Multi-Agent Constrained Policy Optimization for Conflict-Free Management of Connected Autonomous Vehicles at Unsignalized Intersections

Rui Zhao, *Member, IEEE*, Yun Li, *Member, IEEE*, Fei Gao, Zhenhai Gao, *Member, IEEE*, and Tianyao Zhang

***Abstract*— Autonomous Intersection Management (AIM) systems present a new paradigm for conflict-free cooperation of connected autonomous vehicles (CAVs) at road intersections, the aim of which is to eliminate collisions and improve the traffic efficiency and ride comfort. Given the challenges of current centralized coordination methods in balancing high computational efficiency and robust safety assurance, this paper proposes an innovative conflict-free management scheme for CAVs at unsignalized intersections, leveraging safe multi-agent deep reinforcement learning (MADRL). Firstly, we formulate the safe MADRL problem as a constrained Markov game (CMG) and then transform the AIM problem into a CMG by carefully designing state, action, reward, and cost functions. Subsequently, we propose the Multi-Agent Constrained Policy Optimization (MACPO), specifically tailored to solve the CMG problem. MACPO incorporates safety constraints that further restrict the trust region formed by the Kullback-Leibler (KL) divergence, facilitating reinforcement learning policy updates that maximize performance while keeping constraint costs within their limit bounds. This leads us to introduce the MACPO-based AIM Algorithm. Finally, we train an AIM policy and compare its computation time, ride comfort, traffic efficiency, and safety with management schemes based on Model Predictive Control (MPC), Mixed Integer Programming (MIP), and non-safety-aware reinforcement learning. According to the results, compared with the MPC and MIP methods, our method has increased computational efficiency by 65.22 times and 731.52 times respectively, and has improved traffic efficiency by 2.41 times and 1.80 times respectively. In contrast to the non-safety awareness RL methods, our method achieves a zero collision rate for the first time, while also enhancing ride comfort, highlighting the advantages of using MACPO.**

***Index Terms*— Conflict-free management, connected autonomous vehicles, safety reinforcement learning, multi-agent constrained policy optimization, unsignalized intersections.**

Manuscript received 30 January 2023; revised 25 July 2023 and 19 September 2023; accepted 7 November 2023. Date of publication 20 November 2023; date of current version 31 May 2024. This work was supported by the National Science Foundation of China under Grant 52202494 and Grant 52202495. The Associate Editor for this article was G. Wu. *(Corresponding author: Fei Gao.)*

Rui Zhao is with the College of Automotive Engineering, Jilin University, Changchun 130025, China (e-mail: rzhao@jlu.edu.cn).

Yun Li is with the Graduate School of Information and Science Technology, The University of Tokyo, Tokyo 113-8654, Japan (e-mail: li-yun@g.ecc.u-tokyo.ac.jp).

Fei Gao, Zhenhai Gao, and Tianyao Zhang are with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130025, China (e-mail: gaofei123284123@jlu.edu.cn; gaozh@jlu.edu.cn; tianyaz@jlu.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3331723

I. Introduction

With the significant enhancement of autonomous driving and internet of vehicles technology, vehicle-infrastructure collaboration has become a promising traffic management solution to provide safe, effective and comfortable transportation experience [1], [2]. In recent years, various vehicle-road collaborative applications have emerged successively [3], [4], [5]. As the particularly risky areas in urban environments, road intersections have drawn extensive attentions in dealing with serious traffic accident and severe congestion. Autonomous Intersection Management (AIM) systems are aimed to efficiently manage multi-connected autonomous vehicles (CAVs) at intersections, eliminate collisions, and optimize overall traffic efficiency as well as ride comfort [6]. Traditionally, these AIMs handle potential conflicts based on control strategies such as rule-based, optimization-based or machine learning-based methods to prevent anticipated conflicts from occurring [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35].

In AIM systems, conflict identification methods typically include tile-based, conflict point-based, and vehicle-based approaches. The initial method produces a grid within the intersection, ensuring vehicles' trajectories do not coincide within the same grid cell simultaneously. This concept was advanced by Dresner and Stone [7] who introduced a reservation-based approach where each tile in a finely segmented grid could be claimed by only a single vehicle per time unit. Vehicles would request a specified arrival time, securing tiles along their intended path. This reservation-based model was reformulated by Yu et al. [8] into an optimization-based approach. Concurrently, Dai et al. [9] developed an intersection control model, linearizing the grid cell conflict avoidance constraints. Xu et al. [10] focused on determining the optimal sequence for vehicle passage in AIM, transforming it into a tree-search problem. Alternatively, Wu et al. [11] proposed a decentralized coordination approach, promoting safety by limiting the number of grid cells into which a vehicle could move.

A different class of methods substitutes tiles with conflict points (CPs), representing intersections between varying turning paths. Kamal et al. [6] introduced the notion of a CP ensuring vehicles would not reach the CP simultaneously to

enhance vehicular safety. Zhang et al. [12] devised a scheduling mechanism for autonomous vehicles that is governed by state transitions. Meanwhile, He et al. [13] broadened the idea of CPs to encompass regions throughout the routes. Due to the spatial continuity of vehicle trajectories, in the collision avoidance method consifering CPs, building a mathematical prediction model is also more common. For instance, Kamal et al. [6] introduced a vehicle-intersection coordination scheme (VICS) based on a model predictive control (MPC) framework, enforcing CP avoidance constraints. Likewise, Katriniok et al. [14] employed a distributed MPC scheme to tackle the AIM problem. Chen et al. [15] implemented a graph-based approach, managing vehicle dispatch from various lanes and preventing simultaneous approaches to CPs. To simplify scheduling mechanism, Wang et al. [16] exploited detecting zone and control zone for vehicle motion control. Additionally, Yao et al. [17] designed a two-stage method to optimize timing schedules and trajectories for CAVs at intersection that combines the tile-based and conflict point-based approaches. A similar scheme can also be found in [18].

The vehicle-based approach provides complete freedom of movement within the intersection, allowing vehicles to choose their route to their exit lane. Mirheli et al. [19] formulated the cooperative trajectory planning problem as mixed-integer non-linear programs that aim to minimize travel time of each vehicle, while avoiding near-crash conditions. He et al. [13] addressed this by formulating the AIM problem as an optimal control model, improving computational efficiency through discretization and enumeration of variable values.

When conflicts are encountered, various control strategies intervene by influencing the state of vehicles, such as speed, acceleration, and route, and manage the right-of-way for vehicles at intersections. These strategies encompass a range of methods, including rule-based, optimization-based, and machine learning-based approaches. In rule-based methods, the vehicle that arrives at the intersection first, fastest, or is in the longest entry queue is given the highest priority, namely First-Come First-Served (FCFS) [20], [21], [22], Fast First Service [23], or Long Queue Priority [11]. However, such predefined policies are usually difficult to obtain optimal solutions for a highly dynamic AIM system where the traffic environment changes over time [24]. As pointed out by Levin et al. [25], FCFS based AIM may not outperform traffic signals in terms of travel time and emission.

The optimization-based method emerged to bridge the limitations of rule-based methods and also changed the way of the reservation to the assignment. These methods using some heuristics like Dynamic Programming or Mixed Integer Programming (MIP) where given a series of equations and conditions is used to solving AIM problem [6], [8], [13], [14], [15], [16], [17], [18], [19], [26], [27], [28], [29], [30], [31]. In the context of optimal control theory, Bichiou et al. [26] crafted an intersection control system that assigns time slots to each vehicle, adhering to both dynamic and static system constraints. Lu et al. [27] proposed a MIP-based Intersection Coordination Algorithm (MICA) to obtain the fastest crossing discrete-time trajectories for the CAVs at intersection. To enhance solution efficiency, the AIM problem is framed as a Mixed Integer Linear Programming problem [8], [17], [18], [19], [28]. Moreover, Choi et al. [29] and Xu et al [30] employed the genetic algorithm to filter the optimal vehicle passing order. Xu et al. [31] crafted trajectory planning algorithms to determine the optimal passing order using Monte Carlo tree. However, due to the long inference time associated with such methods, coupled with the fact that time consumption for solving these problems escalates almost exponentially with increasing complexity, makes it challenging to meet the real-time control demands when these methods are deployed at complex intersections under high traffic density conditions [32].

The latest advancements in machine learning, especially Deep Reinforcement Learning (DRL), are gradually used for optimizing traffic management [24], [32], [33], [34], [35]. Multi-Agent Deep Reinforcement Learning (MADRL) successfully combines the advantages of Multi-Agent Reinforcement Learning algorithms and Deep Neural Networks, has the potential to solve low computation efficiency and suboptimal challenges encountered by the current AIM methods. In view of a reward function that describes the goal of the problem, RL agents explore their environments to learn optimal policies that maximize the sum of future rewards by trial and error. To enhance travel efficiency and safety at signalized intersections, Boukerche et al. [32] and Zhou et al. [33] employed Deep Q-learning and Deep Deterministic Policy Gradient algorithms to realize the optimized signal control, respectively, to achieve optimized signal control. Furthermore, Guan et al. [34] first used RL algorithm named model accelerated proximal policy optimization (MA-PPO) to automatically manage vehicles at unsignalized intersections, wherein the reward function was formulated as the summation of a substantial collision penalty for safety considerations, and a stepped penalty coupled with a large passing reward for efficiency. The balance between safety and efficiency was modulated by adjusting the weighting of their respective rewards. Similarly, Xu et al. [35] and Antonio et al. [24] employed a non-safety-aware DRL approach, driven by a singular reward function, to address the AIM problem.

Nonetheless, RL policies that are purely optimised for reward maximization are rarely applicable to safety-critical autonomous driving applications. This presents two issues: *i*) There is no perfect trade-off between "desired safety requirements" and "optimized system performance", and it can not be checked before running an RL algorithm. If the penalty is designed to be very small, the agent will learn unsafe actions. If the penalty is designed to be very severe, the agent may fail to learn anything. *ii)* A fixed trade-off, even one that leads to the best hazard-avoiding policy, does not account for a requirement to satisfy safety requirements throughout training and deployment. In the AIM system, the collaboration of CAVs should optimize other non-safety performances such as efficiency and comfort under the premise of ensuring safety, instead of combining safety and non-safety performance metrics into the same reward function to seek its maximization. Namely, safety is a prerequisite constraint, rather than an aspect of system performance.

In order to bridge the aforementioned gap, we propose a conflict-free management scheme for CAVs at unsignalized intersections using safe MADRL. Firstly, we formulate the safe MADRL problem as a Constrained Markov

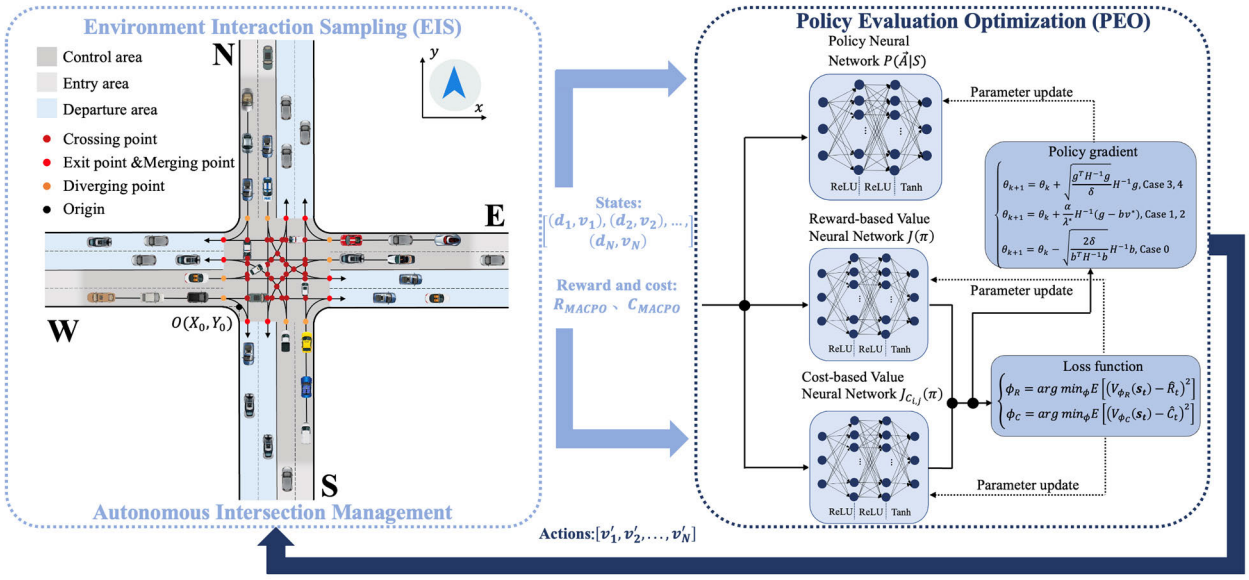


Fig. 1. Structure of the proposed MACPO-based AIM system, comprised of two parts - EIS and PEO, designed for the monotonic improvement of intersection throughput performance including safety, efficiency, and comfort, while concurrently satisfying safety cost constraints.

Game (CMG), and then present the design of state, action, reward, and cost to shape AIM into a CMG problem. Subsequently, we introduce multi-agent constrained policy optimization (MACPO) to tackle it. Inspired by the single-agent-oriented constrained policy optimization (CPO) [36], our approach integrates a safety constraint-satisfying eare, ensuring policy updates maximize performance without violating safety constraints. Lastly, we train a collaborative management policy in a simulation environment and benchmark its computing time, ride comfort, traffic efficiency, and safety against management schemes based on MPC and MIP methods, as well as a non-safety-aware RL method. Results indicate that, compared to MPC and MIP methods, our approach achieves 65.22 and 731.52 times higher computational efficiency, respectively, and traffic efficiency increases by 2.41 and 1.80 times. Additionally, in contrast to the non-safety-aware RL method, our method attains a zero collision rate along with enhanced passenger comfort.

The main contributions of this paper are summarized as follows:

1) Firstly, we introduce CMG, an extension of standard Markov Games, with embedded safety constraints restricting the set of acceptable policies. This framework serves as a formulation for the MADRL problem, necessitating safety considerations. We convert the AIM problem into a CMG one by defining CMG's essential elements. The reward function aims to improve traffic efficiency, comfort, and safety, while the cost function focuses on reducing potential collision risks and improving traffic safety.

2) Secondly, we present MACPO, a safety-aware MADRL approach, to solve the CMG problem. It incorporates safety-constrained to further constrain the trust region formed by Kullback-Leibler (KL) divergence, thus facilitating RL policy updates that maximize performance while keeping constraint costs within their limit bounds. Subsequently, we introduce the MACPO-based AIM algorithm to ensure the safety, efficiency, and ride comfort of CAVs' cooperative traffic behavior.

3) Thirdly, extensive simulation is conducted and comparisons with classical control methods VICS [6], MIP [27] and non-safety aware RL method MAPPO [34], showcasing the superiority of our method.

The structure of this paper is as follows: Section II clarifies our system model and problem statement for AIM. Section III explicates the framework of the MACPO-based AIM approach. Section IV introduces how to formulate the AIM problem via the CMG framework. Section V delves into the detailed exposition of the MACPO strategy and its application to the AIM system. Section VI details experimental settings and results. Lastly, Section VII concludes and discusses research future directions.

II. System Model and Problem Statement

The considered unsignalized road intersection scenario is depicted in Fig. 1, and its topology is a typical four-way intersection with multiple lanes. The intersection is systematically divided into three distinct zones: entry area, control area, and departure area. The AIM system exerts control over vehicles within a specified distance from the entrance of the intersection (i.e., the control area). The origin $O\,(X_0, Y_0)$ of the intersection coordinate system is positioned at the southwest corner, and the road width is denoted by D. Vehicles approaching the intersection have three maneuver options: continue straight, make a right turn, or make a left turn. Fig. 2 graphically illustrates all conceivable conflict relationships which can be categorized into three types: crossing conflicts (indicated by dark red dots), merging conflicts (represented by red dots), and diverging conflicts (marked by yellow dots).

Each vehicle $i\ \left(i \leq N_a, i \in N^+\right)$ periodically transmits its dynamic driving data $s_i = \left[x_i, y_i, v_i\right]^T$ to the AIM via V2I wireless communication. Here, $(x_i, y_i) \in \mathbb{R} \times \mathbb{R}$ represents the vehicle's position coordinates, while $v_i \in \mathbb{R}$ denotes the current velocity. Imbued with a safety-aware MADRL algorithm, MACPO, the AIM system modulates the timing of vehicles traversing road intersections by controlling each

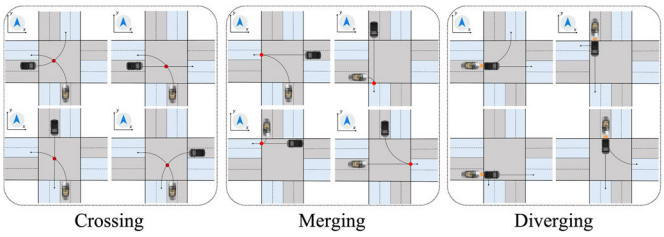


Fig. 2. Collision style.

vehicle's velocity v_i' in real-time within the range $[v_{min}, v_{max}]$, where v_{min} and v_{max} are the minimum and maximum velocities, respectively. Then, the motion control layer of each vehicle can generate the accelerator throttle opening and brake pad force required for the vehicle to travel according to the desired velocity v_i' given by the AIM system.

The problem is defined as follows: At each time step, given the static road information of the intersection and the dynamic states of all vehicles, including positions and velocities, the MACPO-based AIM system determines the joint action (i.e., desired velocities) of the vehicles in the control area in real-time. This system aims to coordinate a collision-free and efficient traffic flow for all vehicles at the intersection, simultaneously enhancing occupant comfort.

III. Framework of the Autonomous Intersection Management Based on MACPO

The proposed MACPO-based AIM approach sets up three neural networks: the policy neural network, along with the reward and cost-based 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 cost-based value networks are used to evaluate the expected performance of reward and safety cost under the current policy. Fig. 1 illustrates the MACPO-based AIM framework, which is comprised of two parts: Environment Interaction Sampling (EIS) and Policy Evaluation Optimization (PEO).

The EIS is tasked with acquiring updated RL policy and value network neural parameters, then employing these parameters to sample experience data from a road intersection environment. This component employs a CMG, a safe MADRL formulation, to formally express the process where multiple vehicles with safety constraints explore within the AIM system's environment. This process subsequently generates discrete time-series trajectory data, inclusive of states, actions, rewards, and safety costs. Such data serve as the foundation for optimizing the policy neural network and the reward and cost-based value neural networks.

The PEO operates in tandem with the EIS, utilizing the data collected from the EIS to update the policy and value neural networks. It then synchronizes the updated parameters with the EIS for the next iteration of sampling and optimizing in a cyclical process that continues until the desired intersection traffic performance is attained. This component features a MACPO method that solves the CMG problem by incorporating safety-constrained costs to further constrain the trust region formed by KL divergence. This approach facilitates policy updates to maximize AIM performance while ensuring that constraint costs are kept within pre-defined cost bounds.

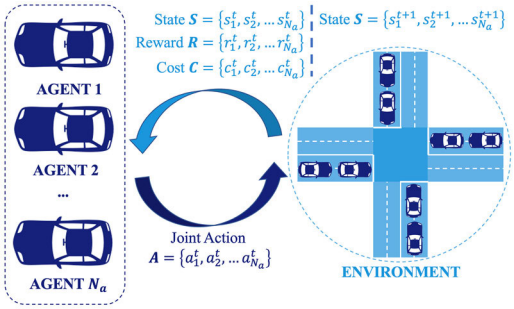


Fig. 3. CMG paradigm, including global state space, joint action space, joint reward function, and safety cost function set.

IV. Transforming AIM Into a Safe MADRL Problem for Environment Interaction Sampling

This section initially formulates the safe MADRL problem as the CMGs, and then transforms the AIM problem into a CMG problem by defining the fundamental elements of CMG, which includes the global state space, joint action space, joint reward function, and safety cost function. Consequently, the proposed safety-aware MADRL method, MACPO, can be effectively deployed to address the AIM problem.

A. Constrained Markov Games

Markov games (MGs), widely recognized for formally expressing the process by which multiple agents traverse the environment, fundamentally fall under the category of a discrete-time decision-making architecture. Within the MG framework, multiple agents engage in gameplay for cooperative or competitive objectives within the global environment, aiming to achieve the maximum total expected rewards. However, when safety constraints are introduced, a standard MG proves inadequate for describing the environment. Therefore, this section introduces the concept of CMGs. This represents an extension of MGs, integrated with constraints that restrict the set of allowable policies, as illustrated in Fig. 3. A CMG for N_a agents is defined by a tuple $\{\mathcal{S}, \mathcal{A}, R, C, P, \mu\}$, where

- $\mathcal{S}$ represents the global state space, consisting of the concatenation of the local states observed by all agents and an optional global non-redundant state;
- $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_{N_a})$ represents the set of joint action space of all agents, where each agent i performs an action $a_t^i \in \mathcal{A}^i$ at the discrete time step t;
- $R : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_{N_a} \times \mathcal{S} \to \mathbb{R}$ represents the joint reward function that describes the instant reward from a state s_t by taking a joint action $\boldsymbol{a_t}$ to the next state s_{t+1};
- $\mathcal{C} = \{C_i^j\}_{i=1,\ldots,N_a, j=1,\ldots,N_{i,c}}$ represents the set of cost functions defined by the specific environment safety constraints (each agent has $N_{i,c}$ cost functions), $C_i^j : S_i^t \times A_i^t \times S_i^{t+1} \to \mathbb{R}$ maps the transition tuples to safety cost with thresholds $d_i^1, d_i^2, \ldots, d_i^{N_{i,c}}$;
- $P : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_{N_a} \times \mathcal{S} \to [0, 1]$ represents the transition probability distribution from a state $s_t \in \mathcal{S}$ by taking a joint action $\boldsymbol{a_t} \in \mathcal{A}$ to the next state $s_{t+1} \in \mathcal{S}$ at the discrete time step t;
- μ: $S \to$ [1,0] represents the initial state distribution.

Following the CMG model, the agents interact with the environment in discrete time steps. At each time step t, the agents produce state $s_t \in \mathcal{S}$ by interacting with the environment, and each agent i performs an action $a_t^i \sim \pi(\cdot|s_t)$ according to the joint policy π. After all agents have taken their joint action $\boldsymbol{a}_t = (a_t^1, a_t^2, \ldots, a_t^{N_a})$, they receive a reward $R(s_t, \boldsymbol{a}_t, s_{t+1})$ and their respective cost $C_i^j(s_t, a_t^i, s_{t+1})$. The environment then transits to a new state $s_{t+1} \sim P(\cdot|s_t, \boldsymbol{a}_t)$. Upon reaching a terminal state, the agents start a new episode with an arbitrary state $s_0 \sim \mu$. When the trajectory $\tau = (s_0, \boldsymbol{a}_0, s_1, \ldots)$ from an epoch is gathered, the policy undergoes an update, subsequently enabling the agents to continue interaction with the environment utilizing this newly updated policy. The purpose of safety MADRL is to enable the agents to learn the opimal policy π^* that maximizes the expected return of rewards while keeping constraint costs within their limit bounds, through continuous policy updates.

Let $J_{C_i^j}(\pi)$ denotes the expected discounted return $J_{C_i^j}(\pi)$ of policy π with respect to cost function C_i^j is:

$$J_{C_i^j}(\pi) = E_{\tau\sim\pi}[\sum\nolimits_t \gamma^t C_i^j(s_t, \boldsymbol{a_t}, s_{t+1})] \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor. With the above conditions, the feasible policy set of the CMG model is:

$$\Pi_C \triangleq \left\{\pi \in \Pi : \forall i, j, J_{C_i^j}(\pi) \leq d_i^j\right\} \tag{2}$$

Given that the expected discounted return of policy π with respect to reward function is $J(\pi) = E_{\tau\sim\pi}\sum_t \gamma^t R(s_t, \boldsymbol{a}_t, s_{t+1})$, the optimal policy π^* with the largest expectation value of the reward function under the CMG model is:

$$\pi^* = arg\ max_{\pi\in\Pi_C} J(\pi) \tag{3}$$

B. Transforming AIM Into a Safe MADRL Formulation via CMG

1) State and Action Space: Given the dynamic driving information of multiple vehicles entering the intersection, the AIM system determines the desired velocities of vehicles in real time. Intuitively, the state space needs to include the speed information of each vehicle and its position information at the intersection, usually marked by the horizontal and vertical coordinates (x_i, y_i), $i \in [1, \ldots, N_a]$. In the intersection scenario, each vehicle has a fixed lateral trajectory corresponding to its traffic pattern, and reducing the redundant state space dimension can help improve the efficiency and stability of RL. Thus, this paper sets the state space of the vehicles as $\boldsymbol{S} = [d_1, d_2, \ldots, d_{N_a}, v_1, v_2, \ldots, v_{N_a}]$, where d_i represents the distance of the i-th vehicle to the exit point of the intersection, and v_i represents the vehicle's velocity. Beyond that, the action space is set to the desired velocities $\boldsymbol{A} = \left[v_1', v_2', \ldots, v_N'\right]$ of all vehicles.

In order to obtain more accurate environmental information, this paper calculates the distance from the exit point of the intersection for vehicles with different driving directions.

For the Left Turn Situation: When the vehicle has not yet entered the intersection, as shown in Fig. 4 (a), the distance d_{left}^1 from the vehicle to the exit point of the intersection can be expressed as:

$$d_{left}^1 = d_{ent} + d_{turn} = Y_0 - y_i + \frac{\pi}{2} \times \frac{5D}{8} \tag{4}$$

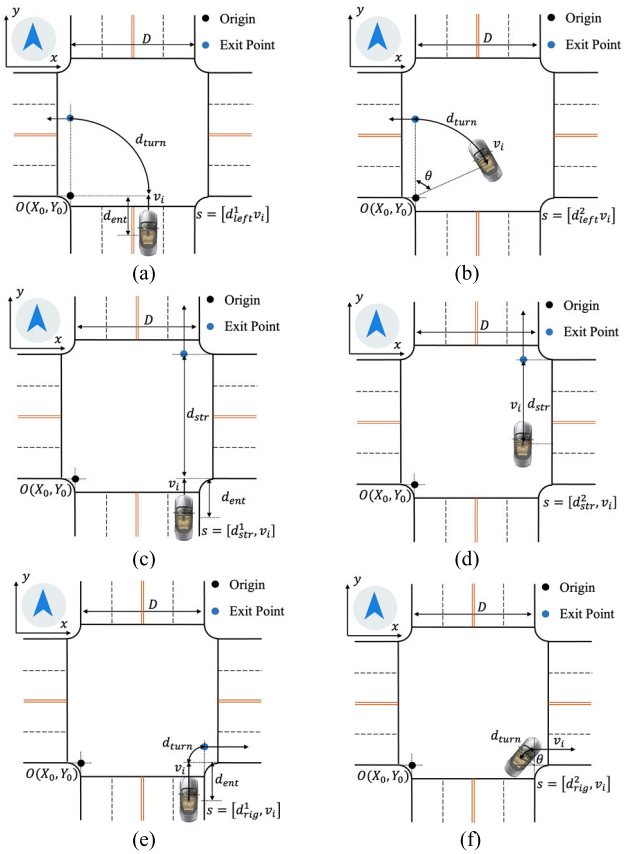


Fig. 4. Distances from the exit point of the intersection for vehicles with different driving directions. (a)-(b) Left Turn Situation; (b)-(c) Straight Situation; (c)-(d) Right Turn Situation.

where d_{ent} represents the distance from the vehicle's current location to the entrance point of the intersection, and d_{turn} represents the distance from the entrance point of the intersection to the exit point of the intersection. Y_0 the y-coordinate of the origin, y_i is the location of vehicle i in y-axis. When the vehicle has entered the intersection, as shown in Fig. 4 (b), the distance d_{left}^2 from the vehicle to the exit point of the intersection can be expressed as:

$$d_{left}^2 = d_{turn} = arcsin\left(\frac{8\,(x_i - X_0)}{5D}\right) \times \frac{5D}{8} \tag{5}$$

For the Straight Situation: When the vehicle has not yet entered the intersection, as shown in Fig. 4 (c), the distance d_{str}^1 from the vehicle to the exit point of the intersection can be expressed as:

$$d_{str}^1 = d_{ent} + d_{str} = Y_0 - y_i + D \tag{6}$$

where d_{str} represents the distance from the entrance point of the intersection to the exit point of the intersection. When the vehicle has entered the intersection, as shown in Fig. 4 (d), the distance d_{str}^2 from the vehicle to the exit point of the intersection can be expressed as:

$$d_{str}^2 = d_{str} = Y_0 - y_i + D \tag{7}$$

For the Right Turn Situation: When the vehicle has not yet entered the intersection, as shown in Fig. 4 (e), the distance

d^1_{rig} from the vehicle to the exit point of the intersection can be expressed as:

$$d^1_{rig} = d_{ent} + d_{turn} = Y_0 - y_i + \frac{\pi}{2} \times \frac{D}{8} \tag{8}$$

When the vehicle has entered the intersection, as shown in Fig. 4 (f) the distance d^2_{rig} from the vehicle to the exit point of the intersection can be expressed as:

$$d^2_{rig} = d_{turn} = arcsin\left(\frac{8\,(X_0 + D - x_i)}{D}\right) \times \frac{D}{8} \tag{9}$$

where X_0 is the location of point O in x-axis and x_i is the location of vehicle i in x-axis.

2) Reward and Cost Settings: Following the CMG framework, we set up not only reward function that characterizes the degree of overall performance optimization but also cost function that characterizes the degree of system safety performance to ensure that policy updates could improve overall performance most without violating safety constraints of each individual CAV and their joint traffic behaviour.

In order to better evaluate the quality of the policy, we comprehensively consider dense and sparse evaluation items, including dense and sparse reward functions as well as dense and sparse cost functions. The dense evaluation item is used to evaluate the performance at each timestep, like the velocities, accelerations of CAVs. While the sparse evaluation item is used to evaluate the performance of the vehicle at each episode, and it is only triggered when some special events occur. For instance, in a case of collision (terminal state), all vehicles pass through the intersection successfully (terminal state) or a certain vehicle passes through the intersection (intermediate state).

The cost function emphasizes the traffic safety and the potential collision risk. For this reason, this paper designs the collision safety distance threshold c_s. When the distance between two vehicles with the possibility of collision is less than c_s, the value of the cost function is increased by ε_R. Therefore, the dense loss function C_d can be defined as:

$$C_d = \sum\nolimits_{t=1}^{N_t} \sum\nolimits_{i=1}^{N_a} \sum\nolimits_{i'=1}^{N_a - 1} \varepsilon_R \delta_{risk_{t,i,i'}} \tag{10}$$

where $\delta_{risk_{t,i,i'}} = 1$ indicates that there is a collision risk between the i-th vehicle and the i'-th vehicle at the t-th time step, otherwise $\delta_{risk_{t,i,i'}} = 0$. At the same time, the value of the cost function is increased by ε_c after collision, and the sparse cost function C_s can be defined as:

$$C_s = \varepsilon_c \delta_{collision} \tag{11}$$

where $\delta_{collision} = 1$ indicates that collision occurs, otherwise $\delta_{collision} = 0$. The total cost function C_{MACPO} is defined as the summation of the dense cost function C_d and the sparse cost function C_s.

$$C_{MACPO} = C_d + C_s \tag{12}$$

The reward function focuses on comprehensively measuring the traffic efficiency, safety and comfort of drivers and passengers at intersections. The reward function of AIM consists of two parts: dense and sparse rewards. The dense reward function R_d is set as:

$$R_d = \sum\nolimits_{t=1}^{N_t} \sum\nolimits_{i=1}^{N_a} \varepsilon_v v_{t,i} - \varepsilon_a a_{t,i} \tag{13}$$

where $v_{t,i}$ and $a_{t,i}$ represent the velocity and absolute acceleration of the i-th vehicle at time step t, respectively, and ε_v and ε_a represent weights. The sparse reward function R_s is defined as:

$$R_s = \sum\nolimits_{i=1}^{N_a} \varepsilon^1_p \delta_{pass\_single} + \varepsilon^2_p \delta_{pass\_all} \tag{14}$$

where $\delta_{pass\_single} = 1$ indicates that one single vehicle passes successfully, otherwise $\delta_{pass\_single} = 0$; $\delta_{pass\_all} = 1$ indicates that all vehicles pass successfully, otherwise $\delta_{pass\_all} = 0$, and ε^1_p and ε^2_p are weights. The total reward function R_{MACPO} is set as:

$$R_{MACPO} = R_d + R_s - C_{MACPO} \tag{15}$$

After transforming AIM into a safe MADRL formulation via CMG, in the road intersection environment as shown in Fig. 1, multiple vehicles obtain the mapping from state $\boldsymbol{S} = [d_1, d_2, \ldots, d_{N_a}, v_1, v_2, \ldots, v_{N_a}]$ to action $\boldsymbol{A} = \left[v'_1, v'_2, \ldots, v'_N\right]$ so as to continuously explore the environment. Furthermore, the policy neural network as well as the reward and cost-based value neural networks can be updated by collecting the entire epoch trajectory in the PEO component until the expected intersection traffic performance is achieved.

V. Autonomous Intersection Management Based on MACPO

In this section, we first introduce MACPO, a safety-aware MADRL policy optimization method designed to solve the CMG problem. We further propose the MACPO-based AIM algorithm, where MACPO, as the core of the PEO component, optimizes the policy neural network based on the CMG-style trajectory data of multiple vehicles collected by the EIS component. This ensures that policy updates can maximize AIM performance with minimum safety constraints violations.

A. Multi-Agent Constrained Policy Optimization

Policy optimization algorithm solves the CMG problem by searching for the optimal feasible policy within a set Π_θ of parametrized policies with neural network parameters θ, and the policy is iteratively updated by maximizing the expected discounted reward return $J(\pi)$ over a local neighborhood $\Pi_\theta \cap \Pi_C$ of the most recent iteration π_k:

$$\begin{aligned} \pi_{k+1} = & \; argmax_{\pi \in \Pi_\theta} J(\pi) \\ & s.t.\; J_{C^j_i}(\pi) \le d^j_i,\;\; i = 1, \ldots, N_a,\;\; j = 0, \ldots, N_{i,c} \\ & D_{KL}(\pi \| \pi_k) \le \delta. \end{aligned} \tag{16}$$

This update is difficult to implement in practice, as the safety constraint functions need to be evaluated to determine whether a policy π is feasible. The off-policy evaluation is known to be challenging when constrained policy updates are computed using samples collected over π_k. In this work, we have replaced the objective and safety constraints with easy-to-evaluate surrogate functions developed by CPO [36] so as to obtain a good local approximation to Equation (16).

The on-policy value function for reward, expressed as $V^\pi(s) = E_{\tau\sim\pi}[R(\tau_\pi)|s_0 = s]$, represents the expected return starting from state s under policy π. The action-value function, denoted as $Q^\pi(s, \boldsymbol{a}) = E_{\tau\sim\pi}\left[R(\tau_\pi)|s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}\right]$, represents the expected return starting from state s,

taking the joint action $\boldsymbol{a}$, and following policy π thereafter. The advantage function, denoted as $A^{\pi}(s,\boldsymbol{a})$, measures the advantage of taking action a relative to the mean. It is defined as the difference between the action-value function and the value function: $A^{\pi}(s,\boldsymbol{a})Q^{\pi}(s,\boldsymbol{a}) - V^{\pi}(s)$. Similarly, the on-policy value functions, action-value functions, and advantage functions for the auxiliary costs are defined as $V^{\pi}_{C_i^j}$, $Q^{\pi}_{C_i^j}$, $A^{\pi}_{C_i^j}$, respectively. Let d^{π} denote the discounted future state distribution, defined by $d^{\pi}(s) = (1-\gamma)\sum_t \gamma^t P(s_t = s|\pi)$. Equation (16) is thus approximated as:

$$\begin{aligned}\pi_{k+1} = argmax_{\pi\in\Pi_\theta} & E_{\substack{s\sim d^{\pi_k}\\ a\sim\pi}}\left[A^{\pi_k}(s,\boldsymbol{a})\right] \\ s.t.\ & J_{c_i^j}(\pi_k) + \frac{1}{1-\gamma}E_{\substack{s\sim d^{\pi_k}\\ a\sim\pi}}\left[A^{\pi_k}_{c_i^j}(s,a)\right] \le d_i^j \quad \forall i,j \\ & \bar{D}(\pi\|\pi_k) \le \delta \end{aligned} \tag{17}$$

The above formula can increase the expected reward return and satisfy the specified constraints C_i^j. However, for neural network based policies with high-dimensional parameter spaces, Equation (17) may be impractical to solve directly due to the computational cost. Given that for a small step size δ, the objectives and safety constraints of policy π can be well approximated by a linear function around the current policy π_k, and the KL divergence constraint can be well approximated by a second-order expansion, Equation (17) is approximated as:

$$\begin{aligned}\theta_{k+1} = arg\ max_\theta & g^T(\theta-\theta_k) \\ s.t.\ & c_i^j + {b_i^j}^T(\theta-\theta_k) \le 0, \quad \forall i,j \\ & \frac{1}{2}(\theta-\theta_k)^T\boldsymbol{H}(\theta-\theta_k) \le \delta \end{aligned} \tag{18}$$

where $g \triangleq \nabla_\theta E\left[A^{\pi_k}(s,\boldsymbol{a})\right]$ represents the gradient of the objective, $b_i^j \triangleq \nabla_\theta E[A^{\pi_k}_{C_i^j}(s,\boldsymbol{a})]$ represents the gradient of safety constraint C_i^j, H represents Hessian of the KL-divergence, and $c_i^j \triangleq J_{c_i^j}(\pi_k) - d_i^j$ represents the difference between the expectation of the j-th safety constraint function C_i^j of the i-th agent under the policy π_k and its corresponding limit d_i^j.

Due to the fact that the Hessian matrix H in the formula is always positive semi-definite, the policy optimization problem is a convex. When the convex optimization problem is feasible, Equation (18) can be solved efficiently using duality:

$$\max_{\lambda\ge0,v\ge0} -\frac{1}{2\lambda}\left(g^T\boldsymbol{H}^{-1}g - 2r^Tv + v^TSv\right) + v^Tc - \frac{\lambda\delta}{2} \tag{19}$$

where

$$\begin{aligned} r \triangleq\ & g^T\boldsymbol{H}^{-1}\Big[b_1^1, b_1^2, \ldots, \\ & b_1^{N_{1,c}}, b_2^1, b_2^2, \ldots, b_2^{N_{2,c}}, \ldots, b_{N_a}^1, b_{N_a}^2, \ldots, b_{N_a}^{N_{N_a,c}}\Big], \\ S \triangleq\ & \Big[b_1^1, b_1^2, \ldots, b_1^{N_{1,c}}, b_2^1, b_2^2, \ldots, b_2^{N_{2,c}}, \ldots, b_{N_a}^1, \\ & b_{N_a}^2, \ldots, b_{N_a}^{N_{N_a,c}}\Big]^T \\ & \times \boldsymbol{H}^{-1}\Big[b_1^1, b_1^2, \ldots, b_1^{N_{1,c}}, b_2^1, b_2^2, \ldots, b_2^{N_{2,c}}, \ldots, b_{N_a}^1, \\ & b_{N_a}^2, \ldots, b_{N_a}^{N_{N_a,c}}\Big], \end{aligned}$$

the total number of cost function is $N_m = \sum_i N_{i,c}$. When the number of constraints is small by comparison to the dimension of neural network parameters θ, this is much easier to solve than (17). If λ^*, v^* are a solution to the dual, the solution to the primal is:

$$\begin{aligned} \theta_{k+1} = \theta_k + & \frac{1}{\lambda^*}\boldsymbol{H}^{-1}g - \frac{v^*}{\lambda^*}\boldsymbol{H}^{-1}\Big[b_1^1, b_1^2, \ldots, b_1^{N_{1,c}}, b_2^1, b_2^2, \\ & \ldots, b_2^{N_{2,c}}, \ldots, b_{N_a}^1, b_{N_a}^2, \ldots, b_{N_a}^{N_{N_a,c}}\Big] \\ v^* = & \left(\frac{\lambda^* c_i^j - r}{S}\right)_+ \\ \lambda^* = & argmax_{\lambda\ge0}\begin{cases} f_a(\lambda) \triangleq \frac{1}{2\lambda}\left(\frac{r^2}{S} - q\right) + \frac{\lambda}{2}\left(\frac{{c_i^j}^2}{S} - \delta\right) \\ -\frac{rc_i^j}{S}, \qquad\qquad if\ \lambda c_i^j - r > 0 \\ f_b(\lambda) \triangleq -\frac{1}{2}\left(\frac{q}{\lambda} + \lambda\delta\right), \\ \qquad\qquad else \end{cases} \end{aligned} \tag{20}$$

After acquisition of v^* and λ^*, the conjugate gradient method can be used to calculate the policy update direction x_k:

$$x_k = \boldsymbol{H}^{-1}\left(g - \hat{b}v^*\right) \tag{21}$$

The current policy can be updated by the following formula:

$$\theta_{k+1} = \theta_k + \frac{\alpha}{\lambda^*}x_k \tag{22}$$

where α is obtained by backtracking linear search.

When the convex optimization problem is infeasible, it indicates that it is impossible to find suitable neural network parameters to make the value of the safety cost function less than the preset limit while improving the reward value. Therefore, we turn to find the semi-optimal policy that minimize the safety cost. In order to obtain the global minimum safety cost function, we ignore the effects of the reward function during the policy updating:

$$\begin{aligned}\theta_{k+1} = arg\ min_\theta & \left[c_i^j + {b_i^j}^T(\theta-\theta_k)\right] \\ s.t.\ & \frac{1}{2}(\theta-\theta_k)^T\boldsymbol{H}(\theta-\theta_k) \le \delta \end{aligned} \tag{23}$$

Let the policy change $x = \theta - \theta_k$, the dual function of formula (23) is given by $L(x,\lambda) = c_i^j + {b_i^j}^T x + \lambda\left(\frac{1}{2}x^T\boldsymbol{H}x - \delta\right)$. The optimal solution for policy update is obtained when the partial derivative of the dual function with respect to policy change is 0:

$$\frac{L}{\partial x} = {b_i^j}^T + \lambda(\boldsymbol{H}x) = 0 \tag{24}$$

The obtained $x = -\frac{1}{\lambda}\boldsymbol{H}^{-1}{b_i^j}^T$ above should satisfy the trust-region constraint determined by KL-divergence:

$$\frac{1}{2}\left(-\frac{1}{\lambda}\boldsymbol{H}^{-1}{b_i^j}^T\right)^T\boldsymbol{H}\left(-\frac{1}{\lambda}H^{-1}{b_i^j}^T\right) \le \delta \tag{25}$$

Therefore, we have $\sqrt{\frac{b_i^{jT} H^{-1} b_i^j}{2\delta}} \leq \lambda$, and the update rule in case of in-feasibility is:

$$\theta_{k+1} = \theta_k - \sqrt{\frac{2\delta}{b_i^{jT} H^{-1} b_i^j}} \boldsymbol{H}^{-1} b_i^{jT} \tag{26}$$

Equations (22) and (26) are used to implement MACPO.

Algorithm 1 MACPO-based AIM

1: Initialize ϕ_R, ϕ_C, π, set $cost_d$, γ, kl_{max}, ρ, lr_r, lr_c, N_b, N_e, N_t, N_m, ζ, ϑ.
2: **for** epoch $k = 1, 2, \ldots, N_e$ **do**
3: **for** $t = 1, 2, \ldots, N_t$ **do**
4: For each vehicle $i = 1, 2, \ldots, N_a$, receive state $s_t = [d_i, v_i]$, choose an action $a_t = [v_i']$ according to current policy π_k.
5: Execute global actions $\boldsymbol{a_t}$, get reward r_t, cost $\mathbf{c_t}$, next state s_{t+1}, w.r.t. current policy and exploration in EIS.
6: $s_t = s_{t+1}$
7: **end for**
8: Collect trajectories $\boldsymbol{\tau_{\pi_k}} = (s_t, \boldsymbol{a_t}, r_t, \mathbf{c_t}, s_{t+1})$
9: Calculate advantage function of reward and cost function: $\hat{A}^{\pi}(s_t, \boldsymbol{a_t})$, $\hat{A}^{\pi}_{C_i^j}(s_t, \boldsymbol{a_t})$
10: Calculate $\hat{g}$, $\hat{b}$, $\hat{c}$, B: $\hat{g} = \nabla_\theta J(\pi)$, $\hat{b} = \nabla_\theta J_{C_i^j}(\pi)$, $\hat{c} = \sum_i \sum_j \hat{c}_i^j$, $B = \delta - \frac{\hat{c}^2}{\hat{b}^T \boldsymbol{H}^{-1} \hat{b}}$
11: **if** case 1 **or** case 2 **then**
12: update policy network as: $\theta_{k+1} = \theta_k - \sqrt{\frac{2\delta}{\hat{g}^T \boldsymbol{H}^{-1} \hat{g}}} \boldsymbol{H}^{-1} \hat{g}$
13: **elseif** case 3 **or** case 4 **then**
14: solve convex dual problem, get v^*, λ^*
15: solve α by backtracking line search, update policy network as: $\theta_{k+1} = \theta_k + \frac{\alpha}{\lambda^*} \boldsymbol{H}^{-1} \left(\hat{g} - \hat{b} v^*\right)$
16: **else** update policy network as: $\theta_{k+1} = \theta_k - \sqrt{\frac{2\delta}{\hat{b}^T \boldsymbol{H}^{-1} \hat{b}}} \boldsymbol{H}^{-1} \hat{b}$
17: **end if**
18: Update ϕ_R, ϕ_C as: $\phi_R = argmin_\phi E\left[\left(V_{\phi_R}(\boldsymbol{s_t}) - \hat{R}_t\right)^2\right]$, $\phi_C = argmin_\phi E\left[\left(V_{\phi_C}(\boldsymbol{s_t}) - \hat{C}_t\right)^2\right]$
19: **end for**

B. MACPO-Based AIM Algorithm

Based on the multi-agent constrained policy update method, we provide the following algorithm that guarantees both AIM monotonic performance improvement and safety cost constraints satisfaction. As shown in Fig. 1, the algorithm sets up three neural networks: the policy neural network and the reward and cost-based value neural networks. The input of the algorithm refers to the local states of all vehicles at the current time step. After mapping by the policy network, the output is the joint actions of the vehicles at the next time step. Given that the vehicle control parameters belong to the continuous action space, we adopt a diagonal Gaussian policy whose output is the mean $\mu_\theta(\mathbf{s})$ of the vehicles' actions. In the early stage of training, the exploratory requirements of the policy are stronger, and the stability requirement of the policy is stronger as the number of training iterations increases. Therefore, we define the dynamic attenuation standard deviation σ_θ to obtain better training effect and faster convergence speed:

$$\sigma_\theta = \zeta \times e^{\vartheta \times z}, \quad z = 1, 2, \ldots, N_T \tag{27}$$

where ζ and ϑ represent the coefficient and decrease index of the standard deviation σ_θ, repectively, z is the time step index, and N_T is the total number of time steps. Consequently, the joint action of the multi-agent is $\boldsymbol{a} = [\mu_\theta(s_1) + \sigma_\theta, \mu_\theta(s_2) + \sigma_\theta, \ldots, \mu_\theta(s_N) + \sigma_\theta]$. Our algorithm evaluates the performance of the current policy according to the reward and cost-based value network. Apart from that, a linear decay learning rate is used to update the reward and cost-based value networks:

$$\begin{aligned} lr_r &= lr_{r0} \times \frac{i}{E}, \quad i = [1, N_E] \\ lr_c &= lr_{c0} \times \frac{i}{E}, \quad i = [1, N_E] \end{aligned} \tag{28}$$

where N_E represents the number of epochs. In the process of continuously updating the reward and cost-based value network through the gradient descent, the policy is updated toward increasing the reward and decreasing the cost, and finally achieving the desired performance.

MACPO-based AIM algorithm is presented in algorithm 1. The first line of the algorithm is used to initialize network parameters and algorithm parameters, including the random reward-based value network ϕ_R, cost-based value network ϕ_C, and policy network π with θ, set $cost_d$, γ, maximum KL divergence kl_{max}, GAE coefficient ρ, value function learning rate lr_r, cost value function learning rate lr_c, batch size N_b, epoch N_e, total steps per batch N_t, total cost functions N_m, policy std ζ, policy std decrease index ϑ.

The main loop of the algorithm begins at the second line, which initially gathers trajectory data composed of state-action pairs for the multiple vehicles, as well as the reward and cost values in the EIS component (lines 2-8). Subsequently, within the PEO component (lines 9-19), the policy network and the reward and cost-based value networks are evaluated and updated separately.

For the policy network, in order to further improve the computational efficiency, our algorithm comprehensively considers the constraint satisfaction of current policy as well as the feasibility of the next policy, estimates the possibility of a policy update that causes the CAVs to be in a dangerous situation, and proposes the corresponding update solution (lines 9-17). More specifically, we define all feasible policies around the current policy π_k that satisfy the preset KL divergence and the safety constraints as the local policies to be searched (see the light blue ellipse area in Fig. 5). During the local policy search process, our algorithm decides how to update the policy by calculating the gradient of the safety constraint cost function, so that the policy can obtain the maximum reward function value as much as possible while satisfying both the safety constraint and the KL divergence constraint. Let $\hat{b}_i^j = \nabla_\theta J_{C_i^j}(\pi_k) = \nabla_\theta E[\sum_t \gamma^t C_i^j(s_i^t, a_i^t, s_i^{t+1})]$,

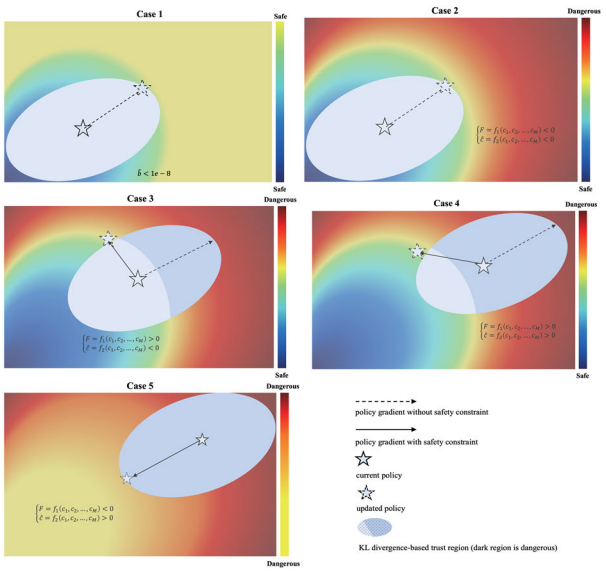


Fig. 5. Judgement of constraint satisfaction of current policy as well as the feasibility of the next step policy when updating the MACPO-based AIM algorithm.

$i \in [1, \ldots, N_a]$, $j \in [0, \ldots, N_{i,c}]$ denotes the policy gradient estimation of the safety constraint function C_i^j, $\hat{c}_i^j = E[\sum_t \gamma^t C_i^j(s_i^t, a_i^t, s_i^{t+1})] - \frac{d_i^j}{1-\gamma}$ denote the proximity of the agents' safety cost on constraint C_i^j to its limit d_i^j following current policy (also known as the dangerous potential energy), we define the overall estimations as $\hat{b} = \sum_i \sum_j \hat{b}_i^j$ and $\hat{c} = \sum_i \sum_j \hat{c}_i^j$, and use $F = \delta - \frac{\hat{c}^2}{\hat{b}^T \boldsymbol{H}^{-1} \hat{b}}$ to measure whether the next step policy of the current policy π_k is feasible (line 10). Based on the parameters $\hat{b}$, $\hat{c}$ and F, as shown in Fig. 5, the situation of a policy update that whether causes the CAVs to be in a dangerous situation is divided into the following cases:

Case 1: If the value of the safety gradient $\hat{b}$ is extremely small ($\hat{b} < 1e-8$), it indicates that the overall dangerous potential energy of the current policy is at an extremely low 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, which is labeled as Case 1.

Case 2: If $F < 0$ and $\hat{c} < 0$, the trust region constrained by the KL divergence is completely in the safety region, it indicates that the overall dangerous potential energy of the current policy is at a low to medium level (i.e. the overall safety cost is lower than the given threshold to varying degrees), but updating policies within KL divergence based trust region will not lead to a dangerous situation, which is labeled as Case 2.

Case 3: If $F > 0$ and $\hat{c} < 0$, and the trust region constrained by the KL divergence is divided into safe and non-safe parts by the safety constraints, it indicates that the policy update without safety constraints (the direction of the dotted line in the figure) will make the dangerous potential energy be at the high level, causing cost value exceeding the limit value and entering a dangerous situation, which is labeled as Case 3.

Case 4: If $F > 0$ and $\hat{c} > 0$, and the there is an intersection between trust domains satisfying KL divergence and safety constraints, it indicates that the current policy is already in an area with high dangerous potential energy, but the updated policy may return to a region with low dangerous potential energy, which is labeled as Case 4.

Case 5: If $F < 0$ and $\hat{c} > 0$, and there is no feasible policy region that satisfies both KL divergence constraints and safety constraint, it indicates that the overall dangerous potential energy of the current policy is at a high level, the updated policy cannot return to the region with low dangerous potential energy, which is labeled as Case 5.

Our algorithm performs corresponding policy updates for the above different cases. When case 1 or 2 is detected, given that none of the local policies in the trust region constrained by the KL-divergence will cause the agent to fall into a dangerous situation, the policy update can directly adopt the traditional Trust region policy optimization (TRPO) [37] method:

$$\theta_{k+1} = \theta_k - \sqrt{\frac{2\delta}{\hat{g}^T \boldsymbol{H}^{-1} \hat{g}}} \boldsymbol{H}^{-1} \hat{g} \tag{29}$$

When case 3 or 4 is detected, the policy update needs to satisfy both KL-divergence and safety constraints and there is a feasible policy region that satisfies both constraints. Therefore, Equation (22) derived above is used to update the current policy. When case 5 is detected, there is no feasible policy region that satisfies both KL divergence constraints and safety constraints. Hence, Equation (26) derived above is used to update the current policy.

For the reward and cost-based value networks, their gradients are used to update the respective network parameters (line 18):

$$\begin{aligned} \phi_R &= argmin_\phi E\left[\left(V_{\phi_R}(\boldsymbol{s_t}) - \hat{R}_t\right)^2\right] \\ \phi_C &= argmin_\phi E\left[\left(V_{\phi_C}(\boldsymbol{s_t}) - \hat{C}_t\right)^2\right] \end{aligned} \tag{30}$$

After updating the three neural networks, the algorithm employs the new policy network to collect trajectories in the environment again, and then evaluates and updates the neural networks until N_e epochs are completed.

VI. Performance Evaluation

A. Experiment Settings

In order to evaluate the proposed AIM system based on MACPO in experimental tests, two sets of experiments are performed in this section. To be specific, the first experiment illustrates the training process of MACPO, MAPPO proposed by Guan et al. [34] and MAPPO-SC with the same safety constraints as MACPO but in the form of reward function penalties, which shows that our algorithm has better overall performance, especially in terms of safety, and reaches zero collision for the first time. The second experiment compares the performance of MACPO-trained policy with MAPPO, MAPPO-SC, MPC-based method VICS [6] and MIP-based MICA [27] in terms of computation time, ride comfort, traffic efficiency, and safety.

All the experiments are performed in a simulation environment, in which Carla 0.9.12 is used to build the road intersection scenarios and the MADRL model is built based

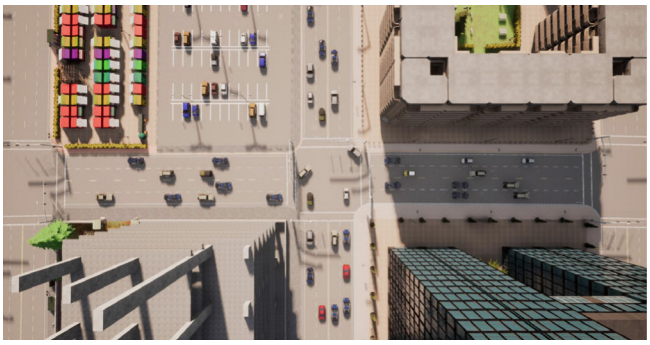


Fig. 6. Simulation Scenario.

on the PyTorch framework. Apart from that, CARLA's built-in sensors are used to transmit vehicle status information in real time, and the class BasicAgent is adopted to generate vehicle traffic trajectories at intersections. The expected vehicle speed output by the policy neural network is converted into the throttle control signal and brake control signal for controlling the vehicle through the built-in PID controller of CARLA. In addition, the GPU is NVIDIA GeForce RTX 3090, and the operating system is Ubuntu 18.04.

In this experiment, the four-way dual-lane signal-free intersection in CARLA TOWN 05 is chosen as the training and testing scenario for the RL model, as shown in Fig. 6. The road width and lane width are 14.2 meters and 3.5 meters, and the lengths of the East-West and North-South lanes (i.e., the departure areas) are 65 meters and 50 meters, respectively. Considering the characteristics of the roads in the CARLA map and the range of V2I communication, the lengths of the East-West and North-South control areas are set to 70 meters and 60 meters, respectively. To simulate real traffic flow, this experiment selects a variety of vehicle types in the simulation environment. The length range of the vehicles is 3.6-5.4 meters, the width range is 1.8-2.2 meters, the height range is 1.5-2 meters, and the arrival of vehicles is assumed to follow a Poisson distribution. Based on the set average traffic flow λ per hour for a single lane, a random number λ' is generated using the Poisson function in the numpy package. This number is used to calculate the time interval at which vehicles enter the intersection on this lane. Further, considering the free-driving speed of vehicles before they enter the control area, the distance between two adjacent vehicles is obtained. Subsequently, vehicle position coordinates are generated continuously, enabling the creation of a continuous traffic flow in CARLA that adheres to a Poisson distribution.

Consistent with the actual vehicle control cycle, the time step is set to 0.1 s. We all employ multiple-layer perceptron with two hidden layers as the approximate functions of the policy and the value functions for MACPO, MAPPO and MAPPO-SC algorithms, all of which have 128 units in each hidden layer. For the MACPO algorithm, we additionally employ a four-layer neural network with two hidden layers with 128 neurons as the cost function. The structure of the policy network is $2N_a \times 128 \times 128 \times N_a$, where N_a represents the number of vehicles that the AIM system needs to coordinate simultaneously, with this study supporting up to 60 vehicles. Beyond that, the structures of the reward and cost value networks are both $2N_a \times 128 \times 128 \times 1$. For each policy iteration, 2048 timesteps are collected, and the Adam optimizer is used for policy update. The learning rate decays linearly from 1e-3 to 0, the standard deviation decays exponentially from 1 to 0, and the training algorithm stops after 1024 epochs. Furthermore, the parameter settings of the MPC-based VICS algorithm and MIP-based MICA algorithms are the same as those in their original papers. The main parameters of the experiments are presented in Table I.

TABLE I
MAIN PARAMETERS OF THE EXPERIMENTS

Parameters	Value
CARLA Simulator	
Time-step $\boldsymbol{\tau}$	0.1 s
Control distance in the East-West direction	$70m$
Control distance in the North-South direction	$60m$
Road width and Lane width	$14.2m, 3.5m$
Center median	$0.2m$
MACPO & MAPPO-SC	
Discount factor $\boldsymbol{\gamma}$	0.99
Learning rate	1e-3→0 (linearly)
Max KL divergence $\boldsymbol{\delta}$	0.001
Damping coefficient	0.01
Timesteps $\boldsymbol{N_t}$	2048
Epoch $\boldsymbol{N_e}$	1024
Cost limit	1
Hidden layer number	2
Hidden layer units	128
Policy std $\boldsymbol{\sigma_\theta}$	1→0 (exponentially)
Policy std decrease index $\boldsymbol{\vartheta}$	-1.5e-6
Coefficient of std $\boldsymbol{\zeta}$	1
Optimizer	Adam
Collision safety threshold $\boldsymbol{c_s}$	$8m$
GAE coefficient $\boldsymbol{\rho}$	0.97
Weights of collision $\boldsymbol{\varepsilon_c}$, velocity $\boldsymbol{\varepsilon_v}$ and acceleration $\boldsymbol{\varepsilon_a}$	50, 0.05, 0.05
Weights of single pass $\boldsymbol{\varepsilon_p^1}$ and all pass $\boldsymbol{\varepsilon_p^2}$	10,50
MAPPO	
Learning rate	3e-4→0 (linearly)
Clip range $\boldsymbol{\epsilon}$	0.2
Minibatch size	64
VICS	
Predictive horizon $\boldsymbol{T}$	5
Velocity range	[0 m/s, 15 m/s]
Target velocity $\boldsymbol{v_t}$	15 m/s
Weights of velocity $\boldsymbol{\omega_v}$ and acceleration $\boldsymbol{\omega_a}$	1,5
Weight of risk $\boldsymbol{H}$	1000
Risk parameter $\boldsymbol{\alpha}$	0.005
MICA	
Maximum velocity $\boldsymbol{v_{max}}$	15 m/s
Planning time step	800
Large constant number M	1000

B. Performance Comparison of MACPO, MAPPO, MAPPO-SC Algorithms During Training Process

In this experiment, we have not only trained three policies through MACPO, MAPPO and MAPPO-SC algorithms, but also compared their performance in terms of traffic efficiency, safety, and ride comfort. Among them, MACPO is the algorithm proposed in this paper, and MAPPO is the RL algorithm for the cooperative control of vehicles at intersections presented in [34]. Although MAPPO-SC has the same safety-related constraints as MACPO, it uses MAPPO's policy update method and parameters. Namely, the constraints guide the policy update in the form of reward function penalty items

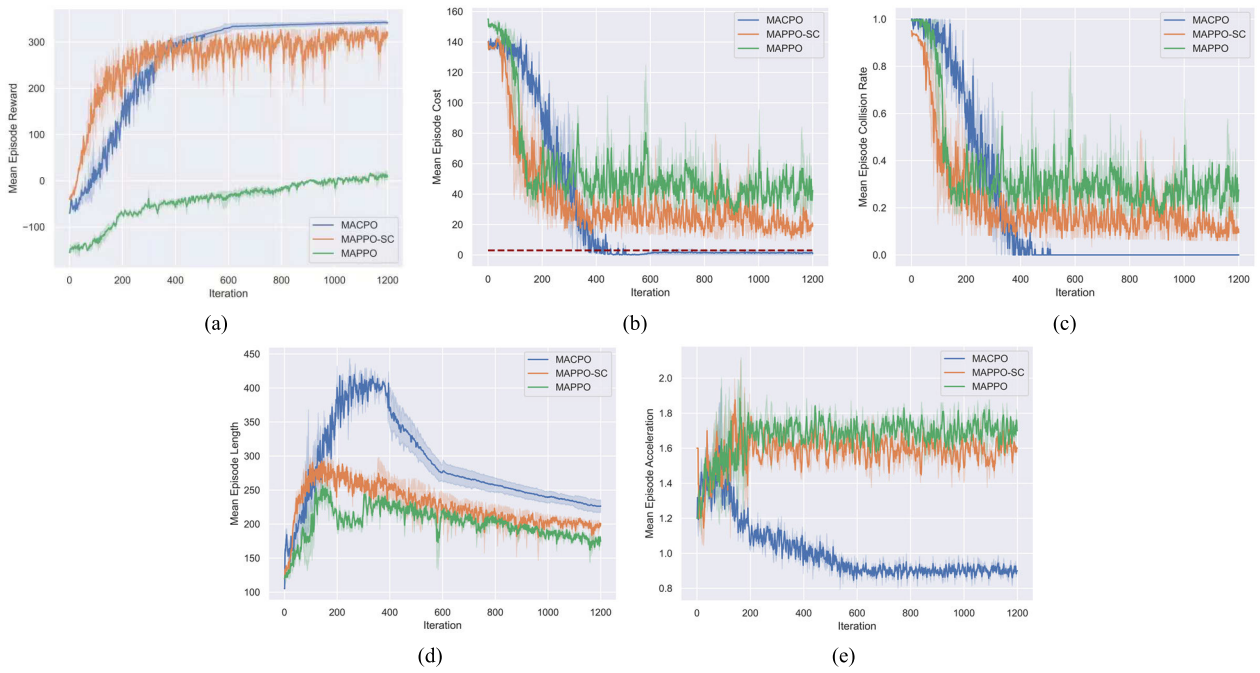


Fig. 7. Performance Comparison of MACPO policy with MAPPO, MAPPO-SC during training process. (a) Mean episode reward. (b) Mean episode cost. (c) Mean episode collision rate. (d) Mean episode length. (e) Mean episode acceleration.

together with other reward items, rather than as the separate constraints that must be satisfied. Fig. 7 dipits the performance comparison of three algorithms during training, in which the solid line represents the average value of the training curves, and the shaded part represents the variance.

The episode rewards of the three algorithms are shown in Fig. 7 (a). In terms of reward value, due to the different reward function settings, the reward function value of the MAPPO algorithm is generally low, and after convergence, the reward value of the MACPO algorithm with the same reward function setting is higher than that of MAPPO-SC. The reason for this gap lies in that the MACPO algorithm can keep the safety overhead value at 0, whereas the MAPPO-SC algorithm is difficult to control the safety penalty item in the reward function to 0 due to the lack of a separate safety cost neural network and policy update method satisfying safety constraints. Regarding the stability of the policy after convergence, the MACPO algorithm can keep the value of the reward function stable at a high level after convergence, thanks to the strict restriction of the safety cost function on the potential collision risk. However, the reward value obtained by the MAPPO-SC algorithm based on the same safety considerations but without constrained policy updates fluctuates greatly when the policy is updated about 300 times. At the same time, the reward value of the MAPPO algorithm that does not consider safety constraints also fluctuates greatly (the shaded area representing the variance of its reward value is the largest). Both of them are difficult to meet people's needs for stability and safety of driverless and intelligent transportation systems.

Fig. 7 (b) shows the episode safety cost that represents the potential collision risk of the traffic environment, in which the red dotted line represents the safety cost function threshold preset in this paper. The cost function includes dense (potential collision risk) and sparse (direct collision) costs associated with safe passage at intersections. The lack of cost neural network makes the MAPPO-SC and the MAPPO algorithm unable to obtain the ideal safety cost value. Due to the fact that the safety penalty is considered in the reward function, the cost function value of the MAPPO-SC algorithm after convergence is smaller than that of the MAPPO algorithm without safety considerations. Using the cost constraint neural network and constraint satisfaction policy update, the MACPO algorithm can always constrain the cost function value near the preset threshold in this paper. In this way, the potential collision risk in the traffic scene is significantly less than that of the MAPPO-SC and MAPPO algorithms.

The episode collision rate, as the most direct indicator for evaluating the safety of the algorithms, is shown in Fig. 7(c). Corresponding to the cost function value that characterizes the collision risk, the algorithm with a lower cost function value curve tends to have a lower episode average collision rate. The MACPO algorithm uses the cost function to constrain the collision risk in the traffic scene, which achieves zero collision rate after convergence, and keeps it from the 500th update to the end of the policy update. The collision rate of MAPPO-SC is smaller than that of MAPPO algorithm. Obviously, even with safety constraints, the MAPPO-SC algorithm cannot achieve zero collision rate during the entire training process, which further proves the safety limitations of the traditional single reward function-driven type RL algorithm.

Fig. 7 (d) shows the episode length, which reflects the traffic efficiency of each algorithm in road intersection, and the lower the value, the higher the traffic efficiency. The MAPPO algorithm without safety constraints obtains the highest traffic efficiency, and the traffic efficiency of MAPPO-SC and MACPO algorithms with safety constraints is relatively low. The reason is that the policy trained by the MACPO algorithm

TABLE II
PERFORMANCE COMPARISON OF MACPO POLICY WITH MAPPO, MAPPO-SC POLICIES, VICS AND MICA AFTER DEPLOYMENT

Test Scenarios	Methods	Mean Episode Length (s)	Safety Distance Violation ($-$)	Collision Rate ($-$)	Mean Episode Acceleration (m/s^2)	Mean Jerk (m/s^3)	Mean Inference Time (s)
Low Traffic Demand	MAPPO	8.31	16	0.15	1.53	0.64	0.62e-2
	MAPPO-SC	9.12	7	0.10	0.97	0.53	0.87e-2
	VICS	9.65	0	0	0.95	0.57	0.11
	MICA	19.93	0	0	0.64	0.16	1.18
	MACPO	9.52	0	0	0.31	0.66	0.91e-2
Medium Traffic Demand	MAPPO	21.37	20	0.20	1.62	1.79	0.62e-2
	MAPPO-SC	22.05	7	0.10	1.61	1.78	0.89e-2
	VICS	44.61	0	0	1.51	1.18	0.25
	MICA	38.86	0	0	0.93	1.05	2.67
	MACPO	22.98	0	0	0.91	0.93	0.92e-2
High Traffic Demand	MAPPO	27.72	22	0.25	1.60	1.79	0.64e-2
	MAPPO-SC	27.65	8	0.15	1.62	1.80	0.89e-2
	VICS	73.09	0	0	1.64	1.22	0.60
	MICA	54.63	0	0	0.96	1.01	6.73
	MACPO	30.27	0	0	0.93	0.95	0.92e-2

controls each vehicle to strictly maintain the safe distance from the surrounding vehicles, so as to limit the risky and aggressive traffic of the vehicles and reduce the potential collision risk in the traffic scene. It can also be observed that the traffic efficiency of the MACPO algorithm is not far behind that of the MAPPO-SC and MAPPO algorithms, but it achieves high safety by sacrificing a small amount of traffic time (around 3 seconds).

Fig. 7 (e) shows the episode acceleration, which reflects the riding comfort of the occupants. A lower average acceleration value indicates that the occupants in the car have higher riding comfort. Obviously, the acceleration value of the MACPO algorithm is still at a relatively low level after about 600 policy updates. At the same time, as the number of training increases, the acceleration fluctuation value remains small. The average acceleration values of MAPPO and MAPPO-SC algorithms are at a high value, and the average acceleration values fluctuate greatly. According to the results, the MACPO algorithm, which separates the reward function from the cost function, can achieve high ride comfort and ensure driving safety.

C. Performance Comparison of MACPO Policy With MAPPO, MAPPO-SC Policies, MPC-Based VICS and MIP-Based MICA After Deployment

In this experiment, we implement the policies trained by MACPO, MAPPO, MAPPO-SC, MPC-based VICS and MIP-based MICA. To evaluate the performance of these different algorithms under varying traffic complexities, we conduct experiments focusing on various metrics: Mean Episode Length, Safety Distance Violation, Collision Rate, Mean Episode Acceleration, Mean Jerk, and Mean Inference Time. These evaluations are done in low, medium, and high-density traffic scenarios, specifically, low traffic flow $\lambda_1 = 600veh/h/lane$, medium traffic flow $\lambda_2 = 1200veh/h/lane$, and high traffic flow $\lambda_3 = 1800veh/h/lane$. The results are presented in Table II.

In terms of travel efficiency, the neural network-based MACPO, MAPPO, and MAPPO-SC algorithms performed within the same order of magnitude in mean episode length. In contrast, the optimization-based VICS and MICA methods lagged significantly behind in this same metric. As traffic complexity increased, these optimization-based methods take longer to solve problems, consuming more time and resources without necessarily obtaining a globally optimal solution. Regarding the MPC-based VICS algorithm specifically, maintaining continuous control proved challenging due to the heavy computational demands. Therefore, we reduced the prediction length to 5 to ensure that vehicles have available input to maintain continuous control before the next set of solutions is generated.

In terms of safety, indicated by safety distance violation and collision rate, the optimization-based VICS and MICA algorithms achieved high safety, however, this heightened safety comes at the expense of significantly reduced traffic efficiency. Among the three neural network-based algorithms, only MACPO achieves a zero collision rate and no safety distance violations, indicating a significantly reduced or even minimized collision risk. In contrast, MAPPO-SC, which incorporates safety constraints into the reward function, has a higher collision rate and safety distance violation than MACPO but is safer than MAPPO, which does not consider safety at all.

In terms of ride comfort, indicated by mean episode acceleration and mean jerk, the optimization-based methods such as VICS and MICA recorded relatively high performance, and the performance of MICA is higher than VICS algorithm. The superior performance of the MICA algorithm is due to its simultaneous control over the leading vehicle in a lane, with subsequent vehicles following in a platoon mode, which

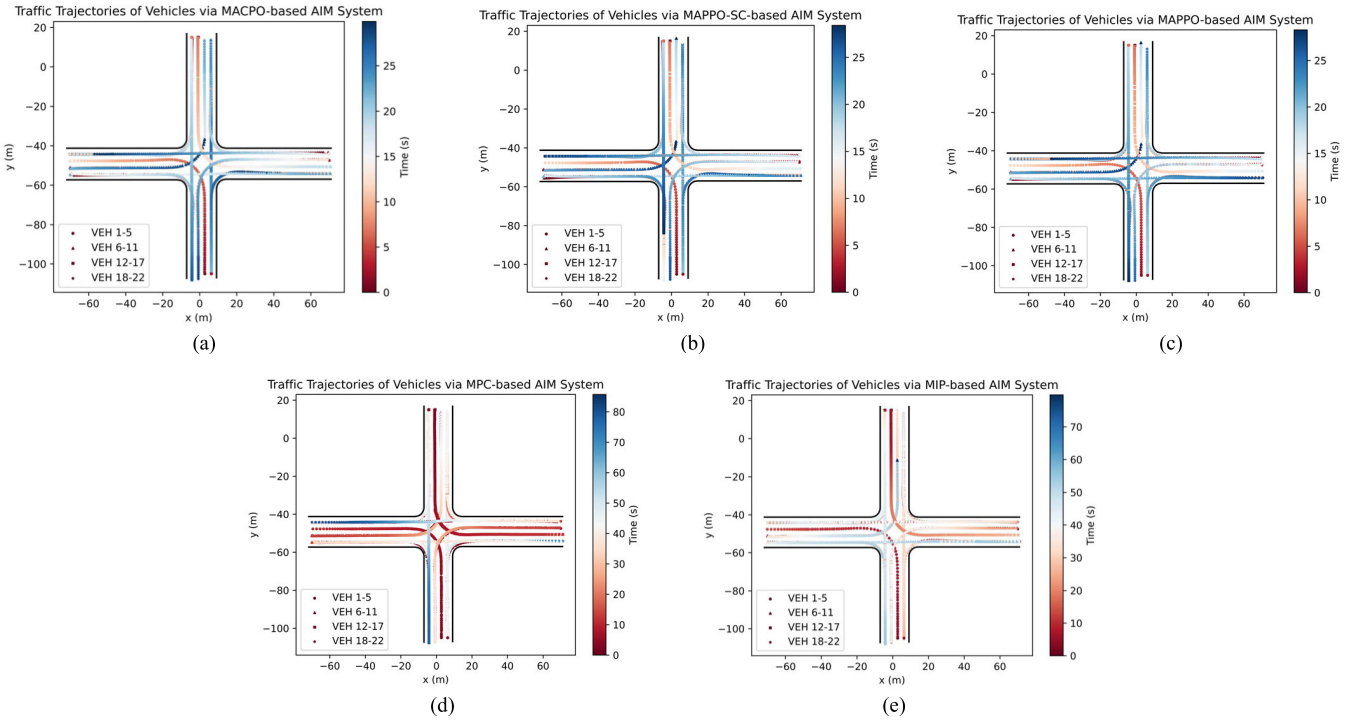


Fig. 8. Traffic Trajectories vs Timestep. (a) MACPO, (b) MAPPO, (c) MAPPO-SC (d) MPC-based VICS, (e) MIP-based MICA.

reduces the acceleration. Conversely, the VICS algorithm independently controls each vehicle in the traffic scenario. In cases where a global optimal solution cannot be guaranteed for all vehicles, substantial fluctuations in acceleration may occur. Among the three neural network-based algorithms, the MACPO algorithm, due to its additional safety constraint network, possesses the smallest action space, resulting in less variation in vehicle speeds in the traffic environment. This ensures ride comfort while maintaining safety and efficiency. The MAPPO-SC algorithm, lacking the additional neural network constraint, has a larger action space and hence shows relatively greater acceleration than MACPO. On the other hand, the MAPPO algorithm, which does not have any safety constraints, exhibits the most randomness in behavior, resulting in the highest acceleration values and thereby leading to a comparatively less comfortable riding experience.

In the aspect of computation time, symbolized by mean inference time, the RL methods greatly outperform the optimization-based methods, reducing computation time by several orders of magnitude. This is mainly attributed to the ability of RL methods to map traffic environment states to actions directly through neural connections, bypassing the need for extensive calculations. On the other hand, optimization-based methods require solving multiple constrained optimization problems given the input states, which demand a substantial amount of computational resources.

Finally, Fig. 8 presents the temporal scatter plots of vehicle passage within the range controlled by these methods. By recording the real-time position of vehicles in the simulation environment at each simulation step, we can observe the pattern Overall, according to the results, MPC-based VICS algorithm and MIP-based MICA algorithm have high safety performance and MIP-based MICA algorithm has high comfort performance, but extremely low computational efficiency, making them difficult to be widely applied in complex and dynamic traffic scenarios. The MAPPO algorithm that does not consider safety constraints has the highest traffic efficiency, but its safety performance is poor, so that it is not suitable for safety-critical autonomous driving and intelligent transportation fields. The MAPPO-SC algorithm adds safety constraints to the reward function in the form of penalty items, which slightly improves its safety performance. Nonetheless, there is still a large collision risk, and the traffic efficiency is slightly lower than that of MAPPO. Given that MACPO algorithm uses the separate cost neural network and constraint satisfaction policy update approach, it can not only take account of traffic efficiency and ride comfort but also eliminate potential collision risks as much as possible. In contrast, it is more suitable for autonomous driving and intelligent transportation system.

Moreover, the proposed algorithm possesses certain generalization capabilities and can be extended to other intersection scenarios, such as three-way intersections (T or Y type) and roundabouts, as well as other irregular environments. Prior to applying the algorithm to a new environment, it is necessary to fine-tune the hyper-parameters of the model, re-adjust the coefficients of the reward and cost functions, vehicle turning formulas, and control area size, among other environment-specific parameters. For scenarios with significantly increased complexity, more state variables (such as heading angle, travel direction, etc.) may be introduced to describe more complete traffic conditions, and more action options (such as passage permissions, routes, etc.) to achieve more optimized traffic flow control. Simultaneously, increasing the number of neural network layers and the number of neurons in each layer can enhance model flexibility to accommodate more

complex problems. While these expansions may lead to longer model learning times, they will not significantly increase the actual computation time after model deployment. Thus, with thoughtful design and reasonable optimization, the proposed method holds the promise of delivering superior performance even in more complex situations.

VII. Conclusion

In this paper, a conflict-free management scheme of CAVs at unsignalized intersection has been proposed by using safe multi-agent deep reinforcement learning (MADRL) so as to address the challenges of current centralized coordination methods in balancing high computational efficiency and robust safety assurance. Firstly, we have approached the safe MADRL problem by formulating it as a constrained Markov game (CMG), and have further transformed the AIM problem into a CMG. Following this, we introduced the Multi-Agent Constrained Policy Optimization (MACPO), a methodology specifically developed to address the CMG problem. MACPO is notable for its integration of safety constraints, which serve to further limit the trust region defined by the Kullback-Leibler (KL) divergence, thereby facilitating reinforcement learning policy updates that strive for maximum performance while still maintaining constraint costs within pre-defined bounds. Subsequently, we introduced the MACPO-based AIM Algorithm to ensure the safety, efficiency, and occupant comfort of CAVs' cooperative traffic behavior. Finally, we implemented the policies trained by MACPO, MAPPO and MAPPO-SC as well as the coordination schemes based on MPC and MIP methods, and compare their performance. According to the results, compared with the MPC and MIP methods, our method has increased computational efficiency by 65.22 times and 731.52 times respectively, and has improved traffic efficiency by 2.41 times and 1.80 times respectively. In comparison to the non-safety awareness RL methods, our method achieves not only zero collision rate for the first time but also better passenger comfort.

However, our work still needs to be improved. Firstly, only the longitudinal dynamics of vehicles is considered, but in the real world, the lateral and longitudinal dynamics are coupled, which will lead to unsatisfactory driving stability. Secondly, the dynamic change of the vehicle's expected driving direction in the control area is not considered, yet there may be a temporary change of driving intention in the actual intersection scene. In order to solve these problems, we will introduce stability constraints and states representing driving intent into the original problems in the future work, so as to enable the policy to improve driving stability and adapt to the intention dynamics.

References

[1] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng, "Eco-departure of connected vehicles with V2X communication at signalized intersections," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5439–5449, Dec. 2015, doi: 10.1109/TVT.2015.2483779.

[2] S. Djahel, R. Doolan, G. M. Muntean, and J. Murphy, "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 125–151, Jan./Mar. 2015, doi: 10.1109/COMST.2014.2339817.

[3] K. C. Dey et al., "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC)," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 491–509, Feb. 2016, doi: 10.1109/TITS.2015.2483063.

[4] F. Navas, V. Milanés, C. Flores, and F. Nashashibi, "Multi-model adaptive control for CACC applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1206–1216, Feb. 2021, doi: 10.1109/TITS.2020.2964320.

[5] Y. Zhu, D. Zhao, and Z. Zhong, "Adaptive optimal control of heterogeneous CACC system with uncertain dynamics," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1772–1779, Jul. 2019, doi: 10.1109/TCST.2018.2811376.

[6] M. A. S. Kamal, J.-I. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1136–1147, Jun. 2015, doi: 10.1109/TITS.2014.2354380.

[7] K. Dresner and P. Stone, "Multiagent traffic management: An improved intersection control mechanism," in *Proc. 4th Int. Joint Conf. Auton. Agents Multiagent Syst.*, vol. 3, Jul. 2005, pp. 530–537.

[8] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: From reservation-to optimization-based methods," *Transp. Res. B, Methodol.*, vol. 122, pp. 416–435, Apr. 2019, doi: 10.1016/j.trb.2019.03.002.

[9] P. Dai, K. Liu, Q. Zhuge, E. H.-M. Sha, V. C. S. Lee, and S. H. Son, "Quality-of-experience-oriented autonomous intersection control in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1956–1967, Jul. 2016, doi: 10.1109/TITS.2016.2514271.

[10] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng, "A bi-level cooperative driving strategy allowing lane changes," *Transp. Res. C, Emerg. Technol.*, vol. 120, Nov. 2020, Art. no. 102773, doi: 10.1016/j.trc.2020.102773.

[11] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 103, pp. 246–260, Jun. 2019, doi: 10.1016/j.trc.2019.04.012.

[12] K. Zhang, D. Zhang, A. de La Fortelle, X. Wu, and J. Grégoire, "State-driven priority scheduling mechanisms for driverless vehicles approaching intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2487–2500, Oct. 2015, doi: 10.1109/TITS.2015.2411619.

[13] Z. He, L. Zheng, L. Lu, and W. Guan, "Erasing lane changes from roads: A design of future road intersections," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 2, pp. 173–184, Jun. 2018, doi: 10.1109/TIV.2018.2804164.

[14] A. Katriniok, B. Rosarius, and P. Mähönen, "Fully distributed model predictive control of connected automated vehicles in intersections: Theory and vehicle experiments," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18288–18300, Oct. 2022, doi: 10.1109/TITS.2022.3162038.

[15] C. Chen, Q. Xu, M. Cai, J. Wang, J. Wang, and K. Li, "Conflict-free cooperation method for connected and automated vehicles at unsignalized intersections: Graph-based modeling and optimality analysis," *IEEE Trans. on Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21897–21914, Jun. 2022, doi: 10.1109/TITS.2022.3182403.

[16] J. Wang, X. Zhao, and G. Yin, "Multi-objective optimal cooperative driving for connected and automated vehicles at non-signalised intersection," *IET Intell. Transp. Syst.*, vol. 13, no. 1, pp. 79–89, Jan. 2019, doi: 10.1049/iet-its.2018.5100.

[17] Z. Yao, H. Jiang, Y. Jiang, and B. Ran, "A two-stage optimization method for schedule and trajectory of CAVs at an isolated autonomous intersection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3263–3281, Mar. 2023, doi: 10.1109/TITS.2022.3230682.

[18] H. Jiang, Z. Yao, Y. Jiang, and Z. He, "Is all-direction turn lane a good choice for autonomous intersections? A study of method development and comparisons," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 8510–8525, Mar. 2023, doi: 10.1109/TVT.2023.3250957.

[19] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 161–176, Mar. 2019, doi: 10.1016/j.trc.2019.01.004.

[20] E. Lukose, M. W. Levin, and S. D. Boyles, "Incorporating insights from signal optimization into reservation-based intersection controls," *J. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 250–264, May 2019, doi: 10.1080/15472450.2018.1519706.

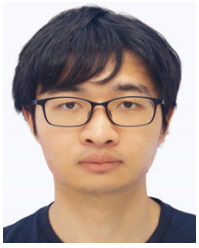
[21] K. Dresner and P. Stone, "Human-usable and emergency vehicle-aware control policies for autonomous intersection management," in *Proc. 4th Int. Work. Agents Traffic Transp. (ATT)*, Hakodate, Japan, vol. 12, May 2006, p. 14.

[22] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated intersection control," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2259, no. 1, pp. 223–232, Jan. 2011, doi: 10.3141/2259-21.
[23] X. Qian, F. Altché, J. Grégoire, and A. Fortelle, "Autonomous intersection management systems: Criteria, implementation and evaluation," *IET Intell. Transp. Syst.*, vol. 11, no. 3, pp. 182–189, Apr. 2017, doi: 10.1049/iet-its.2016.0043.
[24] G.-P. Antonio and C. Maria-Dolores, "Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7033–7043, Jul. 2022, doi: 10.1109/TVT.2022.3169907.
[25] M. W. Levin, S. D. Boyles, and R. Patel, "Paradoxes of reservation-based intersection controls in traffic networks," *Transp. Res. A, Policy Pract.*, vol. 90, pp. 14–25, Aug. 2016, doi: 10.1016/j.tra.2016.05.013.
[26] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1908–1916, May 2019, doi: 10.1109/TITS.2018.2850335.
[27] Q. Lu and K.-D. Kim, "A mixed integer programming approach for autonomous and connected intersection crossing traffic control," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6, doi: 10.1109/VTCFall.2018.8690681.
[28] G. Lu, Z. Shen, X. Liu, Y. M. Nie, and Z. Xiong, "Are autonomous vehicles better off without signals at intersections? A comparative computational study," *Transp. Res. B, Methodol.*, vol. 155, pp. 26–46, Jan. 2022, doi: 10.2139/ssrn.3812649.
[29] M. Choi, A. Rubenecia, and H. H. Choi, "Reservation-based traffic management for autonomous intersection crossing," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 12, Dec. 2019, Art. no. 1550147719895956, doi: 10.1177/1550147719895956.
[30] L. Xu, J. Lu, B. Ran, F. Yang, and J. Zhang, "Cooperative merging strategy for connected vehicles at highway on-ramps," *J. Transp. Eng., A, Syst.*, vol. 145, no. 6, Jun. 2019, Art. no. 04019022, doi: 10.1061/jtepbs.0000243.
[31] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4563–4571, Nov. 2020, doi: 10.1109/TITS.2019.2940641.
[32] A. Boukerche, D. Zhong, and P. Sun, "A novel reinforcement learning-based cooperative traffic signal system through max-pressure control," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1187–1198, Feb. 2022, doi: 10.1109/TVT.2021.3069921.
[33] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 433–443, Jan. 2020, doi: 10.1109/TITS.2019.2942014.
[34] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12597–12608, Nov. 2020, doi: 10.1109/TVT.2020.3026111.
[35] Y. Xu, H. Zhou, T. Ma, J. Zhao, B. Qian, and X. Shen, "Leveraging multiagent learning for automated vehicles scheduling at nonsignalized intersections," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11427–11439, Jul. 2021, doi: 10.1109/JIOT.2021.3054649.
[36] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. Int. Conf. Mach. Learn.*, Aug. 2017, pp. 22–31.
[37] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 1889–1897.

Rui Zhao (Member, IEEE) was born in Liaoyuan, Jilin, China, in 1986. She received the B.S. degree in computer science and technology from Northeast Normal University in 2009 and the Ph.D. degree in computer science and technology from Jilin University, Changchun, China, in 2017.

She is currently an Associate Professor with the College of Automotive Engineering, Jilin University. She has authored about 30 journal articles and ten patents in China. She authored the monograph *Cyber Security Technology for Intelligent Automotive*. Her research interests include cooperative control, functional safety, cybersecurity, and safety reinforcement learning for connected and automated vehicles.

Prof. Zhao is a member of the Society of Automotive Engineers.



Yun Li (Member, IEEE) received the B.S. degree in vehicle engineering from Jilin University in 2021 and the M.S. degree from the Department of Information and Communications Engineering, Tokyo Institute of Technology, in 2023. He is currently pursuing the Ph.D. degree with the Graduate School of Information and Science Technology, The University of Tokyo.

He completed a Progressive Graduate Minor in data science and artificial intelligence with the Department of Information and Communications Engineering, Tokyo Institute of Technology. He has gained practical research experience through an internship with Japan Data Science Research Laboratories, NEC Corporation. He has contributed to the field with a patent submission and a paper presented at the IEEE Vehicular Technology Conference (VTC). His research interests include reinforcement learning, autonomous driving, large language models, and wireless communications.

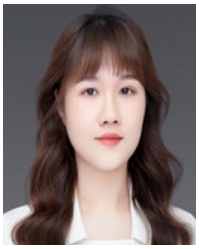
Fei Gao received the Ph.D. degree in automotive engineering from Jilin University, China, in 2017.

From 2014 to 2015, she was a Visiting Student with the University of California at Berkeley, Berkeley, CA, USA. She is currently an Associate Professor with the State Key Laboratory of Automotive Simulation and Control Automotive Engineering, Jilin University. Her research interests include automotive human engineering.

Zhenhai Gao (Member, IEEE) was born in Changchun, Jilin, China, in 1973. He received the Ph.D. degree in automotive engineering from Jilin University.

He is currently the Deputy Dean of automotive engineering and the Director of the State Key Laboratory of Automotive Simulation and Control Automotive Engineering, Jilin University. He is the coauthor of three books. More than 100 papers have been published and 20 invention patents have been authorized. His research interests include autopilot technology and human engineering.

Prof. Gao is a Distinguished Member of the Expert Committee Intelligent Connected Vehicle Innovation Alliance, the Chairperson of the Industrial Design Association in Jilin Province, and an Editorial Board Member of *International Journal of Human Factors Modelling and Simulation*.



Tianyao Zhang was born in Changchun, Jilin, China. She received the bachelor's degree from Jilin University in 2016 and the master's degree in automotive engineering from Clemson University in 2018. She is currently an Engineer with the National Key Laboratory of Vehicle Simulation and Control, Jilin University. Her research interests include the HMI design of intelligent cockpits.