Risk-Aware Reinforcement Learning for Autonomous Driving: Improving Safety When Driving through Intersection

Bo Leng, Ran Yu, Wei Han, Lu Xiong, Zhuoren Li and Hailong Huang

***Abstract*—Applying reinforcement learning to autonomous driving has garnered widespread attention. However, classical reinforcement learning methods optimize policies by maximizing expected rewards but lack sufficient safety considerations, often putting agents in hazardous situations. This paper proposes a risk-aware reinforcement learning approach for autonomous driving to improve the safety performance when crossing the intersection. Safe critics are constructed to evaluate driving risk and work in conjunction with the reward critic to update the actor. Based on this, a Lagrangian relaxation method and cyclic gradient iteration are combined to project actions into a feasible safe region. Furthermore, a Multi-hop and Multi-layer perception (MLP) mixed Attention Mechanism (MMAM) is incorporated into the actor-critic network, enabling the policy to adapt to dynamic traffic and overcome permutation sensitivity challenges. This allows the policy to focus more effectively on surrounding potential risks while enhancing the identification of passing opportunities. Simulation tests are conducted on different tasks at unsignalized intersections. The results show that the proposed approach effectively reduces collision rates and improves crossing efficiency in comparison to baseline algorithms. Additionally, our ablation experiments demonstrate the benefits of incorporating risk-awareness and MMAM into RL.**

***Index Terms*—autonomous vehicles, reinforcement learning, safety, intersection.**

I. INTRODUCTION

As one of the most challenging autonomous driving (AD) tasks, navigating through intersection brings inevitable interactions that require a comprehensive consideration of safety, efficiency, timing, and other factors. Traditional rule-based approaches prone to overly conservative or inconsistent driving strategies in this complex condition, making it difficult to pass through safely and efficiently [1].

Recent advancements in reinforcement learning (RL) have highlighted its potential to surpass human driving capabilities, owing to its superior handling of high-dimensional state spaces and adaptability to complex scenarios [2]. RL technologies have been extensively explored in various scenarios, including highway, merging, intersection, etc. [3], [4]. RL optimizes policies by maximizing expected rewards. However, classic RL agent may exhibit unsafe behaviors due to a lack of safety consideration. For safety-critical driving tasks, it's essential to not only maximize rewards but also to incorporate safety guarantees to prevent accidents [5]. Consequently, the effective application of RL while ensuring safety has become an urgent challenge for promoting AD.

To address this issue, Safe RL is introduced to ensuring compliance with safety constraints while maximizing rewards. Some approaches enhance safety by introducing safety factors or risk measures into the objective or reward function [6], [7]. The essence of this category is to modify the policy gradient to update the policy in the direction of the feasible region, thus improving the safety of agent [8]. While they can enhance safety to some extent, their safety performance significantly deteriorates when confronted with intricate scenarios, resulting in a higher incidence of constraint violations. Other approaches identify unsafe actions during the agent's exploration phase and project them onto a safe set, thereby ensuring fewer or even zero constraint violations during training [9], [10]. While these methods offer better state-wise safety, they typically require accurate system dynamics or other prior knowledge [5], [11], or are designed for specific applications with constraints of a particular form [12], [13].

In unsignalized intersection scenarios, autonomous vehicles (AVs) encounter traffic from multiple directions, creating potential conflict risks. To represent the surrounding environment, existing studies typically concatenate the AV's state with environmental information into a feature vector and implement policy mapping through a multi-layer perception (MLP) [14], [15]. However, these methods face two major challenges: *dimension sensitivity*, where traditional MLPs rely on fixed-dimensional feature vector inputs, making it difficult to adapt to the dynamic number of traffic participants, and *permutation sensitivity*, where irregular traffic flows lead to abrupt changes in interacting objects and spatial relationships between adjacent time steps, further complicating state characterization and decision-making [16]. Some approaches use grid maps [17] or bird's eye view (BEV) [18] to represent environmental features and address these issues. Nevertheless, the discretization of grid divisions and downsampling in image encoding can result in the loss of fine-grained information. In [16], an encoding sum and concatenation (ESC) method

Manuscript created 2024; This work was developed by the IEEE Publication Technology Department.

This work was supported by the National Natural Science Foundation of China under Grant 52325212, the Science and Technology Commission of Shanghai under Grant 21DZ1203802, and the Shanghai Automotive Industry Science and Technology Development Foundation under Grant 2203. *(corresponding author: Zhuoren Li).*

Bo Leng, Ran Yu, Wei Han, Lu Xiong and Zhuoren Li are with the School of Automotive Studies, Tongji University, Shanghai 201804, China (email: lengbo@tongji.edu.cn; 2433113@tongji.edu.cn; tjhanwei@tongji.edu.cn; xiong_lu@tongji.edu.cn; 1911055@tongji.edu.cn).

Hailong Huang is with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong, China (email: hailong.huang@polyu.edu.hk).

is proposed, where an MLP maps each surrounding vehicle (SV) to an feature vector, and they are added element-wise to form the surrounding state representation. However, equal-weight summation struggles to filter out information that is strongly correlated with the ego vehicle (EV). Attention mechanism has been widely used in RL policy construction to capture relationships between features, thereby improving the policy's ability to understand environmental information [19], [20], [21]. Inspired by the transformer architecture [22], we incorporate the attention mechanism into the network to effectively handle dynamic traffic flow.

The inability to handle the dynamic changes in the number and permutation of surroundings traffic participants may make it difficult for AVs to identify potential risks and adopt unsafe strategies. Moreover, the complex information at these intersections requires the AV to identify pivotal data to ascertain the timing of passage. To improve safety and efficiency for driving through intersection, a risk-aware RL approach is proposed in this paper and the main contributions are summarized as follows:

- Safe critics are constructed to evaluate driving risk and work in conjunction with the reward critic to update the actor. A Lagrangian relaxation method is incorporated to generate approximate safe actions, which are projected into a feasible safe region with safety iterative correction by cyclic gradient descent.
- A **M**ulti-hop and **M**ulti-layer perception mixed **A**ttention **M**echanism (**MMAM**) integrated into the actor-critic network enables the policy to adapt to dynamic traffic and overcome permutation sensitivity challenges, enhancing scene understanding and improving decision-making timing when navigating intersections.
- The proposed approach is evaluated through comparative experiments, as well as ablation studies, demonstrating its effectiveness in terms of safety and efficiency.

II. Related Works

A. Safe RL Methods

Algorithms based on Safe RL aim to constrain risks within a given threshold or to avert constraint violations. Many approaches use the Lagrange multipliers to transform the constrained optimization problem into an unconstrained one [23], [24], or utilize the trust region approach, which ensures policy feasibility and stability by constructing an approximate objective at each iteration and restricting the update range [25], [26]. These methods addresses constraints implicitly, but it can only ensure limited safety and is still prone to severe constraint violations in complex scenarios. Another notable branch ensures the safety during training by preventing the agent from exploring risky behaviors. Some algorithms leverage control theory by constraining the agent to a designated feasible region. Zhang *et al*. [9] and Cheng *et al*. [10] integrate RL with Lyapunov functions and Control Barrier Functions (CBF), respectively, to ensure that state trajectories remain within a safe feasible region. However, these algorithms often require manual specification of safety constraint functions, and accurately determining these functions can be challenging. Dalal *et al*. [12] introduce a safety layer that directly modifies the output actions, linearly mapping the original policy to a safe set to ensure safety. However, methods based on linearization assumptions may not accurately represent system dynamics and can lead to approximation errors.

B. Safe RL Methods for Autonomous Driving

In safety-critical tasks such as autonomous driving, ensuring safety is essential to prevent catastrophic accidents. Some algorithms enhance safety by incorporating additional safety constraint objectives. For instance, Li *et al*. [27] evaluate driving risks using probabilistic models that account for position uncertainty and distance-based safety metrics. Similar initiatives have introduced risk assessment and trade-offs within DRL [28], [29]. While the above methods can improve security, it is challenging to avoid the decline in safety performance caused by constraint violations. Krasowski *et al*. [11] propose a framework based on vehicle trajectory prediction, which incorporates a safety layer to mask unsafe actions. Similarly, Chen *et al*. [19] develop a lightweight safety layer designed to identify and eliminate unsafe actions in advance. Wang *et al*. [30] modify the actions during exploration to obtain approximate safe actions and used them to train safe strategies. Moreover, some algorithms employ reachability analysis to assess the safety of vehicle trajectories. Notably, Wang *et al*. [31] introduce an online reachability analysis algorithm that calculates the occupancy of both the vehicle and surrounding trajectories, ensuring the safety of the vehicle's path.

III. Preliminaries

A. Constrained Markov Decision Process

In this paper, Safe RL is modeled as a Constrained Markov Decision Process (CMDP), which extends the standard MDP to a heptuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{C}, \rho, \gamma)$. $\mathcal{S}$ and $\mathcal{A}$ are denoted as state space and action space respectively. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, which represents the system dynamic. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. $\mathcal{C} : \mathcal{S} \times \mathcal{A} \mapsto [0, +\infty]$ maps the state action transition tuple into a cost value and reflects the constraint violation. $\rho : \mathcal{S} \to [0, 1]$ is the initial state distribution and γ is the discount factor for future reward and cost. Policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a map from given states to a probability distribution over action space. In standard MDP, the goal is to optimize the policy by maximizing the agent's cumulative discounted reward:

$$\mathcal{J}_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t\mathcal{R}(s_t, a_t)\right], \tag{1}$$

where, $\tau = [s_0, a_0, s_1, \cdots]$, and $\tau \sim \pi$ stands for the stochastic trajectory distribution depended on $s_0 \sim \rho, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$. CMDP is required to optimize the agent's rewards while guaranteeing that the agent satisfies safety constraints. Hence, CMDP can be formulated as the following constrained optimization problem:

$$\max_{\pi\in\Pi_S} \mathcal{J}_R(\pi), s.t. \mathcal{J}_C(\pi) \leq b, \tag{2}$$

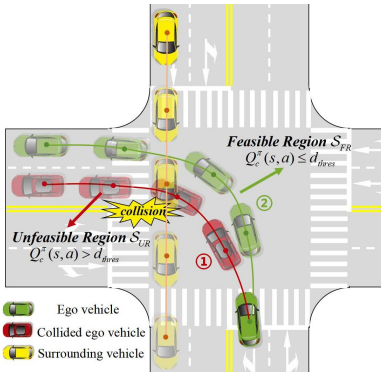


Fig. 1. Diagram of feasible region $\mathcal{S}_{FR}$ and unfeasible region $\mathcal{S}_{UR}$.

where, Π_S is the set of policies π, $b \in \mathbb{R}$ is the constraint threshold. The goal of CMDP is to find a feasible policy set satisfying the cost constraints, i.e., $\Pi_C = \{\pi \in \Pi_S \mid \mathcal{J}_C(\pi) \leq b\}$. Similar to the definition of $Q^\pi(s,a)$ in standard RL, safe critic $Q_c^\pi(s,a) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t'=t}^{\infty} \gamma^t c_{t'}\right] \leq b$ represents the cost-return over a certain time horizon.

B. Safety Correction

The objective is to find the feasible policy that satisfies the safety constraint d_{thres} by defining a safe critic $Q^\pi(s,a)$ as state-wise constraints:

$$\pi^* = \arg\max_\pi \left[Q^\pi(s,a)\right] \quad \text{s.t.} Q_c^\pi(s,a) \leq d_{thres}. \tag{3}$$

As shown in Fig.1, the feasible region $\mathcal{S}_{FR}$: $Q_c^\pi(s,a) \leq d_{thres}$ is defined, and any state s_t within the feasible region $\mathcal{S}_{FR}$ satisfies:

$$\forall s_t \in \mathcal{S}_{FR}, s_{t+i} \in \mathcal{S}_{FR}, \forall i \in \mathbb{N}_+. \tag{4}$$

Similarly, unfeasible region $\mathcal{S}_{UR}$ is defined as the region where $Q_c^\pi(s,a) > d_{thres}$, and the whole state space $\mathcal{S}_{All} = \mathcal{S}_{UR} \bigcup \mathcal{S}_{FR}$. When the ego vehicle is in $\mathcal{S}_{UR}$ and continues to execute its current policy, there is a significant probability of a collision occurring. To circumvent the aforementioned issue, a natural idea would be to correct the original unsafe actions a_{old} towards the feasible region while attempting to minimize the discrepancy between the new and old actions to the greatest extent possible:

$$\begin{aligned} &\arg\min \|a_{new} - a_{old}\| \\ &\text{s.t. } Q_c^\pi(s,a) \leq d_{thres}. \end{aligned} \tag{5}$$

C. Dimension Sensitivity and Permutation Sensitivity

1) Dimension Sensitivity: For the observation set $\mathcal{S}$, it typically contains EV-related information $\mathcal{S}_{EV} \in \mathbb{R}^{1\times d_{EV}}$, and SV-related information $\mathcal{S}_{SV} \in \mathbb{R}^{N_{SV}\times d_{SV}}$,where d_{EV}, d_{SV} denote the feature dimension of EV and SVs respectively, and $N_{SV} \in [1, N] \cap \mathbb{N}$ is the number of observed SVs, which changes dynamically with the traffic flow. That is, $\mathcal{S} = [\mathcal{S}_{EV}, \mathcal{S}_{SV}]$. Since architectures such as MLP typically require fixed input dimensions, many works use a fixed dimension feature vector, i.e., by specifying a potentially observed number of SVs, M_{SV}. However, if $M_{SV} < N_{SV}$, the additional vehicles will not be observable, leading to information loss; conversely, when $M_{SV} > N_{SV}$, it results in information redundancy.

2) Permutation Sensitivity: For the driving policy π, we expect it to be *permutation-invariant*. That is, for any two possible permutations ζ_1 and ζ_2 of the surrounding traffic participants, the policy π should output the same decision, namely:

$$\begin{aligned} &\pi(\cdot|(\mathcal{S}_{EV}, \mathcal{S}_{SV}^{\zeta_1(1)}, \ldots, \mathcal{S}_{SV}^{\zeta_1(N_{SV})})) \\ &= \pi(\cdot|(\mathcal{S}_{EV}, \mathcal{S}_{SV}^{\zeta_2(1)}, \ldots, \mathcal{S}_{SV}^{\zeta_2(N_{SV})})) \quad \forall \zeta_1, \zeta_2 \in \mathfrak{S}_{N_{SV}}. \end{aligned} \tag{6}$$

Conversely, if there exist two permutations ζ_1 and ζ_2 that make the output of the policy inconsistent, this is called *permutation sensitivity*:

$$\begin{aligned} &\pi(\cdot|(\mathcal{S}_{EV}, \mathcal{S}_{SV}^{\zeta_1(1)}, \ldots, \mathcal{S}_{SV}^{\zeta_1(N_{SV})})) \\ &\neq \pi(\cdot|(\mathcal{S}_{EV}, \mathcal{S}_{SV}^{\zeta_2(1)}, \ldots, \mathcal{S}_{SV}^{\zeta_2(N_{SV})})) \quad \exists \zeta_1, \zeta_2 \in \mathfrak{S}_{N_{SV}}. \end{aligned} \tag{7}$$

IV. Methodologies

Directly correcting an unsafe action to a safe one is quite challenging. Hundreds of iterations may be required to project an unsafe action into $\mathcal{S}_{FR}$. Moreover, if the initial action is far from $\mathcal{S}_{FR}$, multiple iterations may still result in the action remaining in $\mathcal{S}_{UR}$. Therefore, we use a Lagrangian relaxation approach to obtain approximate safe action a_{init}. Then, Safety Iterative Correction is applied to a_{init} to obtain a feasible safe solution a_{new} when $Q_c^\pi(s, a_{init}) > d_{thres}$.

The overall framework is depicted in Fig. 2, which includes two pairs of reward critics $Q_{1,2}^\omega$ and target critics $Q_{1,2}^{\omega^-}$, as well as two pairs of safe critics $Q_{c_{1,2}}^\psi$ and target safe critics $Q_{c_{1,2}}^{\psi^-}$, all of which contribute to actor's π_θ policy updates and action risk evaluation. The pseudo-code is shown in Algorithm 1.

A. Approximate Safe Action Generation

The initial solution a_{init} is derived through the construction of a Lagrangian function for constrained policy optimization:

$$\begin{aligned} &\max_{\lambda\geq 0} \min_\theta \mathcal{L}(\theta, \lambda) \\ &= \max_{\lambda\geq 0} \{\min_\theta \mathbb{E}_{s_t\sim\mathcal{D}, a_t\sim\pi_\theta}\big[\alpha \log \pi_\theta(a_t \mid s_t) - Q^\omega(s_t, a_t) \\ &+\lambda\big(ReLU(Q_c^\psi(s_t, a_t) - d_{thres})\big)\big]\}, \end{aligned} \tag{8}$$

where λ is the Lagrange multiplier, α is the temperature parameter that dictates the relative significance of the entropy term compared to the reward and $\log \pi_\theta(\cdot)$ is entropy of policy π_θ. By employing a dual ascent strategy, the algorithm alternately updates the policy and the Lagrange multipliers, thereby gradually converging to the saddle point of the minimax problem:

$$\begin{aligned} \lambda &\leftarrow \lambda + \alpha_\lambda \nabla_\lambda \mathcal{L}(\theta, \lambda), \\ \theta &\leftarrow \pi - \alpha_\theta \nabla_\theta \mathcal{L}(\theta, \lambda), \end{aligned} \tag{9}$$

where α_λ, α_θ are the step sizes for the parameters λ, θ respectively. Lagrange multiplier functions operate analogously to penalty coefficients, enabling the policy to gradually converge within the constraints.

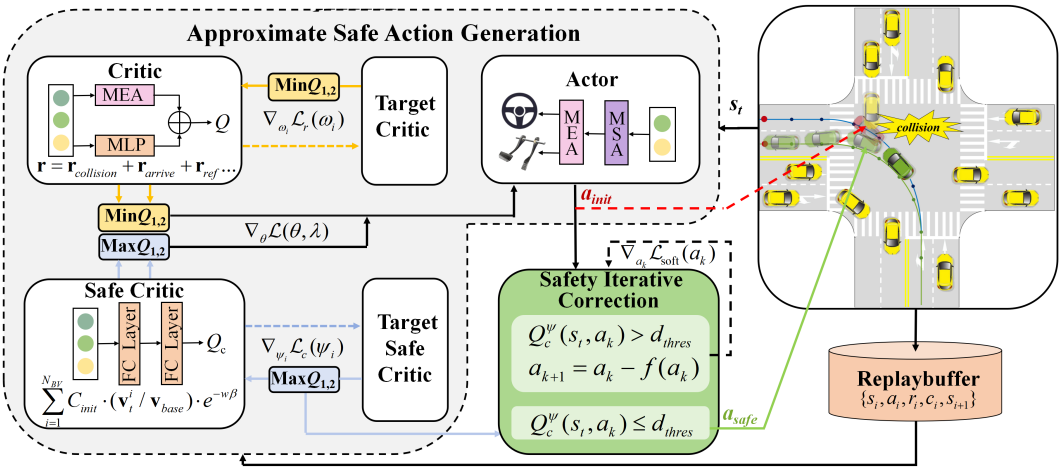


Fig. 2. Schematic of proposed framework. MEA and MSA stand for multi-head ego-attention and multi-head self-attention, respectively. $f(a_k) = \frac{\eta}{\mathcal{N}_k}\nabla_{a_k}\mathcal{L}_{\text{soft}}(a_k)$.

A dual-critic is employed in both reward critic and safe critic, to mitigate positive bias during the policy improvement step to address the overestimation issue. The larger $Q_c^\pi(s,a) = \max_{i=1,2} Q_{c,i}^\pi(s,a)$ is selected to reduce the risk of underestimation. Although it may lead to an overestimation of the $Q_c^\pi(s,a)$, it effectively increases the safety margin for overall reliability. Therefore, reward critic network Q^ω and safe critic network Q_c^ψ can be updated by as follows:

$$\begin{aligned}&\mathcal{L}_r(\omega_i) = \frac{1}{2}\left\{Q_i^\omega(s_t,a_t) - \left(r(s_t,a_t) + \gamma V_r^{\omega^-}(s_{t+1})\right)\right\}^2\\ &with \quad V_r^{\omega^-}(s_{t+1})\\ &= \min_{j=1,2} Q_{r,j}^\omega(s_{t+1},a_{t+1}) - \alpha\log\pi(a_{t+1}|s_{t+1}),\end{aligned} \tag{10}$$

$$\begin{aligned}&\mathcal{L}_c(\psi_i) = \frac{1}{2}\left\{Q_{c,i}^\psi(s_t,a_t) - \left(c(s_t,a_t) + \gamma V_c^{\psi^-}(s_{t+1})\right)\right\}^2\\ &with \quad V_c^{\psi^-}(s_{t+1}) = \max_{j=1,2} Q_{c,j}^\psi(s_{t+1},a_{t+1}),\end{aligned} \tag{11}$$

The temperature parameter can be adjusted adaptively. As the policy becomes more definitive in the later stages of training, the exploration capability can be appropriately reduced. Specifically, the update of the temperature parameter is guided by the following optimization objective:

$$\mathcal{L}(\alpha) = \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[-\alpha\log\pi(a_t \mid s_t) - \alpha\mathcal{H}_0], \tag{12}$$

where ρ_π is the state distribution under policy π and $\mathcal{H}_0$ is the target entropy.

B. Safety Iterative Correction

Based on (5), we refine the initial solution to ensure safety using the following soft loss function:

$$\begin{aligned}\mathcal{L}_{\text{soft}}(a_k) = &\frac{1}{2}\|a_k - a_{init}\|^2\\ &+ \lambda_a\big(ReLU(Q_c(s,a_k) - d_{thres})\big),\end{aligned} \tag{13}$$

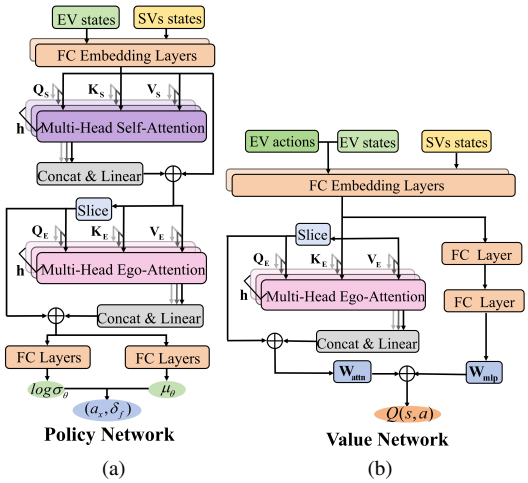


Fig. 3. Multi-hop and MLP-mixed Attention Mechanism (MMAM).

where k denotes the k-th iteration and λ_a represents the coefficient of the constraint. The corresponding gradient for k-th action a_k is:

$$\begin{aligned}\nabla_{a_k}\mathcal{L}_{\text{soft}}(a_k) = &\, a_k - a_{init}\\ &+ \lambda_a\frac{\partial\big(ReLU(Q_c(s,a_k) - d_{thres})\big)}{\partial a_k}.\end{aligned} \tag{14}$$

Ultimately, the gradient descent method is utilized to update the action a_k:

$$a_{k+1} = a_k - \frac{\eta}{\mathcal{N}_k}\nabla_{a_k}\mathcal{L}_{\text{soft}}(a_k), \tag{15}$$

where hyper-parameter η determines the update magnitude for each iteration and $\mathcal{N}_k = \|\nabla_{a_k}\mathcal{L}_{\text{soft}}(a_k)\|_\infty$ is the scaling factor normalize the gradients on a_k. a_k is expected to converge to the optimal action a_k^* as $k \to \infty$. Due to time constraints, a more practical approach is to set a maximum iteration limit N_{iter}. If this limit is reached without satisfying the constraints, the iteration will be terminated.

Algorithm 1 Risk-Aware Soft Actor-Critic

Initialize: parameters ω_1, ω_2, ψ_1, ψ_2, θ, λ; $\omega_1^- \leftarrow \omega_1, \omega_2^- \leftarrow \omega_2$, $\psi_1^- \leftarrow \psi_1, \psi_2^- \leftarrow \psi_2$; replay buffer $\mathcal{D} \leftarrow \emptyset$; learning rate $\alpha_r, \alpha_c, \alpha_\theta, \alpha_\lambda, \beta_\alpha$.

for each episode e **do**
  for each time-step t **do**
    Get state s_t and select action: $a_t \sim \pi(a_t|s_t)$.
    Get safe critic value:
    $Q_c^\psi(s_t, a_t) = \max_{i=1,2} Q_{c,i}^\psi(s_t, a_t)$.
    if $Q_c^\psi(s_t, a_t) > d_{thres}$ **then**
      for each iteration $k = 1 \rightarrow N_{iter}$ **do**
        $a_k \leftarrow a_k - \frac{\eta}{\mathcal{N}_k} \nabla_{a_k} \mathcal{L}_{\text{soft}}(a_k)$.
        if $Q_c^\psi(s_t, a_k) \leq d_{thres}$ **then**
          Break.
    Execute a_t, receive next state s_{t+1}, reward r_t and cost c_t. Store the transition $(s_t, a_t, r_t, c_t, s_{t+1})$ in replay buffer $\mathcal{D}$.
  for each training epoch i **do**
    Sample N transitions from replay buffer $\mathcal{D}$.
    Update the critic network and safe critic network:
    $\omega_j \leftarrow \omega_j - \alpha_r \nabla_{\omega_j} \mathcal{L}_r(\omega_j)$, for$j \in \{1, 2\}$,
    $\psi_j \leftarrow \psi_j - \alpha_c \nabla_{\psi_j} \mathcal{L}_c(\psi_j)$, for$j \in \{1, 2\}$.
    Update actor network and Lagrange multiplier:
    $\theta \leftarrow \theta - \alpha_\pi \nabla_\theta \mathcal{L}(\theta, \lambda)$, $\lambda \leftarrow \lambda + \alpha_\lambda \nabla_\lambda \mathcal{L}(\theta, \lambda)$.
    Update temperature parameter:
    $\alpha \leftarrow \alpha - \beta_\alpha \nabla_\alpha \mathcal{L}(\alpha)$.
    Soft update the target network ω_j^-, ψ_j^-.

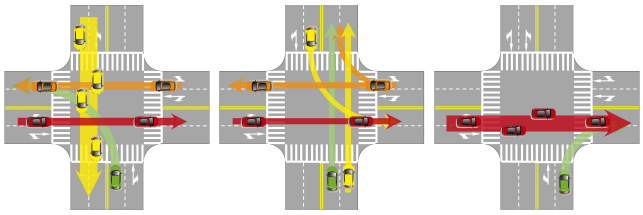
C. Attention embedded Actor-Critic Network

To tackle the dimension and permutation sensitivity, MMAM is incorporated into the actor-critic network, as shown in Fig. 3, which enhances the extraction of scene information and improves the scene comprehension capabilities of EV, allowing them to focus on potential risks more effectively.

With regard to the detailed architecture of the policy network, $\mathcal{S}_{EV} \in \mathbb{R}^{1\times d_{EV}}$ and $\mathcal{S}_{SV} \in \mathbb{R}^{N_{SV}\times d_{SV}}$ are processed through their respective fully connected embedding layers, then concatenated and mapped to the latent input matrix $\mathbf{Z}_1 \in \mathbb{R}^{\mathbf{N}\times d}$, where d represents the hidden size of networks, $\mathbf{N}$ is the total number of vehicles. Then, $\mathbf{Z}_1$ can be further transformed into query, key, and value matrices $\mathbf{Q_S}$, $\mathbf{K_S}$, $\mathbf{V_S} \in \mathbb{R}^{\mathbf{N}\times d}$, respectively, using a linear transformation operator $\mathcal{T} \in \mathbb{R}^{d\times d}$. Multi-head attention is subsequently employed to focus on different parts of the $\mathbf{Z}_1$ and the outputs of each head are then merged and transformed back to their original dimensions, as shown below:

$$\begin{aligned}
&\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})\mathbf{W}^O \\
&\text{where head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \\
&\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V},
\end{aligned} \tag{16}$$

where parameter matrices $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d\times d/h}$ and $\mathbf{W}^O \in \mathbb{R}^{d\times d}$. Subsequently, the output $\mathbf{Z}_1'$ is augmented with a residual connection, sliced, into and then linearly transformed to form the query $\mathbf{Q_E} \in \mathbb{R}^{1\times d}$ for ego-attention, in order to capture the interaction between EV and SVs. Ego-attention is a variant of self-attention wherein the query $\mathbf{Q}$ solely contains EV's features. This configuration establishes a 2-hop attention structure in conjunction with self-attention, facilitating the iterative integration and extraction of additional feature information through the sequential processing of the query and the latent matrix [32]. The parameters of the embedding and attention layers are independent of $\mathbf{N}$, allowing the models to adapt to dynamic input. Meanwhile, since the final result is the dot product of values and key similarities, the model is permutation invariant.



(a) (b) (c)

Fig. 4. Driving tasks and main conflicts at unsignalized intersection. (a) LT task, EV primarily encounters conflicts with oncoming traffic and some crossing traffic. (b) GS task with mixed traffic flow. (c) RT task with crossing traffic, EV needs to perform a right merge.

The value network incorporates the EV's actions as input. The input feature matrix $\mathbf{Z}_2 \in \mathbb{R}^{(N+2)\times d}$ is processed through both the ego-attention branch and the MLP branch, thereby enabling the model to capture relationships between the EV and SVs, as well as global information about states and environment. Let $\mathbf{Y}_{\text{attn}}, \mathbf{Y}_{\text{mlp}} \in \mathbb{R}^{1\times d}$ represent the outputs of the attention and MLP branches, respectively. A weighted sum of these outputs is computed using learnable weight vectors $\mathbf{W}_{\text{attn}}, \mathbf{W}_{\text{mlp}} \in \mathbb{R}^{d\times 1}$, yielding the final Q-value:

$$Q(s, a) = \mathbf{Y}_{\text{attn}} \cdot \mathbf{W}_{\text{attn}} + \mathbf{Y}_{\text{mlp}} \cdot \mathbf{W}_{\text{mlp}}. \tag{17}$$

V. Implementation

A. Environment Settings

We constructed a bidirectional four-lane intersection scenario based on Highway-Env [33] and designed three driving tasks: left-turn (LT), go-straight (GS), and right-turn (RT). As graphed in Fig. 4, to avoid sparse traffic flow caused by the random generation of SVs, which would simplify the task to a path-following problem, we specially design the difficulty of the driving task. To reduce collisions from random SV generation, we applied an improved intelligent driver model (IDM) [34] strategy that follows traffic rules. Each SV predicts its heading and position for the next 2 seconds, yielding to potentially colliding vehicles based on established road priorities. Each time the scenario is reset, 10 SV will be initialized and generated. The initial velocity of each SV is randomly generated within the range of [6 m/s, 10 m/s]. The minimum distance between vehicles is 15 m, and vehicles that do not meet this requirement will be removed. The EV is initialized with a random velocity and positioned in a lane where no collisions will occur at that moment. The simulation frequency f_s is 15 Hz, with the policy execution frequency f_π set to 5 Hz during training and 10 Hz during testing. The maximum length of each episode is 125 time steps (25s).

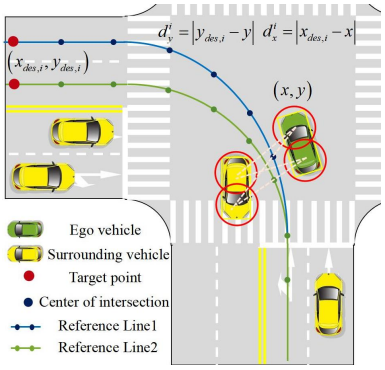


Fig. 5. Design of the scenario. d_x^i, d_y^i are the relative distances of the EV to the target point along the x-axis and y-axis. $i = 1, 2$ represents the indices of reference line1 and reference line2.

B. CMDPs Design

1) observation space and action space: The observation space is constituted by two components: EV's states $\mathcal{S}_{EV}$ and SVs's states $\mathcal{S}_{SV}$, namely: $\mathcal{S} = [\mathcal{S}_{EV}, \mathcal{S}_{SV}^1, ..., \mathcal{S}_{SV}^{N_{SV}}]$, where $S_{EV} = [\mathbb{I}_{veh}, x, y, v_x, v_y, \phi, \omega, d_{veh}, d_{des}]$, $S_{SV}^j = [\mathbb{I}_{veh}, \Delta x, \Delta y, \Delta v_x, \Delta v_y, \phi]$. The N_{SV} vehicles situated in the closest proximity to EV are selected, with the distance measured from 70 m in front of the EV to 30 m behind it. In the Highway-Env simulator, the fixed-dimensional constraint of the observation space inherently contradicts the dynamic nature of traffic scenarios. To address this limitation, we construct an observation tensor with a constant dimension by predefining a maximum number of SVs $M_{SV} > N_{SV} \in [1, N] \cap \mathbb{N}$, ensuring that observation redundancy is always present. Additionally, an indicator function $\mathbb{I}_{veh} \in \{0, 1\}$ is used to indicate whether a vehicle is actually observed ($\mathbb{I}_{veh} = 1$) or is a redundant vehicle ($\mathbb{I}_{veh} = 0$). To mitigate the impact of redundant features on the network, we apply masks to both the feature and attention layers: redundant parts of the feature vector are zero-padded, and when computing the attention weights, a large negative bias (-1e9) is applied to the $\mathbf{QK}^\top$ similarity of the redundant vehicles. This takes advantage of the exponential decay property of the softmax function, causing the normalized weight to approach zero. $\Delta x, \Delta y, \Delta v_x, \Delta v_y$ represent the position and velocity of the SVs relative to the EV. ϕ and ω is the heading angel and yaw rate of vehicle, respectively. As illustrated in Fig. 5, $d_{des} = \min_{i=1,2}(d_x^i + d_y^i)$ is the shortest Manhattan distance from EV to the target points. $d_{veh} = d_{min} - r_{EV} - r_{SV}$ is the shortest distance from the EV to SVs, where d_{min} represents the minimum distance among the four inter-center distances computed between the circular, $r_{veh} = \sqrt{(l_{veh}/4)^2 + (w_{veh}/2)^2}, veh \in [EV, SV]$, l_{veh} and w_{vhe} are the length and width of the vehicle. We directly control the vehicle's front wheel steering angle δ_f and longitudinal acceleration a_x, so the continuous action can be expressed as $a = [a_x, \delta_f]$.

2) reward function: Our reward function is comprised primarily of sparse $\mathbf{r}_{sparse}$ and dense rewards $\mathbf{r}_{dense}$. The sparse rewards, which are used to penalize collisions and encourage reaching the target points, are illustrated as follows:

$$\begin{aligned} \mathbf{r}_{sparse} &= \mathbf{r}_{collision} + \mathbf{r}_{arrive\_goal}, \\ \mathbf{r}_{collision} &= -50 \cdot \mathbb{I}_{collision}, \\ \mathbf{r}_{arrive\_goal} &= 100 \cdot \mathbb{I}_{arrive\_goal}. \end{aligned} \tag{18}$$

To determine the dense reward function, we consider factors such as reference line information, action smoothness, the distance to destination and safety distance:

$$\begin{aligned} \mathbf{r}_{dense} &= \frac{2}{1 + \mathbf{r}_{ref}} + \mathbf{r}_{smooth} + \mathbf{r}_{des} + \mathbb{I}_{RS}\mathbf{r}_{safe}, \\ \mathbf{r}_{ref} &= \max_{i=1,2}(\mathbf{x}_{t,i}^{\text{ref}} - \mathbf{x}_t)^\top Q(\mathbf{x}_{t,i}^{\text{ref}} - \mathbf{x}_t), \\ \mathbf{r}_{act} &= -(a_t^\top R_u a_t + R_\Delta \Delta a_t), \\ \mathbf{r}_{des} &= -d_{des}^2 = -[\min_{i=1,2}(d_x^i + d_y^i)]^2, \\ \mathbf{r}_{safe} &= \begin{cases} 0.0 & \text{if } d_{veh} > 0.5, \\ -(1.0 - d_{veh}) & \text{if } 0.2 < d_{veh} \le 0.5, \\ -3 \times (1.0 - d_{veh}) & \text{if } d_{veh} \le 0.2. \end{cases} \end{aligned} \tag{19}$$

where $\mathbf{x}_t^{\text{ref}} = [x^{\text{ref}}, y^{\text{ref}}, v_x^{\text{ref}}, 0, \phi^{\text{ref}}, 0]^\top$. In order to guarantee safety, vehicles should adhere to a speed limit of 30 or 40 km/h when approaching and traversing intersections. Consequently, v_x^{ref} is set at 9 m/s. For $\mathbf{r}_{ref}$, calculate both reference lines simultaneously and select the maximum value to encourage the EV to stay close to the reference line. $\mathbf{r}_{act}$ is employed to encourage the EV to save energy and smooth the trajectory. The weight coefficient matrix or vector $Q = diag(400.0, 400.0, 20.0, 20.0, 2.0, 0.5)$, $R_a = diag(0.05, 0.02)$, $R_\Delta = [0.10, 0.10]$. $\mathbf{r}_{safe}$ is used exclusively in the baseline algorithm based on reward shaping, i.e., when $\mathbb{I}_{RS} = 1$.

3) cost function: To evaluate the existing risk of collision and facilitate the autonomous vehicles' capacity to proactively anticipate potential collision threats, we proposes a cost function based on vehicle trajectory prediction. The predicted positions and headings $\{\mathbf{X}, \mathbf{Y}, \mathbf{\Phi}\}_{EV,SV}$ derived from vehicle dynamic model $\mathcal{F}_{EV}$ and kinematic model $\mathcal{F}_{SV}$ (Sec. V-C) employed in the calculation of the vehicle's corner points $\mathcal{V}_{EV,SV}$. The separating axis theorem (SAT) is then employed to detect collisions [35]. Considering that the uncertainty of trajectory predictions increases with distance, vehicle expansion coefficient $\beta \in [1.20, 1.50]$ is introduced to enlarge the vehicle's rectangular bounding box. The process of the cost function is illustrated in Algorithm 2.

C. Vehicle Model

We use a kinematic model to describe the motion of SVs, while employing a more precise dynamic model to simulate the motion of the EV. Due to the singularity of conventional dynamic models at low speed, we introduce a discrete dynamic bicycle model inspired by the backward Eulerian method that is feasible at any low speed (i.e., less than 15 m/s) [36]. This model has been demonstrated to be numerically stable and to have lower prediction errors compared to kinematic models. The transition model of EV and SVs are depicted as follows:

$$\mathbf{x}_{t+1} = \mathcal{F}_{EV}(\mathbf{x}_t, \mathbf{u}_t), \mathbf{x}_{t+1}^j = \mathcal{F}_{SV}(\mathbf{x}_t^j), \tag{20a}$$

$$\mathcal{F}_{EV} = \begin{bmatrix} x + T_s(v_x \cos\phi - v_y \sin\phi) \\ y + T_s(v_x \sin\phi + v_y \cos\phi) \\ v_x + T_s(a_x + v_y\omega) \\ \frac{mv_xv_y + T_s[(L_fC_f - L_rC_r)\omega - C_f\delta_f v_x - mv_x^2\omega]}{mv_x - T_s(C_f + C_r)} \\ \phi + T_s\omega \\ \frac{-I_z\omega v_x - T_s[(L_fC_f - L_rC_r)v_y - L_fC_f\delta v_x]}{T_s(L_f^2C_f + L_r^2C_r) - I_z v_x} \end{bmatrix}, \quad (20b)$$

$$\mathcal{F}_{SV} = \begin{bmatrix} x^j + T_s(v_x^j \cos\phi^j - v_y^j \sin\phi^j) \\ y^j + T_s(v_x^j \sin\phi^j + v_y^j \cos\phi^j) \\ v_x^j \\ 0 \\ \phi^j + T_s\omega_{\text{const}}^j \end{bmatrix}, \quad (20c)$$

where the EV's feature vector $\mathbf{x}_t = [x, y, v_x, v_y, \phi, \omega]^\top$, control vector $\mathbf{u}_t = [a_x, \delta_f]^\top$, SV's feature vector $\mathbf{x}_t^j$ is $[x, y, v_x, v_y, \phi]^\top$, x, y are the position coordinates of the vehicle's center of gravity, v_x, v_y are the longitudinal and lateral velocities, ϕ is the heading angle, ω is the yaw rate. L_f, L_r are the front and rear wheelbases, respectively. C_f, C_r are the front and rear axle equivalent sideslip stiffness, respectively. m is the mass of vehicle, I_z is the inertia of vehicle's center of gravity. $j \in [1, N_{SV}]$, where N_{SV} represents the total number of SVs. Sampling time $T_s = \frac{1}{f_s}$. The future states of SVs are predicted under constant velocity and yaw rate. Considering actuator saturation, we restrict continuous actions to $a_x \in [-5.0, 5.0]\text{m/s}^2$ and $\delta_f \in [0.6, 0.6]\text{rad}$, respectively.

Algorithm 2 Cost function with trajectory prediction

Initialize: EV's velocity $\mathbf{v}_t$, vehicle expansion coefficient β, prediction horizon T, initial cost value C_{init}, base velocity $\mathbf{v}_{base}$, weight coefficient $\mathbf{w}$.

Get the predicted positions and headings of EV and SVs:
$\{\mathbf{X}, \mathbf{Y}, \mathbf{\Phi}\}_{EV} \leftarrow \mathcal{F}_{EV}, \{\mathbf{X}, \mathbf{Y}, \mathbf{\Phi}\}_{SV} \leftarrow \mathcal{F}_{SV}$.
for each SV $j = 1 \rightarrow N_{SV}$ **do**
  for each time-step $i = 1 \rightarrow T$ **do**
    Get the corner points of EV and SV:
    $\mathcal{V}_{EV} \leftarrow polygon(\{x_i, y_i, \phi_i\}_{EV}, \beta)$,
    $\mathcal{V}_{SV} \leftarrow polygon(\{x_i, y_i, \phi_i\}_{SV}, \beta)$,
    Collision detection based on SAT($\mathcal{V}_{EV}, \mathcal{V}_{SV}$).
    if have collision **then**
      $C_i \leftarrow C_i + C_{init} \cdot \frac{\mathbf{v}_t}{\mathbf{v}_{base}} \cdot e^{-\mathbf{w}\beta}$
Return: C_i / N_{SV}

D. Comparison Baselines and Metrics

We compare Attention embedded and Risk-aware Soft Actor Critic (ARSAC) to the following baselines: SAC-RS [37], PPO-RS [38], which incorporate an auxiliary reward $\mathbf{r}_{safe}$ compared to standard SAC and PPO; SAC-Lag [23] and CPO [25]. The implementation of SAC-Lag and CPO are based on Omnisafe [39]. For algorithms that do not use MMAM, the hidden layers of their policy and value networks are unified to three. The detailed hyper-parameters of the above algorithm and the one we propose are listed in Table I, with recommended values used for hyper-parameters not specified in the tables below.

TABLE I
HYPER-PARAMETERS

Algorithm	Hyper-parameter	Value
Shared	Network hidden size	256
	Activation function	GELU
	Actor learning rate α_π	3e-4→1e-5
	(Safe) Critic learning rate $\alpha_{r,c}$	3e-3→1e-4
	Discount factor γ	0.99
	safety threshold d_{thres}	0.05
SAC-RS	Temperature factor τ	0.005
	Buffer size	1e5
	Batch size	256
	Alpha learning rate β_α	3e-4
	Target entropy $\bar{\mathcal{H}}$	-dim($\mathcal{A}$)
SAC-Lag	Initial Lagrangian multiplier	1.0
	Lagrangian multiplier learning rate α_λ	1e-4
PPO-RS	GAE parameter λ_{GAE}	0.95
	Clip parameter	0.20
	Batch size	4096
	Mini-batch size	256
	Activation function	TANH
CPO	Conjugate gradients iterations	15
ARSAC	maximum iteration N_{iter}	50
	Update step-size η	0.02
	Attention heads	4

For each algorithm, five training runs with different random seeds are conducted, each spanning 10,000 episodes in randomly generated intersection scenarios featuring three driving tasks. Subsequently, each algorithm is tested with 500 episodes for LT, RT, and GS driving tasks. To evaluate the performance of EV in these intersection scenarios, this study designed the following metrics:

(a) **Collision rate** (CR): As the cost function devised with safety as a primary consideration, this study employs the mean collision rate as a statistical metric. Within an episode, if EV collides with other vehicles or exceeds the traversable area of the road, it is considered a collision.

(b) **Success rate** (SR): If EV safely reaches the target point without any collisions, it is considered a success.

(c) **Frozen rate** (FR): If EV neither collides nor reaches the target point within a limited time (25s), it is considered to be 'frozen'. This phenomenon is typically observed in instances where EV is operating under overly conservative strategies, which can have a detrimental impact on the overall efficiency of traffic flow. The frozen rate can be calculated by:

$$FR = 1 - CR - SR. \quad (21)$$

(d) **Average episode cumulative reward** (AER): reflects the performance of each algorithm.

(e) **Average episode velocity** (AEV): reflects the average speed of the EV when navigating through intersections and its impact on traffic efficiency.

E. Results Analysis

1) Comparison experiment: The learning curves compared with baseline algorithms such as SAC-RS, PPO-RS, SAC-Lag and CPO are shown in Fig. 6 and test results are in Table II. Results indicate that the proposed ARSAC algorithm outperforms or matches all other baseline algorithms across three driving tasks in terms of the final performance. For

TABLE II
COMPARE PERFORMANCE ON THREE DRIVING TASKS.

Tasks	Algorithms	CR(%)	SR(%)	FR(%)	AER	AEV(m/s)
LT	CPO	38.4±10.8	61.6±10.8	**0.0±0.0**	-91.2±48.3	14.12±0.82
	PPO-RS	31.4±6.6	68.6±6.6	**0.0±0.0**	-78.3±38.3	13.56±0.48
	SAC-RS	17.6±5.1	80.8±5.1	1.6±0.3	-49.4±30.1	12.69±0.55
	SAC-Lag	17.4±4.5	80.4±4.4	2.2±1.1	-37.1±32.4	12.16±0.49
	ARSAC	**2.8±1.7**	**95.8±1.3**	1.4±0.5	46.32±19.9	**8.51±0.76**
GS	CPO	20.8±8.1	79.2±8.1	**0.0±0.0**	41.32±31.6	**8.82±1.62**
	PPO-RS	14.4±6.0	85.6±6.0	**0.0±0.0**	54.80±35.9	7.82±0.32
	SAC-RS	12.8±3.0	86.2±3.1	1.0±0.6	46.53±23.9	7.54±0.88
	SAC-Lag	11.6±4.4	87.2±4.0	1.2±0.8	48.42±30.1	8.01±0.46
	ARSAC	**1.6±1.0**	**98.4±1.0**	**0.0±0.0**	**76.62±17.4**	8.23±0.28
RT	CPO	18.6±7.1	81.0±7.1	0.4±0.1	41.5±37.2	**9.10±0.86**
	PPO-RS	16.8±5.7	82.4±5.6	0.8±0.1	58.5±35.8	8.32±0.47
	SAC-RS	10.6±3.7	87.8±3.7	1.6±0.6	52.1±26.5	8.46±0.55
	SAC-Lag	8.8±2.8	89.6±2.8	1.6±0.3	49.9±22.6	7.68±1.13
	ARSAC	**0.4±0.3**	**99.6±0.3**	**0.0±0.0**	**90.4±12.4**	8.27±0.35
MEAN	CPO	25.9±8.8	73.9±8.8	**0.1±0.0**	-2.8±41.2	**10.68±1.02**
	PPO-RS	20.9±6.2	78.9±6.1	0.3±0.0	11.67±36.9	9.90±0.39
	SAC-RS	13.7±4.2	84.9±4.0	1.4±0.4	16.43±27.1	9.56±0.68
	SAC-Lag	12.6±4.1	85.7±4.2	1.7±0.7	20.41±24.8	9.28±0.89
	ARSAC	**1.6±0.9**	**97.9±0.9**	0.5±0.1	**71.13±16.8**	8.37±0.45

[1] Bold: best performance; Underline: undesirable high values.
[2] Policy update frequency $f_\pi = 10$ Hz.

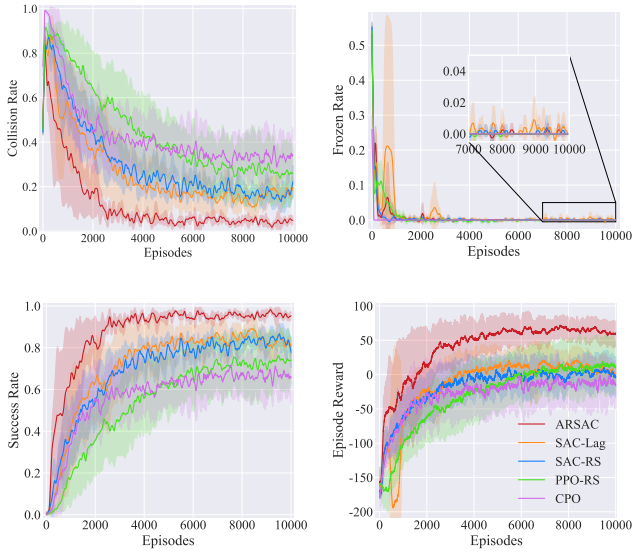


Fig. 6. Training curves on comparison experiment. Solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over 3 runs.

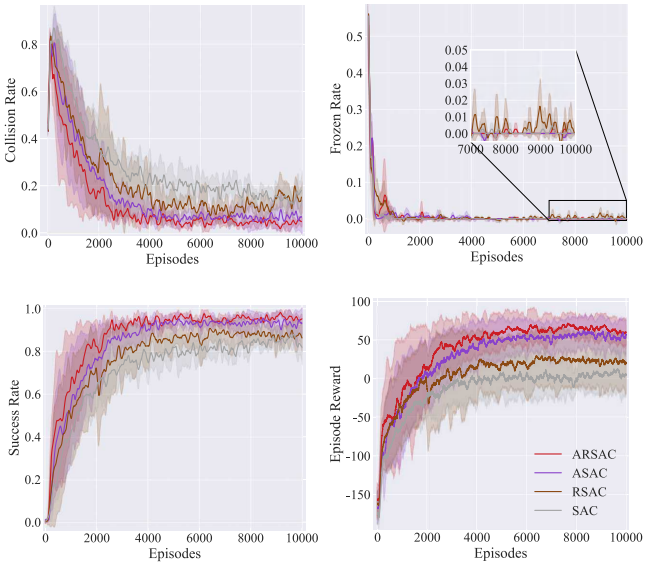


Fig. 7. Training curves on ablation studies. Solid lines correspond to the mean and the shaded regions correspond to 95% confidence interval over 3 runs.

instance, in the RT task, ARSAC achieves 18.2%, 16.4%, 10.2%, and 8.4% lower CR compared to CPO, PPO-RS, SAC-RS, and SAC-Lag respectively, while demonstrating superior performance in AER with significantly lower variance. In the LT task, SAC-RS, PPO-RS, SAC-Lag and CPO demonstrate higher velocity, yet their AER are negative and significantly lower than that of ARSAC. In this scenario, EV needs to adopt a competitive strategy to efficiently identify suitable gaps in fast-moving traffic. Due to insufficient understanding of the scenario, SAC-RS, PPO-RS, SAC-Lag and CPO struggle to achieve higher rewards, resulting in mostly negative outcomes. To maximize cumulative rewards, these algorithms tend to adopt higher driving speeds to avoid accumulating negative rewards in future timesteps, leading to higher AEV and undesirable behaviors such as failure to reach the destination. Although CPO and PPO-RS achieve or surpass ARSAC's performance in FR in both the LT and GS tasks, they lack safety and perform worse than ARSAC in terms of AER. Additionally, in Table II, we present the mean statistics that evaluate the average performance of each method across three testing conditions. We find that ARSAC outperforms or matches the baselines across the three tasks and demonstrates exceptional performance in the more challenging unprotected left-turn scenario.

2) Ablation studies: We additionally perform an ablation study to compare the effects of the safety module and the structure of MMAM on algorithm performance. As shown in Table III, RSAC is a variant of ARSAC that excludes the MMAM, while ASAC is a version of ARSAC that omits the risk-aware component. Compared to SAC, RSAC demonstrates a lower collision rate across three driving tasks. Although RSAC encounters performance degradation in LT task due to limited scene comprehension. Safety iterative correction mitigates collisions by projecting risky actions back towards the feasible region $\mathcal{S}_{FR}$ through cyclic gradient descent, guided by safe critics' evaluations. The collision rates in LT/GS/RT task are reduced by 2.0%, 3.0%, and 3.2% for ARSAC compared to ASAC, while the collision rates are reduced by 10.4%, 3.8%, and 3.8% in comparison to RSAC, respectively. These results indicate that better scene understanding allows the safe critic to more accurately assess risky actions. Consequently, the safe actions corrected by gradient projection are more likely to fall within $\mathcal{S}_{FR}$, thereby enhancing overall safety. As illustrated in Fig. 7, after training convergence,the frozen rate of RSAC exhibits fluctuations around 1%, while ARSAC exhibits a notable decline in its frozen rate. Furthermore, as indicated in Table III, the AEV of RSAC is 1.7% and 6.5% lower than

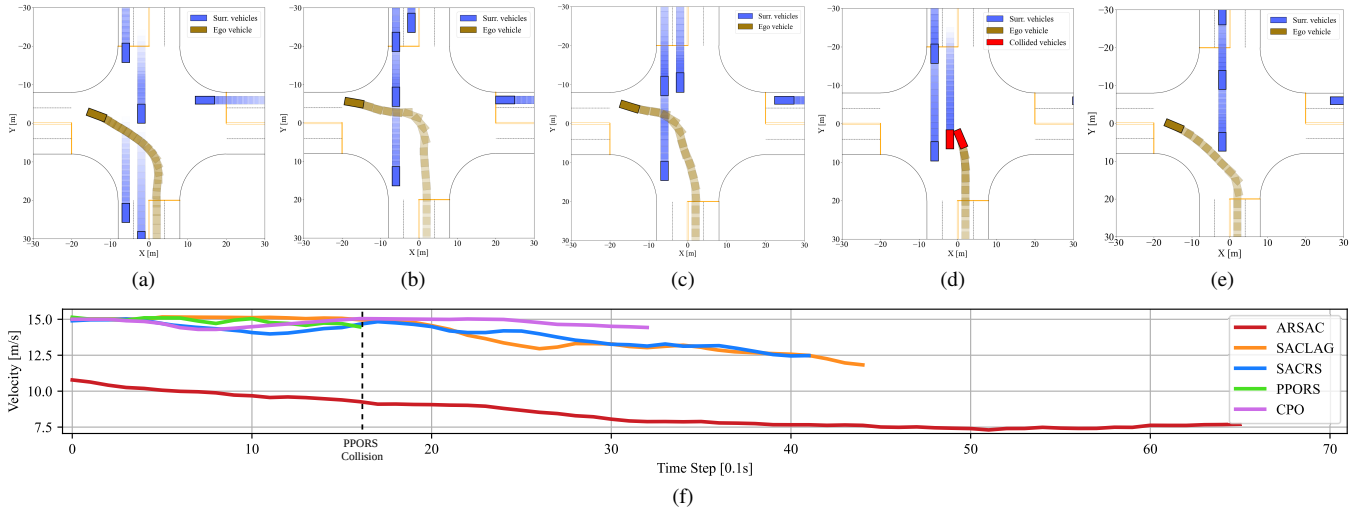


Fig. 8. Vehicle trajectories visualization. Blue rectangles are SVs, brown rectangles are EVs. (a) ARSAC. (b) SAC-Lag. (c) SAC-RS. (d) PPO-RS. (e) CPO. (f) The velocity of the EV driving inside intersection.

TABLE III
ABLATION STUDY ON THREE DRIVING TASKS.

Tasks	Algorithms	CR(%)	SR(%)	FR(%)	AER	AEV(m/s)
LT	SAC	20.4±5.4	79.2±5.3	0.4±0.0	-42.24±31.2	12.28±0.69
	RSAC	13.2±4.8	84.2±4.6	2.6±0.3	-34.28±27.8	10.96±0.56
	ASAC	4.8±1.2	95.2±1.2	**0.0±0.0**	41.43±24.9	**8.63±0.68**
	ARSAC	**2.8±1.7**	**95.8±1.3**	1.4±0.5	46.32±19.9	8.51±0.76
GS	SAC	14.6±5.9	83.2±5.5	2.2±0.4	51.02±34.3	7.95±0.46
	RSAC	5.4±1.6	93.2±1.2	1.4±0.3	66.74±29.1	7.69±0.37
	ASAC	4.6±0.9	95.4±0.9	**0.0±0.0**	69.42±23.6	**8.42±0.33**
	ARSAC	**1.6±1.0**	**98.4±1.0**	**0.0±0.0**	**76.62±17.4**	8.23±0.28
RT	SAC	12.6±4.8	85.6±4.8	1.8±0.3	58.59±31.7	7.84±0.46
	RSAC	4.2±1.4	94.6±1.3	1.2±0.1	76.84±24.2	8.13±0.33
	ASAC	3.6±0.7	96.4±0.7	**0.0±0.0**	86.85±13.2	**8.50±0.61**
	ARSAC	**0.4±0.3**	**99.6±0.3**	**0.0±0.0**	**90.4±12.4**	8.27±0.35
MEAN	SAC	15.2±5.2	83.3±5.1	1.5±0.3	22.46±32.2	9.36±0.52
	RSAC	7.6±3.3	90.7±3.1	1.7±0.5	36.43±25.2	8.93±0.41
	ASAC	4.3±0.8	95.7±0.8	**0.0±0.0**	65.90±19.2	**8.52**±0.49
	ARSAC	**1.6±0.9**	**97.9±0.9**	0.5±0.1	**71.13±16.8**	8.37±0.45

[1] Bold: best performance; Underline: undesirable high values.
[2] Policy update frequency $f_\pi = 10$ Hz.

that of ARSAC in RT and GS tasks, respectively. In contrast, ASAC achieves AEV that are 2.8% and 2.3% higher than those of ARSAC. These findings illustrate that MMAM effectively captures the relationship between EV and SVs, filtering out non-conflicting vehicles and other disruptive factors, thereby enhancing overall traffic efficiency.

F. Driving Behavior Analysis

We apply the trained policies of the compared baselines and our proposed algorithm, ARSAC, to three driving tasks and visualize the trajectories of both EVs and SVs. The basic environment settings are consistent with those described in Sec. V-A.

1) Left-turn Case: As illustrated in Fig. 8, ARSAC, SAC-Lag, SAC-RS and CPO are able to pass through the scene without collisions, while PPO-RS encounters a collision. ARSAC exhibits similar driving behaviour to a human driver when faced with oncoming traffic in parallel lanes. When approaching an intersection, ARSAC first slows down and then performs a pre-steer maneuver to the right, allowing the vehicle to turn left more fluidly and safely. During the turn, it decelerates appropriately to find an optimal moment to pass through, and once the oncoming traffic is cleared, it accelerates again to improve traffic flow. Throughout the turn, the ego's trajectory follows a smooth arc. In contrast, the trajectory of SAC-RS is not smooth, and although SAC-Lag also exhibits a tendency to pre-steer rightward, its trajectory is similarly not smooth. Due to its high-speed performance, CPO maneuvers left to avoid oncoming vehicles. However, in comparison to ARSAC, its trajectory deviates from the reference line. It can be seen that ARSAC enables the vehicle to effectively perceive its surroundings, thereby enhancing safety and providing improved opportunities for better passage.

2) Go-straight Case: In the GS task, the EV encounters the challenge of traffic coming from all directions. If the EV fails to respond expeditiously to potential risks in its surroundings, the likelihood of a collision occurring is increased. As shown in Fig. 9(c), Although SAC-RS attempts to avoid collisions with oncoming lateral traffic by reducing speed, the utilization of $\mathbf{r}_{safe}$ as an auxiliary reward alone is inadequate to ensure collision avoidance. In this case, CPO made significant accelerations and decelerations to avoid a collision. Although it successfully navigated through the intersection, its trajectory was less smooth compared to ARSAC, and the larger speed fluctuations could result in a decrease in comfort. Fig. 9(f) demonstrates that the velocity changes of ARSAC, SAC-Lag, and PPO-RS are characterized by a relatively gentle slope. Nevertheless, due to its limited scene understanding, SAC-Lag is compelled to execute preemptive steering maneuvers to avoid collisions with oncoming lateral traffic, causing the EV's trajectory to shift left. In contrast, ARSAC effectively seizes the opportunity to pass through, making minimal adjustments to avoid collisions while maintaining a smooth trajectory.

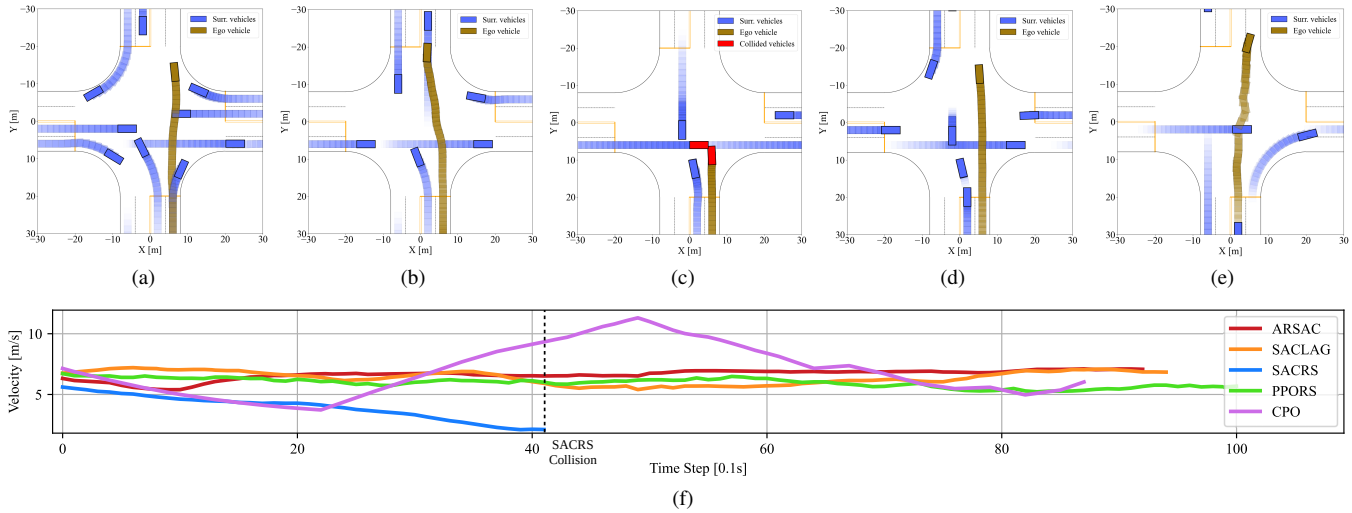


Fig. 9. Vehicle trajectories visualization. Blue rectangles are SVs, brown rectangles are EVs. (a) ARSAC. (b) SAC-Lag. (c) SAC-RS. (d) PPO-RS. (e) CPO. (f) The velocity of the EV driving inside intersection.

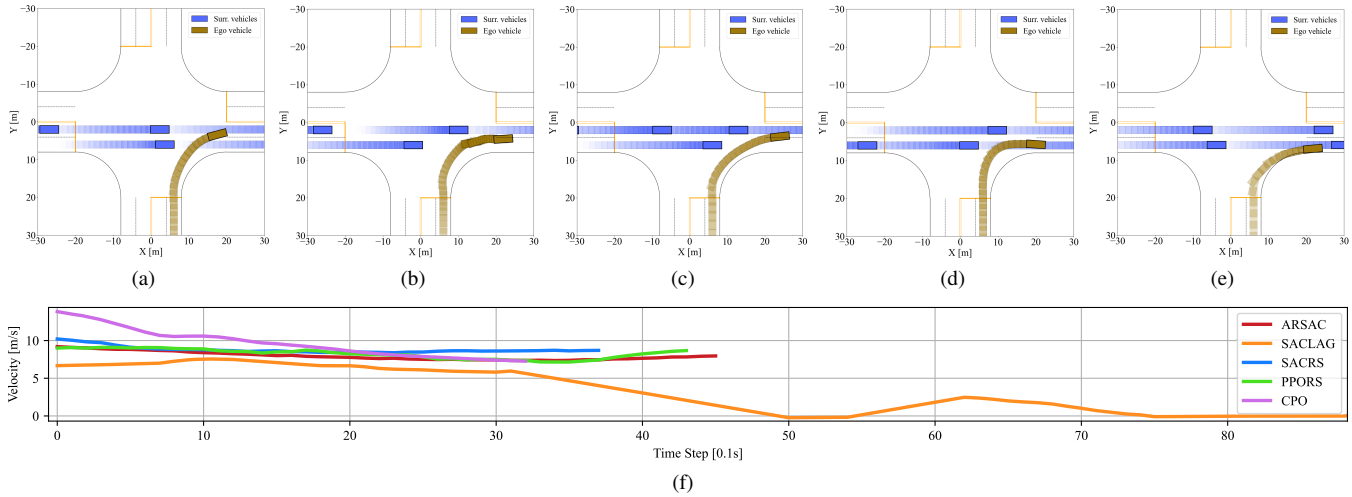


Fig. 10. Vehicle trajectories visualization. Blue rectangles are SVs, brown rectangles are EVs. (a) ARSAC. (b) SAC-Lag. (c) SAC-RS. (d) PPO-RS. (e) CPO. (f) The velocity of the EV driving inside intersection.

3) Right-turn Case: In the RT case, all algorithms except SAC-Lag are able to produce safe, collision-free, and smooth trajectories, as depicted in Fig. 10. Fig. 10(f) illustrates that SAC-Lag attempts to merge into the traffic by reducing its speed. However, due to its inability to accurately gauge the optimal timing for merging, it adopts an overly cautious approach to avoid collisions, which ultimately reduces traffic efficiency. CPO enters the intersection at a speed of approximately 15 m/s. To avoid a collision, it rapidly decelerates to 10 m/s within about 0.7 seconds, then gradually reduces speed after 1.3 seconds to complete the merging maneuver. In contrast, ARSAC maintains a consistently stable speed throughout, highlighting its ability to accurately assess the optimal timing for merging while ensuring safety.

VI. Conclusion

In this paper, we propose a risk-aware reinforcement learning algorithm to ensure that autonomous vehicles can safely and efficiently traverse intersection scenarios. Safe critics are designed to assess driving risks and work in conjunction with the reward critic to update the actor. Building on this, a Lagrangian relaxation method and cyclic gradient iteration are employed to project actions into a feasible safe region. Furthermore, a multi-hop, MLP-mixed attention mechanism is integrated into the actor-critic network, enabling the policy to adapt to dynamic traffic and overcome permutation sensitivity challenges, thereby allowing it to more effectively focus on surrounding potential risks while enhancing the identification of passing opportunities. Experimental results for left-turn, right-turn, and go straight driving tasks demonstrate that our algorithm effectively reduces collision rates and improves the efficiency of EV compared to baseline algorithms. Nonetheless, it is well known that autonomous vehicles share the road with various traffic participants, such as cyclists and pedestrians in the real-world environment. Therefore, our future work will extend to mixed traffic flows. In addition,

we will use more accurate time series forecasting models and integrate predictive features to better assess risk. We also plan to use offline datasets for training and testing, while extending the approach to a wider range of scenarios.

REFERENCES

[1] Z. Li, J. Hu, B. Leng, L. Xiong, and Z. Fu, "An integrated of decision making and motion planning framework for enhanced oscillation-free capability," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 5718–5732, 2024.

[2] H. Deng, Y. Zhao, Q. Wang, and A.-T. Nguyen, "Deep reinforcement learning based decision-making strategy of autonomous vehicle in highway uncertain driving environments," *Automot. Innov.*, vol. 6, no. 3, pp. 438–452, 2023.

[3] G. Li, W. Zhou, S. Lin, S. Li, and X. Qu, "On-ramp merging for highway autonomous driving: An application of a new safety indicator in deep reinforcement learning," *Automot. Innov.*, vol. 6, no. 3, pp. 453–465, 2023.

[4] Y. Ren, G. Zhan, L. Tang, S. E. Li, J. Jiang, K. Li, and J. Duan, "Improve generalization of driving policy at signalized intersections with adversarial learning," *Transp. Res., C, Emerg. Technol.*, vol. 152, p. 104161, 2023.

[5] Z. Li, L. Xiong, B. Leng, P. Xu, and Z. Fu, "Safe reinforcement learning of lane change decision making with risk-fused constraint," in *IEEE Int. Conf. Intell. Transp. Syst.*, 2023, pp. 1313–1319.

[6] H. Yuan, P. Li, B. Van Arem, L. Kang, H. Farah, and Y. Dong, "Safe, efficient, comfort, and energy-saving automated driving through roundabout based on deep reinforcement learning," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*. IEEE, 2023, pp. 6074–6079.

[7] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Int. Conf. Machin. Learn., ICML*, vol. 99, 1999, pp. 278–287.

[8] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *J. Mach. Learn. Res.*, vol. 18, no. 167, pp. 1–51, 2018.

[9] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, and Y. Zhao, "Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5435–5444, 2021.

[10] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 3387–3395.

[11] H. Krasowski, Y. Zhang, and M. Althoff, "Safe reinforcement learning for urban driving using invariably safe braking sets," in *IEEE Int. Conf. Intell. Transp. Syst.* IEEE, 2022, pp. 2407–2414.

[12] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," *arXiv preprint arXiv:1801.08757*, 2018.

[13] H. Odriozola-Olalde, M. Zamalloa, N. Arana-Arexolaleiba, and J. Perez-Cerrolaza, "Towards robust shielded reinforcement learning through adaptive constraints and exploration: The fear field framework," *Eng. Appl. Artif. Intell.*, vol. 144, p. 110055, 2025.

[14] K. Yang, X. Tang, S. Qiu, S. Jin, Z. Wei, and H. Wang, "Towards robust decision-making for autonomous driving on highway," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11 251–11 263, 2023.

[15] Z. Hu, F. Yang, Z. Lu, and J. Chen, "Enhancing autonomous lane-changing safety: Deep reinforcement learning via pre-exploration in parallel imaginary environments," *IEEE Trans. Ind. Inform.*, vol. 20, no. 10, pp. 12 385–12 395, 2024.

[16] J. Duan, D. Yu, S. E. Li, W. Wang, Y. Ren, Z. Lin, and B. Cheng, "Fixed-dimensional and permutation invariant state representation of autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9518–9528, 2022.

[17] Z. Bai, W. Shangguan, B. Cai, and L. Chai, "Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 8600–8605.

[18] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5068–5078, 2021.

[19] G. Chen, Y. Zhang, and X. Li, "Attention-based highway safety planner for autonomous driving via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, 2023.

[20] H. Seong, C. Jung, S. Lee, and D. H. Shim, "Learning to drive at unsignalized intersections using attention-based deep reinforcement learning," in *IEEE Intell. Transp. Syst. Conf. (ITSC)*, 2021, pp. 559–566.

[21] W. Xiao, Y. Yang, X. Mu, Y. Xie, X. Tang, D. Cao, and T. Liu, "Decision-making for autonomous vehicles in random task scenarios at unsignalized intersection using deep reinforcement learning," *IEEE Trans. Veh. Technol.*, 2024.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. neural inf. proces. syst.*, vol. 30, 2017.

[23] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan, "Learning to walk in the real world with minimal human effort," in *Proc. Conf. Robot Learn.*, vol. 155. PMLR, 16–18 Nov 2021, pp. 1110–1120.

[24] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by pid lagrangian methods," in *Int. Conf. Machin. Learn., ICML*, 2020, pp. 9133–9143.

[25] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Int. Conf. Machin. Learn., ICML*, 2017, pp. 22–31.

[26] Y. Zhang, Q. Vuong, and K. Ross, "First order constrained optimization in policy space," *Adv. neural inf. proces. syst.*, vol. 33, pp. 15 338–15 349, 2020.

[27] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. E. Li, "Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness," *Transp. Res., C, Emerg. Technol.*, vol. 134, p. 103452, 2022.

[28] G. Li, Y. Qiu, Y. Yang, Z. Li, S. Li, W. Chu, P. Green, and S. E. Li, "Lane change strategies for autonomous vehicles: A deep reinforcement learning approach based on transformer," *IEEE Trans. Intell. Veh.*, vol. 8, no. 3, pp. 2197–2211, 2022.

[29] L. M. Schmidt, S. Rietsch, A. Plinge, B. M. Eskofier, and C. Mutschler, "How to learn from risk: Explicit risk-utility reinforcement learning for efficient and safe driving strategies," in *IEEE Int. Conf. Intell. Transp. Syst.* IEEE, 2022, p. 1913–1920.

[30] X. Wang, J. Zhang, D. Hou, and Y. Cheng, "Autonomous driving based on approximate safe action," *IEEE Trans. Intell. Transp. Syst.*, 2023.

[31] X. Wang and M. Althoff, "Safe reinforcement learning for automated vehicles via online reachability analysis," *IEEE Trans. Intell. Veh.*, 2023.

[32] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Trans. Knowl. Data. Eng*, vol. 35, no. 4, pp. 3279–3298, 2023.

[33] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[34] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, p. 1805–1824, Jul 2002.

[35] N. Lyu, J. Wen, Z. Duan, and C. Wu, "Vehicle trajectory prediction and cut-in collision warning model in a connected vehicle environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 966–981, 2022.

[36] Q. Ge, Q. Sun, S. E. Li, S. Zheng, W. Wu, and X. Chen, "Numerically stable dynamic bicycle model for discrete-time control," in *Proc. IEEE Intell. Veh. Symposium (IV)*, 2021, pp. 128–134.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Int. Conf. Machin. Learn., ICML*, 2018, pp. 1861–1870.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[39] J. Ji, J. Zhou, B. Zhang, J. Dai, X. Pan, R. Sun, W. Huang, Y. Geng, M. Liu, and Y. Yang, "Omnisafe: An infrastructure for accelerating safe reinforcement learning research," *J. Mach. Learn. Res.*, vol. 25, no. 285, pp. 1–6, 2024.