,if } (s,a) \text{ is dangerous} \\ r + \gamma Q(s', a_{\max}(s';\theta); \theta_{targ}) & \text{,otherwise} \end{cases}$
            Calculate the gradient of $\| y_j - Q(s,a;\theta) \|^2$ to update the main value network;
            $\theta_{targ} \leftarrow \theta$ per $N_{targ}$
        end
    end
end
```

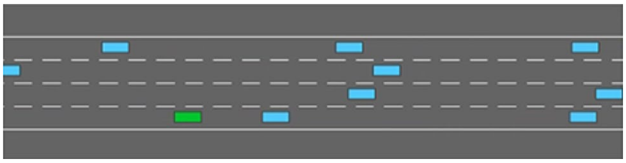


Figure 4 Four-lane high-speed lane change scenario

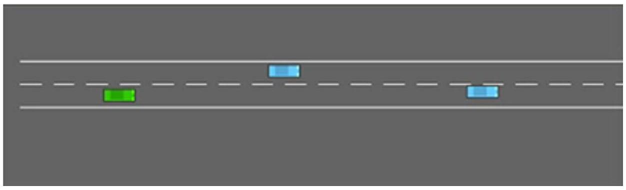


Figure 5 Four-lane high-speed lane change scenario

Secondly, in the experience storage stage, Risk-fused DDQN sets two experience buffers for storing the safe and dangerous action experience separately. The state s' at the next moment can be obtained after the agent executes the action. If the executed action is a corrected action, it means that the state s' has no relationship with the previous dangerous action a, and the agent needs to be notified original action a is dangerous so that s' does not need to be recorded. Accordingly, the action state pair (s, a) is given a fixed penalty $r_{\text{dangerous}}$ instead of calculating rewards based on changes in motion states between adjacent moments, and then the state transition information (s, a, $r_{\text{dangerous}}$, *) is stored in the dangerous experience buffer. If the executed action is an uncorrected action, the reward r will be calculated normally according to the DDQN process, and the state transition information (s, a, $r_{\text{dangerous}}$, s') will be stored in the safe experience buffer.

Thirdly, in the trajectory sampling stage, when a sufficient amount of data has been collected in the safe and dangerous experience buffer, experience replay is performed by fixed proportion sampling. Suppose the length of the trajectory to be sampled is N_t and the sampling proportion of the safe experience is k, then kN_t and (1-k) N_t pieces of state transition information are randomly selected from the safe experience buffer and the dangerous experience buffer respectively, and the two batches of data are combined as the object for calculating the loss function in batch processing.

Finally, when applying the Bellman Equation for the target value calculation, special treatment is required for the data from the dangerous experience buffer. The state-action value function corresponding to the next state is not calculated. On one hand, since the dangerous action is not actually executed, there is no way to obtain the state at the next moment after it has been executed. On the other hand, it can be considered that there is a high probability of collision after the action, which is judged to be dangerous by the risk assessment mechanism, has been taken. In other words, the cumulated reward of the dangerous action after the current episode is considered to be zero. The treatment of dangerous actions is similar to that of collision cases. If the experience data are from

Table 1 Parameters of four-lane high-speed lane change scenario

Fixed parameters	**Number of lanes $n_{\text{lane}} = 4$, lane width $w = 4$ m, total road length $l = 1000$ m, Vehicle rectangle length × width = 5 m × 3 m, speed limit $v_{\min} = 20$ m/s, $v_{\max} = 30$ m/s**				
Parameter name	Number of traffic participants n	Set of initial lanes $\{lane\_index_{\text{init}}\}_n$	Set of initial speed $\{v_{\text{init}}\}_n$	Set of initial longitudinal positions $\{s_{\text{init}}\}_n$	Speed expansion factor for IDM control model ξ
Parameter determination	$n_{\max} = 80$	satisfies the uniform distribution of the set {1,2,3,4}, obtained by random sampling	$v_{\text{init}} \sim N(0.8v_{\max}, (0.7v_{\max})^2)$	$\Delta s_{\text{init}} = l/n + k_s v_{init}$ $s_{\text{init}} = s_{\max} + \Delta s_{\text{init}}$ where k_s is the correction factor	randomly generated within [3.5,4.5]

the safe experience buffer, the target value is calculated normally according to the Bellman Equation.

In conclusion, the rule-based risk assessment guides the exploration in reinforcement learning. It helps the agent to recognize the dangerous action before the actual collision occurs and learn to extract more information about the surrounding traffic conditions, and such information enables the neural network to understand the future dangers.

3 Scenario Construction and Agent Training

3.1 Vehicle Trajectory Planning and Control

After receiving the target lane information from the decision-making module, a quantic polynomial trajectory cluster is generated using longitudinal target points on the center-line of the target lane, and Lattice-based path planning method is performed. The Stanley path tracking control is adopted in this study for lateral control.

Since training in reinforcement learning process often requires hundreds and thousands of experience acquisitions. In order to reduce the computational effort and training time, the planning and control module directly specifies the target velocity v_{target} in the decision-making cycle and tracks v_{target} accordingly. A fixed increment Δv_{acc} and a fixed increment Δv_{dcc} are set for the acceleration and the deceleration decisions respectively. The target speed is the current speed v plus the corresponding speed increment, which is truncated according to the road speed limit interval $[v_{\min}, v_{\max}]$. The acceleration is defined as the controlled variable for longitudinal velocity control, and a proportional control is implemented with coefficient $K_{p,a}$ to obtain the acceleration $a = K_{p,a}(v_{target}$ $-v)$.

3.2 Decision-making Model for Traffic Vehicles

In this study, the longitudinal decision-making of traffic participants adopts the Intelligent Driver Model (IDM) [29], which sets the target speed v_0 for each traffic participant vehicle and calculates the corresponding acceleration value according to the relative motion information of the front vehicle.

In the lane keeping scenario, the discrete decisions of the traffic participants are calculated using the Minimizing Overall Braking Induced by Lane changes (MOBIL) algorithm [30], the core concept of which is a strategy to minimize the overall braking caused by lane change. The MOBIL algorithm is combined with the "comity coefficient" to consider the impact of acceleration on following vehicles in lane change decision-making on the basis of using the acceleration as the utility function.

3.3 Construction of Random Traffic Flow

In this study, two typical structured road scenarios were constructed based on Highway Env simulation platform [31]. The first one is a simulated motorway driving scenario in which the autonomous vehicle completes its lane change and overtaking. The second one is a simulated low-speed urban driving scenario in which the decision-making model needs to choose an appropriate time to complete the lane change, overtaking and merging under the interference of the left vehicle. The above two scenarios are shown in Figures 4 and 5 respectively, where the green square represents the autonomous vehicle, and the blue squares represent other traffic vehicles. The direction of the traffic flow is from left to right. Parameters for the scenario setting are listed in Tables 1 and 2.

4 Simulation and Experimental Test

4.1 Simulation Results

In order to verify the effectiveness of the Risk-fused DDQN algorithm proposed in this paper, the DQN, the DDQN and the Dueling DQN decision-making algorithms, which are based on the value function, as well as the Risk-fused DDQN algorithm were applied for simulation and comparison in the constructed four-lane high-speed scenario and two-lane low-speed lane change scenario, respectively.

All of the DRL model comprises one evaluated Q network and one target Q network. Each Q network is a fully connected neural network, consisting of four layers: the first layer is the input layer with 25 input dimensions, layers 2 and 3 are the hidden layers with 128 units, and the layer 4 is the output layer which outputs the Q-value of each feasible action. The activation function is ReLU and the learning rate is set to 0.2.

In addition, short-term behavioral decisions may cause behavioral oscillations or overly conservative problems. The decision-making task of lane change also needs to take a long period of time to execute. Thus, the decision interval in the paper is set to 1s, and the control interval is 50 ms. Each model is trained by 10000 episodes and tested by 1000 episodes, where the max time of each episode is set to 35 s.

1) Four-lane High-speed Lane Change Scenario:

A total of 10000 episodes are set in this scenario, with a maximum episode length of 40 s. If the vehicle completes the entire episode, the simulation environment will be reset and start the next episode.

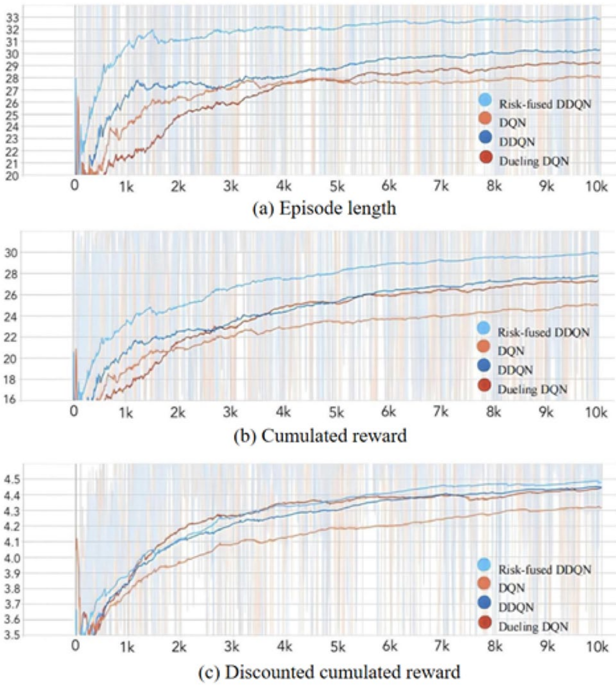
The comparison training results of the Risk-fused DDQN algorithm and several traditional DQN-based RL decision-making algorithms in the four-lane high-speed lane change scenario are shown in Figures 6 and 7. The original data has been linearly smoothed. As shown in Figure 6(a), the Risk-fused DDQN algorithm has a longer episode length at the beginning of the training compared with the traditional reinforcement learning decision-making algorithms. The reason is that the dangerous action correction mechanism in Section 3.3 works to steer the agent quickly towards safe travel in the early stage of training. After 10000 episodes, the policy networks of all four algorithms have largely converged and their episode lengths have stabilized at around 32 s, 30 s, 29 s and 28 s respectively. As shown in Figure 6(b) and (c), the supervised fusion algorithm based on risk assessment leads the other algorithms in both undiscounted and discounted rewards. The detailed data comparison at the end of the training of 10000 episodes is shown in Table 3. Compared with DQN, DDQN, and dueling DQN, Risk-fused DDQN has improved convergence values of all three indicators, which are episode length, cumulated reward and discounted cumulated reward. In addition, Risk-fused DDQN reduces the number of episodes required for the reward function to reach the same value by 52.2%.

2) Two-lane Low-speed Lane Change Scenario:

The comparison training results in the two-lane low-speed lane change scenario are shown in Figure 8 and Figure 9. In terms of episode length, due to the protection of the behavioral decision correction mechanism

Table 2 Parameters of two-lane low-speed lane change scenario

Fixed parameters	**Number of lanes $n_{\text{lane}} = 2$, lane width $w = 4$ m, total road length $l = 1000$ m,** **Vehicle rectangle length × width = 5 m × 3 m, speed limit $v_{\min} = 0$ m/s, $v_{\max} = 15$ m/s**			
Parameter name	Initial speed of the left vehicle, the front vehicle, and the ego-vehicle: v_{left}, v_{right}, v_{left},	Initial longitudinal positions of the left vehicle, the front vehicle, and the ego-vehicle: s_{left}, s_{right}, s_{left},	Initial lateral positions of the left vehicle, the front vehicle, and the ego-vehicle: l_{left}, l_{right}, l_{ego},	Distance and speed difference between the ego-vehicle and the front or the left vehicle
Parameter determination	$\{v_{ego}\}$: 10 discrete values generated uniformly within $[v_{\min}, v_{\max}]$; $\{\Delta v_{front}\}$: 2 positive values and 8 negative values; $\{\Delta v_{left}\}$: 5 positive values and 5 negative values; $v_{front} = v_{ego} + \Delta v_{front}$ $v_{left} = v_{ego} + \Delta v_{left}$	$\{s_{ego}\}$, $\{\Delta s_{front}\}$ and $\{\Delta s_{left}\}$: 10 discrete values generated uniformly within [100, 300], [20, 100] and [-50, 50], respectively $s_{front} = s_{ego} + \Delta s_{front}$ $s_{left} = s_{ego} + \Delta s_{left}$	$l_{left} \sim N(l_{left\_lane}, (0.4w)^2)$ $l_{right} \sim N(l_{right\_lane}, (0.4w)^2)$ $l_{ego} \sim N(l_{ego\_lane}, (0.4w)^2)$ where l_{left_lane} and l_{right_lane} are the lateral positions of the lane center line.	Obtained by randomly sampling.
Additional remarks	v_{ego}, Δv_{front}, Δv_{left}, s_{ego}, Δs_{front} and Δs_{left} are obtained by randomly sampling from the corresponding sets, respectively.			

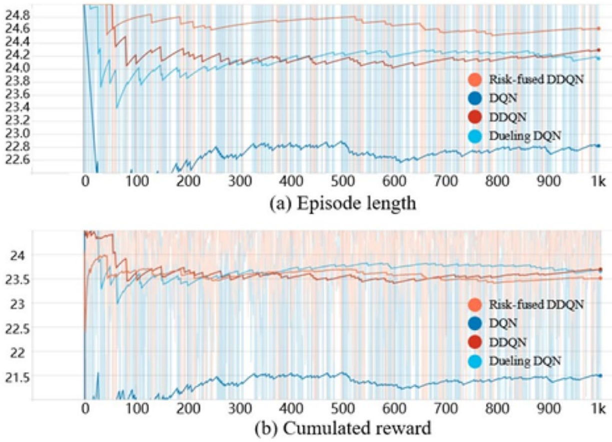


(a) Episode length

(b) Cumulated reward

(c) Discounted cumulated reward

Figure 6 Comparison of the training process of the four-lane high-speed lane change scenario



(a) Episode length

(b) Cumulated reward

Figure 7 Comparison of the testing process of the four-lane high-speed lane change scenario

based on risk assessment, Risk-fused DDQN rose rapidly at the beginning of the training. After 9000 episodes, the episode length of Risk-fused DDQN converged to around 29 s, while DQN, DDQN and Dueling DQN converged to around 28 s, 26.5 s and 27 s, respectively. In terms of cumulated reward, Risk-fused DDQN led the other algorithms at the beginning of the training. In terms of discounted cumulated reward, the four algorithms showed little difference in performance, and the difference in convergence values between the four algorithms did not exceed 1%, among which the Dueling DQN algorithm still got the highest convergence value. The detailed data comparison at the end of the training of 9000 episodes is shown in Table 4. Risk-fused DDQN algorithm improves convergence values of episode length and cumulated reward. Discounted cumulated reward of Risk-fused DDQN algorithm is essentially the same as the other three algorithms. In addition, Risk-fused DDQN reduces the number of episodes required for the reward function to reach the same value in two constructed scenarios by 66.8%. Furthermore, Table 5 shows the simulation testing results in both two scenarios, which shows the proposed Risk-fused DDQN can significantly improve the safety of lane changing decision making with small changes in average speeds.

3) Real-time performance test: In addition, the real-time performance is tested in 500 episodes of each scenario, and the comparison results are listed in Table 6. It can be seen that DRL method has really great real-time performance by generating actions through DNN. Even with the addition of a rules-based risk assessment module, the computational time of the proposed Risk-fused DDQN still meets the current deployment requirements of autonomous driving.

4.2 Real Vehicle Test Results

To further test our approach, we conducted real-vehicle tests under the two-lane low-speed lane change scenario described in Section 3.3. The real vehicle platform is shown in Figure 10, which is configured with a 128-line LIDAR and two 16-line complementary blind LIDARs.

Table 3 Comparison of Risk-fused DDQN algorithm and traditional reinforcement learning decision-making algorithms in the four-lane high-speed lane change scenario in simulation training

Comparison approach	Convergence value of episode length	Compared to DQN	Convergence value of cumulated reward	Compared to DQN	Convergence value of discounted cumulated reward	Compared to DQN
DQN	28.06	-	25.03	-	4.318	-
DDQN	30.28	7.9%	27.78	11.0%	4.449	3.0%
Dueling DQN	29.28	4.3%	27.30	9.1%	4.445	2.9%
Risk-fused DDQN	**32.82**	**17.0%**	**29.88**	**19.4%**	**4.479**	**3.7%**

Bold values indicate the effect of the method proposed in this paper

Table 4 Comparison of Risk-fused DDQN algorithm and traditional reinforcement learning decision-making training algorithms in the two-lane low-speed lane change scenario in simulation training

Comparison approach	Convergence value of episode length	Compared to DQN	Convergence value of cumulated reward	Compared to DQN	Convergence value of discounted cumulated reward	Compared to DQN
DQN	28.08	-	24.27	-	4.323	-
DDQN	26.41	-5.4%	24.51	-1.0%	4.289	-0.8%
Dueling DQN	27.01	-3.8%	23.45	-3.4%	4.352	0.7%
Risk-fused DDQN	**28.93**	**3.0%**	**25.05**	**3.2%**	**4.304**	**-0.4%**

Bold values indicate the effect of the method proposed in this paper

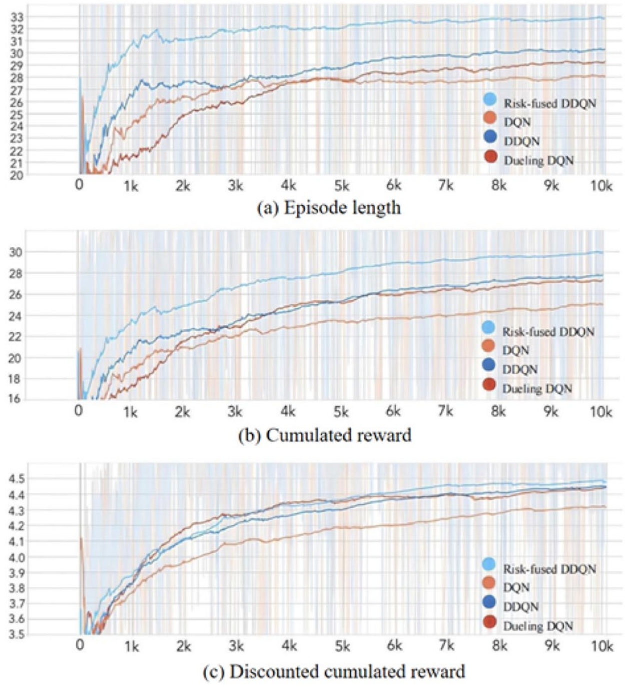
Table 5 Comparison of Risk-fused DDQN algorithm and traditional reinforcement learning decision-making algorithms in simulation test

Scenario	Four-lane high-speed lane change				Two-lane low-speed lane change			
Comparison approach	DQN	DDQN	Dueling DQN	Risk-fused DDQN	DQN	DDQN	Dueling DQN	Risk-fused DDQN
Average speed (m/s)	29.32	29.67	29.58	**29.01**	10.6	10.0	11.03	**11.08**
Lane change decision times	22983	24248	24176	**24638**	29166	29370	28791	**29167**
Collision times	221	61	60	**30**	49	42	112	**7**
Collision rate	0.97%	0.25%	0.25%	**0.12%**	0.17%	0.14%	0.39%	**0.02%**

Bold values indicate the effect of the method proposed in this paper

The localization is provided by a high-precision differential localization device, and the computational unit is an industrial control machine with GPU. We extracted the parameters of the neural network trained in Python environment and reconstructed the network structure using OpenCV in C++ environment.

The obstacle feature data detected by LiDAR and the state data of the ego vehicle are used as state space to input into the reconstructed network structure, and then the required decision action for lane change can be obtained. The lower-level control algorithm will follow the corresponding reference road according to the

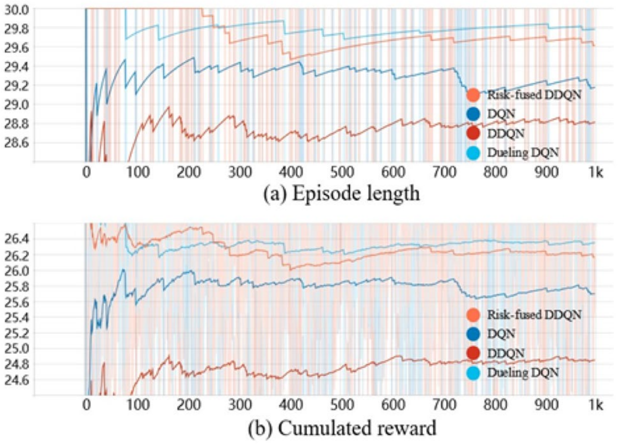


(a) Episode length

(b) Cumulated reward

(c) Discounted cumulated reward

Figure 8 Comparison of the training process for the two-lane low-speed lane change scenario



(a) Episode length

(b) Cumulated reward

Figure 9 Comparison of the testing process of the two-lane low-speed lane change scenario

Table 6 Comparison of the real-time performance

Scenario	four-lane high-speed lane change		two-lane low-speed lane change	
Computation Time (ms)	Average	Max	Average	Max
DQN	6.32	16.84	6.31	10.01
DDQN	6.38	11.36	6.31	12.66
Dueling DQN	9.10	12.54	9.17	12.39
Risk-fused DDQN	**20.69**	**34.89**	**20.59**	**39.14**

Bold values indicate the effect of the method proposed in this paper

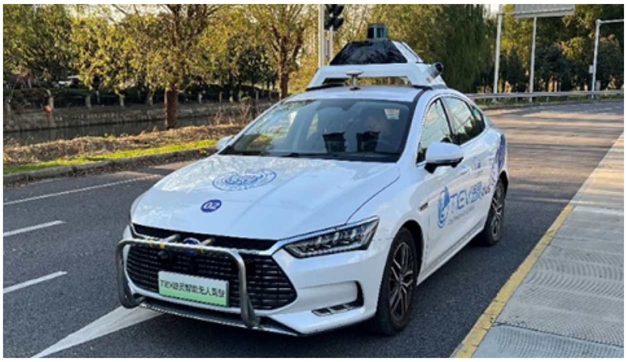


Figure 10 Real vehicle test platform

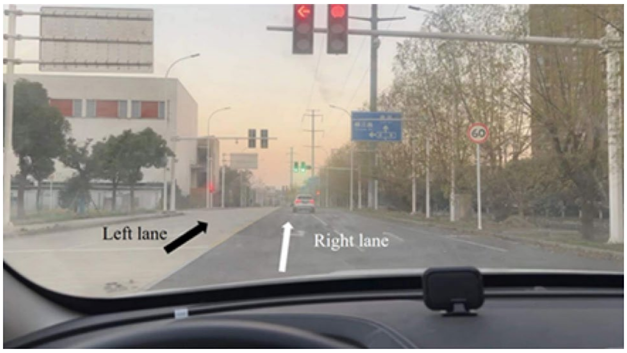


Figure 11 Road scene from the driver's view of the autonomous vehicle

Table 7 Comparison of driving data for scenarios 3 and 4

Scenario	Scenario 3		Scenario 4	
Comparison approach	DDQN	Risk-fused DDQN	DDQN	Risk-fused DDQN
Whether overtaking succeeded	Failed	**Successful, but failed to right lane change**	Successful	**Successful**
Average speed during left lane change (m/s)	-	**4.45**	4.46	**5.11 (14.6%↑)**
Average speed during right lane change (m/s)	-	**-**	2.42	**2.74 (13.2%↑)**
Average speed during the driving (m/s)	-	**4.80**	4.59	**5.17 (12.6%↑)**
Ego-vehicle speed when changing lane to the left (m/s)	-	**5.02**	5.98	**6.49 (8.5%↑)**
Distance from the front vehicle when changing lane to the left (m)	-	**22.21**	22.80	**28.84 (26.5%↑)**
Front vehicle speed when changing lane to the left (m/s)	-	**2.55**	3.10	**2.43 (21.6%↓)**
Time headway when changing lane to the left (s)	-	**8.71**	3.81	**4.44 (16.5%↑)**
TTC with the front vehicle when changing lane to the left (s)	-	**8.99**	7.92	**7.10 (10.4%↓)**
Longitudinal distance from the front vehicle in the target lane when changing lane to the left (m)	-	**14.2**	16.86	**22.32 (32.4%↑)**
Speed of the front vehicle in the target lane when changing lane to the left (m/s)	-	**7.88**	6.80	**6.96 (2.4%↑)**
Ego-vehicle speed when changing lane to the right (m/s)	-	**-**	6.15	**6.95 (13.0%↑)**
Longitudinal distance from the rear vehicle in the target lane when changing lane to the right (m)	-	**-**	11.82	**16.94 (43.3%↑)**
Time headway of the rear vehicle in the target lane when changing lane to the right (s)	-	**-**	3.55	**5.20 (46.5%↑)**

Bold values indicate the effect of the method proposed in this paper

decision action. The two-vehicle scenario was divided into two sub-scenarios of the front vehicle at rest and the front vehicle at a low speed, and the three-vehicle scenario was divided into four sub-scenarios according to the speed of the left interfering vehicle and its initial position relative to the ego-vehicle. There were total of six sub-scenarios. In each sub-scenario, Risk-fused DDQN and traditional DDQN decision-making models were tested once, respectively. The experimental road was a four-lane urban structured road. For safety reasons, the middle two lanes of the four-lane road were chosen as the experimental lanes in the real vehicle tests. The road environment from the driver's view of the ego-vehicle is shown in Figure 11.

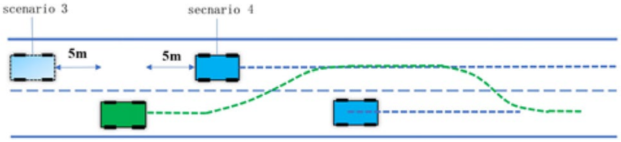


Figure 12 Schematic diagram of the scenarios where the left vehicle passes quickly

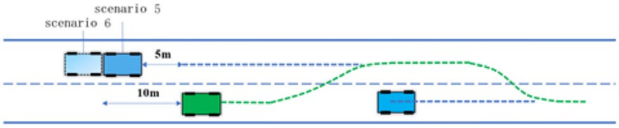


Figure 13 Schematic diagram of the scenarios where the left vehicle yields at a low speed

The initial distance from the front of the ego-vehicle to the rear of the front vehicle was all set to be 50 m. Ideally, the ego-vehicle accelerates from standstill and gradually approaches the front vehicle, and then at the appropriate time, the ego-vehicle is given a left lane change command and moves to the left lane to overtake. Afterwards, the ego-vehicle is given a right lane change command and returns to the right lane.

1) Two-vehicle scenario:

Only the autonomous vehicle was in the right lane. There was a single obstacle in front of the autonomous vehicle in the current lane and no vehicle in the left lane.

Scenario 1: The front obstacle is stationary and the ego-vehicle changes lane to avoid it.

Scenario 2: The front vehicle moves at a slow speed and the ego-vehicle changes lane to overtake and merge. Since the traffic participant vehicles were driven by human beings, it was difficult to achieve precise control. The front vehicle was set to slowly accelerate from standstill to 10 km/h and then maintained a constant speed, and after its front was overtaken by the rear of the ego-vehicle, it gradually accelerated to 15 km/h, in order to make it more difficult for the ego-vehicle to change lane to the right and merge.

Table 8 Comparison of driving data for scenario 5 and scenario 6

Scenario	**Scenario 5**		**Scenario 6**	
Comparison approach	**DDQN**	**Risk-fused DDQN**	**DDQN**	**Risk-fused DDQN**
Whether overtaking succeeded	Successful after losing the rear target vehicle	**Successful**	Successful after losing the rear target vehicle	**Successful**
Average speed during left lane change (m/s)	4.76	**5.79 (21.6%↑)**	3.32	**5.60 (68.7%↑)**
Average speed during right lane change (m/s)	6.58	**7.03 (6.8%↑)**	-	**7.15**
Average speed during the driving (m/s)	4.63	**5.21 (12.5%↑)**	4.00	**5.32 (33.0%↑)**
Ego-vehicle speed when changing lane to the left (m/s)	2.41	**6.09 (152.7%↑)**	3.70	**5.88 (58.9%↑)**
Distance from the front vehicle when changing lane to the left (m)	13.48	**33.08 (145.4%↑)**	17.22	**33.73 (95.9%↑)**
Front vehicle speed when changing lane to the left (m/s)	1.92	**2.54 (32.3%↑)**	3.16	**2.31 (26.9%↓)**
Time headway when changing lane to the left (s)	5.59	**5.43 (2.9%↓)**	4.65	**5.74 (23.4%↑)**
TTC with the front vehicle when changing lane to the left (s)	3.91	**9.29 (137.6%↑)**	31.89	**9.45 (70.4%↓)**
Longitudinal distance from the rear vehicle in the target lane when changing lane to the left (m)	-	**13.29**	-	**17.61**
Speed of the rear vehicle in the target lane when changing lane to the left (m/s)	-	**5.35**	-	**2.30**
Time headway of the rear vehicle when changing lane to the left (s)	-	**2.49**	-	**7.66**
Ego-vehicle speed when changing lane to the right (m/s)	6.93	**6.91 (0.3%↓)**	-	**6.95**
Longitudinal distance from the rear vehicle in the target lane when changing lane to the right (m)	10.77	**11.10 (3.1%↑)**	-	**9.94**
Time headway of the rear vehicle in the target lane when changing lane to the right (s)	3.24	**3.81 (17.6%↑)**	-	**4.12**

Bold values indicate the effect of the method proposed in this paper

In scenario 1, DDQN failed to change lane. When Risk-fused DDQN issued a left lane change command, the ego- vehicle speed was 4.39 m/s, the distance from the front vehicle was 42.23 m and the headway was 9.6 s. In this way, the safety margin was large, and the sense of crisis for the occupants of the autonomous vehicle was very small. When the right lane change command was issued, the ego-vehicle speed was 6.36 m/s and the distance from the rear vehicle was 6.95 m, which fully met the safety requirements.

In scenario 2, Risk-fused DDQN issued a lane change command 17.83 m ahead of DDQN.

2) Three-vehicle scenario:

In the three-vehicle scenario, the ego-vehicle was initially in the right lane. There was a slow-moving vehicle in front of the ego-vehicle and an interfering vehicle in the left lane driving in the same direction.

In the three-vehicle scenario, the left vehicle speed relative to the ego-vehicle would have a significant impact on the lane change decision-making of the ego-vehicle, so the scenario was further divided according to how fast the left vehicle was driving relative to the ego-vehicle:

Scenarios 3 and 4: The front vehicle drives at a low speed and the left vehicle passes quickly. Ideally, the autonomous vehicle firstly yields the left vehicle before the lane change. As shown in Figure 12, in order to check whether the decision-making models would make unreasonable lane change decisions at different longitudinal relative distances, it was divided into two scenarios according to the initial longitudinal distance between the left vehicle and the ego-vehicle. That is, the left vehicle was 5 m behind the ego-vehicle and 5 m ahead of the ego-vehicle, which corresponds to scenario 3 and scenario 4, respectively. The front vehicle drove in the same way as in scenario 2. The left vehicle was set to rapidly accelerate from standstill to 30 km/h and then maintained a constant speed.

Scenario 5 and 6: The front vehicle drives at a low speed, and the left vehicle yields at a low speed. Ideally, the autonomous vehicle firstly overtakes the left vehicle before the lane change. As shown in Figure 13, it is similarly divided into two scenarios according to the relative longitudinal distance between the left vehicle and the ego-vehicle. That is, the left vehicle was 5 m behind the ego-vehicle and 10 m behind the ego-vehicle, which correspond to scenarios 5 and 6, respectively. The front vehicle drove in the same way as in scenario 2. The left vehicle was set to slowly accelerate from standstill to 10 km/h and then maintained a constant speed. Then the left vehicle accelerated to 15 km/h after the ego-vehicle changed lane and drove ahead of it.

In scenarios 3–6, Risk-fused DDQN decision-making model also achieved good decision-making results. The data comparison is shown in Tables 7 and 8.

The average speeds during the six lane changes and the average speeds during the driving for Risk-fused DDQN were faster than that for DDQN. At the moment the left lane change command was issued in scenarios 4–6, for Risk-fused DDQN, the distance between the ego-vehicle and the front vehicle was larger, and the ego-vehicle speed was higher. The above comparison shows that Risk-fused DDQN can find the time to change lane in a timelier manner and avoid the speed reduction caused by the too short following distance. In addition, Risk-fused DDQN improved the time headway by at least 16.5%, compared with DDQN, though the headway in scenario 5 for Risk-fused DDQN was slightly reduced (2.9%). Scenario 4 is a scenario where the left vehicle passes quickly, so when the decision-making model issued the left lane change command, the left vehicle had already driven in front of the ego-vehicle and had become the front vehicle in the target lane. As can be seen from Table 8, Risk-fused DDQN issued the left lane change command when the speed of the front vehicle in the target lane was higher and the longitudinal distance from it was larger. In this way, the safety margin between the ego-vehicle and the front vehicle in the target lane was higher. Finally, at the moment the right lane change command was issued in scenarios 4 and 5, for Risk-fused DDQN, the longitudinal distance from the rear vehicle in the target lane was larger, and the headway of the rear vehicle in the target lane was also larger. In this way, the safety margin between the ego-vehicle and the rear vehicle in the target lane was higher.

To sum up, the success rate of left lane change and right lane change for Risk-fused DDQN in the six scenarios was 91.7%, but that for DDQN was 58.3%, improving 57.3%. In addition, the average speed during the driving for Risk-fused DDQN was improved by 19.4%, compared with DDQN. And when the lane change command was issued, the distances and time headways between the ego-vehicle and the front (or rear) vehicle in the current(or target) lane, were also larger, with the time headway improved at least by 16.5%, making the lane change decisions more secure.

5 Conclusions

To address problems of poor security and low training efficiency in conventional reinforcement-learning-based decision-making algorithms, this study proposes a risk assessment algorithm and constructs a reinforcement learning algorithm called Risk-fused DDQN, which is based on DDQN and integrates a safety evaluation mechanism for lane change scenarios. Firstly, the safety of behavioral decisions is judged on the basis of trajectory prediction and risk assessment, and then the dangerous actions are corrected, which effectively increases the average episode length in the training process. Apart from that, the algorithm sets a separate experience buffer for dangerous experiences for sampling and punishment, helping the agent to detect and record potential dangers before actual collisions. In both high-speed and low-speed lane change scenarios, compared with DDQN, the proposed Risk-fused DDQN improves the convergence value of cumulated reward by 7.6% and 2.2%, respectively, and reduces the number of episodes required for the reward function to reach the same value by 52.2% and 66.8%, respectively, verifying the higher training efficiency of Risk-fused DDQN. In real vehicle tests, Risk-fused DDQN improves the success rate of lane change by 57.3%, the average speed during the driving by 19.4%, the time headway by 16.5%, verifying the higher scenario adaptability and higher security of Risk-fused DDQN.

Acknowledgements
Not applicable.

Author Contributions
LX, ZL, PX, CT developed the idea for the study, ZL and PX implemented the algorithm; DZ and CT wrote the manuscript; ZL, LX and CT revised the manuscript. All authors read and approved the final manuscript.

Funding
Supported by National Key Research and Development Program of China (Grant No. 2022YFE0117100), National Science Foundation of China (Grant No. 52102468, 52325212), Fundamental Research Funds for the Central Universities.

Data availability
The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing Interests
The authors declare no competing financial interests.

Received: 2 February 2023 Revised: 11 December 2024 Accepted: 18 December 2024
Published online: 17 March 2025

References

[1] Y Wang, J Hu, F Wang, et al. Tire road friction coefficient estimation: review and research perspectives. *Chinese Journal of Mechanical Engineering*, 2022, 35: 6. https://doi.org/10.1186/s100-33-021-00675-z.
[2] S Li, K Shu, C Chen, et al. Planning and decision-making for connected autonomous vehicles at road intersections: A review. *Chinese Journal of Mechanical Engineering*, 2021, 34: 133. https://doi.org/10.1186/s10033-021-00639-3.
[3] Hongyan Guo, Jiaming Zhang, Jun Liu, et al. Generation of a scenario library for testing driver-automation cooperation safety under cut-in working conditions. *Green Energy and Intelligent Transportation*, 2022, 1(2): 100004, https://doi.org/10.1016/j.geits.2022.100004.
[4] G. Lucente, R. Dariani, J. Schindler, et al. A Bayesian approach with prior mixed strategy nash equilibrium for vehicle intention prediction. *Automotive Innovation*, 2023, 6(3): 425–437.
[5] Han Zhang, Chang Liu, Wanzhong Zhao. Segmented trajectory planning strategy for active collision avoidance system. *Green Energy and Intelligent Transportation*, 2022, 1(1): 100002, https://doi.org/10.1016/j.geits.2022.100002.
[6] H Jia, P Liu, L Zhang, et al. Lane-changing decision model development by combining rules abstract and machine learning technique. *Journal of Mechanical Engineering*, 2022, 58(4): 212–221.
[7] H. Deng, Y. Zhao, Q. Wang. Deep reinforcement learning based decision-making strategy of autonomous vehicle in highway uncertain driving environments. *Automotive Innovation*, 2023, 6(3): 438–452.
[8] Z Li, J Hu, B Leng, et al. An integrated of decision making and motion planning framework for enhanced oscillation-free capability. *IEEE Transactions on Intelligent Transportation Systems*, 2024, 25 6): 5718-5732.
[9] J Nilsson, J Fredriksson, E Coelingh. Rule-based high- way maneuver intention recognition. *IEEE International Conference on Intelligent Transportation Systems*, 2015: 950–955.
[10] J Nilsson, J Silvlin, M Brannstrom, et al. If, when, and how to perform lane change ma- neuvers on highways. *IEEE Intelligent Transportation Systems Magazine*, 2016, 8(4): 68–78.
[11] A Constantin, J Park, K Iagnemma. A margin-based approach to threat assessment for autonomous highway navigation. *Proceedings of IEEE Intelligent Vehicles Symposium*, 2014: 234–239.
[12] S Brechtel, T Gindele, R Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2014: 392–399.
[13] E Galceran, A G Cunningham, R M Eustice, et al. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 2017, 41(6): 1367–1382.
[14] M Bahram, A Lawitzky, J Friedrichs, et al. A game-theoretic approach to replanning-aware interactive scene prediction and planning. *IEEE Transactions on Vehicular Technology*, 2016, 65(6): 3981–3992.
[15] S Byrne, W Naeem, S Ferguson. Improved APF strategies for dual-arm local motion planning. *Transactions of the Institute of Measurement and Control*, 2015, 37(1): 73–90.
[16] W Chen, X Wu, Y Lu, et al. An improved path planning method based on artificial potential field for a mobile robot. *Cybernetics and Information Technologies*, 2015, 15(2): 181–191.
[17] H. Li, S. Li, F. Xia, et al. Driving risk field modeling and the influencing factors analysis for intelligent connected vehicle. *Proceedings of International Conference on Intelligent Traffic Systems and Smart City (ITSSC)*, 2022, 12165, SPIE: 532–538.
[18] L Yu, Y Wei, S Huo. The method and application of intelligent vehicle path planning based on mcpddpg. *Control and Decision*, 2021, 36(4): 835–846.
[19] C J Hoel, K Wolff, L Laine. Automated speed and lane change decision making using deep reinforcement learning. *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2018: 2148–2155.
[20] T Schaul, J Quan, I Antonoglou, et al. Prioritized experience replay. *arXiv preprint*, 2015, arXiv:1511.05952.
[21] J Wang, Q Zhang, D Zhao, et al. Lane change decision-making through deep reinforcement learning with rule-based constraints. *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2019.
[22] T Shi, P Wang, X Cheng, et al. Driving decision and control for automated lane change behavior based on deep reinforcement learning. *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019: 2895– 2900.
[23] Z Li, L Xiong, B Leng, et al. Safe reinforcement learning of lane change decision making with risk-fused constraint. *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2023: 1313-1319.
[24] Y Chen, H Yu, J Zhang, et al. Lane-exchanging driving strategy for autonomous vehicle via trajectory prediction and model predictive control. *Chinese Journal of Mechanical Engineering*, 2022, 35(1): 71.
[25] Y Ye, X Zhang, J Sun. Automated vehicle's behavior decision making using deep reinforcement learning and high- fidelity simulation environment. *Transportation Research Part C: Emerging Technologies*, 2019, 107: 155–170.
[26] C Tang, Y Liu, H Xiao, et al. Integrated decision making and planning framework for autonomous vehicle considering uncertain prediction

of surrounding vehicles. *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2022: 3867–3872.

[27] H Chen, X Wang, J Wang. A trajectory planning method considering intention-aware uncertainty for autonomous vehicles. *Proceedings of Chinese Automation Congress (CAC)*, 2018: 1460–1465.

[28] W Zhao, T He, R Chen, et al. State-wise safe reinforcement learning: A survey. *arXiv preprint*, 2023, arXiv: 2302.03122.

[29] M Treiber, A Hennecke, D Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 2000, 62(2): 1805.

[30] A Kesting, M Treiber, D Helbing. General lane-changing model mobil for car-following models. *Transportation Research Record*, 2007, 1999(1): 86–94.

[31] E Leurent. An environment for autonomous driving decision making. *GitHub Repository*, 2018, Available: https://github.com/eleurent/high-way-env.

Lu Xiong, born in 1978, received the PhD degree in vehicle engineering from *Tongji University, China*, in 2005. He is currently the vice dean and professor with the *School of Automotive Studies, Tongji University, China*. Dr. Xiong won the first prize of Shanghai Science and Technology Progress Awards in 2013, 2019 and 2022. He was the recipient of the National Science Fund for Distinguished Young Scholars. His research interests include dynamic control of distributed drive electric vehicles, decision making, motion planning and control of intelligent vehicles.

Zhuoren Li, born in 1996, received the B.E. degree in the *School of Aerospace Engineering and Applied Mechanics, Tongji University, China,* in 2019. He is currently pursuing his PhD degree with *the School of Automotive Studies, Tongji University, China*. His research interests include interaction decision making, motion planning, and safe reinforcement learning of autonomous vehicles.

Danyang Zhong, born in 1999, received the B.E. degree in vehicle engineering from the *Tongji University, China*, in 2022. She is currently pursuing the M.E. degree with the *School of Automotive Studies, Tongji University, China*. Her research interests include behavioral decision making, and motion control of autonomous vehicles.

Puhang Xu, born in 1997, received the B.E. degree in vehicle engineering from the *Harbin Institute of Technology, China*, in 2019. He received the B.E. degree with the *School of Automotive Studies, Tongji University*, *China* in 2022. His research interests include behavioral decision-making, trajectory planning, and autonomous vehicles.

Chen Tang, born in 1986, received the PhD degree in mechanical and mechatronics engineering from the *University of Waterloo, Canada*, in 2018. From 2019 to 2021, he was a Postdoctoral Researcher with the *University of Waterloo, Canada*. He is currently an Associate Professor with *School of Automotive Studies, Tongji University, China*. His research interests include advanced chassis systems, vehicular active safety systems, and intelligent transportation systems.