2023 IEEE 26th International Conference on
Intelligent Transportation Systems (ITSC)
24-28 September 2023. Bilbao, Bizkaia, Spain

Safe Reinforcement Learning of Lane Change Decision Making with Risk-fused Constraint

Zhuoren Li, Lu Xiong, Bo Leng, Puhang Xu and Zhiqing Fu

***Abstract*—Deep reinforcement learning (DRL) has become a powerful method for autonomous driving while often lacking safety guarantees. In this paper, we propose a Risk-fused Constraint Deep Reinforcement Learning (RCDRL) with D3QN network for safe decision making in lane change maneuver. The problem is formulated as a state-wise MDP (SCMDP), which embeds a rule-based risk-fused Constraint module. We map the decision action to the trajectory layer via a polynomial curve-based trajectory planner, which is combined with the predicted trajectories of surrounding vehicles to assess future risk and correct the unsafe action. Therefore, the proposed method can deal with unsafe decision actions when training the policy network. To investigate the decision performance, the trained RCDRL policy is tested and validated under different traffic densities. In particular, we implement real vehicle tests to validate the effectiveness of the proposed method. Simulation and real vehicle tests demonstrated that the proposed RCDRL method achieves better performance, especially in safe decision. In addition, the framework can be extended with other advanced DRL networks.**

I. Introduction

Autonomous driving technologies have shown their great potential in reducing traffic jams and accidents [1]. Decision making is an important part of autonomous driving technology, which is regarded as the brain of the intelligent vehicle. The proper decision behaviors are planned and sent to the trajectory planning or motion control module, according to the information provided by the environment perception module [2]. The performance of the decision making directly affects the safety, efficiency and comfort of the autonomous driving [3]. Many approaches have been applied to the decision making of autonomous driving. At present, the decision making approaches can be roughly divided into two types: 1) artificial priori rule-based, 2) learning-based [4].

Rule-based decision making intuitively builds mathematical models of behavior based on traffic regulations, driving experience, etc. Many approaches including logic rules library [5], Finite State Machines (FSM) [6], Decision Tree [7], Partially Observable Markov Decision Processes (POMDP) [8] and Game Theory [9] are widely used to construct rule-based decision models. These methods have achieved good results in many single similar cases, however, they are difficult to adjust the hyperparameters to be compatible with changing scenarios [10].

This work is supported by the National Natural Science Foundation of China under Grant 52002284, the National Key R&D Program of China under Grant 2022YFE0117100, the Science and Technology Commission of Shanghai under Grant 21DZ1203802 and Fundamental Research Funds for the Central Universities.

Zhuoren Li, Lu Xiong, Bo Leng, Puhang Xu and Zhiqing Fu are with the School of Automotive Studies, Tongji University, Shanghai, China (corresponding author: Bo Leng; e-mail: lengbo@tongji.edu.cn).

Learning-based approaches make it easier to identify the specific attributes of different scenarios by obtaining a large amount of environmental and behavioral data for self-learning and training. With the development of deep representation learning, deep reinforcement learning (DRL) has become another powerful method for autonomous driving, which learn the complex policies in high dimensional environments [11] and achieved impressive progress, especially in control tasks [12-14]. RL agents are not told how to act instead of evaluate themselves by the reward function. It can obtain the corresponding decision policy with maximized rewards by exploring the expected utility of different state-action pairs [15]. The performance of RL driving policy has been proved in many scenarios such as highway [16], roundabout [17], intersection [18], parking [19], and et. al. However, the unexplainable and incomplete nature make DRL can not guarantee the validity of the generated policy in inexperienced scenarios, which is the most critical problem that hinders the practical application of DRL [20]. RL agents may sometimes prefer maximizing the reward over ensuring safety, which can lead to unsafe or even disastrous accidents [21].

Based on the analysis of these limitations of two types decision methods, some scholars aim to construct a DRL model with the embedded rule-based method to improve safety. Conservative Safety Critics (CSC) is proposed to ensure the safe exploration during policy training by resampling the remaining candidate action. However, the sampling procedure is time-consuming and may be detrimental to reward performance [22]. [23] uses longitudinal MPC to predict the future trajectory according to the DRL policy. For the trajectory evaluated as unsafe, ST-solver is used to controlling the vehicle instead of DRL. This attempt is inspiring, but the accuracy of longitudinal model prediction is insufficient. [24] uses historical driving data training RNN to predict the future motion states of surrounding vehicles while training DRL, but the additional deep neural networks likewise bring new inexplicability.

With these inspirations above, this paper proposes a Risk-fused Constraint DRL method (RCDRL) to enhance the safety of DRL, and implement it for lane change decision making. We use a hierarchical scheme and ensure that the RL policies during the training process satisfy the safe constraints. We design a planner to map decision policies into trajectories and establish safety constraints through risk assessment. The decision action that exceeds the safety constraint will not be executed and replaced with another safe action. It enhances the safety of decision making and extends the episode to improve training efficiency while gaining different experiences. The main contributions of this work are the following:

2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3503-9946-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITSC57777.2023.10422331

- We propose an RCDRL framework for safe autonomous driving that can improve training efficiency and make high-quality decisions adaptively in lane change scenarios.
- Rule-based risk assessment and motion correction by mapping DRL policies into trajectories.
- We validate the capability of the proposed method with different traffic densities in a simulation and in real experiments.

The remainder of this paper is structured as follows: In Section II, we introduce the problem definition and formalize the RCDRL framework. Section III introduces the methodology of our method, including detailed illustrations of the DRL scheme, trajectory planning and prediction, and Risk-fused Constraint module including risk assessment and safety correction. Section IV provides the simulation and experiment results, which demonstrate our improvement over the previous work. The last is the conclusion in Section V.

II. Problem Formulation

A. Preliminaries

The reinforcement learning decision making problem can be formulated as a finite horizon Markov decision process (MDP) [25], which is specified by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{Z}, \mathcal{R}, \gamma, h >$:

- $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space.
- $\mathcal{Z} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function from a current state-action pair (s, a) to a new state s' at the next time step with corresponding probability $P(s'|s, a)$.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function.
- $0 \leq \gamma \leq 1$ denotes the discount factor.
- The finite decision horizon $h \in \mathbb{N}$.

The output policy $\pi : \mathcal{S} \rightarrow \mathcal{Z}$ $(\mathcal{A})$ maps the states to a probability of selecting actions, i.e. $\pi(a|s)$, which represents the driving decision under the surrounding environment in this work. The set of all policies is denoted by Π. Subsequently, π_θ is denoted as the policy parameterized by θ. The standard goal is to find the corresponding optimal policy π^* that maximizes the expected discount reward.

$$\pi^* = \arg\max_{\pi} \mathcal{Q}(\pi) = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{h} \gamma^t \mathcal{R}(s_t, \pi(a_t \mid s_t))\right] \quad (1)$$

where $\mathcal{Q}$ is called the state-action value function. RL agent can learn the state-action value function estimates directly [26].

For complex problems with large state space dimensionality, the optimal state-action function is commonly approximated by a neural network (NN) and the corresponding RL paradigm is called deep reinforcement learning (DRL). The neural network finds a fitting function that performs well on a given data set by adjusting the weight parameters of the neurons, and it predicts the values of all state-actions without using any domain-specific information or manually designed features. Thanks to the powerful parameter fitting capabilities of neural networks, the DRL algorithms have achieved human-level or even higher performance.

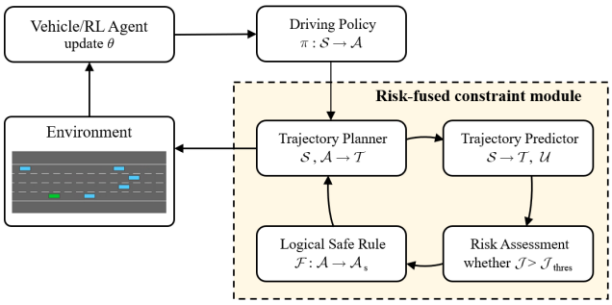


Fig. 1. The hierarchical framework of the proposed RCDRL

B. Risk-fused Constraint DRL (RCDRL)

During the MDP problem, the security specification as a hard constraint is consistently met at every step, which is called state-wise MDP (SCMDP). SCMDP evaluates the instantaneous cost $\mathcal{J}$ of state action transition by setting the constraint set $\mathcal{C}$ and requiring the cost $\mathcal{J}$ for policy set Π to satisfy the hard constraints at every step. Therefore, the feasible policy set of SCMDP ($\Pi_\mathcal{C}$) is defined as:

$$\prod\nolimits_{\mathcal{C}} = \left\{\pi \in \prod \middle| \forall (s_t, a_t, s_{t+1}), \mathcal{C}(s_t, a_t, s_{t+1}) \leq \mathcal{W}\right\} \quad (2)$$

where $\mathcal{W} \in \mathbb{R}$ is the corresponding constraint threshold set. The objective for SCMDP is to find the optimal feasible policy from $\Pi_\mathcal{C}$,

$$\pi^* = \arg\max_{\pi} \mathcal{Q}(s_t, a_t), \text{ s.t. } \pi \in \prod\nolimits_{\mathcal{C}} \quad (3)$$

In this work, the instantaneous cost $\mathcal{J}$ is the security risk with an evaluation of the estimated future state $\hat{s}_{t+1\sim t+hp}$ (h_p is the prediction horizon, which is equal to the planning horizon). Trajectory planning is designed to map the decision action $\mathcal{A}$ of the ego vehicle (EV) to a planned trajectory $\mathcal{T}^{(\text{ev})}$. Meanwhile, the prediction module (from our previous work [27]) could give the possible future trajectories $\mathcal{T}^{(\text{sv})}$ of the surrounding vehicles (OVs) and their positional uncertainties $\mathcal{U}^{(\text{sv})}$, which can be represented as:

$$\begin{cases} \mathcal{T}^{(ev)} = Plan(s_t, a_t) \\ (\mathcal{T}^{(sv)}, \mathcal{U}^{(sv)}) = Predict(s_t) \end{cases} \quad (4)$$

As a result, the security risk cost $\mathcal{J}$ can be defined as

$$\mathcal{J}(\pi(a_t \mid s_t)) = V_{risk}(s_{t+1\sim t+hp}) = V_{risk}(\mathcal{T}^{(ev)}, \mathcal{T}^{(sv)}, \mathcal{U}^{(sv)}) \quad (5)$$

With a given threshold value $\mathcal{J}_{thres}$, Equation (2) can be rewritten as,

$$\prod\nolimits_{\mathcal{C}} = \left\{\pi \in \prod \middle| \forall (s_t, a_t, \hat{s}_{t+1\sim t+hp}), \mathcal{J}(\pi(a_t \mid s_t)) \leq \mathcal{J}_{thres}\right\} \quad (6)$$

The unsafe action is identified when $\mathcal{J}(\pi(a_t|s_t)) \geq \mathcal{J}_{thres}$, and the corresponding $(s_t, a_t, \hat{s}_{t+1}, \mathcal{R}_t)$ is recorded separately. The unsafe policy will be reselected as $\mathcal{A}_s$ based on logical safe rule $\mathcal{F}$ and the agent will continue to interact with the environment for training. The hierarchical RCDRL framework is shown in Fig. 1.

III. Methodology

In this section, an Double DQN algorithm using Dueling Networks (namely D3QN [28]) is implemented in the RCDRL lane change decision making for autonomous driving, where

the dueling networks are introduced to enhance the training performance of the RL agent. The update of the state-action value function in (3) can be denoted as,

$$\mathcal{Q}(s_t,a_t)\leftarrow(1-\alpha)\mathcal{Q}(s_t,a_t) +\alpha\left[\mathcal{R}_t+\gamma\mathcal{Q}(s_{t+1},\arg\max_a\mathcal{Q}(s_{t+1},a;\theta_e);\theta_t)\right] \quad (7)$$

where α is the learning rate, θ_e and θ_t denote the parameters of the evaluation network and the target network, respectively. The driving task is to travel as efficiently as possible at the desired speed through lane changing behavior.

A. State Space and Action Design

We assume the observed feature results from direct perception as the state representation under the Frenét Coordinates. State space $\mathcal{S}=(\mathcal{S}_{ev}\ \mathcal{S}_{sv}(i))^T$ (i=1~6), where, $\mathcal{S}_{sv}(i)$ represents the two SVs closest to EV on each surrounding lane (shown in Fig. 2). The state of EV $\mathcal{S}_{ev}$ includes the position (s_{ev}, l_{ev}), velocity ($v_{ev}^{(s)}$, $v_{ev}^{(l)}$) and the presence marker p_{ev} with a constant value of 1. The state of SVs $\mathcal{S}_{sv}(i)$ includes the position ($\Delta l(i)$, $\Delta s(i)$) and speed ($\Delta v^{(s)}(i)$, $\Delta\ v^{(l)}(i)$) relative to EV, as well as the presence marker $p_{sv}(k)$. In summary, the state vector in RCDRL can be expressed as

$$\begin{cases}\mathcal{S}_{ev}=\left(p_{ev}\quad s_{ev}\quad l_{ev}\quad v_{ev}^{(s)}\quad v_{ev}^{(l)}\right)^T\\ \mathcal{S}_{sv}(i)=\left(p_{sv}(i)\quad \Delta s(i)\quad \Delta l(i)\quad \Delta v^{(s)}(i)\quad \Delta v^{(l)}(i)\right)^T\\ i=1\sim 6,\ i\in\mathbb{N}\end{cases} \quad (8)$$

The action space a set of semantic behaviors of EV containing two parts as $\mathcal{A}=(\mathcal{A}_{lat},\mathcal{A}_{lon})^T$. Concerning the latera action $\mathcal{A}_{lat}$, we consider three action options along the lateral direction, namely, {change lane to left, lane keep, change lane to right}. For the longitudinal action $\mathcal{A}_{lon}$, there are also three actions{accelerate maintain, decelerate}.

B. Trajectory Planning and Prediction

We map the semantic decision actions of the ego vehicle to the trajectory layer by designing a planner which can be expressed as,

$$\begin{aligned}&\text{find } \mathcal{T}^{(ev)}\in\mathcal{T}^{(c)}=\left\{\mathcal{T}_i^{(ev)},i=1\sim m\right\}\\ &\min\mathcal{G}(\mathcal{T}^{(ev)})=\eta_1\frac{L_i}{\max(L_i)}+\eta_2\frac{a_{y\max,i}}{\max(a_{y\max,i})}+\eta_3\frac{\varepsilon_i}{\max(\varepsilon_i)}\\ &\text{s.t. } \mathcal{T}^{(c)}=\left\{(s,l)\middle|l(s)=b_5s^5+b_4s^4+b_3s^3+b_2s^2+b_1s+b_0\right\}\\ &\left|a_y\right|\le A_{y\max},\ \left|\delta_f\right|\le\delta_{f,\max}\\ &\begin{cases}(s(0),l(0))=(s_{ev},l_{ev}),\ (s(h_p),l(h_p))=(s^{(c)},l^{(c)})\\ l'(0)=\tan\varphi_{ev},l'(n)=0\\ \kappa(0)=\kappa_{ev},\kappa(n)=0\\ \dfrac{1}{|\kappa|}\ge\max\left(\dfrac{v^2}{A_{y\max}},\dfrac{l_{rf}}{\tan\delta_{\max}}\right)\end{cases}\end{aligned} \quad (9)$$

where $\mathcal{T}^{(c)}$ is the candidate trajectory set, which is generated by quintuple polynomial curves with different sampling end poses ($s^{(c)}$, $l^{(c)}$) on the target lane. $\mathcal{T}^{(c)}$ contains m number of candidate trajectories, and the optimal trajectory is selected

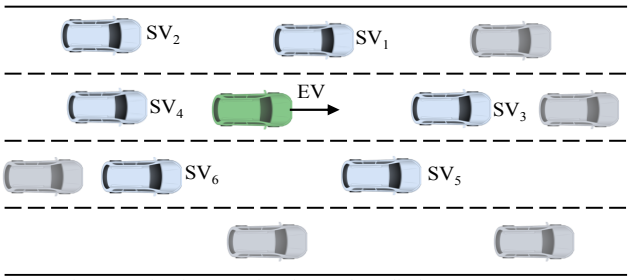


Fig. 2. Diagram of trajectory planning and prediction:

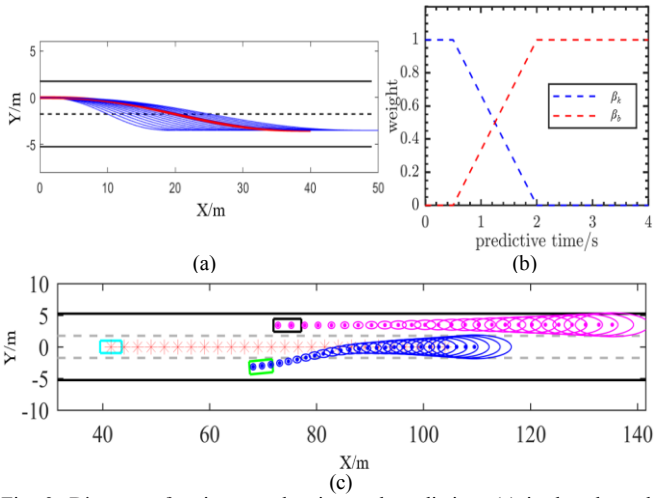


Fig. 3. Diagram of trajectory planning and prediction: (a) is the planned trajectory with m candidates, (b) is the fuzzy confidence coefficient β_k and β_b, (c) shows the planned trajectory of EV and predicted trajectories of SVs with their position uncertainty under 0.9 confidence.

based on three aspects: the path length L, the maximum lateral acceleration of the trajectory point $a_{y\max}$, and the degree of trajectory continuation ε. Here, ε represents the deviation from the historical trajectory of EV after extending the candidate trajectory in the negative direction with ΔL length. We normalize these 3 metrics and then form the selection objective $\mathcal{G}$ with corresponding weight coefficients η_1, η_2, η_3. $s(\cdot)$, $l(\cdot)$), $l'(\cdot)$ and $\kappa(\cdot)$ represent the position constraints at the start and end of the $\mathcal{T}^{(c)}$. In addition, the absolute value of curvature $|\kappa|$ at each trajectory point also needs to satisfy the constraint of minimum steering radius according to the limit of maximum lateral acceleration $A_{y\max}$ and front wheel steering angle $\delta_{f,\max}$.

Also, a predictor is introduced which estimates the possible future trajectories $\mathcal{T}^{(sv)}$ of surrounding vehicles. The future trajectories of SVs are designed as,

$$\mathcal{T}^{(sv)}=\beta_k\mathcal{T}_k^{(sv)}+\beta_b\mathcal{T}_b^{(sv)} \quad (10)$$

where $\mathcal{T}_k^{(sv)}$ and $\mathcal{T}_b^{(sv)}$ are the predicted trajectories given by the kinematics model and the behavior model, and β_k and β_b are the confidence coefficients calculated from the Mamdani fuzzy system. Each prediction trajectory consists of future poses $\zeta_i(k)$ (including longitudinal and lateral positions ($s_i(k)$, $l_i(k)$ and heading $\varphi_i(k)$) and velocities ($v_{si}(k)$, v_{li} (k))) in the prediction horizon h_p,

$$\mathcal{T}^{(sv)}(i)=\left\{\zeta_i(k)\right\}^T,\ i=0\sim n\in\mathbb{N},k=0\sim h_p\in\mathbb{N} \quad (11)$$

where i in (11) represents the number of SVs. Then, we construct Gaussian uncertainty distributions $\mathcal{U}^{(sv)}\sim N(\mathcal{T}^{(sv)},\Sigma^{(sv)})$

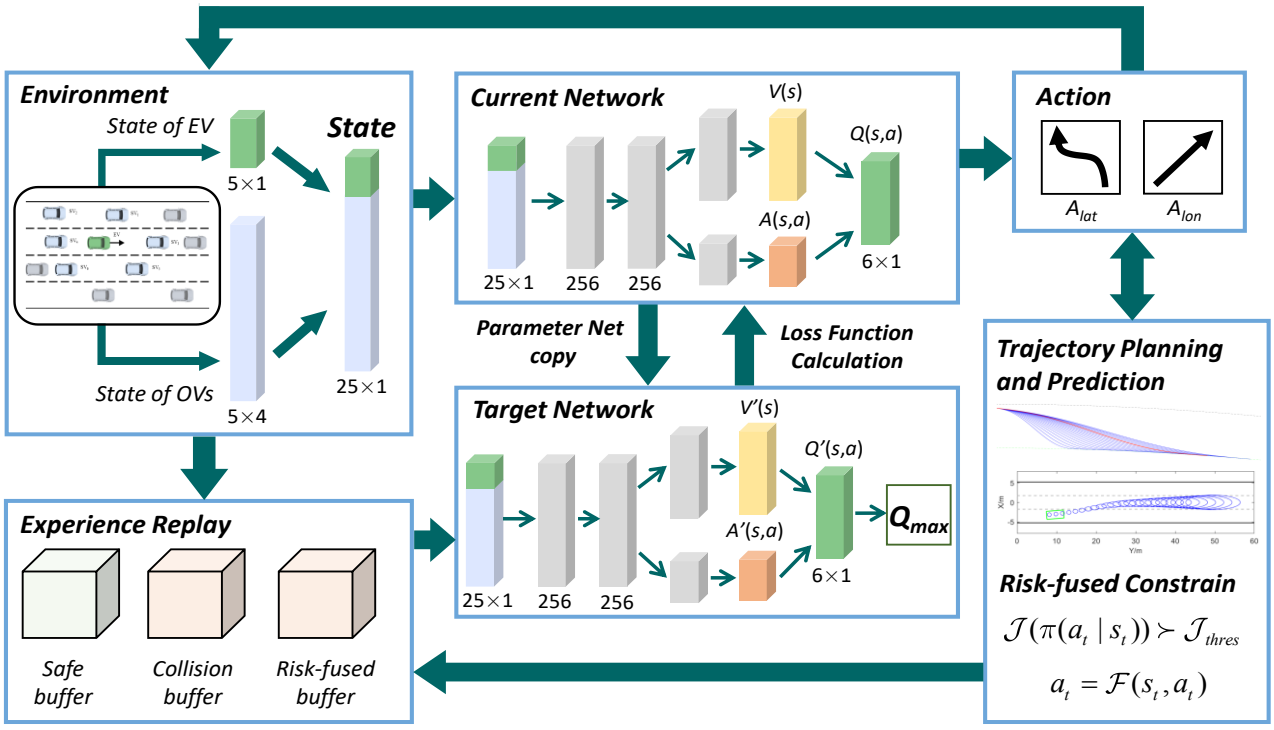


Fig. 4. The whole structure of RCDRL algorithm.

for the predicted trajectories using the standard approximation explained in detail in [29]. The schematic diagram of trajectory planning and prediction is shown in Fig. 3.

C. Risk-fused Constraint

Based on the $\mathcal{T}^{(ev)}$, $\mathcal{T}^{(sv)}$ and $\mathcal{U}^{(sv)}$, the security risk cost $\mathcal{J}$ can be expressed as,

$$\begin{aligned}\mathcal{J}(\pi(a_t \mid s_t)) &= V_{risk}(\mathcal{T}^{(ego)}, \mathcal{T}^{(pre)}, \mathcal{U}) \\ &= \omega_1 \frac{n_{risk}}{h_P} + \omega_2 P_{c\max} + \omega_3 \frac{T_p}{TTP}\end{aligned} \tag{12}$$

where n_{risk} is the number of high-risk trajectory points, $P_{c\max}$ is the peak of the collision probability $P_c(k)$, and TTP (Time to Peak) the time to reach the $P_{c\max}$. (12) is normalized through h_p and predicted duration T_p=$h_p \cdot \Delta T$ (ΔT is the sample step).

The collision probabilities are calculated by the Monte-Carlo sampling method. The collision probabilities are calculated by Monte Carlo sampling. The collision probability between the k^{th} point $\zeta_{ev}(k)$ of $\mathcal{T}^{(ev)}$ and $\mathcal{T}^{(sv)}$ with uncertainty $\mathcal{U}^{(sv)}$ can be expressed as,

$$P_c(k) = \left(\frac{1}{i} \sum_{i=1}^{n} \sum_{j=1}^{N_s} I_j(\zeta_{ev}(k), \zeta_i(k), \mathcal{U}_i^{(sv)}(k)) \right) \tag{13}$$

where $I_j(\zeta_{ev}(k), \zeta_i(k)) = \{0,1\}$ indicates whether $\zeta_{ev}(k)$ and $\zeta_i(k)$ have a collision in j^{th} sampling according to the intersection of rectangles generated by the two trajectory points. N_s is the is the maximum sampling size. if $P_c(k) > P_{c0}$, $\zeta_{ev}(k)$ is regarded as the high-risk trajectory point.

In the early stages of training, to avoid the over constraining and not taking full advantage of the dangerous training data, $\mathcal{J}_{thres}$ is designed to gradually decrease as the episode increased. When $\mathcal{J}(\pi(a_t \mid s_t)) > \mathcal{J}_{thres}$, the current decision action a_t is regarded as unsafe and should be reselected by the logical safe rule $\mathcal{F}$ (detail in Algorithm 1).

Algorithm 1: logical safe rule $\mathcal{F}$ for unsafe decision action a_t

define $TTC_m(a_i)$: the min TTC between the front and rear SVs on the target lane $\mathcal{L}_i$ corresponding to a_i.
define $(i - j) = \Delta(a_i - a_j)$: the change size of a_i and a_j.
define N_e is the total number of episodes, t is the current episode number

$\mathcal{J}_{thres} = (N_e / t) \cdot \mathcal{J}_0$
while $(\mathcal{J}(\pi(a_t|s_t)) > \mathcal{J}_{thres})$
  if $a_t \in \mathcal{A}_{lat}$
    remove a_t from $\mathcal{A}_{lat}$
    for $a_i \in \mathcal{A}_{lat}$
    a_t = argmax $(TTC_m(a_i))$
  if $a_t \in \mathcal{A}_{lon}$
    remove a_t from $\mathcal{A}_{lon}$
    for $a_i \in \mathcal{A}_{lon}$
    a_t = argmin $(\Delta(a_i - a_t))$
return a_t

D. Reward Design

The reward function considers three aspects of driving efficiency, safety and traffic manners.

The efficiency reward r_e mainly considers the velocity of EV, which can be calculated as,

$$r_e = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \tag{14}$$

where $v_{\max}$ and $v_{\min}$ are the speed limit of EV.

The safety reward r_s mainly considers the longitudinal gap d_i between EV and SVs on the target lane with a_t, in addition

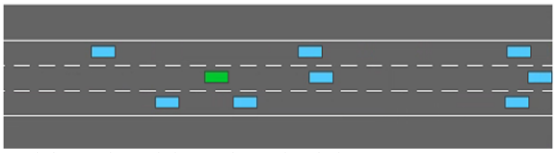


Fig. 5. High-env for training and test simulation

to giving a large negative reward when $\mathcal{J}(\pi(a_t \mid s_t)) > \mathcal{J}_{thres}$ or collision happened. Especially, we design d_e as the emergency distance between two vehicles and d_c = 1 as the collision distance. Thus, r_s can be calculated as,

$$r_s = \begin{cases} -0.1 & \text{if } d_i < d_e \\ -0.5 & \text{if } d_i < d_c \\ -0.3 & \text{if} \mathcal{J}(\pi(a_t \mid s_t)) > \mathcal{J}_{thres} \\ -1 & \text{if collided} \\ 0 & \text{else} \end{cases} \tag{15}$$

$$d_e = \frac{1}{2A_{\max}}(v^2 - v_i^2) + vT_{react}$$

Besides, we do not want EV to make frequent and ineffective lane changes or ac/deceleration, which is impolite and affects the driving comfort of SVs. The traffic manners reward r_t is calculated as,

$$r_t = \frac{\Delta(a_t - a_{t-1})}{2} \tag{16}$$

The total reward can be expressed as,

$$r = k_e r_e + k_s r_s + k_t r_t \tag{17}$$

where the corresponding weights [k_e, k_s, k_t] are set as [0.5, 1, 0.1] in this work.

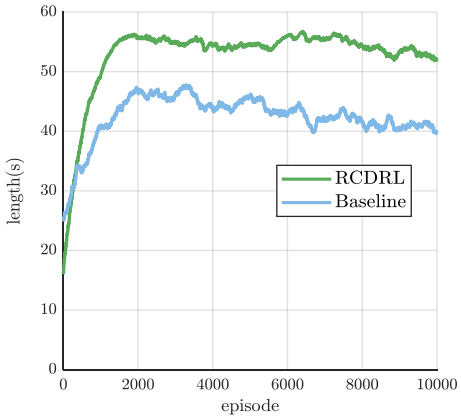
E. Network Architecture

The whole structure of the risk-fused D3QN network is introduced in Fig. 4. The input states are converted into a 25×1 vector and the output policy is a 6×1 action vector. The common feature is extracted through two fully connected layers (256×256). After that, it is divided into value function ($V(s)$) and advantage function ($A(s,a)$=$Q(s,a)$-$V(s)$) streams which solve the overestimation problem. We set up three experience buffers to store state transition information < s, a, $\mathcal{R}$, s' > of the common cases, collision cases and risk-fused cases. During the training and testing, the trajectory of EV is implemented by a kinematic controller, and the traffic vehicles are controlled by the controller from [30]. Each decision action will be implemented in 15 time-steps.

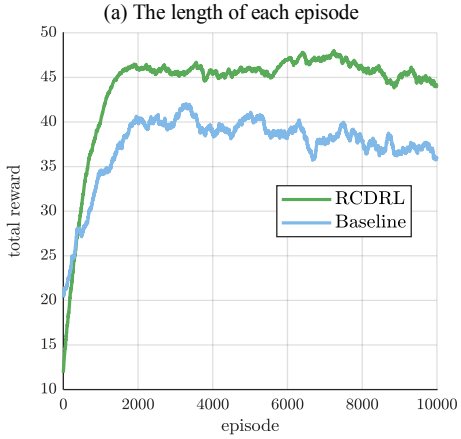
IV. Experiments and Evaluation

A. Simulation Results

We demonstrate the proposed method using an open-source autonomous driving simulation environment High-env [31]. As shown in Fig. 5, the EV (agent) drives on a 3-lanes unidirectional highway with different traffic densities. We trained the RCDRL with D3QN network and compared it to the baseline D3QN. The performance of the agent is evaluated by comparing the learning and testing curves, as well as the collision rate and risk fuse rate, etc. TABLE. I. lists the environment parameters, where the traffic density $\mu \in [0,1]$ represents the congestion of vehicles on the road.



(a) The length of each episode



(b) The total reward of each episode

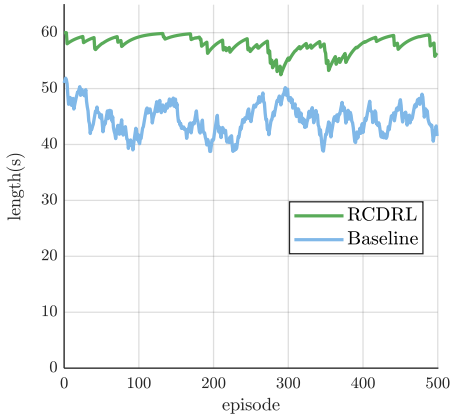
Fig. 6. Learning curves of the two policy during training.

TABLE I. Environment Parameters

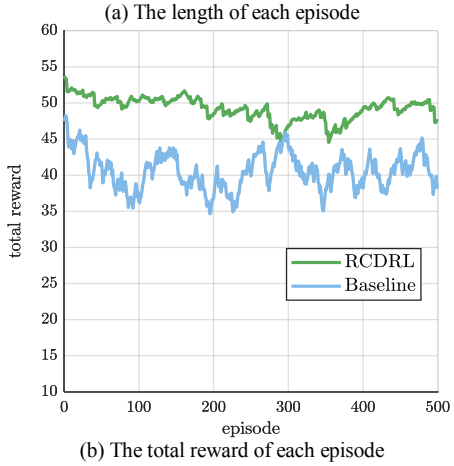
Parameters	Value	
EV speed (m/s)	[0,15]	
Traffic vehicles' speed (m/s)	[5,10]	
Max episode length (s)	60	
Total number of episodes	training	10000
	test	500
Traffic density μ	training	0.5
	test	0.3/0.4/0.5/0.6
Learning rate α	0.2	
Gamma γ	0.9	
Activation function	ReLU	

Fig. 6. shows the learning curves of the proposed RCDRL. During training, the RCDRL is able to obtain longer average episode length and cumulative returns faster and since the introduction of Risk-fused Constraint, while the Baseline ends the episode earlier due to collisions. This performance difference is better reflected by the learning curves during testing with μ=0.5, as shown in Fig. 7

We test four groups with different traffic densities of 500 episodes respectively and summarize the testing results in TABLE. II. and III. Specifically, the RCDRL eliminated the Risk-fused Constraint module during the testing to intuitively demonstrate the performance of the generated policy.



(a) The length of each episode



(b) The total reward of each episode

Fig. 7. Learning curves of the two policy during testing.

TABLE II. TESTING PERFORMANCE RESULTS OF EACH POLICY

Traffic density		0.3	0.4	0.5	0.6
Average speed (m/s)	RCDRL	12.5	11.1	10.8	10.2
	Baseline	12.9	12.8	12.2	11
Average length of episodes (s)	RCDRL	58.9	58.5	57.8	56.1
	Baseline	51.3	49.7	44.2	34.7
Success rate (%)	RCDRL	95.8	93.2	89.8	84.2
	Baseline	65.2	58.6	40.2	25.8
Total reward	RCDRL	51.1	51.6	49	46.2
	Baseline	45.5	45.8	40.2	30.6
Number of collisions	RCDRL	12	26	36	52
	Baseline	175	209	303	373
Collision rate (%)	RCDRL	0.04	0.08	0.12	0.17
	Baseline	0.58	0.70	1.01	1.24

TABLE. II. shows the testing performance of the proposed RCDRL and Baseline. The average length of episodes and the success rate improved significantly. It can be shown that the RCDRL greatly reduces the collision rate, resulting in only a 0.1025% probability of unsafe decisions and lower risk with surrounding vehicles when making lane change decisions under the same environment. The average speed of RCDRL is only slightly reduced compared to Baseline, which shows that the proposed safety model does not drive too conservatively

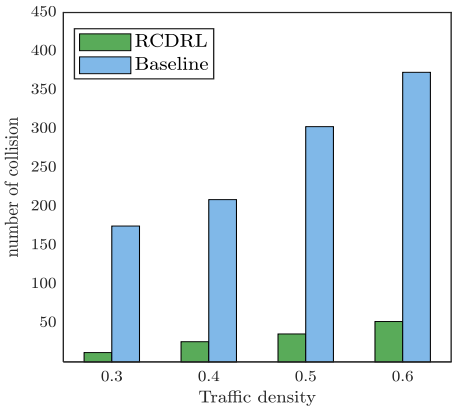


Fig. 8. Number of collisions with different traffic density in simulation test.

Fig. 9. The driving process of overtaking real test.

and impedes the traffic flow. Fig. 8. represents the number of collisions for different traffic densities in the test. The blue bars resent the number of collisions for the Baseline policy, while the green bars report the collision number for the RCDRL policy. As the traffic density increases, the number of collisions also increases as expected, while it is significantly reduced with the policy of Risk-fused Constraint safety module, by average 88.72% compared to Baseline.

B. Real Vehicle Tests

To further validate the policy effectiveness of the proposed method, we conducted a real vehicle test in a lane change overtaking scenario. The test platform is a modified electric vehicle from BYD Qin-Pro, equipped with a high-precision positioning system including GPS/INS. The DRL algorithm is converted from Python to C++ and runs on the industrial computer with RTX 2080 Ti GPU.

The initial distance between EV and SV1 is 50m, while between EV and SV_2 are 10m and 5m, respectively in 2 cases. SV_1 and SV_2 slowly accelerate from 0 km/h to 10km/h and then maintains the speed. As shown in Fig.9., the vehicle in front (SV_1) and on the left (SV_2) are driving in a straight line at low speed. After overtaking SV_2, the EV insert the gap to left lane change and then overtake SV_1.

With the Baseline algorithm, EV starts the left lane change only after losing SV_2 target in all cases, which causes the EV to follow SV_1 and loses a lot of speed. In contrast, RCDRL algorithm normally completes the lane change and overtaking decision with large Time headway in all cases. More results are recorded in the TABLE. III. showing RCDRL can find the time to change lanes more safely and in time.

TABLE III. REAL VEHICLE TEST RESULTS

	Case 1		Case 2	
	RCDRL	**Baseline**	**RCDRL**	**Baseline**
Average speed (m/s)	5.35	3.36	5.21	4.63
left lane change Time headway from SV_1 (s)	8.77	5.14	5.59	5.43
left lane change Time headway from SV_2 (s)	3.03	/	2.49	/
right lane change Time headway from SV_1 (s)	5.02	4.03	3.81	3.24

V. CONCLUSION

In this paper, we propose a Risk-fused Constraint Deep Reinforcement Learning (RCDRL) framework with a D3QN network for safe decision making in lane change maneuver. We specially design a Risk-fused Constraint module with rule-based risk assessment and motion correction by mapping DRL behavior actions into trajectories through trajectory planner and predictor, which can deal with unsafe decision actions when training the policy network. Simulation and real vehicle tests demonstrate that the proposed RCDRL method achieves better performance especially in safe decision making. In addition, this framework can be extended to other advanced DRL networks in the future.

REFERENCES

[1] Z. Li, L. Xiong and B. Leng, "A Unified Trajectory Planning and Tracking Control Framework for Autonomous Overtaking Based on Hierarchical MPC," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 937-944.

[2] Y. Huang, H. Wang, A. Khajepour, H. Ding, K. Yuan, and Y. Qin, "A Novel Local Motion Planning Framework for Autonomous Vehicles Based on Resistance Network and Model Predictive Control," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 55–66, Jan. 2020.

[3] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2076-2087, Apr. 2021.

[4] L. Claussmann, M. Revilloud, D. Gruyer and S. Glaser, "A Review of Motion Planning for Highway Autonomous Driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826-1848, May. 2020.

[5] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh and J. Fredriksson, "If, When, and How to Perform Lane Change Maneuvers on Highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 4, pp. 55–66, Win. 2016.

[6] G. Xiong, Y. Li, S. Wang, X. Li and P. Liu, "HMM and HSS based social behavior of intelligent vehicles for freeway entrance ramp," *Int. J. Control Autom.*, vol. 7, no. 10, pp. 79-90, 2014.

[7] N. Li, H. Chen, I. Kolmanovsky and A. Girard, "An Explicit Decision Tree Approach for Automated Driving," in *Proc. ASME Dyn. Syst. and Control Conf.*, 2017.

[8] C. Xia, M. Xing and S. He, "Interactive Planning for Autonomous Driving in Intersection Scenarios Without Traffic Signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24818–24828, Dec. 2022.

[9] C. Wei, Y. He, H. Tian and Y. Lv, "Game Theoretic Merging Behavior Control for Autonomous Vehicle at Highway On-Ramp," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21127–21136, Nov. 2022.

[10] J. Li, L. Sun, J. Chen, M. Tomizuka and W. Zhan. "A Safe Hierarchical Planning Framework for Complex Driving Scenarios based on Reinforcement Learning," in *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, May. 2021.

[11] H. Lu, C. Lu, Y. Yu, G. Xiong and J. Gong, "Autonomous Overtaking for Intelligent Vehicles Considering Social Preference Based on Hierarchical Reinforcement Learning," *Automot. Innov.*, vol. 5, no. 2, pp. 195-208, Apr. 2022.

[12] N. Brown and T. Sandholm, "Superhuman ai for heads-up no-limit poker: Libratus beats top professionals," *Science.*, vol. 359, no. 6374, pp. 418– 424, Dec. 2017.

[13] D. Silver, A. Huang, C.J. Maddison, et al., "Mastering the game of go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[14] S. Aradi, "Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022.

[15] B. R. Kiran, I. Sobh, V. Talpaert, et al, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[16] Y. Yu, C. Lu, L. Yang, Z. Li, F. Hu, J. Gong, "Hierarchical Reinforcement Learning Combined with Motion Primitives for Automated Overtaking," *in Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020.

[17] Y. Zhang, B. Gao, L. Guo, H. Guo and H. Chen, "Adaptive Decision-Making for Automated Vehicles Under Roundabout Scenarios Using Optimization Embedded Reinforcement Learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5526-6638, Dec. 2021.

[18] W. Zhou, K. Jiang, Z. Cao, N. Deng and D. Yang, "Integrating deep reinforcement learning with optimal trajectory planner for automated driving," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020.

[19] P. Fang, Z. Yu, L. Xiong, et. al., "A Maximum Entropy Inverse Reinforcement Learning Algorithm for Automatic Parking," *in Proc. CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2021.

[20] Z. Cao, S. Xu, H. Peng, D. Yang and R. Zidek, "Confidence-Aware Reinforcement Learning for Self-Driving Cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7419–7430, Jul. 2022.

[21] S. Gu, L. Yang, Y. Du, et al., "A review of safe reinforcement learning: Methods, theory and applications," *arXiv*: 2205.10330, 2022.

[22] W. Zhao, T. He, R. Chen, T. Wei, and C. Liu, "State-wise Safe Reinforcement Learning: A Survey," *arXiv*:2302.03122,2023.

[23] J. Lubars, H. Gupta, S. Chinchali, et al., "Combining Reinforcement Learning with Model Predictive Control for On-Ramp Merging," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2022, pp. 942-947.

[24] A. Baheri, S. Nageshrao, H.E. Tseng, et al., "Deep Reinforcement Learning with Enhanced Safety for Autonomous Highway Driving," *in Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1550-1555.

[25] Y. Ye, X. Zhang, J. Sun, "Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment," *Transp. Res. Part C, Emerg. Technol.*, vol. 107, pp. 155-170, Oct. 2019.

[26] C. J. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, 1992.

[27] L. Xiong, Z. Fu, D. Zeng and B. Leng, "Surrounding Vehicle Trajectory Prediction and Dynamic Speed Planning for Autonomous Vehicle in Cut-in Scenarios," *in Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 987-993.

[28] B. Peng, Q. Sun, S.E. Li, et al., "End to End Autonomous Driving Through Dueling Double Deep Q Network," *Automot. Innov.*, vol. 3, no. 4, pp. 328-337, Aug. 2021.

[29] H. Chen, X. Wang, J. Wang, "A Trajectory Planning Method Considering Intention-aware Uncertainty for Autonomous Vehicles," *in Proc. Chinese Automation Congress (CAC)*, Nov. 2018, pp. 1460-1465.

[30] N. Li, D.W. Oyler, M. Zhang, et al., "Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems," *IEEE Trans. Control Syst. Technol* , vol. 26, no. 5, pp.1782–1797, Sep. 2018.

[31] E. Leurent, "An Environment for Autonomous Driving Decision Making," *GitHub repository*, 2018, Available: https://github.com/eleu-rent/highway-env.