Expert Systems With Applications 267 (2025) 126153

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Centralized cooperative control for autonomous vehicles at unsignalized all-directional intersections: A multi-agent projection-based constrained policy optimization approach ☆

Rui Zhao [a], Kui Wang [b], Yun Li [c], Yuze Fan [a], Fei Gao [d,*], Zhenhai Gao [d]

[a] College of Automotive Engineering, JiLin University, Changchun, 130025, China
[b] School of Mechanical Engineering, Beijing Institute of Technology, Beijing, 10008, China
[c] Department of Information and Communications Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan
[d] State Key Laboratory of Automotive Simulation and Control, Jilin University, Changhun, 130025, China

ARTICLE INFO

Keywords:
Connected and automated vehicles
Safe reinforcement learning
Intersection cooperative control
Multi-agent reinforcement learning
Constrained Markov Games

ABSTRACT

The interest in real-time cooperative control at urban unsignalized intersections has surged recently, with a focus on enhancing the driving safety and traffic throughput for connected and automated vehicles (CAVs). Nonetheless, most existing studies either struggle with computational complexity in high-density traffic scenarios or fail to ensure robust safety for distributions outside of training. To tackle these issues, this study introduces a novel multi-agent Safe Reinforcement Learning framework, SRL-CLCADI, designed for Cross-Longitudinal cooperative Control of CAVs at All-Directional Intersections. Approaching vehicles communicate their kinematic data to the central controller in real-time, hand over their control authorities, and comply with the instructions from the central controller. Specifically, to address the significant computational challenges of existing optimization control methods and the robust safety issues of deep learning methods in practical applications, we reformulate the intricate multi-objective decision-making process into a model-free Constrained Markov Game and introduce a Multi-Agent Projection-based Constrained Policy Optimization (MAPCPO) approach within the framework of safe Reinforcement Learning (RL). The policy neural network features an innovative queue-based dynamic state input design to handle dynamic traffic scenarios at intersections, and the MAPCPO method incrementally optimizes updates within the Kullback–Leibler divergence trust region to enhance performance before projecting them onto the safety constraint boundaries to ensure safety. In simulations across various traffic densities at unsignalized intersections, our method significantly outperformed Model Predictive Control and Mixed Integer Programming methods, improving computational efficiency by 72.79% and 63.16%, traffic efficiency by 36.84% and 32.38%, and energy consumption by 8.22% and 5.48%, respectively. Unlike non-safety-aware RL methods, our approach achieved a zero collision rate, also enhancing ride comfort.

1. Introduction

Autonomous driving is growing rapidly and shows excellent application potential in road traffic provides to improve safety, efficiency, and comfort (Hang, Huang, Hu, & Lv, 2022; Hu et al., 2022; Kuwata et al., 2009; Li, Sun, Cao, He, & Zhu, 2015). However, due to the intricacy of the environment and the unpredictability of anticipated vehicular actions, urban settings present significant challenges for autonomous driving. Among these, unsignalized intersections, which converge multiple directional lanes and lack traffic signal guidance coupled with visual obstructions, are notably difficult. Studies indicate that even human drivers find it challenging to navigate these intersections safely and efficiently, leading to frequent accidents (Li, Gong, Lu, & Yi, 2021; Zhao, Knoop, & Wang, 2023). Therefore, researching safe and efficient passage for autonomous vehicles at unsignalized intersections is of paramount importance for the broader acceptance and adoption of autonomous driving technology. In recent years, Vehicle-to-Everything (V2X) communication has significantly addressed the perception limitations at these intersections. Vehicle-Infrastructure Collaborative driving

☆ This work was supported by the National Natural Science Foundation of China under Grant 52202495 and Grant 52202494.
* Corresponding author.
E-mail addresses: rzhao@jlu.edu.cn (R. Zhao), 3120230321@bit.edu.cn (K. Wang), li-yun@g.ecc.u-tokyo.ac.jp (Y. Li), fanyz23@mails.jlu.edu.cn (Y. Fan), gaofei123284123@jlu.edu.cn (F. Gao), gaozh@jlu.edu.cn (Z. Gao).

https://doi.org/10.1016/j.eswa.2024.126153
Received 11 February 2024; Received in revised form 15 November 2024; Accepted 10 December 2024
Available online 16 December 2024

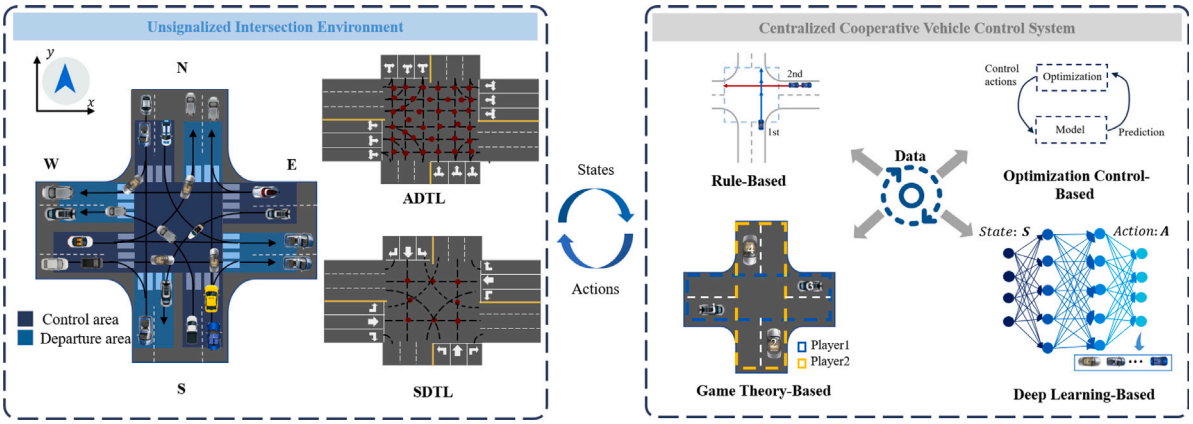


Fig. 1. Signal-Free Intersection Environment and Vehicle-to-Infrastructure Cooperative Control Method.

methods for unsignalized intersections, based on V2X communication, are emerging and represent a significant advancement towards safer and more efficient autonomous intersection management (Abboud, Omar, & Zhuang, 2016; Hafner, Cunningham, Caminiti, & Del Vecchio, 2013; Wu, Chen, & Zhu, 2019). The vehicle-road collaboration process is depicted in Fig. 1, where each connected and automated vehicle (CAV) periodically engages in wireless communication with roadside units (RSUs) to transmit its dynamic form data. The roadside then adjusts the flow of vehicles through the intersection based on the embedded collaborative control policy, affecting various state variables such as speed, acceleration, and trajectory. The motion control layer of each CAV controls its movement at the intersection in accordance with the decision-making instructions from the roadside.

Initially, rule-based collaborative control policies were developed. These methods utilized traffic rules engineered to schedule vehicles efficiently and orderly. Commonly adopted methods included First Come, First Serve (FCFS) (Dresner & Stone, 2006; Lukose, Levin, & Boyles, 2019; Wu et al., 2019), which prioritized vehicles based on their arrival sequence at intersections; Fastest First Service (FFS) (Fajardo, Au, Waller, Stone, & Yang, 2011), which prioritized vehicles according to shorter traversal times; and Long Queue First Service (LQF) (Qian, Altché, Grégoire, & de La Fortelle, 2017), which aimed to improve intersection throughput by prioritizing the longest queues. While effective in simple scenarios, these methods' limited adaptability and coarse temporal granularity can lead to significant performance issues in real and complex intersections. Thus, managing intersections with high traffic density and variability effectively remains a challenging task.

As control optimization control methods have demonstrated outstanding performance, research has been focused on modeling, identifying conflict constraints, and coordinating vehicle passages at unsignalized intersections using optimization approaches. The formulation of conflict constraints involves grid-based, point-based, and vehicle-based approaches; then methods like Dynamic Programming (Guo et al., 2019), Mixed Integer Programming (MIP), Mixed Integer Linear Programming (MILP) (Lu & Kim, 2018), and Model Predictive Control (MPC) (Kamal, Imura, Hayakawa, Ohata, & Aihara, 2014) have been used to compute real-time vehicle control commands that meet these constraints. The conflict-tile method (Bichiou & Rakha, 2018; Dai et al., 2016; Xu, Zhang, Li, & Li, 2019) prevents collisions by limiting vehicle numbers in designated grid areas, although it may underutilize space. The conflict-point method (Kamal et al., 2014; Mirheli, Hajibabai, & Hajbabaie, 2018) controls vehicles near potential collision points, which requires a delicate balance between efficiency and safety. Vehicle-based methods (He, Zheng, Lu, & Guan, 2018; Li & Zhang, 2018; Li, Zhang, Zhang, Jia, & Ge, 2018; Mirheli, Tajalli, Hajibabai, & Hajbabaie, 2019), while coordinating vehicles across the whole intersection, may yield less than optimal outcomes due to computational intensity.

Game-theory-based control methods are also employed for cooperative control at intersections. Here, vehicles act as strategic players in a game seeking to reach Nash equilibria for decision-making. These methods simulate interactions between vehicles at unsignalized intersections, aiming to improve intersection adaptability and efficiency (Li, Yao, Kolmanovsky, Atkins, & Girard, 2020; Tian, Li, Kolmanovsky, Yildiz, & Girard, 2020). However, with increasing traffic, the computational load on coordinating servers surges, posing challenges for their application at busy intersections.

The rapidly evolving Deep Reinforcement Learning (RL) methods (Isele, Rahimi, Cosgun, Subramanian, & Fujimura, 2018) represent significant learning-based approaches and have the potential to tackle computational demand issues while ensuring excellent intersection control performance. Deep RL studies vehicle-environment interactions to learn optimal cooperative control policies at intersections by maximizing rewards. Unlike rule-based methods, which presume a well-understood environment, Deep RL-based techniques train models to map environmental states to actions, adapting to changing conditions. Typical Deep RL algorithms like Proximal Policy Optimization (PPO) (Guan et al., 2020), Deep Deterministic Policy Gradient (DDPG) (Antonio & Maria-Dolores, 2022) and Soft Actor–Critic (SAC) (Al-Sharman et al., 2023) are employed for unsignalized intersection automation training. The reward function balances the scores of performance in safety, comfort, and efficiency. Introducing Deep RL to this domain has yielded promising training outcomes, addressing computational efficiency in high-traffic scenarios and issues with coarse granularity and suboptimality, yet several challenges remain.

First and foremost, single-reward policy updating methods are inadequate for safety-critical applications in autonomous driving, where Preventing accidents is the premise and foundation for optimizing ride comfort and traffic efficiency. Simply combining safety, comfort, and efficiency into one reward function may not correctly prioritize the safety elements for the learning agent. To tackle this, various safe Deep RL methods, such as the Constrained Policy Optimization algorithm (Achiam, Held, Tamar, & Abbeel, 2017) and the Projection-based Constrained Policy Optimization algorithm (Yang, Rosca, Narasimhan, & Ramadge, 2020), have been proposed. However, these algorithms, including their distributed versions, are not yet suitable for real-world deployment at intersections. Distributed methods lack global awareness, potentially leading to safety risks and inconsistent decision-making. Due to the complex nature of coordinating multiple vehicles and the necessity for prompt synchronization, centralized safe reinforcement learning becomes crucial for ensuring traffic safety and efficiency. Furthermore, adaptive policy update methods are advised to manage the complexity of diverse hazardous situations at intersections, striking a balance between learning efficacy and safety. However, there are

currently no centralized safe Deep RL extension algorithms available that adequately fulfill these requirements.

Secondly, the use of static input neural networks in developing multi-vehicle cooperative control policies poses significant challenges to effective continuous learning. Limiting the input quantity results in shortened episodes, complicating the application in real-world intersections where traffic flow is continuous. Increasing the input quantity, on the other hand, leads to an excessive number of network parameters, making learning more difficult. Additionally, the inclusion of blank feature inputs from vehicles that have already passed adversely impacts the model's ability to understand and learn.

Thirdly, regarding training scenarios, current intersection cooperative control methods based on Deep RL often assume the presence of specific-direction turn lanes (SDTL), a presumption that does not match many real-world scenarios, thereby limiting the ability to effectively manage intersection traffic. Moreover, employing discontinuous episodic settings, where new episodes begin at initialized positions, is not conducive to policy applications in intersections with continuous traffic flow.

Fourthly, existing Deep RL or other methodologies have not yet integrated lateral vehicle control, focusing solely on the longitudinal control of vehicles. This limitation restricts the potential for optimizing traffic flow, as comprehensive control over both longitudinal and lateral movements is crucial for enhancing overall traffic efficiency and safety.

This paper proposes a novel multi-agent Safe Reinforcement Learning framework designed for Cross-Longitudinal coordination Control of CAVs within All-Directional Intersections (SRL-CLCADI), thereby addressing the previously identified deficiencies.

The primary contributions of this paper can be summarized as:

(1) Firstly, we formulate the cross-longitudinal cooperative control process of CAVs at intersections with all-directional turn lanes (ADTL) into a model-free Constrained Markov Game (CMG), an extension of standard Markov Game (MG), with embedded safety constraints restricting the set of acceptable policies. The improved Impact Regularization (IR) for cooperative control reward and cost functions is employed, with the goal of preventing irreversible or harmful changes to the environment caused by CAVs when their actions deviate from a baseline state. Additionally, the definition of the state space takes into account communication delays, closely mirroring real-world scenarios.

(2) Secondly, we introduce a centralized Multi-Agent Projection-based Constrained Policy Optimization algorithm (MAPCPO) to solve the CMG problem. This method incorporates safety constraints to further constrain the trust region formed by Kullback–Leibler (KL) divergence, thereby facilitating policy updates that maximize performance while keeping constraint costs within their specified limits across different quantified safe levels.

(3) Thirdly, we design the policy neural network architecture, aiming to enhance control performance and flexibility. The policy input utilizes an innovative queue-based dynamic update mechanism to handle dynamic traffic scenarios at intersections. The network neurons employ Long Short-Term Memory (LSTM) units, enhancing its ability to capture vehicular social interactions and historical state information. The policy output encompasses coupled lateral and longitudinal control of CAVs, providing a broader space for optimization control.

(4) Extensive simulation across various traffic densities at unsignalized intersections is conducted and comparisons with classical control methods Vehicle-Intersection Coordination Scheme (VICS) (Kamal et al., 2014), Mixed integer programming based Intersection Coordination Algorithm (MICA) (Lu & Kim, 2018) and non-safety aware RL method Model Accelerated Proximal Policy Optimization (MAPPO) (Guan et al., 2020), showcasing the superiority of our method.

The structure of this paper is organized as follows: Section 2 provides an overview of the algorithmic framework of the SRL-CLCADI algorithm. In Section 3, we present the design aspects of the SRL-CLCADI algorithm, including the state space, action space, and the design of reward and risk functions. Section 4 offers a detailed exposition on the update process of the SRL-CLCADI algorithm based on the MAPCPO algorithm. Section 5 evaluates the performance of various control algorithms within training scenarios. Lastly, Section 6 concludes the paper and discusses future research directions and prospects.

2. Problem definition and method framework

2.1. Problem definition

The scenario of an unsignalized road intersection, as shown in Fig. 1, represents a standard four-way intersection with several lanes. The intersection is systematically divided into two distinct areas: the control area and the departure area. The intersection control system exerts lateral and longitudinal cooperative control over CAVs within a specified distance from the entrance of the intersection (i.e., the control area). CAVs approaching an intersection are capable of maneuvers in any direction: proceeding straight, turning right, or turning left, and are permitted to change lanes at any time, provided they adhere to traffic regulations. All potential vehicle conflict relationships can be categorized into three types: crossing conflicts, merging conflicts, and diverging conflicts. Each CAV i transmits dynamic data in real-time

$$s^i = \begin{bmatrix} x^i \\ y^i \\ v^i \\ \iota^i \\ \kappa^i \end{bmatrix}$$

to vehicle-road cooperative control system via V2I wireless communication. Here, $\{x_t^i, y_t^i\} \in \mathbb{R} \times \mathbb{R}$ represents the CAV's position coordinates at time step t; v_t^i denotes the velocity, $\iota_t^i \in \mathcal{I} := \{\text{left}, \text{straight}, \text{right}\}$ indicates the anticipated direction of travel at the intersection; $\kappa^i \in \mathbb{R}_+$ represents the sum of the communication delay time from CAV to RSU and from RSU to CAV at last time step. Influenced by factors such as vehicle speed, distance to RSU and flow density, communication delay times dynamically change in real-time.

Equipped with the safety-aware multi-agent deep reinforcement learning algorithm SRL-CLCADI, the centralized cooperative control system modulates the timing of CAVs traversing road intersections by realtime control of each CAV's speed $v^i \in [v_{\min}, v_{\max}]$ and steering angle variation $\Delta\psi^i \in [\psi_{\min}, \psi_{\max}]$, where $v_{\min}$ and $v_{\max}$ represent the minimum and maximum velocities, $\psi_{\min}$ and $\psi_{\max}$ represent the minimum and maximum steering angle, respectively. Subsequently, the motion control layer of each CAV is capable of generating the required throttle opening and brake pad force for longitudinal control of the CAV. The unaltered steering angle ψ^i is determined by advanced trajectory planning and trackers based on the CAV's positional coordinates. The true steering angle $\psi'^i = \psi^i + \Delta\psi^i$ is then derived by adding the steering angle variation to this, allowing the CAV to maintain its original trajectory as much as possible while also possessing the capability to avoid collisions through lateral movement.

2.2. Method framework

The proposed SRL-CLCADI method sets up three neural networks: the centralized policy neural network, along with the reward and safety cost value neural networks. The policy network is employed to map the local states of all vehicles at the current time step to the joint action probability distribution of the vehicles at the next time step, while the reward and safety cost-based value networks are used to evaluate the expected performance reward and safety cost under the current policy. Fig. 2 illustrates the SRL-CLCADI method framework, comprising the Sampler Module and Learner Module.

The task of the SRL-CLCADI sampler is to acquire updated neural parameters for the policy and value networks, and then to use these parameters to sample experience data from the road intersection

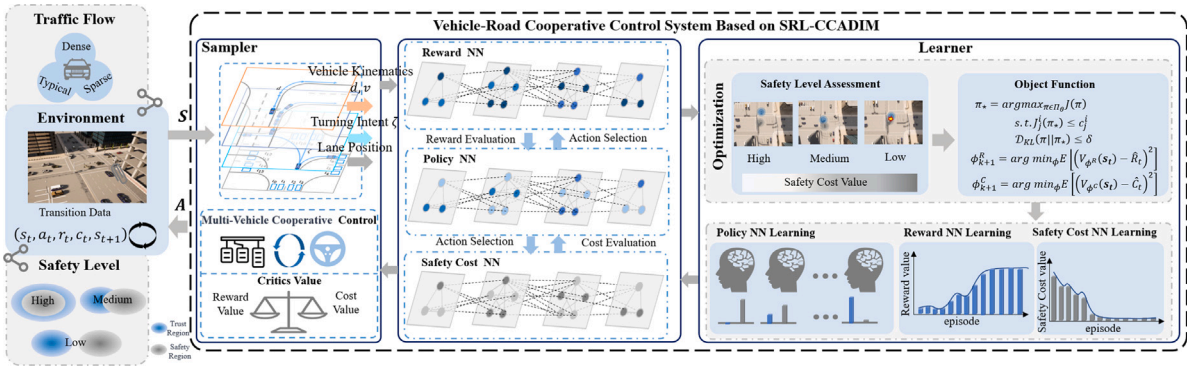


Fig. 2. Structure of the proposed vehicle-road cooperative control system based on SRL-CLCADI, comprised of two parts — Sampler and Learner, designed for the monotonic improvement of intersection throughput performance including safety, efficiency, and comfort, while concurrently satisfying safety cost constraints. The SRL-CLCADI structural framework diagram encompasses three neural networks: the policy neural network, the reward value neural network, and the safety costs value neural network. The Sampler module is responsible for interacting with the environment, collecting state space data from it, and obtaining action space data from the policy neural network. The Learner module is tasked with updating the parameters of three neural networks based on the collected data and two evaluation functions in the update method.

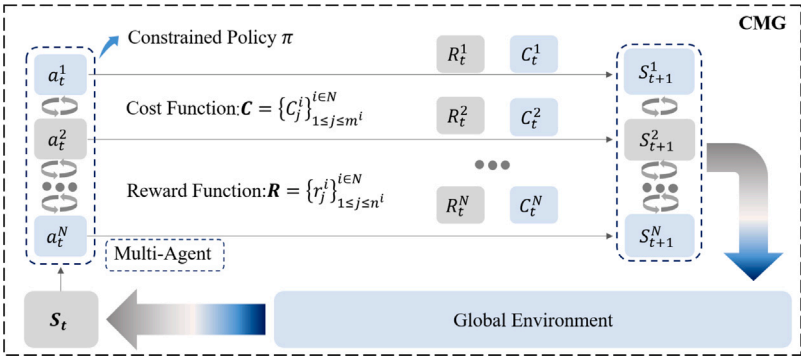


Fig. 3. Structural Framework of the CMG.

environment. This module employs a safe multi-agent RL formulation, CMG, to formally articulate the exploration process of multiple CAVs with safety constraints within the intersection cooperative control system environment. This process subsequently produces discrete time-series trajectory data, which includes states, actions, rewards, and safety costs. This data provides the basis for optimizing the policy neural network and the reward and cost value neural networks.

The SRL-CLCADI learner works in conjunction with the sampler, utilizing the data collected by the sampler to update the policy and value neural networks. It then synchronizes the updated parameters with the sampler to facilitate the next iteration of sampling and optimization in a cyclical process until the desired traffic performance at the intersection is reached. This module employs a MAPCPO method that solves the CMG problem by incorporating safety costs-constrained into the trust region formed by KL divergence. It first maximizes the policy reward value, then projects the policy into the constrained range of safety costs to realize a safe and efficient policy, aiming to optimize intersection cooperative control performance while ensuring that the constrained safety costs remain within predefined boundaries.

3. Sampler module following the CMG formulation

This section introduces a safe Multi-Agent Deep Reinforcement Learning (MADRL) formulation, namely CMG, and transforms the unsignalized Intersection Control issue into a CMG problem by defining the fundamental elements of CMG, including the global state space, joint action space, joint reward function, and a series of safety cost functions for CAVs to safely share the road with each other. This process yields discrete time-series trajectory data, including states, actions, rewards, and safety costs, used for optimizing the policy, reward value, and safety cost neural networks.

3.1. Safe multi-agent reinforcement learning formulation

By incorporating additional constraints to the traditional MG, a CMG model is constructed, which is then used for the safe multi-agent RL problem, as depicted in Fig. 3. The CMG model is defined as a tuple $(\mathcal{N}, S, \mathcal{A}, P, R, C, \rho^0, \gamma)$, where:

- $\mathcal{N} = \{1, \ldots, n\}$ represents a set of n agents;
- $S = \prod_{i=1}^{n} S^i$ denotes the global state space, which is the cartesian product of the state spaces of each agent i;
- $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{n} A^i$ represents the global action space, which is the cartesian product of the action spaces of each agent, where the agents perform the joint action $A_t = (a_t^1, a_t^2, \ldots, a_t^n)$ at the discrete time step t;
- $P : S \times \boldsymbol{\mathcal{A}} \times S \to [0, 1]$ denotes the state transition probability function from a state $s_t \in S$ and joint action $a_t \in \boldsymbol{\mathcal{A}}$ to the next state $s_{t+1} \in S$;
- $R : S \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ represents the joint reward function;
- $\boldsymbol{C} = \{C_j^i\}_{j\le m^i}^{i\in\mathcal{N}}$ denotes the set of safety cost functions for the agents $c_j^i : S \times \boldsymbol{\mathcal{A}}^i \to \mathbb{R}$ (each agent has N_j^i cost functions), with the corresponding safety cost constraint values represented by $\boldsymbol{d} = \{d_j^i\}_{1\le j\le m^i}^{i\in\mathcal{N}}$;
- $\rho^0 : S \to [0, 1]$ is the starting state distribution;
- $\gamma : S \to [0, 1]$ is the discount factor.

According to the CMG model, agents interact with the environment in discrete time steps. At each time step t, agents receive all the information of the current state $s_t \in \boldsymbol{S}$ of the environment, with each agent performing an action $\boldsymbol{a}_t = (a_t^1, \ldots, a_t^n)$. After all agents have taken their joint actions, they receive a joint reward $R(s_t, \boldsymbol{a}_t)$, and each incurs their respective costs $C_j^i(s_t, a_t^i), \forall j = 1, \ldots, m^i$. The environment then

Table 1
Global state space and joint action space.

Notation	Description	Data format	Range
d^i	The distance of vehicle i from the target location	$[float]$	[0, 15] m
v^i	Current velocity of vehicle i	$[float]$	[0, 10] m/s
ζ^i	Current lane of vehicle i	$[bool_1, bool_2, \ldots, bool_l, \ldots]$	0 or 1
l^i	Current driving direction of vehicle i	$[bool_0, bool_1, bool_2]$	0 or 1
κ^i	The sum of the bidirectional communication delay time	$[float]$	(0, $+\infty$) s
v'^i	Expected velocity of vehicle i	$[float]$	[2, 10] m/s
ψ'^i	Expected steering angle of vehicle i	$[float]$	[−0.65, 0.65] rad

transitions to a new state s_{t+1} with probability $P(s_{t+1}|s_t, a_t)$. RL utilizes sample trajectory data obtained from multi-agent interactions with the environment. The purpose of RL is to teach agents to learn a policy π that maximizes the expected return of rewards.

$$J(\pi) = \mathbb{E}_{s_0\sim\rho^0, \boldsymbol{a}_\sim\pi, s_{1:\infty}\sim\boldsymbol{P}}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, \boldsymbol{a}_t)\right] \tag{1}$$

Simultaneously, it also needs to satisfy the expected discounted return $J_{c_j^i}(\pi)$ for each safety cost function C_j^i under the threshold c_j^i:

$$J_{C_j^i}(\pi) = \mathbb{E}_{s_0\sim\rho^0, \boldsymbol{a}\sim\pi, s_{1:\infty}\sim\boldsymbol{P}}\left[\sum_{t=0}^{\infty}\gamma^t C_j^i(s_t, \boldsymbol{a}_t)\right] \le d_j^i \quad \forall i = 1, \ldots, m; j = 1, \ldots, n \tag{2}$$

3.2. Representing the collaborative control problem of CAVs as a safe MARL formulation in CMG

3.2.1. State space and joint action space

The global state space is used to describe the observation information during the interaction between multi-agent and the environment. A reasonable state space is crucial for efficient RL. For the MADRL-based vehicle-road cooperative control system, the ability to effectively extract and reconstruct observation information from the vehicle driving environment in highly complex and dynamic road intersections determines its capability to accurately output appropriate actions (see Table 1).

To align with the realistic and complex scenario of randomized free movement of vehicles at road intersections, the state space comprehensively considers both the fluid characteristics of the intersections and the randomness of vehicle driving paths. In this work, the state space is defined as $\boldsymbol{S} = [\prod_{i=1}^{n}(d^i, v^i, \zeta^i, \iota^i, \kappa^i)]$. Here, d^i represents the distance that the CAV i needs to travel to leave the intersection, v^i denotes its real-time speed, ζ^i represents the road information for CAV i where each lane of every road occupies a distinct indicator position, employing one-hot encoding, ι^i indicates the driving direction of agent i, including left-turn, straight, and right-turn, also using one-hot encoding where each direction occupies a distinct indicator position and κ^i indicates the bidirectional communication delay. The initial value of $\kappa^i > 0$ for each vehicle is set to $\kappa^{i,0}$. The velocity, turning information and communication delay information from the previous time step of the CAV are transmitted in real-time to the road segment by the CAV, while the distance information and lane data are calculated in real-time based on the coordinates sent by the CAV. By combining the information (d^i, ζ^i, ι^i), the specific position of a CAV at an intersection can be encoded. The aforementioned state space representation is capable of adapting to complex and random intersection traffic scenarios, effectively and dynamically characterizing the variable number of passing CAVs, diverse traffic intentions, and dynamic motion information at the intersection.

The action space mapped by the policy neural network is the action command explored by CAVs in a complex intersection environment. This algorithm achieves coordinated passage of CAVs through real-time control of the expected speed v'^i and steering angle variation $\Delta\psi^i$ of CAVs in next time steps. Similar to the global state space, the joint action space is established as the ordered concatenation of each CAV's action $\boldsymbol{A} = \prod_{i=1}^{n}\boldsymbol{A}^i$, with the action space is defined as $\boldsymbol{A}^i = [v'^i, \Delta\psi^i]$. The motion control layer of each CAV generates acceleration throttle opening and deceleration braking force to control the longitudinal movement of the CAV based on the desired speed provided by the vehicle-road cooperative control system. Additionally, it combines the steering angle differential information provided by the system with the current steering angle of the CAV for lateral control.

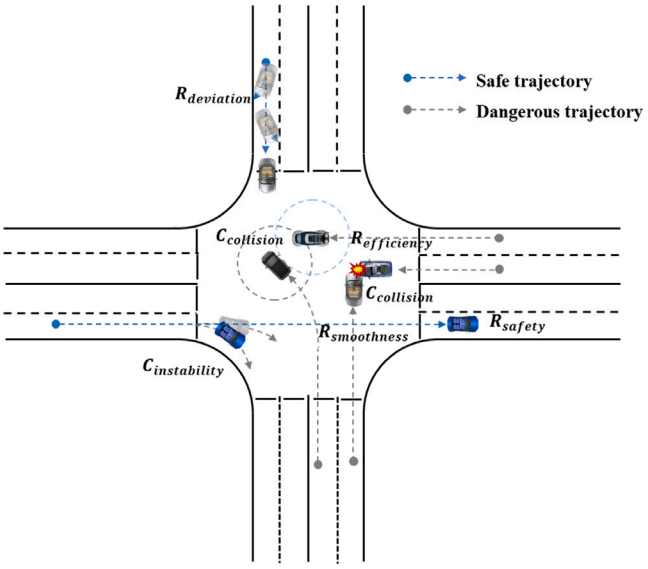


Fig. 4. The intuitive representation of the subitems characterizing intersection throughput performance in the reward and cost functions.

3.2.2. Reward function and safety cost function

Reward and safety cost functions serve as the quantitative feedback received by multi-agent systems from the environment following the execution of joint actions. These functions guide agents in persistently learning policies focused on maximizing rewards within cost constraints, thereby progressively enhancing the asymptotic performance and convergence speed of reinforcement learning algorithms. We have comprehensively and finely designed the reward and safety cost functions for vehicle-road cooperative control system based on, thereby achieving safe, efficient, and comfortable passage for multiple CAVs at intersections.

In traditional RL methods, reward functions are instrumental in guiding agents towards achieving predefined objectives. However, these methods frequently fail to account for the intricate nuances present in complex real-world situations. This oversight can inadvertently prompt agents to exhibit behaviors that were not intended, favor less-than-ideal solutions, or even compromise the integrity of the system in an effort to maximize the specified rewards. To address these challenges and instill finer-grained behaviors or constraints, we introduce the principle of reward regularization. By augmenting the original reward signal with penalty values from a safety cost function,

our objective is to not only guide the agent towards primary safety goals but also to comply with secondary criteria that ensure efficiency and comfort. This method is really effective for dealing with different situations and problems in smart transportation systems and other systems that have a bunch of performance needs.

In the design of integrated reward and cost functions, indifference points are often difficult to determine. The selection of coefficients for various performance sub-items is usually dependent on the designer's personal preferences, making it challenging to balance relationships among multiple performance metrics. Additionally, reward functions that are overly optimized for a specific scenario may lead to overfitting problems, rendering the trained model less adaptable to new scenarios. As a general rule, it is better to design performance metrics according to what one actually wants to be achieved in the environment, rather than according to how one thinks the agent should behave. Therefore, this study separately considers the cost sub-items arising from violations of safety distances $\boldsymbol{C}_{close}$, safety skidding events $\boldsymbol{C}_{skid}$, and safety collision events $\boldsymbol{C}_{collision}$. They reflect the true utility function $J_{C_{i,j}}(\pi)$ without intermediate shaping factors, as shown in Fig. 4. The first two are dense cost evaluation items triggered at each time step, while the last is a sparse cost evaluation item triggered by each event or specific occurrence, representing a delayed value that encapsulates the global collision of the entire event.

In the traffic environment, when the Time to Collision (TTC) between any two vehicles with potential collision risk falls below the preset safety threshold $\boldsymbol{t}_s$ at a given time step, the value of the cost sub-item $\boldsymbol{C}_{risk}$ can be calculated as

$$\boldsymbol{C}_{close} = \sum_{t=1}^{n}\sum_{i=1}^{n}\sum_{i'\neq i}^{n-1}\varepsilon_{close_{t,i,i'}} \tag{3}$$

where the Boolean variable $\varepsilon_{close_{t,i,i'}}$ indicates the presence of a collision risk between CAV i and CAV i' at time step t. The calculation of distance and TTC takes into account three different scenarios. Initially, for vehicles not yet within the intersection area and traveling on the same lane, the focus is on longitudinal safety. distance $d_{i,i'} = |x_i - x_{i'}|$. The longitudinal TTC is $TTC_{lon} = \frac{d_{i,i'}}{v_{x,i}-v_{x,i'}}$, where $v_{x,i}$ and $v_{x,i'}$ represent the longitudinal speeds of vehicles i and i', respectively. If the vehicles are on different lanes before entering the intersection, emphasis shifts to the lateral safety distance $d_{i,i'} = |y_i - y_{i'}|$, with the lateral TTC defined as $TTC_{lat} = \frac{d_{i,i'}}{v_{y,i}-v_{y,i'}}$, where $v_{y,i}$ and $v_{y,i'}$ denote the lateral speeds of vehicles i and i', respectively. Upon a vehicle's entry into the intersection area, the distance calculation must consider both lateral and longitudinal components, where $d_{i,i'} = \sqrt{(x_i - x_{i'})^2 + (y_i - y_{i'})^2}$. The corresponding TTC is then established as $TTC_{cor} = \frac{d_{i,i'}}{v_i - v_{i'}}$, where v_i and $v_{i'}$ denote the speeds of vehicles i and i' along their respective directions of travel. If there is any situation where the TTC is below the threshold, the risk indicator $\varepsilon_{close,t,i,i'}$ is set to 1. Otherwise, $\varepsilon_{close,t,i,i'}$ is set to 0.

After a skid occurs, the value of the cost sub-item $\boldsymbol{C}_{skid}$ can be calculated:

$$\boldsymbol{C}_{skid} = \sum_{t=1}^{n}\sum_{i=1}^{n}\sum_{i'}^{n-1}\varepsilon_{skid_{t,i,i'}} \tag{4}$$

Here, $\varepsilon_{skid_{t,i,i'}}$ is a Boolean variable representing whether a CAV undergoes sideslip. When $\frac{v^2}{2r} > F_f$ indicates that the CAV is experiencing sideslip, $\varepsilon_{skid_{t,i,i'}}$ is set to 1, otherwise it is 0. The turning radius r of the CAV, calculated from the vehicle's current steering angle, represents the vehicle's maneuverability, while F_f signifies the maximum friction force that the CAV's tires can provide.

Then, the value of the cost sub-item $\boldsymbol{C}_{collision}$ can be calculated:

$$\boldsymbol{C}_{collision} = \sum_{i=1}^{n}\sum_{i}^{n-1}\varepsilon_{collision_{t,i,i'}} \tag{5}$$

where the Boolean variable $\varepsilon_{collision_{t,i,i'}}$ indicates the presence of a collision between CAV i and CAV i' at time step t. If a collision occurs, $\varepsilon_{collision_{t,i,i'}}$ is set to 1; otherwise, it is set to 0.

The total cost function $\boldsymbol{C}_{total}$ is defined as the sum of the dense cost sub-items $\boldsymbol{C}_{close}$, $\boldsymbol{C}_{skid}$, and the sparse cost sub-item $\boldsymbol{C}_{collision}$:

$$\boldsymbol{C}_{total} = \delta_r \boldsymbol{C}_{close} + \delta_c \boldsymbol{C}_{skid} + \delta_s \boldsymbol{C}_{collision} \tag{6}$$

where δ_r, δ_c and δ_s refer to the hyper-parameters to balance the safety factor. In SRL-CLCADI system, we use MAPCPO to measure the global influence on the environment caused by multiple CAVs. Such a factor is optimized via a separate safety cost value neural network.

This paper aims to avoid any form of shaping rewards, which include guidance on behaviors that deviate from merely measuring expected outcomes, in order to prevent agents from learning potentially risky behaviors (Hu et al., 2020). To comprehensively depict the environment, the reward function emphasizes on holistically improving traffic efficiency, ride comfort, adherence to the original trajectory, and safety at intersections.

For the traffic efficiency sub-component $\boldsymbol{R}_{efficient}$, we linearly increase the dense reward $\delta_v v_i(t)$ corresponding to each CAV's speed and subtract the time-intensive dense reward δ_t at each time step. Additionally, when a CAV successfully navigates through an intersection, we augment the sparse reward by a value of δ_p, and when all CAVs pass through the intersection smoothly, we increase the sparse reward by a value of δ'_p. The reward sub-component $\boldsymbol{R}_{efficient}$ can be expressed as:

$$\boldsymbol{R}_{efficient} = \sum_{t=1}^{n}\sum_{i=1}^{n}\delta_v v_i(t) - \sum_{t=1}^{n}\delta_t + \sum_{i=1}^{n}\varepsilon_{pass\_single_i}\delta_p + \varepsilon_{pass\_all}\delta'_p \tag{7}$$

where the Boolean variable $\varepsilon_{pass\_single_i} = 1$ indicates that CAV i passes successfully, and $\varepsilon_{pass\_all} = 1$ indicates that all CAVs pass successfully. where the Boolean variable $\varepsilon_{pass\_single_i} = 1$ indicates that CAV i passes successfully, and $\varepsilon_{pass\_all} = 1$ indicates that all CAVs pass successfully.

Moreover, the speed difference (i.e., acceleration) between adjacent time steps of the CAV is used to quantify the occupant comfort reward sub-item $\boldsymbol{R}_{comfort}$:

$$\boldsymbol{R}_{comfort} = \sum_{t=1}^{n}\sum_{i=1}^{n}\delta_a \frac{|v_i(t) - v_i(t-1)|}{\tau} \tag{8}$$

where δ_a represents the score weight of this item and τ represents the time interval between two time steps.

The reward function incorporates a trajectory tracking reward. The policy neural network coordinates both the lateral and longitudinal controls of each vehicle to allow them to pass through intersections comfortably and efficiently without collisions. Given that the policy neural network has the authority to control the steering angle of the vehicle, it is imperative to add a reward $\boldsymbol{R}_{deviation}$, to ensure that the vehicle adheres as closely as possible to the trajectory dictated by the higher-level autonomous driving decision. Assuming that at time step t, the CAV's steering angle following the higher-level autonomous driving decision is ϑ and the steering angle controlled by the policy neural network is ϑ', the trajectory tracking reward is defined as:

$$\boldsymbol{R}_{deviation} = -\sum_{t=1}^{T}\sum_{i=1}^{n}\delta_d\,|\vartheta_i - \vartheta'_i| \tag{9}$$

The safety reward sub-item $\boldsymbol{R}_{safety}$ is quantified directly by the negative value of the cost function $\boldsymbol{C}_{total}$. In summary, the total reward function $\boldsymbol{R}_{total}$ can be articulated as:

$$\boldsymbol{R}_{total} = \boldsymbol{R}_{efficient} + \boldsymbol{R}_{comfort} + \boldsymbol{R}_{deviation} + \boldsymbol{R}_{safety} \tag{10}$$

4. Learner module based on safety situation awareness MAPCPO

The SRL-CLCADI learner operates in parallel with the sampler, utilizing the data collected from the sampler to update both the policy and the value neural networks. Subsequently, it synchronizes the

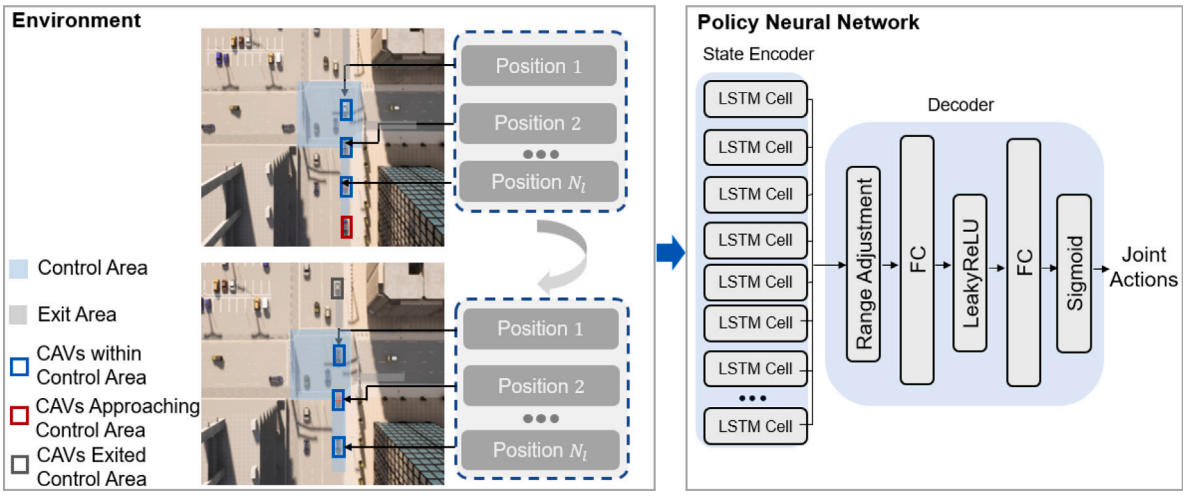


Fig. 5. Policy neural network structure diagram.

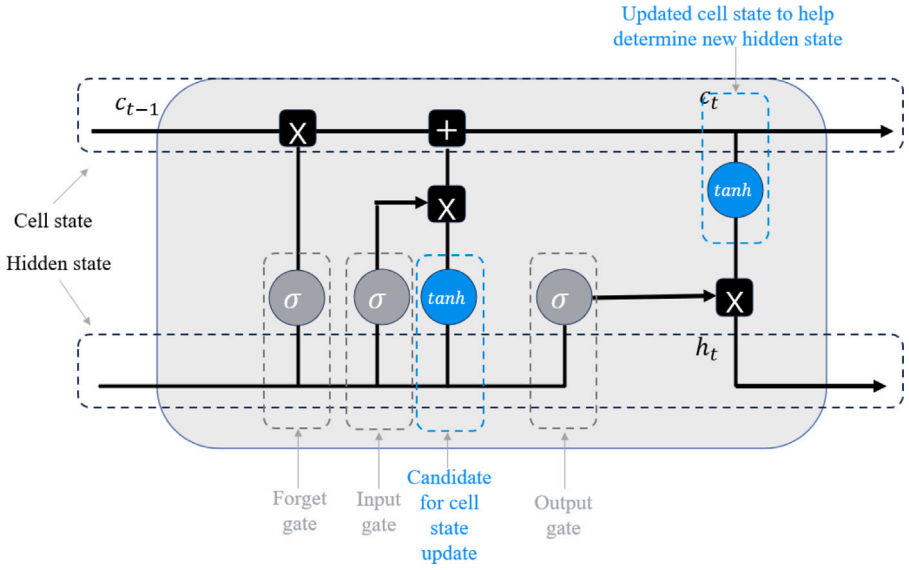


Fig. 6. LSTM unit structure diagram.

updated parameters back to the learner, preparing for the next epoch of sampling and optimization. This iterative process continues until the desired performance in intersection traffic is achieved. This section describes the core methodology of the learner module for addressing the CMG problem, including the design of the DRL network architecture with one actor and two critics, and the Safety Situation Awareness Projection-based Multi-Agent Constrained Policy Optimization Methodology.

4.1. DRL architecture design

The SRL-CLCADI system comprises three neural networks: the policy neural network, the reward value neural network, and the cost value neural network. The policy neural network is responsible for providing collaborative control decision-making for CAV passing through the intersection, mapping the local states of all CAVs at the current time step to their joint actions at the next moment. Meanwhile, as shown in Fig. 2, the reward and cost value networks are used to evaluate the expected reward and safety cost values under the current policy.

In contrast to traditional RL policy neural networks that utilize MLP architectures, the collaborative control policy necessitates the acquisition of both historical motion data and social interaction data of CAVs. This requirement is pivotal as the maximum learning capacity of the policy directly sets the performance boundaries for collaborative control in intersection scenarios. To effectively address this challenge, the SRL-CLCADI approach integrates LSTM neural network structures into the policy neural network, thereby enhancing its capability to process and learn from the complex and dynamic information associated with CAVs. As shown in Fig. 5, the structure of the policy neural network consists of a state encoder and a decoder. The main function of the encoder, which is composed of multiple LSTM units, is to model the current dynamic information of the vehicles. The primary function of the decoder is to adjust the range of the output tensor from the state encoder. Due to the small range of changes in the tensor output by the LSTM, it is not conducive to capturing the differences before and after the policy update in safe RL. This also leads to a slow speed of the agent in exploring the action space, ultimately resulting in a high time cost in learning effective policies. Therefore, a range adjustment module has been added to amplify the changing characteristics of the tensor, thus accelerating the exploration and learning speed of the policy neural network.

The structure of LSTM neurons is shown in Fig. 6, which consists of five parts: the cell state, hidden state, forget gate, input gate, and output gate. When LSTM unit is working, initially, the forget gate

of the LSTM determines which previous state information should be retained and which should be discarded. This is achieved by observing the hidden state from the previous time step h_{t-1} and the current input s_t and applying a Sigmoid function to generate a forget gate activation value F_t ranging between 0 and 1. This value is then multiplied by the previous cell state c_{t-1} to decide the extent to which each state information is retained. Subsequently, the input gate decides which new information from the current step will be saved in the cell state. This also involves two parts: a Sigmoid function determines which information to update, generating an input gate activation value I_t; a Tanh function creates a candidate state vector T_t, which provides the new information that might be added to the cell state. Then, the element-wise product $I_t \times T_t$ represents the new information that will be added to the cell state. Following this, the cell state c_t is updated by combining the outputs from the forget and input gates. The specific update formula is $c_t = F_t \times c_{t-1} + I_t \times T_t$ which indicates that the current cell state is composed of partially forgotten previous states and newly added information. Lastly, the output gate controls the output generated based on the current cell state c_t. A Sigmoid function determines which part of the cell state should be outputted, and a Tanh function normalizes the cell state to a range of -1 to 1. This output is then produced through an element-wise multiplication with the output of the Sigmoid function, generating the hidden state output h_t.

In the considered multi-agent system, the neural network's input S_t represents the set of states of CAVs at time step t. Given that the behavior of CAVs in the intersection scenario is dynamic, the update of the state set inevitably includes the incorporation of new CAVs and the departure of existing CAVs. To achieve this dynamic feature, this work introduces a circular queue mechanism, where the neural network updates the input in a rolling manner. Fig. 5 exemplifies the operation of the circular queue mechanism using a single north-south lane. The upper part of Fig. 5 shows the lane with one CAV about to enter the control area and three CAVs within the control area. The lower part depicts the subsequent movement: the CAV at the head of the queue departs the control area, followed by each CAV in the control area advancing forward by one position, with the newly entered CAV occupying the rear of the queue.

Specifically, the state space of lane l is predefined with N_l positions, where N_l is greater than the maximum capacity of the control area of that lane at the intersection under high-density traffic conditions. Assume at time step t, there are u_l CAVs in the queue P_l. In future time steps when a CAV departs, the position at the head of the queue, $P_l[0]$, will be vacated. Subsequently, all CAVs within the queue move forward sequentially by one position, leaving the last position empty. When a new CAV enters the controlled area, it will occupy the rear position of the queue, $P_l[u_l]$. This method facilitates the maintenance of a distance-ordered arrangement of CAVs within the state space, from farthest to nearest to the intersection, aiding the neural network in understanding the current environmental state. The rule for updating the position within the state space can be formalized as:

$$\begin{cases} \text{Dequeued Element} = P_l[0] \\ P_l[j] = P_l[j+1] & \text{for } j = 0 \text{ to } u_l - 2 \\ P_l[u_l] = \text{Enqueued Element} \end{cases} \tag{11}$$

The reward value neural network and cost value neural network mainly serve to evaluate the current policy neural network's strengths and weaknesses, mapping dynamic state information into expected reward values and expected safety cost values, respectively. Given their lower task complexity, to conserve computational and training data resources, fully connected layer structures are chosen for the reward value neural network and safety costs value neural network structures.

4.2. Policy neural network optimization

4.2.1. Policy optimization problem description

The purpose of policy optimization is to continually update the policy π based on the trajectories $\mathcal{T}_t = (s_t, a_t, r_t, c_t, s_{t+1})$ of multi-agent interactions with the environment sampled by the sampler, enabling the multi-agents to learn to maximize the expected rewards while maintaining the function below a threshold for the optimal policy. For details, refer to Eqs. (1) and (2).

The state–action value function $Q^\pi(s, \boldsymbol{a})$ with respect to reward indicates the quality of taking action $\boldsymbol{a}$ in state s, while the state value function $V^\pi(s)$ intuitively represents the quality of state s. Which can be respectively defined as:

$$Q^\pi(s, \boldsymbol{a}) = \mathbb{E}_{s_1:\infty\sim P, \boldsymbol{a}_0:\infty\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, \boldsymbol{a}_0 = a\right] \tag{12}$$

$$V^\pi(s) = \mathbb{E}_{\boldsymbol{a}\sim\pi}\left[Q(s, \boldsymbol{a})\right] \tag{13}$$

On this basis, the concept of the advantage function $A^\pi(s, \boldsymbol{a})$ is established, which represents the advantage of taking action $\boldsymbol{a}$ relative to the average reward of all actions. It is defined as:

$$A^\pi(s, \boldsymbol{a}) = Q_\pi(s, \boldsymbol{a}) - V_\pi(s) \tag{14}$$

In practical computation, the Generalized Advantage Estimation (GAE) method is commonly used for calculation. Similarly, the state–action Value function and state value function concerning the jth cost constraint value of agent i are respectively defined as:

$$Q^\pi_{c_j^i}(s, \boldsymbol{a}^i) = \mathbb{E}_{s_1:\infty\sim P, \boldsymbol{a}_0:\infty\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t C_j^i(s_t, \boldsymbol{a}_t) | s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}\right] \tag{15}$$

$$V^\pi_{C_j^i}(s) = \mathbb{E}_{\boldsymbol{a}\sim\pi}\left[Q^\pi_{C_j^i}(s, \boldsymbol{a}^i)\right] \tag{16}$$

The cost advantage function for agent i and cost index j is defined as:

$$A^\pi_{C_j^i}(s, \boldsymbol{a}^i) = Q^\pi_{C_j^i}(s, \boldsymbol{a}^i) - V^\pi_{C_j^i}(s) \tag{17}$$

which measures the improvement of cost function after taking actions under the current state and follow policy π. To achieve more stable and efficient learning, practically, it is calculated by GAE method, which is obtained as

$$A^\pi_{C_j^i}(s, \boldsymbol{a}^i) = \sum_{k=0} (\gamma\lambda)^k \left(-C_j^i(s_{t+k}, \boldsymbol{a}_i^{t+k}) + \gamma V^\pi_{C_j^i}(s_{t+k+1}) - V^\pi_{C_j^i}(s_{t+k})\right) \tag{18}$$

where λ is a parameter about GAE, we use it to balance the accuracy of measurements and variance.

4.2.2. Policy network optimization method

The MAPCPO method, presented in this section, addresses the CMG problem by searching for the optimal feasible policy within the region satisfying pre-set KL divergence and safety cost constraints. In the multi-agent environment, each agent has its local KL-divergence trust region, denoted as $\mathcal{T}^i$. The global KL-divergence trust region, $\mathcal{T}$, is the intersection of all local KL-divergence trust regions: $\mathcal{T} = \bigcap_{i=1}^n \mathcal{T}^i$. Under a centralized policy framework, the global KL-divergence trust region is defined as the local neighborhood of the most recent iteration π^k that satisfies the expected KL-divergence being less than a specified step size.

$$\mathcal{T} = \{\pi : \overline{D}(\pi \parallel \pi^k) \le \delta\} \tag{19}$$

Moreover, the policy update must satisfy safety constraints c_j^i. Thus, the safety constraint trust region for policy π^k is:

$$\boldsymbol{C} = \left\{\pi : J_{C_j^i}(\pi) + \frac{1}{1-\gamma}\mathbb{E}(A^\pi_{c_j^i}) \le d_j^i\right\} \tag{20}$$

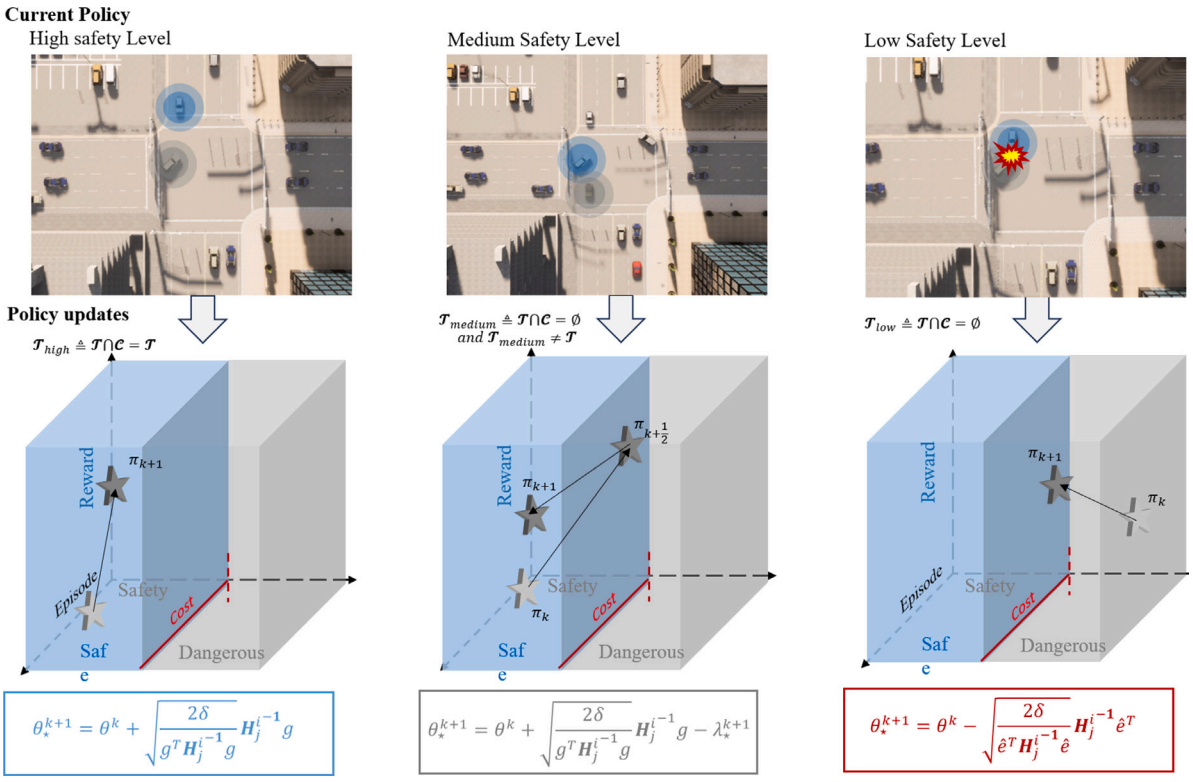


Fig. 7. Policy updating method oriented towards different safety postures.

Constrained by the KL-divergence-based trust region $\mathcal{T}$ and the safety The cost-based constraint region $\boldsymbol{C}$, the objective of policy optimization is to ensure that the policy remains within both constraint regions while obtaining a higher reward function value. As illustrated in Fig. 7, the updated policy is ensured, as much as possible, to maximize the expected reward value within the constraints of safety costs.

$$\pi^{k+1} = \arg\max_{\pi} \mathbb{E}[A^{\pi}(s, \boldsymbol{a})], \quad \pi^{k+1} \in \mathcal{T} \cap \boldsymbol{C} \tag{21}$$

Based on the relative positions of the trust region $\mathcal{T}$ and the constraint region $\boldsymbol{C}$, the likelihood of policy updates affecting the safety levels of multi-agents can be categorized as high safety level, medium safety level, and low safety level.

Let the parameters of the policy neural network be θ, the Hessian matrix of the average KL divergence be $\boldsymbol{H}_j^i = \frac{\partial^2 \mathbb{E}_{s_0 \sim \rho^0, s_{1:\infty} \sim P_{\pi_k}} \left[\overline{D}_{KL}(\pi \| \pi^k)[s]\right]}{\partial \theta_i \partial \theta_j}$, and g denote the policy gradient of the advantage function. The update methods under three different safety situations are as follows.

(1) High-Safety

Fig. 7 (a) illustrates the scenario with high safety level, the reward-guided updatable policies are entirely in the intersection of the trust region and the constraint region, indicating that the overall safety of the current policy is at a high level. At this time, even without imposing safety cost constraints, updating policies within KL divergence-based trust region will not lead to a dangerous situation while maximizing the reward. The policy region can be denoted as:

$$\mathcal{T}_{high} \triangleq \mathcal{T} \cap \boldsymbol{C} = \mathcal{T} \tag{22}$$

Policies within this trust region comply with the safety cost constraints; hence, a situation without an optimal solution is not anticipated. The traditional Trust Region Policy Optimization (TRPO) updating method is employed. For detailed proof, please refer to Schulman (2015).

$$\theta_\star^{k+1} = \theta^k + \sqrt{\frac{2\delta}{g^T \boldsymbol{H}_j^{i-1} g}} \boldsymbol{H}_j^{i-1} g \tag{23}$$

(2) Medium Safety

Fig. 7 (b) illustrates the scenario with medium safety level, the accessible region constrained by KL divergence intersects with the constraint region of safety costs. In this scenario, maximizing expected rewards might lead to the policy being in a risky state. After the initial step of maximizing rewards, the policy update results in a state where the safety cost value exceeds the threshold, necessitating a safety cost value projection step to address the constraint violation. The intersection of the KL divergence constraint range and the safety cost constraint range can be denoted as:

$$\mathcal{T}_{medium} \triangleq \mathcal{T} \cap \boldsymbol{C} \neq \emptyset \quad \text{and} \quad \mathcal{T}_{medium} \neq \mathcal{T} \tag{24}$$

which can be further expressed as:

$$\mathcal{T}_{medium} \triangleq \left\{ D_{\mathrm{KL}}(\pi \parallel \pi_\star) \leq \delta \right.$$
$$\left. \text{and} \quad J_{C_j^i}(\pi) + \frac{1}{1-\gamma} \mathbb{E}_{s_o \sim \rho^0, \boldsymbol{a} \sim \pi_k, s_{1:\infty} \sim \boldsymbol{P}_{\pi_k}} A_{C_j^i}^{\pi}(s, \boldsymbol{a}^i) \leq d^j \right\} \tag{25}$$

Under this safety situation the policy update using the MAPCPO algorithm is divided into two steps: reward enhancement and constraint projection. Initially, the TRPO method is used to maximize the reward of policy π into policy $\pi_\star^{k+\frac{1}{2}}$ without considering safety constraints, which is represented as follows:

$$\pi_\star^{k+\frac{1}{2}} = \arg\max \mathbb{E}_{s_o \sim \rho^0, \boldsymbol{a} \sim \pi_k, s_{1:\infty} \sim \boldsymbol{P}_{\pi_k}} \left[A^{\pi}(s, \boldsymbol{a})\right] \tag{26}$$

$$\text{s.t.} \quad \mathbb{E}_{s_o \sim \rho^0, \boldsymbol{a} \sim \pi_k, s_{1:\infty} \sim \boldsymbol{P}_{\pi_k}} \left[D_{\mathrm{KL}}(\pi \parallel \pi^{k+\frac{1}{2}})[s]\right] \leq \delta$$

Subsequently, policy $\pi^{k+\frac{1}{2}}$ is projected into the safety constraint set, transforming into policy $\pi_\star^{k+1}$, to ensure the safety of the policy optimization process.

$$\pi_\star^{k+1} = \arg\min D(\pi, \pi^{k+\frac{1}{2}}) \tag{27}$$

$$\text{s.t.} \quad J_{C_j^i}(\pi^{k+1}) + \frac{1}{1-\gamma}\mathbb{E}_{s_0\sim\rho^0,\boldsymbol{a}\sim\pi_k,s_{1:\infty}\sim\boldsymbol{P}_{\pi_k}}\left[A^{\pi}_{C_j^i}(s,\boldsymbol{a}^i)\right] \le d_j^i$$

Reward enhancement

Let $\theta_\star^{k+\frac{1}{2}}$ represent the optimal advantage function's largest policy neural network parameters within the KL-divergence trust region. The objective function is linearized within the KL-divergence trust region using a second-order approximation method.

$$\theta_\star^{k+\frac{1}{2}} = \arg\max g^T(\theta-\theta^k) \tag{28}$$

$$\text{s.t.} \quad \frac{1}{2}(\theta-\theta^k)^T\boldsymbol{H}_j^i(\theta-\theta^k) \le \delta$$

As the Hessian matrix is always positive semi-definite, (28) expression represents a convex optimization problem with quadratic inequality constraints. If this optimization problem has a feasible solution, it satisfies the Slater's condition, and strong duality exists. The dual problem can be more efficiently solved using the following dual formulation:

$$\mathcal{L}(\theta^{k+\frac{1}{2}},\lambda^{k+\frac{1}{2}}) = -g^T(\theta-\theta^k) + \lambda\left(\frac{1}{2}(\theta-\theta^k)^T\boldsymbol{H}_j^i(\theta-\theta^k)-\delta\right) \tag{29}$$

Considering the continuously differentiable nature of the original objective function, it can be established that the Karush–Kuhn–Tucker (KKT) conditions serve as both necessary and sufficient criteria for determining the optimal solution $\theta_\star^{k+\frac{1}{2}}$ for the primal problem, and $\lambda_\star^{k+\frac{1}{2}}$ for the dual problem. Consequently, Eq. (28) can be proficiently addressed by employing the KKT conditions:

$$-g+\lambda^{k+\frac{1}{2}}\boldsymbol{H}_j^i\theta^{k+\frac{1}{2}} - \lambda^{k+\frac{1}{2}}\boldsymbol{H}_j^i\theta^k = 0 \qquad \nabla_\theta\mathcal{L}(\theta^{k+\frac{1}{2}},\lambda^{k+\frac{1}{2}}) = 0 \tag{30}$$

$$\left(\frac{1}{2}(\theta^{k+\frac{1}{2}}-\theta^k)\right)^T\boldsymbol{H}_j^i\left(\theta^{k+\frac{1}{2}}-\theta^k\right)-\delta = 0$$
$$\nabla_\lambda\mathcal{L}(\theta^{k+\frac{1}{2}},\lambda^{k+\frac{1}{2}}) = 0 \tag{31}$$

To satisfy the KKT conditions, the primal constraints, dual constraints, and complementary slackness must be met respectively in Eqs. (30) and (31).

$$\frac{1}{2}(\theta^{k+\frac{1}{2}}-\theta^k)^T\boldsymbol{H}_j^i(\theta^{k+\frac{1}{2}}-\theta^k)-\delta \le 0 \tag{32}$$

$$\lambda^{k+\frac{1}{2}} \ge 0 \tag{33}$$

$$\lambda^{k+\frac{1}{2}}\left(\frac{1}{2}(\theta^{k+\frac{1}{2}}-\theta^k)^T\boldsymbol{H}_j^i(\theta^{k+\frac{1}{2}}-\theta^k)-\delta\right) = 0 \tag{34}$$

Ensuring the constraints from the Eqs. (32), (33) and (34) are met, $\theta_\star^{k+\frac{1}{2}}$ can be derived as $\theta_\star^{k+\frac{1}{2}} = \theta^k + \frac{1}{\lambda^{k+\frac{1}{2}}}\boldsymbol{H}_j^{i\,-1}g$ from Eq. (39).

Subsequently, merging $\theta_\star^{k+\frac{1}{2}}$ into the Eq. (30), the optimal solution is $\lambda_\star^{k+\frac{1}{2}} = \sqrt{\frac{g^T\boldsymbol{H}_j^{i\,-1}g}{2\delta}}$

Cost projection

Subsequently, the MAPCPO method projects the intermediate policy $\theta^{k+\frac{1}{2}}$ into the safety cost constrained domain by minimizing the distance between current policy and the safety cost constrained set. The distance between the current policy and the safety cost constrained set is measured using the KL divergence. Using a second-order approximation, the objective function can be linearized:

$$\theta_\star^{k+1} = \arg\min_\theta \frac{1}{2}(\theta-\theta^{k+\frac{1}{2}})^T\boldsymbol{H}_j^i(\theta-\theta^{k+\frac{1}{2}}) \tag{35}$$

$$\text{s.t.} \quad \hat{e}^T(\theta-\theta^{k+\frac{1}{2}})+\hat{c} \le 0$$

where e represents the policy gradient of the safety value function, defined as $\hat{e} \approx \sum_i\sum_j \nabla_\theta J_{c_j^i}(\pi)$ and $\hat{c}$ represents the proximity of the current policy's safety level to the safety cost threshold, which is defined as $\hat{c} \approx \sum_i\sum_j\left(J_{C_j^i}(\pi)-\frac{\hat{c}_j}{1-\gamma}\right), \forall j=1,\dots,h.$

Similar to the reward enhancement step, the Eq. (35) can be solved using the dual approach:

$$\mathcal{L}(\theta^{k+1},\lambda^{k+\frac{1}{2}}) = \frac{1}{2}(\theta-\theta^{k+\frac{1}{2}})^T\boldsymbol{H}_j^i(\theta-\theta^{k+\frac{1}{2}})+\lambda(\hat{e}^T(\theta-\theta^k)+\hat{c}) \tag{36}$$

With $\lambda_\star^{k+1}$ denoting the optimal solution to the dual problem, Eq. (36) can be efficiently addressed using the KKT conditions.

$$\begin{cases}\mathcal{L}\theta^{k+1}-\mathcal{L}\theta^{k+\frac{1}{2}}+\lambda^{k+1}\hat{b}=0\\ \nabla_\theta\mathcal{L}(\theta^{k+1},\lambda^{k+1})=0\\ \nabla_\lambda\mathcal{L}(\theta^{k+1},\lambda^{k+1})=0\end{cases} \tag{37}$$

$$\begin{cases}\hat{e}^T(\theta^{k+1}-\theta^k)+\hat{c}=0\\ \nabla_\lambda\mathcal{L}(\theta^{k+1},\lambda^{k+1})=0\end{cases} \tag{38}$$

Eqs. (37) and (38) must satisfy the primal constraints, dual constraints, and complementary slackness from the KKT conditions, as shown:

$$\hat{e}^T(\theta^{k+1}-\theta^k)+\hat{c} \le 0 \tag{39}$$

$$\lambda^{k+1} \ge 0 \tag{40}$$

$$\lambda^{k+1}(\hat{e}^T(\theta^{k+1}-\theta^k)+\hat{c}) = 0 \tag{41}$$

Deriving from Eq. (41), it follows that $\theta_\star^{k+1} = \theta^{k+\frac{1}{2}}+\lambda_\star^{k+\frac{1}{2}}\hat{e}$. Considering the constraints imposed by Eqs. (39) and (40), the optimal solution for the dual problem can be expressed as $\lambda_\star^{k+1} = \max\left(0, \frac{\hat{e}^T(\theta^{k+\frac{1}{2}}-\theta^k)+\hat{c}}{\hat{e}L^{-1}\hat{e}}\right)$. Therefore, the optimization outcomes that satisfy Eqs. (39) and (41) are:

$$\theta_\star^{k+1} = \theta^{k+\frac{1}{2}} - \lambda_\star^{k+1}\boldsymbol{H}_j^{i\,-1}\hat{e} \tag{42}$$

Finally, combining the reward enhancement step with the cost provides the optimized result:

$$\theta_\star^{k+1} = \theta^k + \sqrt{\frac{2\delta}{g^T\boldsymbol{H}_j^{i\,-1}g}}\boldsymbol{H}_j^{i\,-1}g - \lambda_\star^{k+1}\boldsymbol{H}^{-1}\hat{e} \tag{43}$$

The analysis of the worst-case performance degradation of the system during policy iteration, through the aforementioned two steps of policy update, is as follows.

When the current policy meets the safety cost constraints, given a current policy π^k that satisfies these constraints, under the KL-divergence projection, the upper bounds for reward improvement and constraint violation for each policy update are given as follows (Yang et al., 2020):

$$\begin{cases}J(\pi^{k+1})-J(\pi^k) & \ge -\frac{\sqrt{2\delta}\gamma\vartheta_R^{k+1}}{1-\gamma^2}\\ J_{c_j^i}(\pi^{k+1}) & \le d_j+\frac{\sqrt{2\delta}\gamma\vartheta_C^{k+1}}{1-\gamma^2}\end{cases} \tag{44}$$

where

$$\vartheta_R^{k+1} = \max|\mathbb{E}_{s_0\sim\rho^0,\boldsymbol{a}\sim\pi^k,s_{1:\infty}\sim\boldsymbol{p}^{\pi_k}}\left[A^{\pi_k}(s,\boldsymbol{a})\right]|,$$
$$\vartheta_C^{k+1} = \max|\mathbb{E}_{s_0\sim\rho^0,\boldsymbol{a}\sim\pi^k,s_{1:\infty}\sim\boldsymbol{p}^{\pi_k}}\left[A^{\pi}_{C_j^i}(s,\boldsymbol{a}^i)\right]|.$$

When the current policy does not satisfy the safety cost constraints, given a current policy π^k that meets the constraints, under KL-divergence projection, the upper bounds for reward improvements

Table 2
Main parameters of the experiments.

Parameters	Value
Scenario Parameters	
Time-step	0.1 s
Control distance length	60 m
Road width	14.2 m
Network bandwidth	20 MHz
RSU Tx power	50 dBm
RSU antenna gain	12 dB
CAV antenna gain	3 dB
SRL-CLCADI	
Discount factor	0.99
Learning rate	$1\times10^{-3}\rightarrow 0$ (linearly)
Max KL divergence	0.001
Damping coefficient	0.01
Cost limit	1
Policy std σ_θ	$1\rightarrow 0$ (exponentially)
Policy std decrease index ϑ	$-1.5e-6$
Coefficient of std ζ	1
Collision safety threshold	8 m
GAE coefficient ρ	0.97
$\boldsymbol{N}_e, \boldsymbol{N}_t, \boldsymbol{N}_a$	2000, 2000, 500
c_j^i	$1.5\times e^{-4}$
$\delta_r, \delta_c, \delta_s, \delta_p, \delta_a, \delta_v, \delta_{a_t}$	15, 125, 20, 15, 125, 0.05, 0.05, 0.4
Dimension of $\boldsymbol{S}, \boldsymbol{A}$	252, 48
MAPPO	
Learning rate	$3\times10^{-4}\rightarrow 0$ (linearly)
Clip range ε	0.2
Minibatch size	64
VICS and MICA	
Predictive horizon	5
Target velocity v_t	15 m/s

and constraint violation for each policy update are given by Yang et al. (2020):

$$\begin{cases} J(\pi^{k+1}) - J(\pi^k) & \geq -\dfrac{\sqrt{2(\delta+b^{+2}\alpha_{KL})\gamma}\vartheta_R^{k+1}}{1-\gamma^2} \\ J_{c_j^i}(\pi^{k+1}) & \leq d_j + \dfrac{\sqrt{2(\delta+b^{+2}\alpha_{KL})\gamma}\vartheta_C^{k+1}}{1-\gamma^2} \end{cases} \tag{45}$$

These theoretical outcomes provide mathematical assurances for satisfying safety cost constraints during policy iteration. Concurrently, they establish explicit limits for potential performance degradation during policy updates, thereby ensuring the enhancement of algorithmic performance through the policy update process. where $b^+ = \max(0, J_j^i(\pi^k) - c_j^i)$ and $\alpha_{KL} = \frac{1}{2q^T \boldsymbol{H}_j^{i\,-1} q}$. Here, q represents the gradient of the safety cost advantage function.

(3) Low Safety

Fig. 7 (c) illustrates the scenario with low safety level, the reward-driven updatable policy lies entirely outside the constraint region. There is no intersection between the trust and constraint regions, suggesting the current policy's overall safety is critically low. Thus, any policy update will not return to the constraint region, satisfying the safety cost. The policy region can be represented as:

$$\boldsymbol{\mathcal{T}}_{low} \triangleq \boldsymbol{\mathcal{T}} \cap \boldsymbol{C} = \emptyset \tag{46}$$

When the policy is in a low safety posture, a backtrack search method is utilized to discover updates aimed at finding suitable parameters for the policy update:

$$\theta_\star^{k+1} = \theta^k - \sqrt{\frac{2\delta}{\hat{e}^T \boldsymbol{H}_j^{i\,-1}\hat{e}}}\boldsymbol{H}_j^{i\,-1}\hat{e}^T \tag{47}$$

4.2.3. Value neural network optimization

The reward value neural network and the safety costs value neural network function to assess the efficacy of the newly updated policy neural network, with their current network parameters denoted as θ_R^{k+1} and θ_C^{k+1}, respectively. They calculate the loss based on the discrepancy between the estimated and actual values of reward and safety cost, and subsequently update to obtain new network parameters ϕ_R^k and ϕ_C^k.

$$\theta_R^{k+1} = \arg\min_\theta E\left[\left(V_{\phi_R^k}(s_t) - \hat{R}_t\right)^2\right] \tag{48}$$

$$\theta_C^{k+1} = \arg\min_\theta E\left[\left(V_{\phi_C^k}(s_t) - \hat{C}_t\right)^2\right] \tag{49}$$

where $\hat{R}_t$ represents the true reward value, and $\hat{C}_t$ represents the true safety cost value.

Algorithm 1 SRL-CLCADI

Require: Global state $\boldsymbol{S}$ from environment: $\boldsymbol{S} = [\prod_{i=1}^n (d^i, v^i, \zeta^i, t^i, \kappa^i)]$
Ensure: Expected velocity and steer of each vehicle in the environment: $\boldsymbol{A} = [\prod_{i=1}^n (v_i^t, \psi_i^t)]$
1: Initialize hyper-parameters: θ_R, θ_C, θ_{π^k}, set training parameter $\boldsymbol{N}_e$, $\boldsymbol{N}_t$
2: **for** $e = 1, 2, \ldots, N_e$ **do**
3: **for** $t = 1, 2, \ldots, N_t$ **do**
4: **for** each vehicle $i = 1, 2, \ldots, n$ **do**
5: chose action $\boldsymbol{\mathcal{A}}_t$ according to the current policy.
6: Execute global actions $\boldsymbol{\mathcal{A}}_t = \prod_{i=1}^n (v_i^t, \psi_i^t)$, get reward $\boldsymbol{R}_t$, cost $\boldsymbol{C}_t$ and next state $s_{t+1}(t) = [\prod_{i=1}^n (d^i, v^i, \zeta^i, t^i, \kappa^i)]$
7: $\boldsymbol{S}_t = \boldsymbol{S}_{t+1}$
8: **end for**
9: **end for**
10: Collect trajectories $\boldsymbol{\tau}_{\pi_k} = (\boldsymbol{S}_t, \boldsymbol{a}_t, r_t, \boldsymbol{c}_t, s_{t+1})$
11: Calculate advantage function of reward and risk function: $A^{\pi_k}(s, \boldsymbol{a})$, $A_{C_j^i}^{\pi_k}(s, \boldsymbol{a})$
12: Calculate $\hat{g}$, $\hat{e}$, $\hat{c}$
13: Update policy network as:
14: **if** $\boldsymbol{\mathcal{T}}_{high} \triangleq \boldsymbol{\mathcal{T}} \cap \boldsymbol{C} = \boldsymbol{\mathcal{T}}$ **then**
15: $\theta_\star^{k+1} = \theta^k + \sqrt{\frac{2\delta}{\hat{g}^T \boldsymbol{H}_j^{i\,-1}\hat{g}}}\boldsymbol{H}_j^{i\,-1} g$
16: **else if** $T_{medium} \triangleq T \cap C = \emptyset$ and $\boldsymbol{\mathcal{T}}_{medium} \neq T$ **then**
17: $\theta_\star^{k+1} = \theta^k + \sqrt{\frac{2\delta}{\hat{g}^T \boldsymbol{H}_j^{i\,-1}\hat{g}}}\boldsymbol{H}_j^{i\,-1} g - \lambda_\star^{k+1}$
18: **else**
19: $\theta_\star^{k+1} = \theta^k - \sqrt{\frac{2\delta}{\hat{e}^T \boldsymbol{H}_j^{i\,-1}\hat{e}}}\boldsymbol{H}_j^{i\,-1}\hat{e}^T$
20: **end if**
21: Update ϕ_R^k, ϕ_C^k as:
22: $\theta_R^{k+1} = \arg\min_\theta E\left[\left(V_{\phi_R^k}(s_t) - \hat{R}_t\right)^2\right]$
23: $\theta_C^{k+1} = \arg\min_\theta E\left[\left(V_{\phi_C^k}(s_t) - \hat{C}_t\right)^2\right]$
24: **end for**

4.3. SRL-CLCADI algorithm overview

This section introduces SRL-CLCADI algorithm, based on the MAPCPO algorithm, which includes three safety situation update rules for addressing the cooperative control problem at intersections. This algorithm ensures both monotonic reward performance improvement and safety cost constraints satisfaction, as presented in Algorithm 1. The first line of the algorithm initializes network and algorithm parameters, including the random reward-based value network θ_R, the safety cost-based value network θ_C, and the policy network θ^k. It also sets the total number of training iterations $\boldsymbol{N}_e$ and the maximum number of time steps per iteration $\boldsymbol{N}_t$. The main loop of the algorithm starts from the second line. It first constructs a trajectory $\boldsymbol{\tau}_{\pi^k}$ consisting of state–action pairs for multiple vehicles, along with the associated reward $\boldsymbol{A}_R^\pi(s, \boldsymbol{a})$ and safety cost values $\boldsymbol{A}_{C_j^i}^\pi(s, \boldsymbol{a}^i)$ within the environment (lines 2–12). The algorithm then evaluates and updates the policy network and the reward and safety cost-based value networks independently (lines 13–23). When updating the policy network, the algorithm considers both the current policy's constraint satisfaction and the feasibility of

subsequent policies. It estimates the likelihood of a policy update leading to a safe scenario for the multi-agent system and proposes an appropriate update solution (lines 13–20). Regarding the reward and safety cost-based value networks, their respective gradients are employed to update the network parameters (line 21–23).

5. Experiments

This section describes the experiments conducted to assess the performance of the proposed method. First, the models and parameters used in the experiment are elaborated. Then, the acquired training and testing results are analyzed. Additionally, a sensitivity analysis of the hyperparameters in the cost function is performed, and the vehicle scheduling process at intersections is visualized.

5.1. Experiments setting

The simulation tools used in this experiment include CARLA, SUMO, and NS3. The hardware environment consists of a high-performance computer equipped with an i9-13700KF CPU and an NVIDIA GeForce RTX 3090 GPU, running Ubuntu 18.04 as the operating system. In the simulation setup, CARLA and SUMO are integrated through their respective Python API to perform joint traffic simulation, enabling coordinated modeling of complex traffic scenarios. The RL methods were implemented using the PyTorch framework and network simulation was conducted in NS3 using C++ to accurately emulate vehicular network protocols and network traffic. Additionally, CARLA's built-in sensors were utilized to transmit real-time vehicle status information, including speed, position, and orientation. To generate vehicle traffic trajectories at intersections, we used the BasicAgent class in CARLA, which allows for dynamic movement of multiple autonomous vehicles through the intersection area. For detailed configurations of the intersection scenario and network simulation parameters, please refer to Table 2.

Vehicle speed and turning angle control signals provided by the policy neural network were converted into throttle and brake signals via a PID controller within the simulator. In this experiment, a four-way dual-lane signal-free intersection in CARLA TOWN05 was chosen as the training and testing scenario for the RL model. The road width was 14.2 m, with the East–West and North–South lanes measuring 65 m and 50 m in length, respectively. Taking into account the road characteristics in the CARLA map and the V2I communication range, the control area length was fixed at 70 m. Various vehicle types were selected for the simulation to emulate genuine traffic flow, with vehicle dimensions ranging from 3.6–5.4 m in length, 1.8–2.2 m in width, and 1.5–2 m in height. Vehicle arrivals followed a Poisson distribution, and the traffic flow in the scenario was continuous. Vehicle turning intentions at intersections (left turn, straight, and right turn) are determined randomly, ensuring a more realistic and varied traffic flow within the simulated environment. The time step was set to 0.1 s, consistent with the actual vehicle control cycle.

To evaluate the performance of the proposed SRL-CLCADI algorithm in experimental tests, two sets of experiments were conducted in this section. Specifically, the first set of experiments showcased the RL-Based training process of SRL-CLCADI, MAPPO (Guan et al., 2020) and MAPPO-SC, the latter having the same safety constraints as SRL-CLCADI but utilizing reward function penalties. The experiment compared the variations in rewards, costs, collision rates, number of safety distance violations, and acceleration during the training process of the SRL-CLCADI with those during the training processes of MAPPO and MAPPO-SC methods. And results demonstrate that our algorithm achieves superior overall performance, particularly in terms of safety, successfully reaching zero collisions for the first time. The second set of experiments compares the performance of the SRL-CLCADI-trained policy with MAPPO, MAPPO-SC, the MPC-based method VICS (Kamal et al., 2014) and the MIP-based method MICA (Lu & Kim, 2018) in terms of safety, ride comfort, computational time, and traffic efficiency.

VICS represents a coordination scheme in the MPC framework. The scheme efficiently utilizes the intersection area by preventing each pair of conflicting vehicles from approaching their cross collision point (CCP) at the same time, where the CCP is the intersection of their trajectories. A risk function is proposed to quantify the risk of a collision of a pair of vehicles around their CCP, which is given by

$$R_{i,i'}(t) = H\delta_{i,j}(t)e^{-(\alpha_i d_i^2+\alpha_{i'} d_{i'}^2)} \tag{50}$$

where H is a positive constant indicating the highest possible risk of collision, and $\delta_{i,i'}$ is a binary variable to state whether the vehicles i and i' have a CCP. Besides, d_i and $d_{i'}$ are the distances of the CCP from current position of vehicles i and i' along their trajectories, and α_i and $\alpha_{i'}$ are positive constants. At any time, if two conflicting vehicles are very close to their CCP, the risk function returns a high value, and if at least one vehicle is far from the CCP, it returns a low value. Based on that, a constrained nonlinear optimization problem is constructed as follows:

$$J = \sum_{t=0}^{T-1}\sum_{i=0}^{N} w_v(v_i(t+1)-v_d)^2 + \sum_{t=0}^{T-1}\sum_{i=0}^{N} w_a(a_i(t))^2 + \sum_{t=0}^{T-1}\sum_{i=0}^{N}\sum_{i'=i+1}^{N} w_R R_{i,i'}(t) \tag{51}$$

$$s.t. v_{\min} \le v_i \le v_{\max},$$

$$a_{\min} \le a_i \le a_{\max}.$$

where T is the length of the prediction horizon, v_d is the desired velocity, and w_v, w_a, and w_R are weight coefficients. There are three cost terms in total. The first term denotes the cost related to velocity deviation from the desired value v_d. The second term denotes the cost of acceleration. Minimizing these two terms means comfortable and smooth flow of vehicles. The third term denotes the cost related to the risk of collisions as defined in the risk Eq. (51), which sums up quantified risks at all CCPs for all possible pairs of vehicles considering their predicted trajectories in the horizon.

MICA represents a control scheme within the MIP framework. Define an objective function:

$$J = \sum_i t_{i,out} \tag{52}$$

$$s.t. t_i + M \times b_{i,i'} \le t_{i',in}, \text{if } b_{i,i'} = 1,$$

$$t_i \ge t_{i',out} + M \times (1 - b_{i,i'}), \text{if } b_{i,i'} = 0.$$

where t_i denotes the crossing time for vehicle i, M represents a sufficiently large number, and $b_{i,i'}$ is a binary variable. $t_{i',in}$ and $t_{i',out}$ respectively denote the times at which vehicle i' enters and exits the conflict area.

In the SRL-CLCADI algorithm, the policy function employs three hidden layers, comprising two fully connected layers and one LSTM layer, each containing 512 units. The value function utilizes two fully connected layers, each with 128 units. Both the MAPPO and MAPPO-SC algorithms use a multi-layer perceptron with two hidden layers as approximations for the policy and value functions, each layer containing 128 units. Policy iteration is performed every 2000 timesteps, with the Adam optimizer used for policy updates. The learning rate linearly decreases from 1×10^{-3} to 0, while the standard deviation exponentially decays from 1 to 0, and training stops after 2000 epochs. Moreover, the parameter settings for MAPPO, VICS and MICA algorithms are consistent with those in the original paper. The main parameters of the experiments are referenced in Table 2.

5.2. Performance comparison during training process

We compared the training performance of cooperative control method at Intersections based on RL: SRL-CLCADI, MAPPO, and

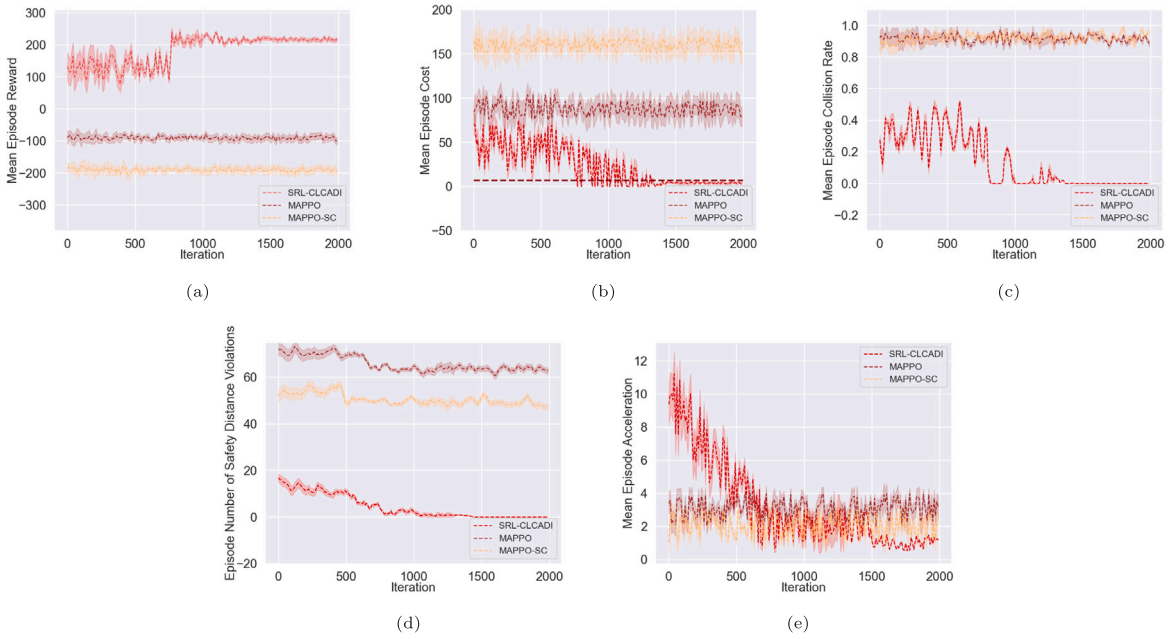


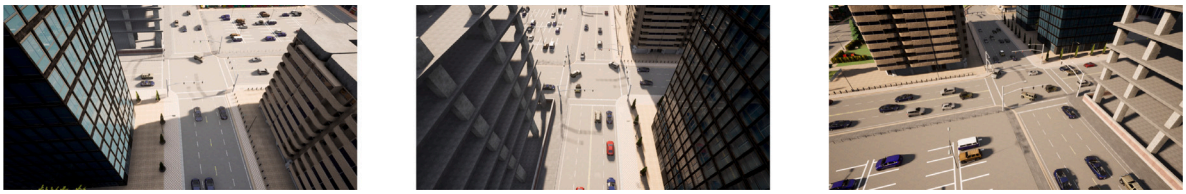
Fig. 8. Performance comparison of SRL-CLCADI policy with MAPPO, MAPPO-SC during training process: (a) shows the variation in average episode rewards, (b) represents the changes in average episode risk values, (d) illustrates the changes in average episode collision rates, and (e) demonstrates the variation in the number of times average episode safety distances are violated.

MAPPO-SC. The MAPPO-SC method features the same reward and safety cost functions as SRL-CLCADI, facilitating the comparison of the performance differences between the two policy updates under identical settings. To achieve optimal training results, each episode was set at $t = 2000$ time steps. Too short episode lengths could lead to trajectory information with high specificity, thereby compromising the robustness of the policy neural network. Conversely, excessively long episode durations demand significant computational and time resources. Additionally, these methods could yield policies with diminished responsiveness to terminal rewards, potentially leading to decreased efficiency in learning environments driven by rewards.

Performance metrics included average episodic reward, average episodic cost, average episodic collision rate, average episodic safety distance violation occurrences and average episodic acceleration. The average episodic reward value represents the overall performance of the policy. The average episodic cost represents the comprehensive safety performance. The average episodic acceleration illustrates the policy's comfort performance. The average episodic collision rate offers a direct insight into the policy's safety, and the average episodic safety distance violation occurrences denote potential collision risks. These metrics are calculated as the average over the four attempts per episode.

From Fig. 8(a), it can be observed that the SRL-CLCADI algorithm (represented by the bright red line) converges after approximately 1250 iterations. Post-convergence, its variance is relatively small, indicating good stability of the SRL-CLCADI algorithm. Due to the presence of stochastic delays in communication between vehicles, fluctuations increased during the training process and minor fluctuations persisted after convergence, yet the overall performance remained stable.The successful convergence of the SRL-CLCADI algorithm is attributed to its design, which integrates dual-value networks based on the reward function and safety expenditure, coupled with risk-aware constrained policy optimization. This enables it to balance optimization objectives with potential risks in addressing complex and uncertain road intersection control problems, thereby demonstrating robust performance. As for the MAPPO algorithm (represented by the dark red line) and the MAPPO-SC algorithm (represented by the yellow line), which adopt the same reward function design as SRL-CLCADI (considering safety, comfort, and efficiency), their performance does not show a significant increase during iterations. This is due to three highly challenging factors in the training scenario. First, the strong correlation between consecutive time steps in intersection scenarios means that future policy choices heavily depend on current decisions, and ordinary fully connected layer neural networks lack the capability to capture historical vehicle state information. Next, as vehicle collisions directly lead to the interruption of the training and truncation of the acquired rewards, and since they do not employ risk-constrained policies during performance optimization, the policies are highly prone to local optima. Finally, the training scenario is a dynamic intersection scenario where the timing of vehicles entering the intersection and their directions after entering are random, and the large dimensionality of the state space presents significant training challenges.

Fig. 8(b) represents the global safety level in the highly stochastic traffic environment established for this study. The safety cost value represents the risks of collision and skidding present in the traffic environment. The red dashed line parallel to the x-axis signifies the cost function limit of the SRL-CLCADI algorithm, serving as a guide for updating the safety cost value network. Given that this limit is a very small value, if the safety cost function falls below it, it can be assumed that there are virtually no risks of collision or skidding in the traffic environment, and almost no likelihood of CAVs coming dangerously close or being prone to skidding. It can be observed that the red curve, representing the SRL-CLCADI algorithm, successfully converges below the limit line after approximately 1250 policy iterations. It then maintains the safety of the traffic scenario with excellent stability, which is noteworthy in intersection scenarios where all vehicles exhibit random turning behaviors. The outstanding performance of the SRL-CLCADI algorithm can be attributed to two key features: an independent safety cost value network and safety cost-constrained policy optimization. These elements guide the policy, gradually enhancing the overall performance while simultaneously improving the system's safety. The MAPPO algorithm, which lacks safety risk reward penalties, and the MAPPO-SC algorithm, which incorporates such penalties, did not show a significant reduction in safety cost values even after repeated fluctuations. This reflects the limitations of traditional reward-driven RL



(a) Sparse (b) Normal (c) Dense

Fig. 9. Different traffic flow simulation scenarios: sparse, normal, and dense.

algorithms in safety-critical road intersection scenarios lacking safety awareness.

Fig. 8(c) provides a detailed depiction of the mean collision rate curve across multiple training iterations, directly reflecting the policy's safety performance. Fig. 8(d) shows the curve for the average number of safety distance violations, a lateral indicator of policy safety, where fewer violations indicate lower potential collision risks. The SRL-CLCADI algorithm, represented by the red curve, achieves both a zero collision rate and zero safety distance violations in the traffic scenarios after approximately 1250 policy iterations, maintaining this performance in subsequent iterations. In contrast, the MAPPO-SC and MAPPO algorithms, relying solely on a single-value network based on rewards, exhibit considerable fluctuation during training, but without a significant downward trend in collision rates and safety distance violations. This highlights the outstanding performance of SRL-CLCADI in longitudinal and lateral safety aspects of autonomous driving, and the inadequacy of traditional single-value, reward-based networks in handling complex scenarios at dynamic intersections.

Fig. 8(e) shows the curve of the mean acceleration per episode over multiple training iterations, a metric reflecting the comfort of passengers during the driving process. A lower value indicates higher comfort. It is observed that the strategy of the SRL-CLCADI algorithm converges to a high level of comfort and maintains this elevated level with low variance after convergence. Initially, due to safety constraints on its policy space, SRL-CLCADI could not maintain a high level of comfort. However, once the safety level of the environment is elevated, the policy network becomes more capable of optimizing driving comfort. In contrast, the MAPPO-SC and MAPPO algorithms exhibit considerable fluctuation during training, and their acceleration rates do not show a significant downward trend. This is attributed to the excessive complexity of the scenario and the lack of a separate safety cost value evaluation network, leading the policies to become trapped in local optima and unable to explore the optimal policy.

5.3. Performance comparison after deployment policy

In this study, the performance at intersections of five methods – SRL-CLCADI, MAPPO, MAPPO-SC, VICS, and MICA – is compared. The first three methods deploy trained policies into models. VICS, based on the MPC control method, serves as a performance baseline for intersection coordination control, whereas MICA, based on the MIP control method, is a classic approach for intersection coordination control. In the experimental scenario, the continuous flow of vehicles arriving at the intersection follows a Poisson distribution. As shown in Fig. 9, three intersection scenarios are selected, characterized by low traffic flow $\lambda_1 = 600$ veh/h/lane, medium traffic flow $\lambda_2 = 1200$ veh/h/lane, and high traffic flow $\lambda_3 = 1800$ veh/h/lane, where λ represents the Poisson number. Performance metrics include collision rate, number of safe distance violations, acceleration, fuel consumption per 100 kilometers, the number of vehicles passing during each policy-environment interaction attempt, and inference time. Collision rate and safe distance violations indicate safety, while acceleration measure comfort. Fuel consumption per 100 kilometers represents energy consumption. The efficiency is reflected through the number of vehicles passing during each policy-environment interaction, and inference time indicates the computational demand and real-time performance of the algorithm. Each method undergoes six sets of experiments, with the average of ten trials taken as the result for each set.

Fig. 10 presents the results for the low traffic flow scenario. As shown in the first two box plots of Fig. 10, in terms of safety, all methods except MAPPO and MAPPO-SC achieve a zero collision rate. This further underscores the inability of RL algorithms without explicit safety constraints to be implemented in safety-critical domains. Similarly, all but MAPPO and MAPPO-SC register zero safe distance violations, with MAPPO-SC showing less variation than MAPPO. This reflects the inherent safety risks of reinforcement learning methods without individual risk constraints, while SRL-CLCADI meets the safety requirements of autonomous driving.

The third box plots in Fig. 10 represent the policy's ride comfort. For the average acceleration per episode, the SRL-CLCADI algorithm exhibits the lowest median and the smallest variance, indicating superior comfort performance through small and stable accelerations. Similarly, in terms of average jerk per episode, the SRL-CLCADI algorithm demonstrates a comparable level of excellence. The fourth box plots in Fig. 10 represent the policy's Fuel Consumption Per 100 kilometers. The SRL-CLCADI method generally exhibits lower energy consumption, benefiting from its smoother acceleration control. The average energy consumption of the MAPPO-SC method is similar to that of the SRL-CLCADI method, but MAPPO-SC exhibits a larger variance, indicating that SRL-CLCADI has better stability in terms of energy consumption.

As shown in the fifth box plot of Fig. 10, in terms of efficiency, SRL-CLCADI possesses the highest single-pass count and the highest average pass count, which can be attributed to its design of regularized rewards and its excellent safety performance. MAPPO-SC has a slightly lower maximum single-pass count than the SRL-CLCADI method, but a significantly lower average pass count, due to its high inconsistency. If a collision occurs, the trial is interrupted, resulting in fewer vehicles passing during those trials. This inconsistency further highlights the higher safety risks associated with MAPPO-SC. The MPC-based VICS method and the MIP-based MICA method exhibit fluctuations similar to those of the SRL-CLCADI method, but their traffic efficiency is significantly lower than that of the SRL-CLCADI method.

As shown in the sixth box plot of Fig. 10, a key advantage of SRL-CLCADI lies in its shorter inference time, reducing the average inference time by approximately 65% compared to traditional control methods. This exceptional performance can largely be attributed to the RL methods' ability to directly correlate traffic scenario states with actions via neural pathways, thereby eliminating the need for exhaustive or extensive computational demands required for solving optimization constraints. The computational power at the endpoint of autonomous driving vehicles is extremely limited, and the high computational demand of traditional methods may lead to the failure to find the optimal solution or the application of suboptimal solutions, potentially causing safety incidents or poor collaborative control effects. This has limited the application of optimization methods in vehicle-road cooperative control system. Our method significantly reduces the inference time required for intersection cooperative control policies, further advancing

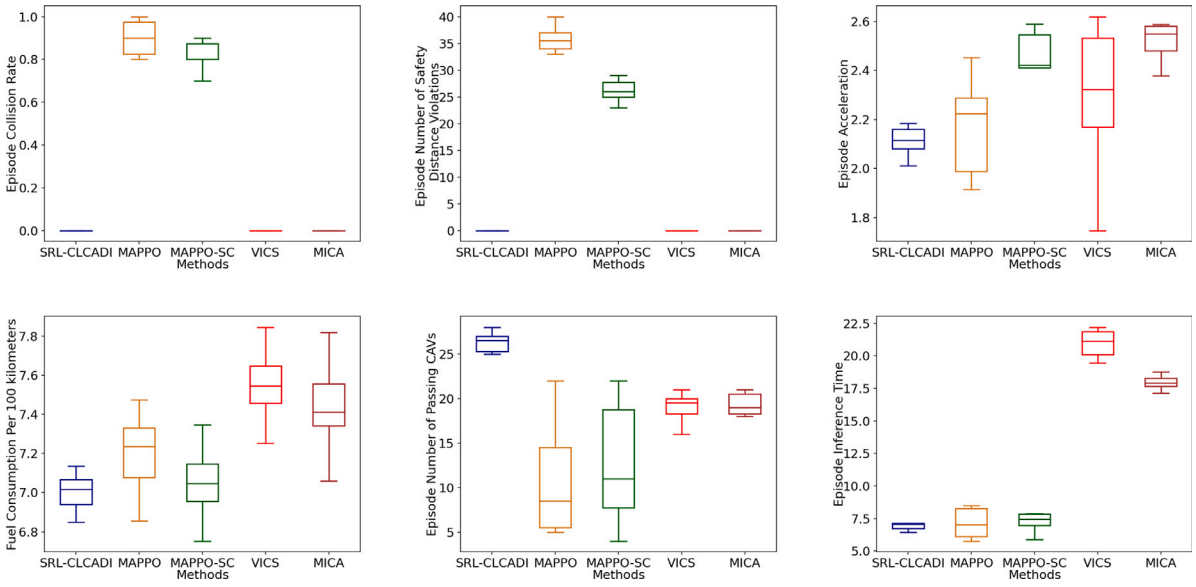


Fig. 10. Performance comparison of SRL-CLCADI policy with MAPPO, MAPPO-SC policies, VICS and MICA after deployment under sparse scenarios.

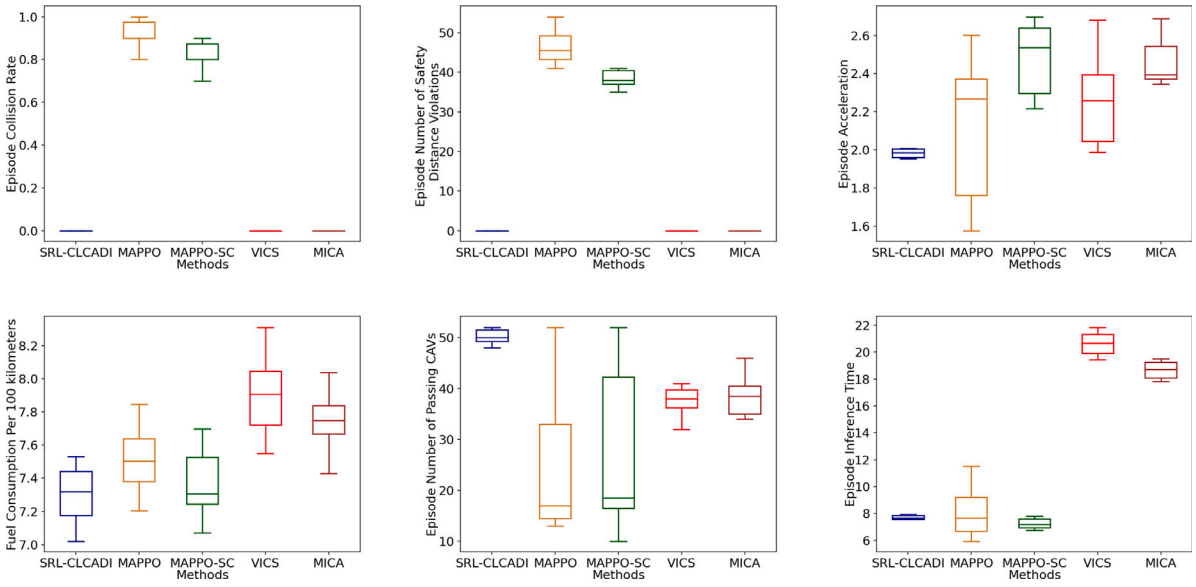


Fig. 11. Performance comparison of SRL-CLCADI policy with MAPPO, MAPPO-SC policies, VICS and MICA after deployment under normal scenarios.

the application of RL methods in vehicle-road cooperative control system.

Fig. 11 and Fig. 12 depict scenarios with medium and high traffic flow, respectively. These scenarios demonstrate that SRL-CLCADI maintains consistent performance characteristics akin to those in low traffic conditions, highlighting its robustness. Even with increased traffic flow, SRL-CLCADI does not exhibit a decline in performance. In both medium and high traffic conditions, SRL-CLCADI effectively sustains a zero collision rate, proving its efficacy in maintaining safety across varying traffic scenarios. In contrast, the MAPPO-SC algorithm, lacking specific safety constraints, exhibits a higher collision rate, underscoring the importance of limited safety cost constraints in RL-based vehicle-road cooperative control system. The collision rate of the MAPPO-SC method is slightly lower than that of the MAPPO method, which can be attributed to an appropriate reward function. The MPC-based VICS and MICA methods, like the SRL-CLCADI method, maintained a zero collision rate. From the perspective of riding comfort, SRL-CLCADI achieves the lowest median and minimal variance across all three traffic volumes, indicating its superiority in terms of ride comfort. As traffic volume increases, energy consumption also rises for all five methods, which is attributable to higher traffic densities implying a certain degree of congestion and greater fluctuations in vehicle acceleration. Regarding inference time, compared to the VICS and MICA methods, SRL-CLCADI shows a significant reduction in both medium and high traffic scenarios, with a greater decrease as the complexity of the scenario increases. The MAPPO method and the MAPPO-SC method have a slightly shorter inference time than the SRL-CLCADI method, which is attributed to the larger parameter size of the LSTM neural network in the SRL-CLCADI method. SRL-CLCADI has the highest traffic efficiency and the least fluctuation, while the MAPPO and MAPPO-SC algorithms exhibit higher peaks but greater variability, due to the lack of safety-related constraints in their update strategies. The VICS and MICA methods show smaller fluctuations, but their mean values are lower. This suggests that optimization-based methods experience a rapid increase in computational demand as traffic, and particularly the number of CAVs, increases, whereas SRL-CLCADI efficiently manages

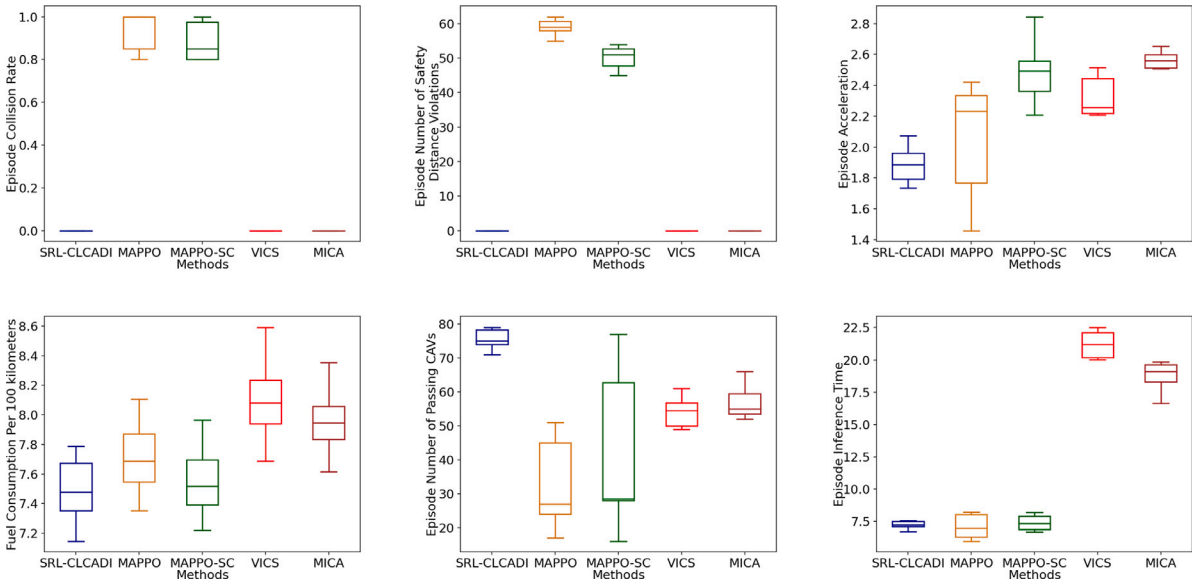


Fig. 12. Performance comparison of SRL-CLCADI policy with MAPPO, MAPPO-SC policies, VICS and MICA after deployment under dense scenarios.

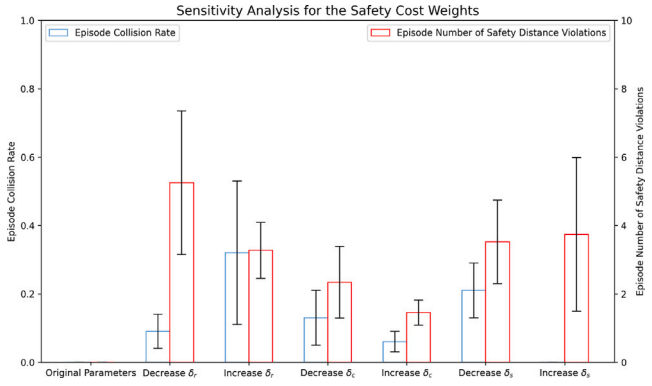


Fig. 13. Sensitivity analysis of safety distance violation, sideslip, and collision term weights. Bars represent: current parameters (1st), ±50% for safety distance violation term δ_r (2nd, 3rd), ±50% for sideslip term δ_c (4th, 5th), and ±50 units for collision term δ_s (6th, 7th).

these demands. The experimental results in medium and high traffic conditions further prove the method's ability to handle increased traffic without impacting key performance and safety metrics. Balancing inference time and safety in high-density traffic scenarios is crucial, further highlighting SRL-CLCADI's advanced capabilities in managing complex traffic conditions.

In summary, SRL-CLCADI offers the most suitable comprehensive performance in the autonomous driving and intelligent transportation domain. In high traffic flow scenarios, RL methods without individual safety constraints struggle to cope. When deployed in models, SRL-CLCADI, based on RL, provides a significant advantage in intersection coordination. Compared to traditional control methods, it delivers similar or even superior performance. However, traditional control policies require high computational power and longer inference times. In contrast, SRL-CLCADI demands minimal computational power and significantly shorter resolution times, representing a considerable improvement in computational efficiency.

5.4. Cost hyperparameter sensitivity analysis

To evaluate how cost weight hyperparameters affect algorithm performance, we conducted a sensitivity analysis of these weights, as shown in Fig. 13. The x-axis shows different parameter combinations, and the y-axes represent the collision rate and the number of safety distance violations per episode. The analysis indicates that adjusting the weight of each term – safety distance violation, sideslip, and collision – affects these performance metrics distinctly. First, when the safety distance violation term weight is reduced (with sideslip and collision weights constant), both the collision rate and safety distance violations increase. This result stems from the model's reduced penalty for safety distance violations during training, limiting the cost function's ability to guide the model towards safer outcomes. Consequently, an increase in safety distance violations also raises the collision rate. Conversely, when this weight is increased, the model places excessive focus on minimizing safety distance violations, which weakens the impact of the collision penalty. This narrow focus on safety distance violations paradoxically results in a higher collision frequency. A similar pattern occurs with adjustments to the sideslip weight. Reducing this weight leads to more frequent sideslip events, causing additional safety distance violations between vehicles, especially in adjacent lanes or intersections, which in turn raises the collision rate. Increasing the sideslip weight likewise causes both metrics to rise due to the diminished relative weight of the collision penalty. Lastly, reducing the collision penalty weight drives both metrics higher, as the collision penalty's diminished importance limits its effectiveness. When the collision penalty weight is maximized, the collision rate drops to zero, but due to a relative decrease in penalties for safety distance violations and sideslip, the frequency of safety distance violations rises. Based on these sensitivity analysis results, we selected the current parameter combination to achieve an optimal balance among the various safety cost metrics.

5.5. Traffic dynamic flow analysis

This section provides a traffic dynamics flow analysis. Two traffic scenarios were selected for trajectory and speed visualization, one with slightly below-average difficulty and one with slightly above-average difficulty. The difficulty considerations for the scenarios include factors such as traffic flow and the number of left-turning vehicles. Fig. 14 presents the visualization for the slightly below-average difficulty scenario, while Fig. 15 presents the visualization for the slightly above-average difficulty scenario. As shown in Fig. 14 (b), in the slightly below-average difficulty traffic scenario, only VEH2 decelerates and waits to pass at the initial stage, and the deceleration process is

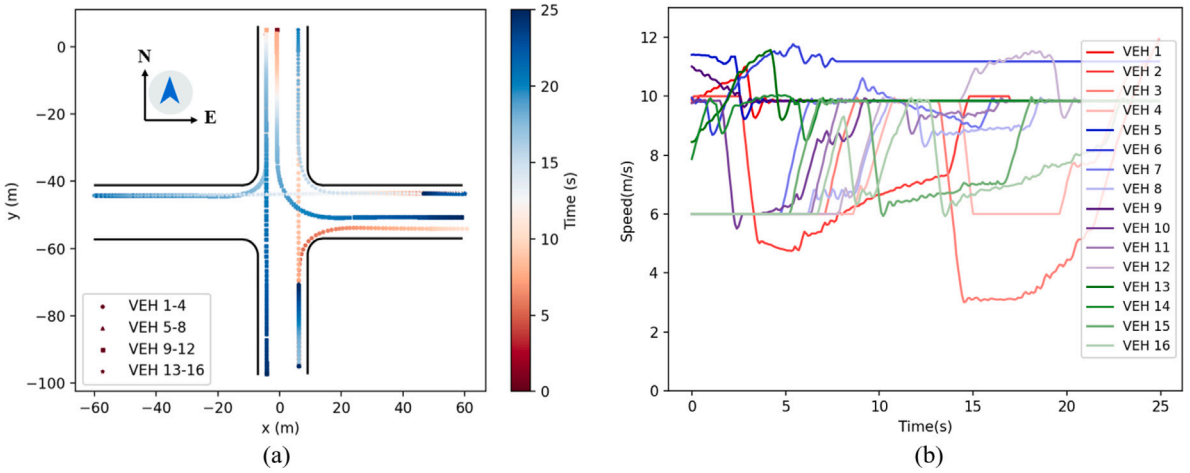


Fig. 14. Visualization of slightly below-average difficulty traffic scenario: (a) represents the traffic trajectory of vehicles, and (b) represents the speed curve of vehicles.

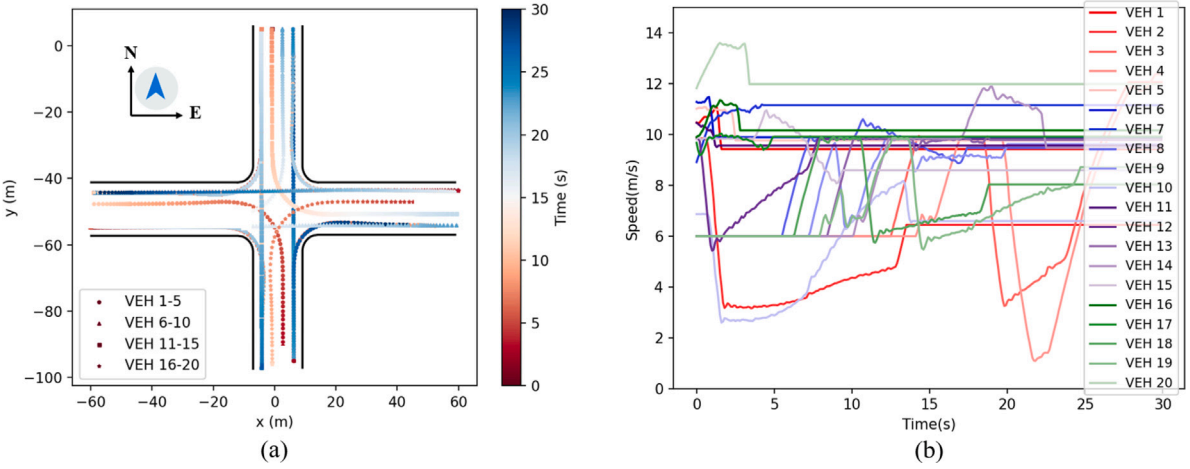


Fig. 15. Visualization of slightly above-average difficulty traffic scenario: (a) represents the traffic trajectory of vehicles, and (b) represents the speed curve of vehicles.

minimal. After the other vehicles have passed and the collision risk is eliminated, VEH2 accelerates and passes through while maintaining a safe TTC. As shown in Fig. 15 (b), in the slightly above-average difficulty scenario with many conflict points, at the beginning, all vehicles, except for VEH1 and VEH10 which decelerate to avoid collision risks, maintain relatively high speeds while passing through the intersection. After VEH1 and VEH10 eliminate collision risks with other vehicles, they begin to accelerate and pass through the intersection quickly, while maintaining a safe TTC. From Fig. 15 (a), it can be observed that VEH10 decelerates to avoid left-turning vehicles coming from the south to the west, while VEH1 decelerates to avoid left-turning vehicles coming from the east to the south. As VEH4 reaches the intersection last, it decelerates and waits before passing through, and after eliminating collision risks with other vehicles, VEH4 quickly passes the intersection. Multiple scenario verifications confirm that the SRL-CLCADI method demonstrates a high road utilization rate, further validating the efficiency and safety of the SRL-CLCADI method.

6. Conclusion and future outlook

This paper introduces the SRL-CLCADI system, a safety reinforcement learning-based approach for vehicle-road cooperative control system. This system collects trajectory information through interaction with the intersection environment and updates the cooperative control policy using the proposed MAPCPO algorithm. Notably, the paper categorizes the conditions of policy updates based on safety levels and incorporates cost value projection steps in corresponding safety levels to coordinate the constraints of cost values. Additionally, the policy neural network in this paper adopts dynamic inputs to more closely resemble real-world intersection scenarios and includes lateral control of vehicles in its output, thereby enhancing the algorithm's performance ceiling. Experiments demonstrate that the proposed SRL-CLCADI outperforms both traditional baseline methods and current state-of-the-art reinforcement learning approaches in terms of performance. While our system exhibits superior performance, the development of cooperative control policies for intersections still holds significant potential. In the future, we aim to improve the structure of the policy neural network to enhance its capability in capturing historical state information and social interaction data, thereby raising its performance ceiling. Furthermore, we will explore more rational update methods to increase the efficiency and potential of policy updates.

Declaration of competing interest

The corresponding author, on behalf of all authors of this submission, hereby declares that there are no financial or personal relationships with other people or organizations that could inappropriately influence our work. This declaration encompasses but is not limited to employment, consultancies, stock ownership, honoraria, paid expert testimony, patent applications/registrations, and grants or other funding. The authors declare that they have no competing interests.

Acknowledgment

All authors approved the final version of the manuscript.

Data availability

Data will be made available on request.

References

Abboud, K., Omar, H. A., & Zhuang, W. (2016). Interworking of DSRC and cellular network technologies for V2x communications: A survey. *IEEE Transactions on Vehicular Technology*, *65*(12), 9457–9470.

Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In *International conference on machine learning* (pp. 22–31). PMLR.

Al-Sharman, M., Dempster, R., Daoud, M. A., Nasr, M., Rayside, D., & Melek, W. (2023). Self-learned autonomous driving at unsignalized intersections: A hierarchical reinforced learning approach for feasible decision-making. *IEEE Transactions on Intelligent Transportation Systems*, *24*(11), 12345–12356.

Antonio, G.-P., & Maria-Dolores, C. (2022). Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections. *IEEE Transactions on Vehicular Technology*, *71*(7), 7033–7043.

Bichiou, Y., & Rakha, H. A. (2018). Developing an optimal intersection control system for automated connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *20*(5), 1908–1916.

Dai, P., Liu, K., Zhuge, Q., Sha, E. H.-M., Lee, V. C. S., & Son, S. H. (2016). Quality-of-experience-oriented autonomous intersection control in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, *17*(7), 1956–1967.

Dresner, K., & Stone, P. (2006). Human-usable and emergency vehicle-aware control policies for autonomous intersection management. In *Fourth international workshop on agents in traffic and transportation (ATT), Hakodate, Japan, vol. 12* (p. 14).

Fajardo, D., Au, T.-C., Waller, S. T., Stone, P., & Yang, D. (2011). Automated intersection control: Performance of future innovation versus current traffic signal control. *Transportation Research Record*, *2259*(1), 223–232.

Guan, Y., Ren, Y., Li, S. E., Sun, Q., Luo, L., & Li, K. (2020). Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization. *IEEE Transactions on Vehicular Technology*, *69*(11), 12597–12608.

Guo, Y., Ma, J., Xiong, C., Li, X., Zhou, F., & Hao, W. (2019). Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation Research Part C: Emerging Technologies*, *98*, 54–72.

Hafner, M. R., Cunningham, D., Caminiti, L., & Del Vecchio, D. (2013). Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1162–1175.

Hang, P., Huang, C., Hu, Z., & Lv, C. (2022). Driving conflict resolution of autonomous vehicles at unsignalized intersections: A differential game approach. *IEEE/ASME Transactions on Mechatronics*, *27*(6), 5136–5146.

He, Z., Zheng, L., Lu, L., & Guan, W. (2018). Erasing lane changes from roads: A design of future road intersections. *IEEE Transactions on Intelligent Vehicles*, *3*(2), 173–184.

Hu, C., Hudson, S., Ethier, M., Al-Sharman, M., Rayside, D., & Melek, W. (2022). Sim-to-real domain adaptation for lane detection and classification in autonomous driving. In *2022 IEEE intelligent vehicles symposium* (pp. 457–463). IEEE.

Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., et al. (2020). Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, *33*, 15931–15941.

Isele, D., Rahimi, R., Cosgun, A., Subramanian, K., & Fujimura, K. (2018). Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE international conference on robotics and automation* (pp. 2034–2039). IEEE.

Kamal, M. A. S., Imura, J.-i., Hayakawa, T., Ohata, A., & Aihara, K. (2014). A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. *IEEE Transactions on Intelligent Transportation Systems*, *16*(3), 1136–1147.

Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., & How, J. P. (2009). Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, *17*(5), 1105–1118.

Li, Z., Gong, J., Lu, C., & Yi, Y. (2021). Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework. *IEEE/ASME Transactions on Mechatronics*, *26*(3), 1339–1349.

Li, X., Sun, Z., Cao, D., He, Z., & Zhu, Q. (2015). Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE/ASME Transactions on Mechatronics*, *21*(2), 740–753.

Li, N., Yao, Y., Kolmanovsky, I., Atkins, E., & Girard, A. R. (2020). Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 1428–1442.

Li, B., & Zhang, Y. (2018). Fault-tolerant cooperative motion planning of connected and automated vehicles at a signal-free and lane-free intersection. *IFAC-PapersOnLine*, *51*(24), 60–67.

Li, B., Zhang, Y., Zhang, Y., Jia, N., & Ge, Y. (2018). Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection. In *2018 IEEE intelligent vehicles symposium* (pp. 1432–1437). IEEE.

Lu, Q., & Kim, K.-D. (2018). A mixed integer programming approach for autonomous and connected intersection crossing traffic control. In *2018 IEEE 88th vehicular technology conference (VTC-fall)* (pp. 1–6). IEEE.

Lukose, E., Levin, M. W., & Boyles, S. D. (2019). Incorporating insights from signal optimization into reservation-based intersection controls. *Journal of Intelligent Transportation Systems*, *23*(3), 250–264.

Mirheli, A., Hajibabai, L., & Hajbabaie, A. (2018). Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment. *Transportation Research Part C (Emerging Technologies)*, *92*, 412–425.

Mirheli, A., Tajalli, M., Hajibabai, L., & Hajbabaie, A. (2019). A consensus-based distributed trajectory control in a signal-free intersection. *Transportation Research Part C: Emerging Technologies*, *100*, 161–176.

Qian, X., Altché, F., Grégoire, J., & de La Fortelle, A. (2017). Autonomous intersection management systems: Criteria, implementation and evaluation. *IET Intelligent Transport Systems*, *11*(3), 182–189.

Schulman, J. (2015). Trust region policy optimization. arXiv preprint arXiv:1502.05477.

Tian, R., Li, N., Kolmanovsky, I., Yildiz, Y., & Girard, A. R. (2020). Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2211–2226.

Wu, Y., Chen, H., & Zhu, F. (2019). DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transportation Research Part C (Emerging Technologies)*, *103*, 246–260.

Xu, H., Zhang, Y., Li, L., & Li, W. (2019). Cooperative driving at unsignalized intersections using tree search. *IEEE Transactions on Intelligent Transportation Systems*, *21*(11), 4563–4571.

Yang, T.-Y., Rosca, J., Narasimhan, K., & Ramadge, P. J. (2020). Projection-based constrained policy optimization. arXiv preprint arXiv:2010.03152.

Zhao, J., Knoop, V. L., & Wang, M. (2023). Microscopic traffic modeling inside intersections: interactions between drivers. *Transportation Science*, *57*(1), 135–155.