This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Safe Quadratic-Enhanced MADRL for Robust and Efficient CAVs Ramp Merging in Mixed Traffic

Haigen Min, *Senior Member, IEEE*, Yong Liu, Xia Wu, Sen Xia, Xiangmo Zhao, *Member, IEEE*, and Guoqiang Mao, *Fellow, IEEE*

***Abstract*—Ramp merging is a critical challenge for the safety of connected autonomous vehicles (CAVs), particularly due to the high incidence of traffic accidents in mixed-traffic environments. To address this, this paper introduces the Safe Quadratic-Enhanced Double Buffer Independent Twin Delayed Deep Deterministic Policy Gradient (SQD-ITD3) method, which aims to achieve robust and safe merging decisions for CAVs in such environments. First, we propose an efficient and scalable multi-agent deep reinforcement learning (MADRL) framework that incorporates parameter sharing, distributed learning, and invalid action masking strategies to address ramp merging problems in mixed traffic conditions. Second, quadratic neurons and an adaptive double buffers mechanism are integrated into the MADRL model to enhance feature representation and improve experience utilization efficiency. Furthermore, a safety strategy based on an extended two-dimensional car-following model is introduced to improve safety during the merging process. Finally, the effectiveness of the proposed method is validated through numerical simulations across traffic scenarios of varying difficulty levels. It is shown that the proposed method consistently outperforms several state-of-the-art benchmarks.**

***Index Terms*—Multi-agent, deep reinforcement learning, ramp merging, connected autonomous vehicles, mixed traffic.**

I. Introduction

IN RECENT years, research on Intelligent Transport Systems (ITS) has become a prominent topic in the field of transportation, significantly enhancing traffic safety and comfort [1]. Within ITS, connected autonomous vehicles (CAVs) play a critical role. By integrating artificial intelligence and automation technologies, CAVs are expected to substantially improve road safety, reduce traffic congestion, and fundamentally change traditional travel modes and transportation management paradigms [2]. However, developing reliable control strategies for CAVs to navigate the complexities of real-world driving remains a significant challenge. This is especially true in mixed traffic environments, where CAVs and human-driven vehicles (HDVs) coexist and are expected to persist for the foreseeable future [3]. CAVs must not only respond to road objects but also consider the behavior of HDVs.

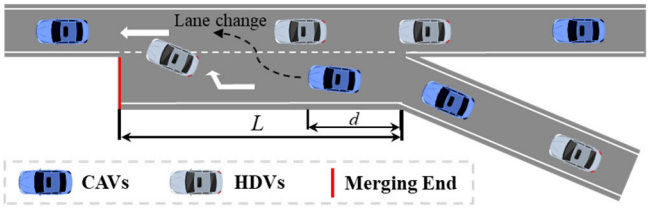


Fig. 1. Example of the considered ramp merging traffic scenario. Merging vehicles need to complete lane changes in a limited acceleration lane in a mixed traffic environment.

Among the many challenging driving scenarios, ramp merging is widely recognized as one of the most safety-critical and technically demanding driving scenarios. Merging vehicles must complete mandatory lane changes within limited acceleration lanes [4], [5], often disturbing the mainline flow and causing congestion or collisions when the gap acceptance is inadequate [6], [7]. As a result, decision-making control for vehicles during the merging process has become a critical area of research to improve both driving efficiency and safety.

The scenario considered in this paper is a parallel ramp merging area, as shown in Fig. 1. It has a merging area in which the CAV can merge at any point. Compared to the tapered ramp merging scenario, where only longitudinal control needs to be considered and the merge point is unique, this type is more complex and prevalent in real-world situations [8]. Ideally, vehicles on the ramp should make autonomous decisions in both lateral and longitudinal directions to merge into the main lane efficiently and safely. Vehicles already in the main lane must coordinate to create sufficient space for merging vehicles from the ramp, thereby avoiding collisions and enhancing traffic efficiency.

Traditionally, rule-based and optimization-based methods have been employed to address ramp merging problems [9], [10], [11]. While these methods yield satisfactory results in some scenarios, they frequently demonstrate insufficient

Received 21 February 2025; revised 24 October 2025, 9 February 2026, and 21 September 2026; accepted 24 September 2026. This work was supported in part by National Natural Science Foundation of China under Grant 52372426, Grant 52302399, and Grant U25B20136; and in part by the Fundamental Research Funds for the Central Universities, Chang'an University under Grant 300102243202 and Grant 300119246501. The Associate Editor for this article was Z. Yao. *(Corresponding authors: Haigen Min; Xia Wu.)*

Haigen Min, Xia Wu, and Xiangmo Zhao are with the School of Information Engineering, Chang'an University, Xi'an 710064, China, and also with the Collaborative Innovation Center for Western China Traffic Safety and Intelligent Cooperative Control, Xi'an 710064, China (e-mail: hgmin@chd.edu.cn; wuxia@chd.edu.cn; xmzhao@chd.edu.cn).

Yong Liu and Sen Xia are with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: yongliu@chd.edu.cn; xiasen@chd.edu.cn).

Guoqiang Mao is with the School of Transportation, Southeast University, Nanjing 211189, China (e-mail: 101013900@seu.edu.cn).

Digital Object Identifier 10.1109/TITS.2026.3740303

1558-0016 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

flexibility to handle intricate and changing traffic conditions [12]. In recent years, data-driven approaches, such as deep reinforcement learning (DRL), have garnered increasing attention due to their ability to effectively manage stochastic environments. DRL continuously learns and adapts to various environmental inputs, uncertainties, and disturbances during the training process, making it highly suitable for autonomous driving tasks [13], [14], [15], particularly in ramp merging scenarios [16], [17], [18]. Despite its promise, DRL algorithms frequently face challenges in achieving steady-state performance during training, with slow convergence and difficulties in model stability.

Additionally, as a closed-box method, DRL is often criticized for its lack of interpretability, which can lead to safety concerns, especially in extreme cases [19]. A common approach to address this issue is to incorporate safety considerations directly into the algorithm design, such as by adding safety functions to the reward structure. However, this typically comes at the cost of compromising other important properties. Recent studies have attempted to address this challenge by using optimization-based safety modules to constrain policy learning [20], [21]. While these approaches aim to solve constrained optimization problems, their improvements in safety performance remain limited, largely because of the significant computational burden associated with managing complex system dynamics. In contrast, although pure reinforcement learning methods demand substantial computational resources during training, they enable low-complexity forward inference once pre-trained, thereby supporting real-time decision-making in complex traffic environments.

Furthermore, while several DRL-based ramp merging studies have considered both longitudinal and lateral maneuvers, most of them either discretize the action space into symbolic control decisions [16] or model lane-changing as a finite state process [22]. Such simplifications reduce control flexibility and hinder the ability to fully capture dynamic vehicle interactions. In contrast, few methods have employed a two-dimensional continuous action space (incorporating both longitudinal and lateral accelerations) to model the coupled maneuvering of CAVs. Bridging this research gap is essential to enhancing the performance of CAVs in mixed-traffic environments and contributing to more efficient transportation systems. To overcome the limitations of existing methods, this paper proposes a novel approach named Safe Quadratic-Enhanced Double Buffer Independent Twin Delayed Deep Deterministic Policy Gradient (SQD-ITD3). The proposed approach introduces a quadratic actor network to enhance feature representation, an adaptive double buffers mechanism to improve sample utilization, and a two-dimensional safety policy to reduce collision risks. Its core goal is to enable CAVs to make robust decisions in ramp merging scenarios, thereby achieving both efficient and safe driving. The key contributions of this study are as follows:

1) A novel and scalable multi-agent deep reinforcement learning (MADRL) framework is proposed, which integrates parameter sharing, distributed learning, and invalid action masking. This framework can be flexibly combined with other classical reinforcement learning algorithms to achieve efficient ramp merging in mixed traffic environments.
2) In the context of ramp merging scenarios, a quadratic actor network is employed to bolster feature representation. Additionally, an adaptive double buffers mechanism is proposed, which dynamically adjusts the composition of samples required for policy updates based on the real-time success rate, thereby improving the efficiency of effective experience sampling.
3) A two-dimensional safety policy, developed as an extension of the Krauss car-following model, is introduced to dynamically constrain unsafe actions, thereby ensuring robust control and effective collision avoidance.

The remainder of this paper is organized as follows: Section II reviews related work. In Section III, we formulate the problem. SQD-ITD3 method is detailed in Section IV. Section V presents the experiments, results, and discussion. Finally, we conclude the paper and outline future work in Section VI.

II. Related Work

A. (MA)DRL for Ramp Merging Vehicle Control

In recent years, DRL has become an influential tool for autonomous vehicle control due to its ability to handle intricate state–action spaces and dynamic environments [23]. As a result, an increasing number of researchers have proposed DRL-based approaches to tackle the challenges of vehicle ramp merging. For example, Bagwe et al. [24] proposed a framework for robust ramp merging of CAVs using enhanced multimodal reinforcement learning. This method improves data utilization efficiency and model generalization by simultaneously incorporating noisy basic safety information and surveillance images for multimodal observations. Ma et al. [25] introduced a novel merging policy training framework for CAVs that reconstructs merging environments and employs a two-stage training approach. The framework enhances realism by augmenting empirical scenarios from the NGSIM dataset, allowing CAVs first to imitate human behavior and then optimize its performance. However, most existing single-agent DRL algorithms focus primarily on training the merging behaviors of individual CAVs in specific environments, while neglecting the control of CAVs movements on the main road.

Given the limitations mentioned above, a growing number of studies are now focusing on MADRL for control applications. For example, Zhang et al. [26] proposed an Independent Proximal Policy Optimization method for the ramp merging task of autonomous vehicles on freeways. The experimental findings indicate that this algorithm substantially enhances the success rate of decisions made during vehicle merging when compared to baseline methods. Chen et al. [16] developed a scalable and efficient control framework to train all CAVs in the ramp merging area, achieving convergence in complex scenarios through curriculum learning. Experimental results show that this framework outperforms state-of-the-art techniques in terms of training efficiency and collision rates.

Despite these advances, existing MADRL methods often suffer from sample inefficiency and safety instability, which

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

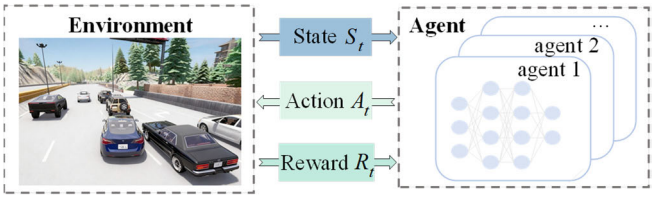


Fig. 2. MDP model for behavioral decision making in CAVs.

hinder their application in safety-critical domains such as ramp merging. Finding ways to overcome these shortcomings while retaining the benefits of DRL remains an important and relevant research challenge.

B. Safety-Integrated (MA)DRL for Autonomous Driving

Safety remains a major concern in DRL-based vehicle control because learned policies may exhibit unpredictable behavior. Integrating explicit safety mechanisms into DRL frameworks has therefore become an active research direction.

For example, Sun et al. [27] present a novel hierarchical framework that integrates model predictive control (MPC) and DRL for ramp merging control. It utilizes a low-frequency MPC for baseline inputs and a high-frequency DRL for real-time adjustments, efficiently managing uncertainties with limited computational resources. Li et al. [17] introduced a new Time Difference to Merging (TDTM) metric, based on the Deep Deterministic Policy Gradient (DDPG), to assess driving safety and help ramp vehicles find suitable merging gaps. Liu et al. [28] proposed the integration of a motion-predictive safety controller into the Discrete Soft Actor-Critic (D-SAC) method to enhance driving safety during vehicle ramp merging. Potential collisions are predicted by estimating the trajectories of the ego vehicle and surrounding vehicles using a kinematic model. While these methods enhance safety awareness, they typically simplify control to one-dimensional longitudinal motion or discrete decision-making, neglecting the continuous coupling of lateral and longitudinal maneuvers inherent in real-world driving. Moreover, the computational cost required for solving such models often limits their practical applicability.

In summary, although existing DRL or MADRL studies have advanced research on ramp merging, they still fall short in robustness, adaptability, and explicit safety modeling under mixed-traffic conditions. Therefore, a method is needed that can more accurately simulate interactions among real vehicles, enhance maneuvering precision, and achieve a better balance between safety and efficiency.

III. Formulation of Ramp Merging Problem

In this section, we formulate the ramp merging problem under mixed traffic as a Markov Decision Process (MDP), as illustrated in Fig. 2. The model mainly includes the state space S_t, action space A_t, and reward function R_t. Throughout the driving process, the CAV selects the action A_t through the state S_t obtained in real time from the environment. The environment feeds back the corresponding reward R_t, so that the CAV continuously improves the strategy and the environment moves to the next state.

TABLE I
Input Status Features

Feature	Description	Unit
x	real-time horizontal axis coordinates	/
y	real-time vertical axis coordinates	/
v_{lon}	real-time longitudinal speed	m/s
v_{lat}	real-time lateral speed	m/s
a_{lon}	real-time longitudinal acceleration	m/s^2
a_{lat}	real-time lateral acceleration	m/s^2

A. State Space

The state space S_t consists of the information that can be sensed and gathered during the movement of CAV. In the ramp merging scenario under mixed traffic conditions, the CAV must consider not only its own real-time position and motion state but also the state information of surrounding traffic vehicles to make informed and accurate decisions. Well-defined state features enable the agent to better understand the environment and improve decision-making. Therefore, interpretable, structured features are used as inputs.

We assume that the ego vehicle can observe only "neighboring vehicles, [30]" defined as the closest vehicles within a 125-meter radius. The state space of each vehicle is represented as a matrix of size $N \times W$, where N is the number of observed vehicles. In this study, we focus on the ego vehicle and four "neighboring vehicles" as the observable set. W represents the number of features used to describe the state of each vehicle, the selected features are detailed in Table I. If no vehicles are detected within the radar's range, zero padding is applied to the state space.

B. Action Space

The action space A_t refers to the real-time control decisions that a vehicle must make in response to its current environment. To more accurately model real-world scenarios, this paper defines the A_t as a two-dimensional continuous space comprising longitudinal and lateral acceleration, similar to [31]. This method effectively couples the driving process of the vehicle. CAV can modify its state by continuously performing lateral and longitudinal maneuvers, thereby avoiding collisions and enhancing driving efficiency in response to traffic conditions.

The range of longitudinal acceleration is $[-4.5\ m/s^2, 3.5\ m/s^2]$, and the range of lateral acceleration is $[-1.2\ m/s^2, 1.2\ m/s^2]$. The overall action space of the system is the joint action of all CAVs. The lower-level controller of the self-driving car incorporates the vehicle's kinematics model to convert the two-dimensional actions generated by the reinforcement learning algorithm into appropriate control commands for lateral and longitudinal vehicle velocities.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

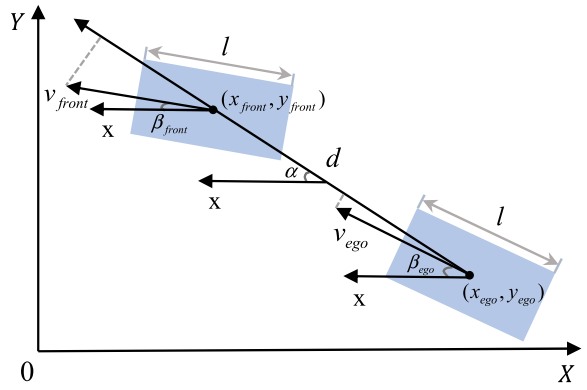


Fig. 3. Schematic of the two-dimensional Time-to-Collision (2D-TTC).

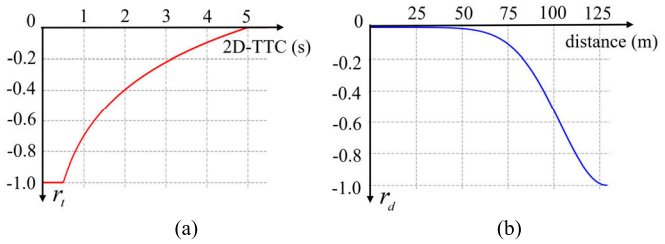


Fig. 4. The function graphs of the reward r_t and r_d. (a) r_t, (b) r_d.

C. Reward Function

A well-structured reward function is crucial for assisting agents in understanding the value of their actions and formulating effective strategies. In the ramp merging scenario, the reward function must account for both driving efficiency and the mitigation of risky behaviors, ensuring that the agent navigates the merging area safely and efficiently. Therefore, the reward function for each agent is designed as follows:

1) Penalty for Safety: Safety is the most critical factor in vehicle operation. To ensure that the CAV learns to drive safely, we introduce a crash penalty function r_c:

$$r_c = \begin{cases} -100, & \text{if collision happens;} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

If a collision occurs, the crash evaluation metric is assigned a value of -100; otherwise, the value remains 0. Given the paramount importance of safety, the penalty value is set to a large magnitude to prioritize collision avoidance.

2) Reward for Efficiency: To enhance efficiency, CAV is incentivized to travel at a higher speed. The speed reward function r_s is defined as:

$$r_s = \begin{cases} \dfrac{\sqrt{v_{lon}(t)^2 + v_{lat}(t)^2} - v_{\min}}{v_{\max} - v_{\min}}, \\ \quad \text{if } v(t) >= v_{\min}; \\ \dfrac{2(\sqrt{v_{lon}(t)^2 + v_{lat}(t)^2} - v_{\min})}{v_{\max} - v_{\min}}, \\ \quad \text{otherwise,} \end{cases} \tag{2}$$

where $v(t)$ represents the speed of the vehicle at time step t. $v_{lon}(t)$ and $v_{lat}(t)$ represent the longitudinal and lateral speed of the vehicle at time t, respectively. $v_{\min}$ and $v_{\max}$ are the set thresholds for the minimum and maximum speed of the vehicle. By conventional safe driving speed settings, we set the minimum and maximum speed to be $v_{\min} = 10\ m/s$ and $v_{\max} = 30m/s$, respectively.

3) Penalty for Risk Assessment: To assess the risk of CAVs in ramp merging scenarios, we use the two-dimensional Time-to-Collision (2D-TTC) metric for risk evaluation during these maneuvers. The metric is calculated as follows:

$$\begin{cases} \alpha = \arccos[(x_{ego} - x_{lead})/d] \\ T_{2d} = \min\left(\dfrac{d - l}{v_{ego}cos(\alpha - \beta_{ego}) - v_{lead}cos(\alpha - \beta_{lead})}, 200\right) \end{cases} \tag{3}$$

where x and y denote the horizontal and vertical coordinates of the current vehicle, respectively. d is the Euclidean distance between the centers of the ego vehicle and the front vehicle. l denotes the length of the vehicle. The steering angles of the current and front vehicles are denoted by β_{ego} and β_{front}, respectively. The angle between these two vehicles is represented by α. v_{ego} and v_{front} refer to the speeds of the current vehicle and the front vehicle, respectively. T_{2d} is the real-time value of 2D-TTC. Further details are shown in Fig. 3.

Based on the proposed 2D-TTC metric, the corresponding penalty function r_t is designed as follows:

$$r_t = \max\left[\min\left(\log\frac{T_{2d}}{\tilde{T}_{2d}}, 0\right), -1\right] \tag{4}$$

where $\tilde{T}_{2d}$ is a threshold value set at 5.0 s. When T_{2d} falls below this threshold, the CAV incurs a progressively increasing penalty. To prevent excessive penalties due to very small values of T_{2d}, a minimum penalty value of -1 is established. The variation of the penalty function is shown in Fig. 4(a).

4) Penalty for Driving Task Completion: To prevent excessive waiting times in the acceleration lane and avoid the formation of deadlock [25], it is necessary to penalize the waiting time of the CAV in the merging lane [16]. Additionally, this penalty helps guide the ramp CAV to merge into the main lane efficiently. The definition of the merging distance cost r_d is as follows:

$$r_d = -\exp\left(\frac{-(d_{acc} - L)^2}{10L}\right) \tag{5}$$

where d_{acc} represents the distance traveled by the vehicle in the acceleration lane, and L denotes the length of the acceleration lane. As the distance traveled increases, the penalty value progressively rises. The variation of the penalty function is shown in Fig. 4(b). Taking all the aforementioned factors into account, the overall reward function is defined as the weighted sum of each reward:

$$R = w_c r_c + w_s r_s + w_d r_d + w_t r_t \tag{6}$$

where w_c, w_s, w_d, and w_t are the weights assigned to various components of the reward function. After applying weighting coefficients for the four distinct driving characteristics based

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

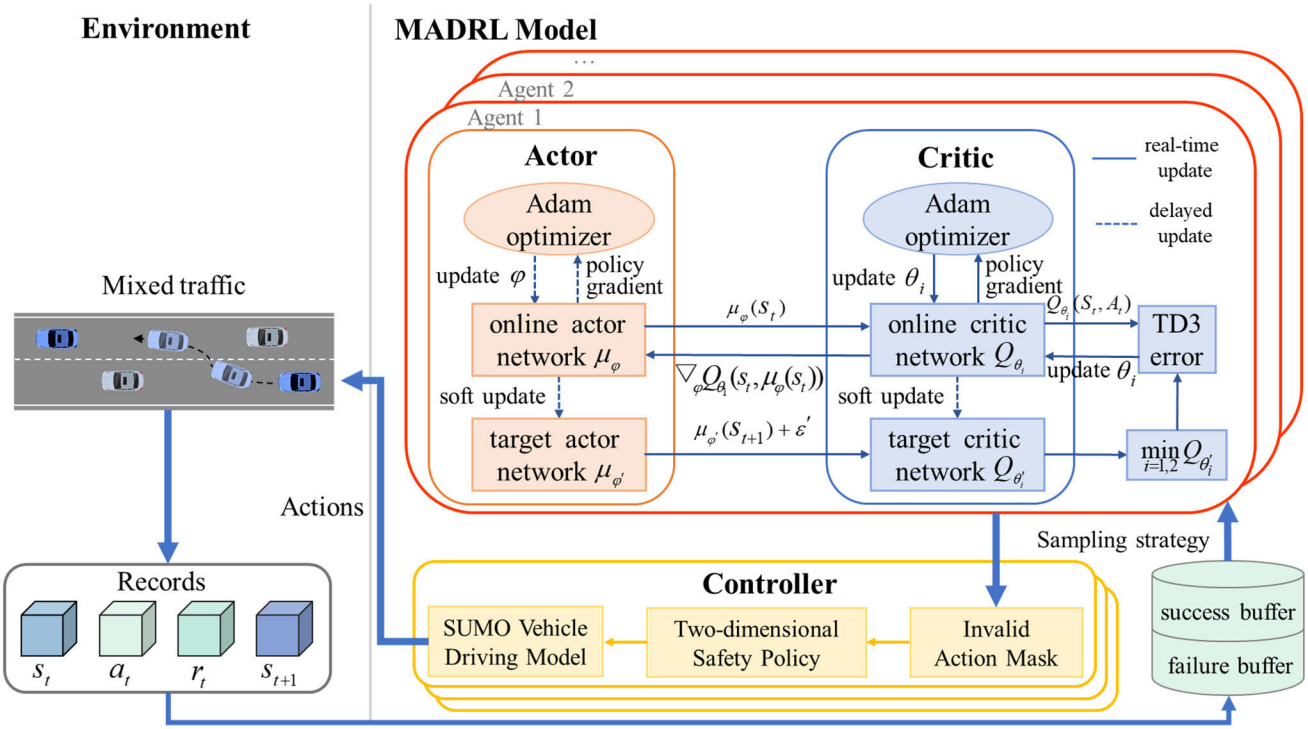


Fig. 5. Architecture of the proposed MADRL method (SQD-ITD3).

on the fine-tuning parameters, their respective values are 1, 1, 3, and 5, which are selected to balance conflicting objectives by prioritizing safety while preserving adequate incentives for efficient merging behavior.

IV. SQD-ITD3 Framework

A. MADRL Framework

For complex driving tasks in ramp merging scenarios, the Independent Twin Delayed Deep Deterministic Policy Gradient (ITD3) is employed as a framework for CAVs in two-dimensional continuous lateral and longitudinal control. ITD3 is a decentralized MADRL framework, which is a multi-agent extension of the TD3 [32] based on the actor-critic paradigm. It transforms a MADRL challenge involving n agents into n individual agent problems, allowing each agent to learn independently while treating others as part of the environment. The agents derive their policies from local observations. The joint action space in centralized joint learning grows exponentially with the increasing number of agents [33], making it difficult to solve the optimal value function. Therefore, this paper adopts the decentralized ITD3 framework, which offers greater scalability and adaptability for scenarios involving multiple agents.

Fig. 5 illustrates the structural framework of the designed SQD-ITD3 method. Each agent requires six networks: the actor network μ_φ with parameter φ, and the critic networks $Q_{\theta_{i=1}}$ and $Q_{\theta_{i=2}}$ with parameters $\theta_{i=1}$ and $\theta_{i=2}$, respectively, along with their respective target networks $\mu_{\varphi'}$, $Q_{\theta'_{i=1}}$, and $Q_{\theta'_{i=2}}$ with parameters φ', $\theta'_{i=1}$, and $\theta'_{i=2}$. The actor network μ_φ and its target network $\mu_{\varphi'}$ output the current action A_t and the target action A'_t based on the current state S_t and the next state S_{t+1}, respectively. When the critic network updates its parameters, the target value y_t is computed by selecting the smaller Q-value between the two critic target networks, as follows:

$$\begin{cases} \Delta\theta_i = \nabla_{\theta_i}\left(Q_{\theta_i}(S_t, A_t) - y_t\right)^2 \\ y_t = r_t + \gamma \min\limits_{i=1,2} Q_{\theta_i'}\left(S_{t+1}, A'_t\right) \\ A'_t = \mu_{\varphi'}(S_{t+1}) + \varepsilon'' \end{cases} \tag{7}$$

where γ is the discount factor and ε' is the added random noise, which follows a truncated normal distribution. The parameters of the actor network and the three target networks are updated according to the following formula:

$$\begin{cases} \Delta\varphi =_\varphi Q_{\theta_1}(S_t, \mu_\varphi(S_t)) \\ \theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i \\ \varphi' \leftarrow \tau\varphi + (1-\tau)\varphi' \end{cases} \tag{8}$$

where τ is the soft update coefficient. In contrast to the critic's action-value function parameters, which are updated once per step, the actor's policy network parameters are updated less frequently. Typically, the update frequency is set to every two steps, i.e., the actor network is updated once every two steps. This method helps reduce the variance of the approximate action-value function.

To further accelerate convergence and improve the learning efficiency of the agents, the parameter sharing technique [33] is employed. By sharing the same set of parameters across multiple agents, the strategies of each agent can influence and adjust one another, which effectively enhances the stability of the learned policies.

During training, the collection of irrelevant or misleading experiences is inevitable. To address this, we propose the

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

invalid action masking method [34], which masks invalid actions output by the actor network and replaces them by recomputing to obtain the correct actions. In this study, we consider the following types of invalid actions:

1) *Lane boundary violations:* The vehicle attempts to move laterally beyond lane boundaries when the vehicle is already at the edge of its lane.
2) *Practical constraints:* Mainline vehicles are restricted from moving into the ramp lane.
3) *Speed limit violations:* When an acceleration is applied and the resulting speed exceeds the vehicle's maximum speed or drops below 0 m/s, the acceleration will be recalculated accordingly.

For practical reasons, we also prohibit vehicles in the acceleration lane from moving to the left, as well as vehicles on the main roadway from moving onto the ramp.

B. Quadratic Actor Neural Network

Standard deep reinforcement learning agents typically employ Multilayer Perceptrons (MLPs) where neurons perform linear aggregation followed by non-linear activation. While effective, these linear neurons struggle to capture complex higher-order interactions between state features efficiently. To address this limitation and enhance the actor network's expressivity, we introduce Quadratic Neurons inspired by the diversity of biological neural systems [35]. Empirical and theoretical studies [35], [36], [37] have demonstrated that such quadratic structures offer significantly enhanced feature extraction capabilities compared to conventional linear neurons by explicitly modeling the interactions between input components.

A standard quadratic neuron processes an input state vector $s \in \mathbb{R}^n$ using a quadratic form, defined as:

$$y_{quad} = \sigma\left(\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} s_i s_j + \sum_{k=1}^{n} b_k s_k\right) = \sigma(s^T W_q s + w_l^T s + b) \tag{9}$$

where $W_q \in \mathbb{R}^{n\times n}$ represents the quadratic weight matrix capturing pairwise interactions, w_l is the linear weight vector, b is the bias, and $\sigma(\cdot)$ is the activation function.

Although powerful, the standard quadratic form in (9) introduces a computational complexity of $O(n^2)$ per neuron, which is computationally prohibitive for high-dimensional state spaces in real-time control. To resolve this, we employ a low-rank approximation strategy, like [37] and [38]. We decompose the weight matrix W_q into the product of two lower-dimensional matrices. This allows the quadratic operation to be computed via the element-wise product (Hadamard product) of two linear branches, reducing complexity to $O(n)$. The simplified operation is formulated as:

$$f_{quad}(s) = (W_1 s + b_1) \odot (W_2 s + b_2) \tag{10}$$

where $f_{quad}(s)$ represents the output of the efficient quadratic function. W_1 and W_2 are the trainable weight matrices for the two parallel linear paths, respectively, while b_1 and b_2 are their corresponding bias vectors. The symbol $\odot$ denotes the element-wise multiplication operator. This formulation allows the network to learn second-order feature interactions efficiently using standard matrix operations.

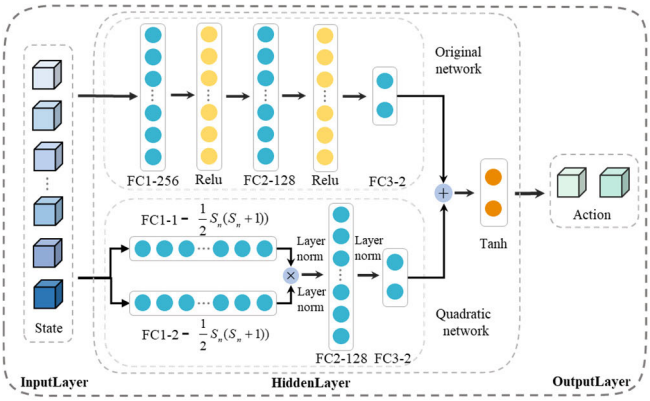


Fig. 6. Actor network architecture with embedded quadratic neurons.

As illustrated in Fig. 6, we integrate this efficient quadratic mechanism into the SQD-ITD3 actor network using a dual-path architecture. The network processes the input state through two distinct parallel branches. The first branch is the quadratic path, which implements Eq. (10) to capture nonlinear feature interactions. It consists of two parallel fully connected layers, denoted as FC_{1-1} and FC_{1-2}, which map the input state s to latent representations. To ensure numerical stability and consistent feature scaling during the interaction process, Layer Normalization is applied before the element-wise multiplication. The output of this quadratic branch, denoted as $Q(s)$, is expressed as:

$$Q(s) = LayerNorm(FC_{1-1}(s)) \odot LayerNorm(FC_{1-2}(s)) \tag{11}$$

To guarantee sufficient capacity for modeling the polynomial expansion of the input features, the number of neurons N_h in the hidden layers of the quadratic branch is determined by the combinatorial expansion of the input dimension S_n. This relationship is defined as:

$$F_n^c = \frac{1}{2} S_n(S_n + 1) \tag{12}$$

where S_n represents the dimension of the input state (i.e., the observation space size), and F_n^c ensures that the network has enough parameters to approximate all possible pairwise combinations of the state features.

The outputs from the quadratic interaction are subsequently processed by linear layers to map the features to the action space dimension. Finally, the network aggregates the representations from both the quadratic path and the original linear MLP path to form the final policy: distribution. A Tanh activation function is applied to the sum of these branches to bound the action range, yielding the final action A_t:

$$A_t = \text{Tanh}(A_{orig} + A_{quad}) \tag{13}$$

where A_{orig} denotes the output feature vector from the original linear MLP path, and A_{quad} denotes the transformed output from the quadratic path.

By explicitly embedding these quadratic neurons, the proposed actor network can directly model the quadratic

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

dependence of the policy on state variables. This architectural design provides significant expressive advantages through the natural incorporation of quadratic layers [39], thereby improving convergence speed and control precision in complex merging scenarios compared to purely linear architectures.

C. Adaptive Double Buffers

The TD3 algorithm employs the traditional experience replay method, which stores exploration-derived experience data in a memory buffer and updates the neural network parameters by randomly sampling from this buffer. This process helps break the temporal correlation between samples, thereby improving the stability of the neural network [32]. However, the quality of the samples selected through random sampling can vary, leading to the inclusion of irrelevant or low-quality samples that negatively impact training performance. To address this issue, this paper introduces an adaptive double buffers mechanism based on the success rate. This mechanism partitions the experience buffer into two sub-buffers, labeled $M_{success}$ and $M_{failure}$, to categorize and store different types of experience data. During training, experience data are selected from each buffer according to different probabilities, which are determined by the current merging success rate.

DRL is inherently delayed, with past experiences often influencing the current state of the CAV. To account for this, in addition to experience buffers $M_{success}$ and $M_{failure}$, each CAV is assigned a temporary experience buffer M_{temp} to store its driving experiences within a given round. Assuming that the CAV fails in this round at moment t, the corresponding traveling data at moment $t-a$ can be determined to have a negative impact on the CAV and is considered failed traveling data. In this study, we classify CAV failure into two types:

1) *Collision failure:* When the CAV collides with another vehicle. Here we set $a = 25$. We regard this window as the most informative for negative learning signals.
2) *Timeout failure:* When the CAV fails to reach its destination within the time limit. For this case, we set $a = 50$. Based on our observations, timeouts typically occur in the early training stage due to excessive exploratory actions, where the CAV may remain idle for extended periods. Therefore, a larger a is adopted to capture these prolonged ineffective behaviors.

If the result of the CAV task is a failure, the driving experience within the corresponding experience buffer M_{temp} during the time interval $[t-a, t]$ will be transferred to experience buffer $M_{failure}$, while the remaining experiences will be stored in experience buffer $M_{success}$. If the task is successful, all experiences will be stored in $M_{success}$. During training, a different number of sample data will be selected from the two pools, as expressed in the following equation:

$$\begin{cases} N_{failure} = \lambda(1-\beta)B + \kappa \\ N_{success} = B - N_{failure} \end{cases} \tag{14}$$

where $N_{failure}$ and $N_{success}$ represent the number of data samples drawn from $M_{failure}$ and $M_{success}$, respectively. β is the success rate over the last 200 rounds. B is the batch size. λ is the discount factor applied to the number of failed experiences, set to 0.25 in the experiments to prevent excessive sampling of failed experiences; and a constant term κ, valued at 5 in the experiments, is added to the failed experiences to avoid under-sampling and ensure diversity in the sample data.

D. Two-Dimensional Safety Policy

DRL-based control models have the potential to surpass human driving behavior by exploring diverse driving strategies through trial and error. However, these models may also generate unexpected or anomalous behaviors that could lead to significant safety risks. To mitigate this issue, we propose an additional 2D safety policy that replaces abnormal actions generated by the proposed MADRL model, based on the *Krauss* [40] car-following model, to accelerate learning and improve the policy's overall effectiveness. The model operates under the assumption that even if the vehicle ahead suddenly brakes, the following vehicle should maintain a speed that allows it to stop safely, thus avoiding a collision. The safe speed $v_{safe}(t)$, is calculated as follows:

$$v_{safe}(t+1) = v_{i-1}(t) + \frac{g(t) - v_i(t)\tau}{\tau - [v_{i-1}(t) + v_i(t)]/2b} \tag{15}$$

where $v_i(t)$ and $v_{i-1}(t)$ represent the longitudinal speeds of vehicle i and the front vehicle $i-1$ at time t, respectively. $g(t)$ is the distance between the two vehicles at time t, and b denotes the maximum deceleration, set to $-4.5\ m/s^2$. Additionally, τ represents the driver reaction time, which is set to 0.5 seconds.

The *Krauss* model prioritizes collision-free driving, which results in a conservative calculation of the safe speed. However, overly cautious driving can result in the stop-and-go problem. Thus, we calculate the safe velocity only when the 2D-TTC, as proposed in Section III-C, is less than the safety threshold, thereby effectively balancing safety and efficiency. The current vehicle identifies the nearest vehicle ahead as the leading vehicle to calculate the safe speed. Based on this, the safety policy is proposed:

$$v_{lon}(t+1) = \begin{cases} min(v_{lon}(t+1), v_{safe}(t+1)), \\ \quad \text{if } T_{2d} < \tilde{T}_{2d_{\max}}; \\ v_{lon}(t+1), \\ \quad \text{otherwise}, \end{cases} \tag{16}$$

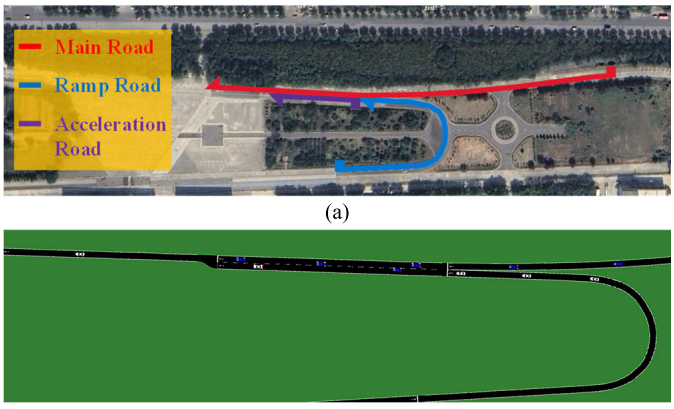
where $v_{lon}(t+1)$ represents the longitudinal speed at the next time step, derived from the longitudinal acceleration produced by the MADRL model at time t. $v_{safe}(t)$ is the safe speed calculated using (15). $\tilde{T}_{2d_{max}}$ denotes the maximum threshold for 2D-TTC, set to 2.5s. This threshold acts only as the online trigger of the safety-policy intervention. This safety policy helps shield against unreasonable behaviors and facilitates faster model convergence while ensuring vehicle safety.

V. Experimental Analysis

In this section, we will introduce the simulation environment used in the experiment and the construction of the ramp merging scenario. An ablation study is performed to assess the performance of various components, and the proposed method is also compared with several state-of-the-art baseline approaches under different mixed-traffic conditions.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)

(b)

Fig. 7. SUMO simulation based on real-world ramp merging area: (a) View from Google Maps at the real-world ramp, (b) SUMO platform.

TABLE II
SETTINGS OF THE SIMULATION SCENARIO

Traffic simulator terms	Value
Main road length	546.5 m
Ramp road length	285.5 m
Acceleration road length	129.2 m
Simulation step	0.2s
Initial speed	[18, 22] m/s
Maximum speed	30 m/s
Maximum acceleration	3.5 m/s^2
Desired deceleration	4.5 m/s^2
Minimum gap	2.5 m
Time headway	2.0 s
Main road spawn points	[5,45,85,125,165,205,245]
Ramp road spawn points	[0,40,80,120,160,200,240]
Random position noise	[-2, 2] m
Easy traffic mode	1-3 CAVs + 1-4 HDVs
Medium traffic mode	2-4 CAVs + 2-5 HDVs
Hard traffic mode	4-6 CAVs + 3-6 HDVs

A. Experimental Design

To enhance the realism of the experiment, we encoded a real-world traffic network in the SUMO simulation platform [41], specifically focusing on the ramp merging area in the CAV test field at Chang'an University. The real-world environment is illustrated in Fig. 7(a), while the simulation environment is depicted in Fig. 7(b). To increase the complexity of the simulated environment, we define three different traffic modes [16], [28] by setting different numbers of initial CAVs and HDVs in the lanes, namely easy, medium and hard traffic densities. The detailed settings of the simulation scenario are shown in Table II.

Specifically, in each training round, the initial speed of the generated vehicles is randomly selected within the range of 18 to 22 m/s, with a maximum speed set at 30 m/s. Random noise is added to the observed state of each CAV. The lateral and longitudinal control models for HDVs are based on the SL2015 [41] and intelligent driver model (IDM) [42], respectively, while the movement of CAVs is governed by the designed method. The simulation step size is set to 0.2 seconds. Vehicles that have completed the driving task, defined as safely traveling the entire length of the road, as well as those involved in collisions, are removed from the simulation.

TABLE III
MAIN HYPERPARAMETER DETAILS

Hyperparameters	Value	Hyperparameters	Value
Hidden units 1	256	Hidden layers activation	ReLU
Hidden units 2	128	Optimizer type	Adam
Discount factor	0.99	Batch size	256
Target update rate	0.005	Start steps	1000
Optimizer learning	0.0005	Max episode steps	200
Actor learning rate	0.0005	Number of episodes	10000
Critic learning rate	0.0005	Memory buffer size	1000000

TABLE IV
THE INFLUENCE OF DIFFERENT w_t'S VALUES ON THE INDICATORS UNDER THE MEDIUM TRAFFIC MODE. NOTE THAT OTHER WEIGHTING FACTORS ARE KEPT UNCHANGED

Metrics	$w_t = 1$	$w_t = 5$(we choose)	$w_t = 10$
Speed (m/s)	**25.92±0.02**	25.68±0.34	25.28±0.67
2D-TTC (s)	**48.18±0.42**	47.24±1.24	45.98±1.61
Hf Times	0.11±0.05	**0.08±0.02**	0.10±0.01
Success Rate	**1.00±0.00**	**1.00±0.01**	0.99±0.02

B. Training Details

To validate the effectiveness of the proposed method, this study designs three types of comparative experiments. The specific evaluation scenarios are summarized as follows:

1) Sensitivity analysis of the reward function: Since the performance of MADRL is sensitive to the scale and balance of reward signals, we first conduct experiments under the medium traffic mode to investigate the impact of different weight combinations on the agent's behavior. This step aims to identify the optimal hyperparameter setting that balances safety and efficiency.
2) Ablation study under medium traffic mode: Using the ITD3 algorithm as the classic baseline, comparisons are made with variant algorithms incorporating different components of our proposed method, including: S-ITD3 (ITD3 + two-dimensional safety policy), Q-ITD3 (ITD3 + quadratic actor neural network), and D-ITD3 (ITD3 + adaptive double buffer mechanism).
3) Comparative experiments across various traffic modes: The proposed SQD-ITD3 method is compared with several state-of-the-art algorithms. Specifically, DDPG [43], SAC [44], and Proximal Policy Optimization (PPO) [45] are selected as baseline algorithms. For adaptation to multi-agent environments, these baseline algorithms are modified using independent learning and parameter

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
TRAINING PERFORMANCE COMPARISON FOR DIFFERENT METHODS IN THREE TRAFFIC MODES

Scenarios	Metrics	IPPO	ITD3	ISAC	IDDPG	SQD-ITD3
Easy Mode	Speed (m/s)	16.60±1.35	22.28±0.63	23.12±1.36	23.28±2.36	**26.97±0.05**
	2D-TTC (s)	48.30±2.02	48.51±1.71	49.13±1.49	45.68±1.39	**59.50±0.62**
	Hf Times	2.12±0.16	0.56±0.15	0.49±0.07	0.96±0.33	**0.03±0.01**
	Success Rate	0.40±0.04	0.85±0.04	0.87±0.05	0.79±0.03	**1.00±0.00**
	Collision Times	0.62±0.05	0.17±0.02	0.14±0.03	0.24±0.05	**0.00±0.00**
Medium Mode	Speed (m/s)	14.57±2.26	18.47±2.06	20.00±1.36	21.27±2.20	**25.68±0.34**
	2D-TTC (s)	31.57±6.18	36.02±1.83	35.72±0.44	33.08±3.44	**47.24±1.24**
	Hf Times	3.18±0.98	1.00±0.25	0.85±0.28	1.47±0.79	**0.08±0.02**
	Success Rate	0.15±0.02	0.64±0.08	0.69±0.12	0.63±0.15	**1.00±0.00**
	Collision Times	1.23±0.14	0.42±0.05	0.43±0.15	0.56±0.27	**0.01±0.00**
Hard Mode	Speed (m/s)	9.02±4.50	20.59±5.72	18.35±2.33	22.14±4.25	**23.89±0.21**
	2D-TTC (s)	30.58± 15.37	28.11±3.46	30.49±2.64	25.15±8.33	**37.22±0.76**
	Hf Times	3.07±1.76	2.30±1.50	1.25±0.49	2.50±2.64	**0.33±0.09**
	Success Rate	0.00±0.00	0.42±0.13	0.62±0.13	0.47±0.38	**0.97±0.03**
	Collision Times	1.89±1.15	0.83±0.17	0.62±0.12	1.04±0.92	**0.05±0.06**

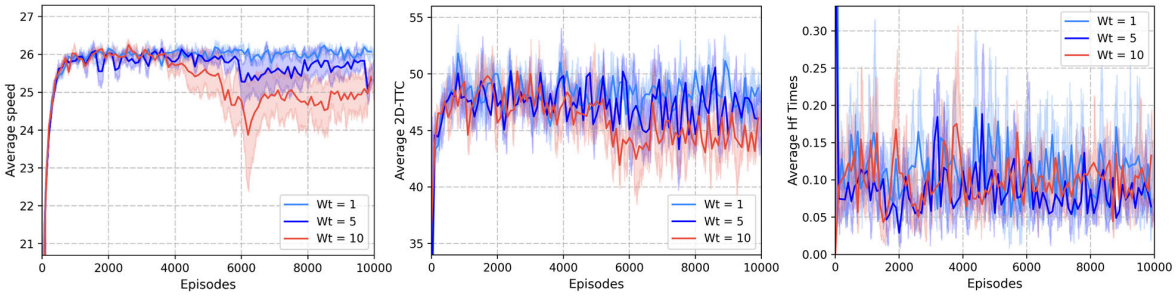


Fig. 8. Functional performance comparison of SQD-ITD3 with different w_t parameter choices in medium traffic mode. (a) Speed, (b) 2D-TTC, (c) Hf Times.

sharing strategies, resulting in three derived algorithms: IDDPG, ISAC, and IPPO. Finally, SQD-ITD3 is compared with both these derived algorithms and ITD3. Notably, a rule-based First-In-First-Out (FIFO) is included under medium traffic conditions for comparison with traditional control strategies.

The main hyperparameters of the proposed method are given in Table III. We utilized three random seeds to train each MADRL algorithm for 10,000 rounds across different traffic difficulty modes. All baselines share the same state/action spaces, the network configuration in Table III (apart from the quadratic branch specific to SQD-ITD3), and identical training budgets and hyperparameters; results in Tables IV and V are reported as mean ± standard deviation over three seeds. The performance of different algorithms is compared by the changes in metrics during training as well as evaluation.

The experiments were conducted on the SUMO simulation platform with the following software/hardware stack: Python 3.7, Ubuntu 18.04, Intel® i7-14700F (28 logical cores), NVIDIA GeForce RTX 4060 Ti GPU, and 32 GB RAM.

C. Sensitivity Analysis

The reward function defined in Eq. (6) consists of four components: collision penalty (w_c), speed reward (w_s), distance cost (w_d), and risk assessment penalty (w_t). To optimize the critical trade-off between driving efficiency (governed by w_s) and safety (governed by w_t), we evaluated three distinct configurations under the medium traffic mode by adjusting the ratio of the risk assessment penalty to the speed reward. Specifically, with w_s fixed at 1, we set w_t to 1, 5, and 10, representing equal, high, and very high safety priorities relative to efficiency, respectively.

The quantitative results are summarized in Fig. 8 and Table IV. In Fig. 8, the solid line represents the mean value smoothed over the most recent 100 rounds, while the shaded area indicates the variability across three random seeds. In Table IV, the best results in each column are highlighted in bold. Hazard Frequency Times (Hf Times) represents the average number of instances per CAV per episode where the real-time 2D-TTC falls below the minimum hazard threshold of 2.0 seconds, distinguishing safe from unsafe events. This threshold is applied only as a post-hoc evaluation criterion and does not trigger the safety policy, whose online intervention threshold remains 2.5 s as defined in Eq. (16).

As the value of w_t increases from 1 to 10, average speed and average 2D-TTC decrease concurrently. The average driving speed falls from 25.92 m/s to 25.28 m/s, while average 2D-TTC drops from 48.18 s to 45.98 s. This trend indicates higher penalty weights induce more conservative driving, yet reduced speed distorts the distance-based 2D-TTC statistic.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

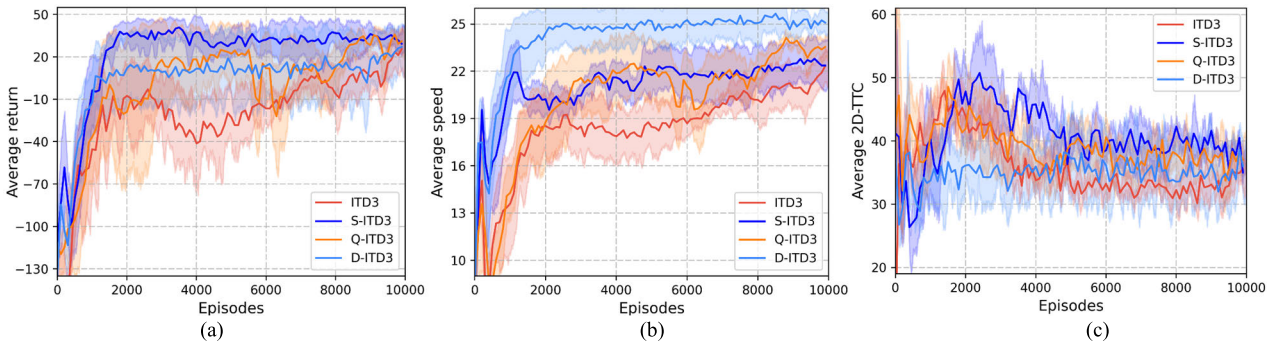


Fig. 9. Functional performance comparison during training in medium traffic mode. (a) Reward, (b) Speed, (c) 2D-TTC.

However, increasing w_t does not lead to a monotonic improvement in overall performance. According to the Hf Times metric, the configuration with $w_t = 5$ achieves the lowest hazard frequency of 0.08, outperforming both $w_t = 1$ with a value of 0.11 and $w_t = 10$ with a value of 0.10. The slight performance degradation observed at $w_t = 10$ can be attributed to over-conservatism, where excessive risk penalties result in hesitant merging behavior and occasional timeout failures, as reflected by a reduced success rate of 0.99.

Overall, the configuration with $w_t = 5$ provides the best balance between safety and efficiency. It achieves a 100% success rate while incurring only a negligible reduction in driving speed. Therefore, $w_t = 5$ is selected for the subsequent experiments.

D. Ablation Study

This section reports experiments to evaluate the different key features proposed in medium traffic modes. Fig. 9 shows the results for reward, average driving speed, and 2D-TTC. The solid line in each figure represents the mean, smoothed over the past 100 rounds, while the shaded area indicates the fluctuations of three random seeds.

As shown in Fig. 9(a), all component-augmented variants achieve higher episodic rewards compared with the baseline ITD3. In particular, S-ITD3 attains the highest final reward of approximately 35, representing an improvement of about 40% over ITD3. This improvement stems mainly from its ability to prevent collision events, thereby avoiding large negative penalties during training. Both Q-ITD3 and D-ITD3 also demonstrate faster convergence and higher peak rewards than ITD3, improving by roughly 20–30%, which confirms that the quadratic representation and adaptive sampling strategy enhance policy learning stability and efficiency.

As illustrated in Fig. 9(b), D-ITD3 attains the highest average speed (approximately 25 m/s), which can be attributed to its more efficient use of high-quality experiences and its ability to exploit positive rewards during training. Q-ITD3 also achieves higher speeds than the baseline, primarily due to its superior feature representation capability that captures complex vehicle interactions. In contrast, S-ITD3 exhibits a slightly lower peak speed compared with the other variants, as expected, because its safety policy enforces corrective actions in potentially unsafe situations. However, this conservative bias effectively suppresses risky acceleration behaviors, resulting in smoother and safer driving performance.

As shown in Fig. 9(c), the safety metric 2D-TTC further confirms that the proposed two-dimensional safety policy provides the most significant enhancement in driving safety. Compared with ITD3, S-ITD3 achieves the largest increase in average 2D-TTC, demonstrating its ability to prevent unsafe proximity and potential collisions. Q-ITD3 and D-ITD3 also perform better than the baseline, but to a lesser extent. This indicates that improved feature representation and enhanced sample quality indirectly contribute to safety by enabling smoother and more anticipatory vehicle control. Overall, all three components achieve a balance across the key performance metrics, including higher episodic rewards, competitive cruising speeds, and greater 2D-TTC gains, which demonstrate more efficient, robust, and safe merging behavior.

E. Comparison Experiment

In this subsection, we test the effectiveness of the proposed SQD-ITD3 method. The experiment compares the overall performance and various metrics of five methods in the task of controlling vehicle ramp merging.

Figs. 10 and 11 illustrate the changes in reward and average speed of the five methods during training across three designed traffic modes with varying difficulty levels. In each figure, the solid line denotes the mean value smoothed over the most recent 100 rounds, while the shaded area represents the fluctuations of three random seeds. Across all traffic scenarios, the proposed SQD-ITD3 algorithm consistently outperforms the baseline methods, exhibiting superior and faster learning performance. Its training curve rises steadily and converges in the shortest time. Moreover, SQD-ITD3 attains a higher and more stable average speed than the comparison algorithms.

Table V presents the final performance comparison of all methods. Collision Times indicate the average number of vehicle collisions per round. In each column of Table V, bold numbers highlight the best.

The training results demonstrate that the overall performance of the proposed SQD-ITD3 method surpasses that of other algorithms across various traffic scenarios, particularly in terms of safety, efficiency, and task completion. Specifically, the speed performance of SQD-ITD3 exceeds that of all other algorithms in every difficulty scenario, contributing to

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

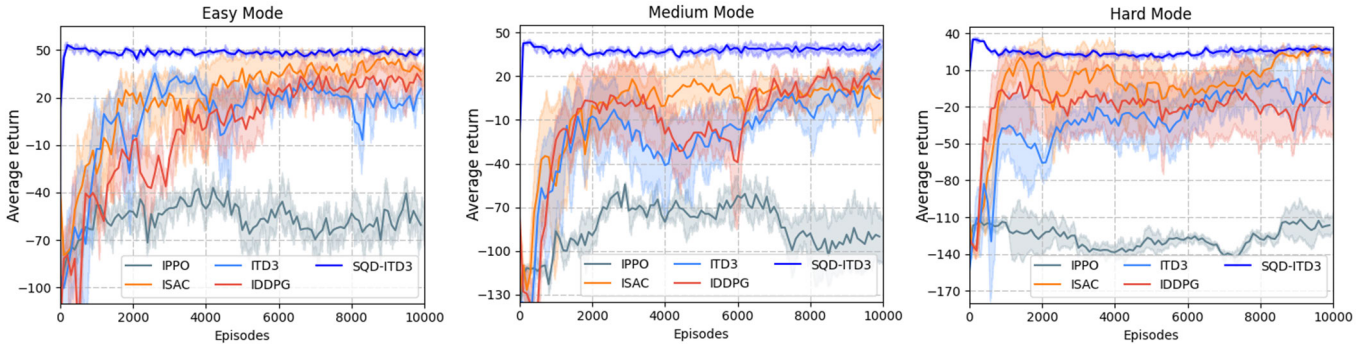


Fig. 10. Average reward comparison between the proposed policy (SQD-ITD3) and four benchmarks in three traffic modes.

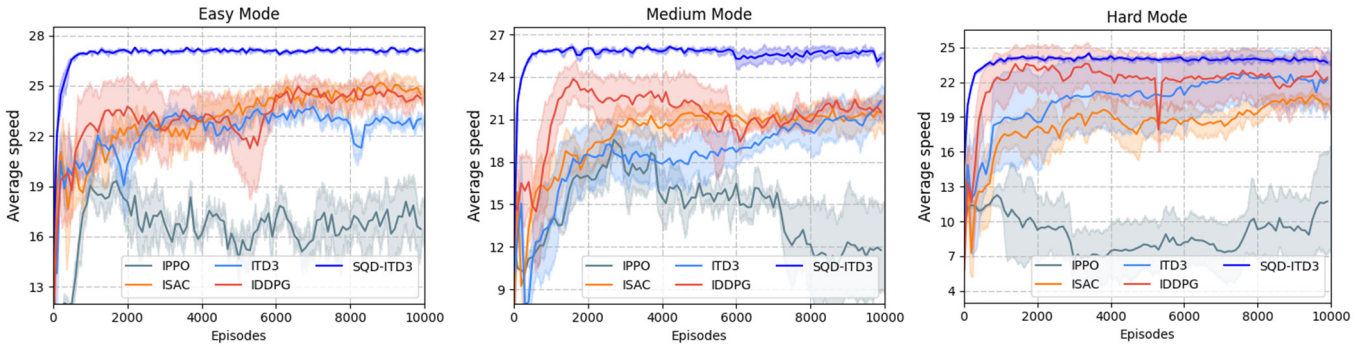


Fig. 11. Average speed comparison between the proposed policy (SQD-ITD3) and four benchmarks in three traffic modes.

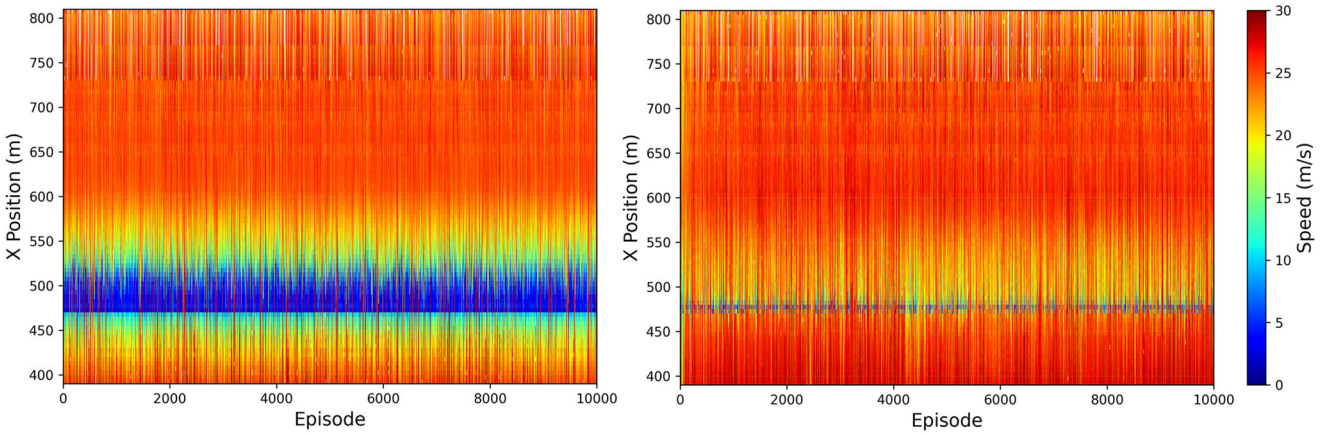


Fig. 12. The velocity-position heatmaps of different algorithms during the training process. (a) FIFO (b) SQD-ITD3.

improved task completion efficiency. Simultaneously, SQD-ITD3 maintains high safety levels, showing the best results in the 2D-TTC and Hf Times metrics, indicating its effectiveness in avoiding dangerous events and maintaining a safe distance.

A near-zero Hf Times indicates that the 2D-TTC rarely drops below the 2.0 s evaluation threshold, and thus seldom approaches the 2.5 s online intervention threshold. Notably, SQD-ITD3 achieves a 100% success rate in both simple and medium modes, with nearly zero collisions, ensuring a collision-free and efficient operational process.

In hard mode, SQD-ITD3 records a collision count of 0.05, a low value but a difference compared to the collision-free outcomes in easy and medium modes. This discrepancy is attributed to high vehicle density in this scenario, resulting in a complex traffic environment. Although CAVs can effectively predict and slow down to avoid danger, surrounding HDVs may not respond in time, leading to collision events. Nevertheless, the success rate remains at 97%, indicating that SQD-ITD3 performs effectively in complex and dense traffic environments in most cases.

Furthermore, we conducted a comparative analysis with the rule-based First-In-First-Out (FIFO) method specifically under the medium traffic mode. As expected, the FIFO method exhibited safety performance comparable to SQD-ITD3, with a mean 2D-TTC of 45.73 s, which can be attributed to the strict safety constraints inherent in its design. However, the rule-based approach revealed significant limitations regarding traffic efficiency. Fig. 12 presents the speed-position heatmaps

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

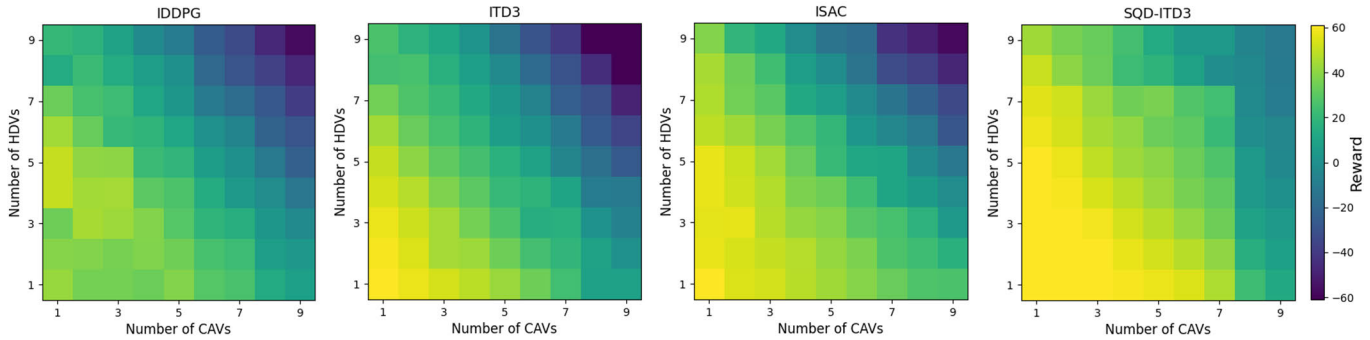


Fig. 13. Evaluation of reward heatmaps for SQD-ITD3 compared to three benchmarks across varying CAV to HDV ratios.

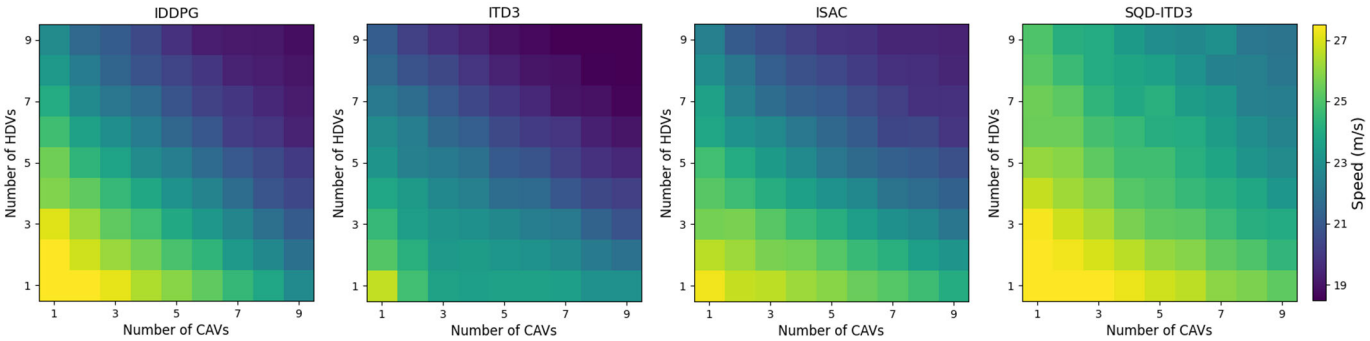


Fig. 14. Evaluation of speed heatmaps for SQD-ITD3 compared to three benchmarks across varying CAV to HDV ratios.

for the two methods. SQD-ITD3 demonstrates smooth speed transitions, with vehicles maintaining high speeds throughout the merging zone. Conversely, FIFO shows pronounced deceleration near the merging point. This is due to FIFO's inability to dynamically regulate gaps; vehicles are compelled to brake heavily to comply with rigid sequencing protocols, thereby causing congestion. Consequently, while the non-learning baseline guarantees safety, SQD-ITD3 achieves a more optimal balance, ensuring safety while significantly surpassing the rule-based method in traffic efficiency.

To evaluate the generalization capability of the policies obtained through training, we saved the optimal network models derived from IDDPG, ITD3, ISAC, and SQD-ITD3 methods under hard mode for further testing. In setting up the evaluation scenario, we modified the original vehicle generation points from 40m intervals to 30m, resulting in nine generation points along both the main road and the ramp. CAVs and HDVs were randomly generated from these points, with an additional uniformly distributed random noise of [-2.0 m, 2.0 m] applied for each turn.

We tested the resulting network model across 81 scenarios, varying the numbers of CAVs and HDVs for evaluation. In each scenario, we conducted 200 evaluation rounds for each model. The 81 scenarios differ only in the numbers of CAVs and HDVs and in the vehicle generation points; the evaluation therefore characterizes generalization to unseen traffic compositions to communication impairments or heterogeneous driving behaviors.

Figs. 13, 14, and 15 illustrate the performance of the various strategies in terms of average reward value, speed, and 2D-TTC for different proportions of mixed traffic. The yellow grids represent high mean values, while the black grids indicate low

values. The similarity in grid colors results in heatmaps that approximate solid colors, suggesting that the corresponding strategies generalize better across unseen traffic compositions. Notably, the SQD-ITD3 strategy displays a more homogeneous heatmap color compared to the IDDPG, ITD3, and ISAC strategies. Furthermore, the SQD-ITD3 strategy maintains optimal speed and 2D-TTC values even in scenarios not encountered during the previous training process. This indicates that the proposed SQD-ITD3 method effectively enhances the driving efficiency of CAVs in entrance ramp merging scenarios while ensuring higher safety.

The final performance of all methods during the evaluation phase is presented in Table VI, demonstrating the generalization capability of the proposed approach. For instance, SQD-ITD3 enhances the average reward value by 144.3%, 139.0%, and 45.5% compared to the IDDPG, ITD3, and ISAC methods, respectively. In terms of efficiency, compared with IDDPG, ITD3, and ISAC methods, the average

TABLE VI
FINAL EVALUATION PERFORMANCE OF DIFFERENT METHODS

	Reward	Speed	2D-TTC	Success Rate
IDDPG	11.70	22.67	33.51	0.67
ITD3	11.96	21.41	35.69	0.81
ISAC	19.65	22.68	36.24	0.82
SQD-ITD3	**28.58**	**24.75**	**45.73**	**0.98**

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

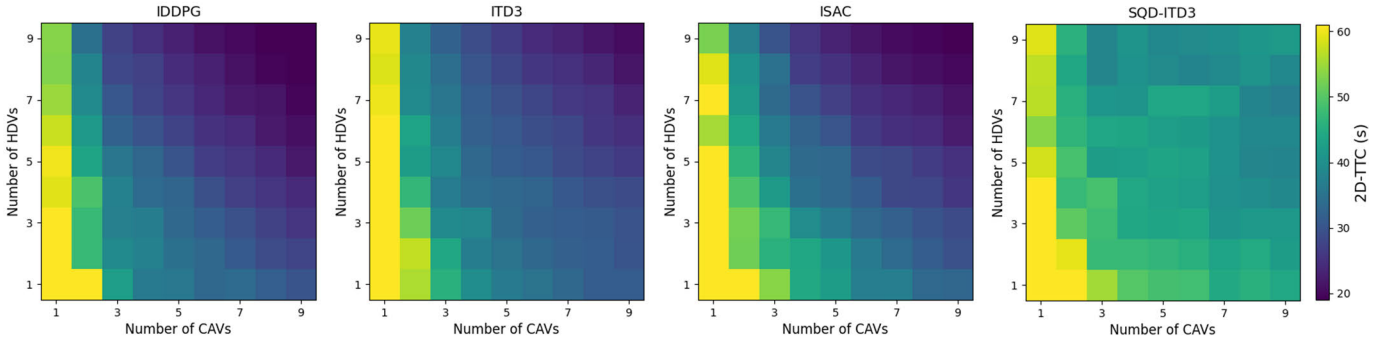


Fig. 15. Evaluation of 2D-TTC heatmaps for SQD-ITD3 compared to three benchmarks across varying CAV to HDV ratios.

speed value of the SQD-ITD3 method is increased by 9.2%, 15.6%, and 9.1%, respectively. In terms of safety, the average 2D-TTC value of the SQD-ITD3 method is improved by 36.5%, 28.1%, and 26.2% compared with IDDPG, ITD3, and ISAC methods, respectively. Furthermore, the average success rate of SQD-ITD3 increases by 46.3%, 21.0%, and 19.5% compared to the IDDPG, ITD3, and ISAC methods, respectively. These results indicate that CAVs utilizing the SQD-ITD3 strategy can reach target points more efficiently than those based on other methods, all while avoiding collisions and maintaining a safe distance. In other words, the policy trained by SQD-ITD3 enables CAVs to perform well even on traffic compositions unseen in training, further demonstrating its generalization capability.

During the evaluation phase, we also measured the end-to-end online inference latency of the trained SQD-ITD3 policies. The average computation time was 0.373 ms, with a maximum of 1.197 ms and a minimum of 0.230 ms. These results indicate that the computational overhead of the proposed algorithm is negligible and that real-time inference can be feasibly achieved on real vehicles with reasonable on-board hardware. This is also an aspect we plan to further explore in future work.

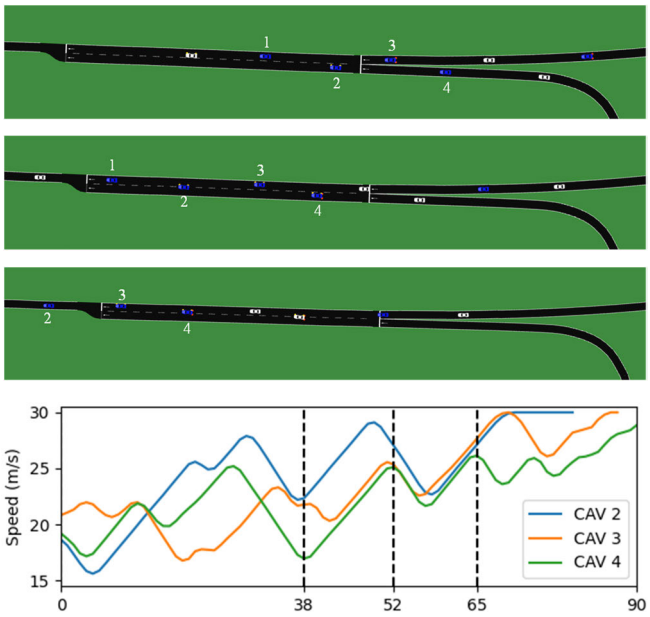


Fig. 16. Frames show the learned policy. The below figure shows the corresponding speed of the CAVs.

F. Policy Interpretation

In this subsection, we present an example to illustrate the effectiveness of the proposed strategy in CAV merging. As shown in Fig. 16, snapshots at time steps 38, 52, and 65 depict the speeds of CAV 2, 3, and 4.

At time step 25, CAV 2 enters the merging area, prompting CAV 3 to decelerate to create a sufficient gap for merging. CAV 2 accelerates and successfully merges into the main lane while maintaining a safe distance from the vehicle ahead. At this same time step, CAV 4 enters the acceleration lane; however, due to the front vehicle deceleration, there is an insufficient merging gap, causing CAV 4 to first decelerate to maintain a safe distance. As CAV 4 approaches the end of the merge, it begins to accelerate to join the main road, consistently maintaining a speed lower than that of the vehicle ahead to ensure safety. By time step 65, CAV 3 completes the merging operation and starts to decelerate to maintain a greater distance from the vehicle in front.

VI. Conclusion

In this paper, we model the problem of CAV ramp merging under mixed traffic as an MDP and propose a novel SQD-ITD3 method to address it. This method incorporates invalid action masking, parameter sharing, and a distributed learning mechanism. Additionally, we develop an actor network structure with quadratic neurons to enhance feature representation capabilities, along with an adaptive double buffers mechanism to improve experience utilization efficiency. Furthermore, a 2D safety policy is introduced to enhance the safety of the merging process.

Training results from the developed mixed traffic flow scenarios, which include three levels of difficulty, demonstrate that the proposed method can achieve safe merging decisions in ramp scenarios. In the evaluation phase, it maintains excellent performance on traffic compositions that did not appear during training. Furthermore, compared to four baseline schemes, the proposed algorithm exhibits excellent training efficiency and success rates.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

While the proposed method has demonstrated promising performance in the simulation environment based on the real-world layout, its performance in real traffic remains an open question. This paper utilizes SUMO's built-in IDM and SL2015 models to simulate human driving behavior; there remains a gap compared to real-world human drivers. Therefore, our future work will focus on developing a more detailed representation of human driving behavior based on data collected from real-world environments and extending the evaluation to real-vehicle experiments. Additionally, we will explore modeling driver heterogeneity by assigning random distributions to IDM parameters to further enhance the distributional robustness of the trained policy. Finally, we plan to investigate object information loss scenarios to further enhance the comprehensiveness of the proposed method.

REFERENCES

[1] G. Ozbilgin, U. Ozguner, O. Altintas, H. Kremo, and J. Maroli, "Evaluating the requirements of communicating vehicles in collaborative automated driving," in *Proc. 4th IEEE Intell. Vehicles Symp.*, Jun. 2016, pp. 1066–1071.

[2] S. Chen, H. Min, Y. Fang, X. Wu, B. Li, and X. Zhao, "Uncertainty-aware sensor data anomaly detection for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp.* (IV), Jun. 2024, pp. 478–483.

[3] Y. Zhou, S. Ahn, M. Wang, and S. Hoogendoorn, "Stabilizing mixed vehicular platoons with connected automated vehicles: An H-infinity approach," *Transp. Res. Proc.*, vol. 132, pp. 152–170, Feb. 2020.

[4] H. Li, Z. Huang, L. Qin, S. Zheng, and Y. Yang, "Modeling and simulation of vehicle group lane-changing behaviors in upstream segment of ramp areas under a connected vehicle environment," *Smart Resilient Transp.*, vol. 1, no. 1, pp. 17–29, May 2019.

[5] Z. Huang, Z. Zhang, H. Li, L. Qin, and J. Rong, "Determining appropriate lane-changing spacing for off-ramp areas of urban expressways," *Sustainability*, vol. 11, no. 7, p. 2087, Apr. 2019.

[6] S. Zheng, H. Li, Z. Huang, K. Li, and L. Qin, "Flow-balanced-based collaborative strategies and simulation of vehicle group behaviors for on-ramp areas," *Simul. Model. Pract. Theory*, vol. 103, Sep. 2020, Art. no. 102093.

[7] T. Mo, K. Li, J. Zhang, L. Qin, Z. Huang, and H. Li, "Collaborative strategies and simulation of vehicle group behaviors for off-ramp areas," *J. Adv. Transp.*, vol. 2020, Dec. 2020, Art. no. 8817364.

[8] H. Wang, S. Yuan, M. Guo, X. Li, and W. Lan, "A deep reinforcement learning-based approach for autonomous driving in highway on-ramp merge," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 235, nos. 10–11, pp. 2726–2739, Sep. 2021.

[9] Z. H. Khattak, B. L. Smith, H. Park, and M. D. Fontaine, "Cooperative lane control application for fully connected and automated vehicles at multilane freeways," *Transp. Res. C, Emerg. Technol.*, vol. 111, pp. 294–317, Feb. 2020.

[10] S. Xia et al., "On-ramp merging strategy integrating dynamic lane-changing decision and quasi-uniform B-spline trajectory optimization," *IEEE Trans. Veh. Technol.*, vol. 73, no. 12, pp. 18333–18347, Dec. 2024.

[11] X. Liao et al., "Game theory-based ramp merging for mixed traffic with unity-SUMO co-simulation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 9, pp. 5746–5757, Sep. 2022.

[12] I. Nishitani, H. Yang, R. Guo, S. Keshavamurthy, and K. Oguchi, "Deep merging: Vehicle merging controller based on deep reinforcement learning with embedding network," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 216–221.

[13] L. Xiong, Z. Li, D. Zhong, P. Xu, and C. Tang, "Rule-guidance reinforcement learning for lane change decision-making: A risk assessment approach," *Chin. J. Mech. Eng.*, vol. 38, no. 1, Mar. 2025, Art. no. 30.

[14] A. Irshayyid, J. Chen, and G. Xiong, "A review on reinforcement learning-based highway autonomous vehicle control," *Green Energy Intell. Transp.*, vol. 3, no. 4, Aug. 2024, Art. no. 100156.

[15] C. Chen, X. Chen, and P. Hang, "Personalized longitudinal motion planning based on a combination of reinforcement learning and imitation learning," *Green Energy Intell. Transp.*, vol. 5, no. 2, Apr. 2026, Art. no. 100321.

[16] D. Chen et al., "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 11623–11638, Nov. 2023.

[17] 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, Aug. 2023.

[18] X. He, B. Lou, H. Yang, and C. Lv, "Robust decision making for autonomous vehicles at highway on-ramps: A constrained adversarial reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4103–4113, Apr. 2023.

[19] G. Shi et al., "Neural lander: Stable drone landing control using learned dynamics," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9784–9790.

[20] Z. Zhang, S. Han, J. Wang, and F. Miao, "Spatial–temporal-aware safe multi-agent reinforcement learning of connected autonomous vehicles in challenging scenarios," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2023, pp. 5574–5580.

[21] B. Hu, L. Jiang, S. Zhang, and Q. Wang, "An explainable and robust motion planning and control approach for autonomous vehicle on-ramping merging task using deep reinforcement learning," *IEEE Trans. Transport. Electrific.*, vol. 10, no. 3, pp. 6488–6496, Sep. 2024.

[22] S. Hwang, K. Lee, H. Jeon, and D. Kum, "Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17594–17606, Oct. 2022.

[23] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3204–3217, Nov. 2017.

[24] G. Bagwe, X. Yuan, X. Chen, and L. Zhang, "RAMRL: Towards robust on-ramp merging via augmented multimodal reinforcement learning," in *Proc. IEEE Int. Conf. Mobility, Oper., Services Technol. (MOST)*, May 2023, pp. 23–33.

[25] Y. Ma, Y. Li, Z. Zheng, and H. Huang, "Developing merging policies for CAVs: A policy training framework combining human experience with reinforcement learning," *IEEE Trans. Intell. Vehicles*, vol. 10, no. 5, pp. 2977–2986, May 2024, doi: 10.1109/TIV.2024.3445334.

[26] X. Zhang, L. Wu, H. Liu, Y. Wang, H. Li, and B. Xu, "High-speed ramp merging behavior decision for autonomous vehicles based on multiagent reinforcement learning," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22664–22672, Dec. 2023.

[27] D. Sun, A. Jamshidnejad, and B. De Schutter, "A novel framework combining MPC and deep reinforcement learning with application to freeway traffic control," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6756–6769, Jul. 2024.

[28] Q. Liu, F. Dang, X. Wang, and X. Ren, "Autonomous highway merging in mixed traffic using reinforcement learning and motion predictive safety controller," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 1063–1069.

[29] M. P. Ronecker and Y. Zhu, "Deep Q-network based decision making for autonomous driving," in *Proc. 3rd Int. Conf. Robot. Autom. Sci. (ICRAS)*, Jun. 2019, pp. 154–160.

[30] C. Yu et al., "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 735–748, Feb. 2020.

[31] S. Wang, Z. Wang, R. Jiang, F. Zhu, R. Yan, and Y. Shang, "A multi-agent reinforcement learning-based longitudinal and lateral control of cavs to improve traffic efficiency in a mandatory lane change scenario," *Transp. Res. C. Emerg. Technol.*, vol. 158, Apr. 2024, Art. no. 104445.

[32] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2018, pp. 1582–1591.

[33] E. Wei and S. Luke, "Lenient learning in independent-learner stochastic cooperative games," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1–42, 2016.

[34] K. Lin, R. Zhao, Z. Xu, and J. Zhou, "Efficient large-scale fleet management via multi-agent deep reinforcement learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1774–1783.

[35] F. Fan, J. Xiong, and G. Wang, "Universal approximation with quadratic deep networks," *Neural Netw.*, vol. 124, pp. 383–392, Apr. 2020.

[36] F.-L. Fan, M. Li, F. Wang, R. Lai, and G. Wang, "On expressivity and trainability of quadratic networks," 2021, *arXiv:2110.06081*.

[37] M. Weissenbacher and Y. Kawahara, "A quadratic actor network for model-free reinforcement learning," 2021, *arXiv:2103.06617*.

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[38] Z. Xu, J. Xiong, F. Yu, and X. Chen, "Efficient neural network implementation with quadratic neuron," 2020, *arXiv:2011.10813*.
[39] J.-X. Liao, H.-C. Dong, Z.-Q. Sun, J. Sun, S. Zhang, and F.-L. Fan, "Attention-embedded quadratic network (Qttention) for effective and interpretable bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.
[40] S. Krauss, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 55, no. 5, pp. 5597–5601, May 1997.
[41] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[42] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.
[43] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[44] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proc. Int. Conf. Mach. Learn.*, Aug. 2017, pp. 1352–1361.
[45] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

Haigen Min (Senior Member, IEEE) received the Ph.D. degree in traffic engineering from Chang'an University, China, in 2018. He is currently a Professor with Chang'an University. His research interests mainly include environment perception, decision making, and motion control based on emerging sensor technologies of connected and autonomous vehicles (AV), and intelligent transportation systems (ITS). He has been selected for Shaanxi Province Young Science and Technology Rising Star and Young Top Talent. He received the best paper awards from COTA International Conference of Transportation Professionals (CICTP). In addition, he serves as an Associate Editor for *International Journal of Transportation Science and Technology*, *China Journal of Highway and Transport*, and *Journal of Traffic and Transportation Engineering*.

Yong Liu received the B.E. degree in the Internet of Things engineering from Chang'an University, China, in 2023, where he is currently pursuing the master's degree in electronic information. His current research interests include connected autonomous vehicles and deep reinforcement learning.

Xia Wu received the Ph.D. degree in traffic information engineering and control from Chang'an University, China, in 2020. She is currently an Associate Professor with Chang'an University. Her current research interests include mixed traffic flow control, trajectory optimization with connected and automated vehicles, and human-driven vehicles.

Sen Xia received the B.E. degree in communication engineering from Xi'an Shiyou University, China, in 2021, where he is currently pursuing the Ph.D. degree in traffic information engineering and control. His current research interest is cooperative control for connected and automated vehicles.

Xiangmo Zhao (Member, IEEE) received the B.S. degree from Chongqing University, China, in 1987, and the M.S. and Ph.D. degrees from Chang'an University, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Engineering, Chang'an University, and the President of Xi'an University of Architecture and Technology. He is also the Vice President of the Joint Laboratory for Connected Vehicles, Ministry of Education, and China Mobile, and Shaanxi Road Traffic Intelligent Detection and Equipment Engineering Technology Research Centre. He is also a Leader of the National Key Subjects-Traffic Information Engineering and Control, Chang'an University. His current research interests include connected vehicles, intelligent transportation systems, and computer science. He is the Director of the Information Professional Committee, a member of the Advisory Expert Group of China Transportation Association, the Vice Chairperson of the Institute of Highway Association on Computer Professional Committee, and the Deputy Director of the Institute of Computer in Shaanxi Province.

Guoqiang Mao (Fellow, IEEE) has published over 200 papers in international conferences and journals, which have been cited more than 8500 times. His research interests include intelligent transport systems, applied graph theory and its applications in telecommunications, the Internet of Things, wireless sensor networks, wireless localization techniques, and network modeling and performance analysis. He is a fellow of IET. He received the "Top Editor" Award for outstanding contributions to IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY in 2011, 2014, and 2015. He was the Co-Chair of IEEE Intelligent Transport Systems Society Technical Committee on Communication Networks. He has served as the chair, the co-chair, and a TPC member of a number of international conferences. He has been an Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (since 2018), IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2014–2019), and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY (since 2010).

Authorized licensed use limited to: Southeast University. Downloaded on October 30,2026 at 12:09:43 UTC from IEEE Xplore. Restrictions apply.