Zhang *et al. Intell. Robot.* **2025**, *5*(3), 695-716
DOI: 10.20517/ir.2025.36

Intelligence & Robotics

Research Article

Open Access

Learning-based cooperative decision-making and control for multiple autonomous vehicles in unsignalized intersections

Ronghua Zhang[1,2,#], Xincheng Xu[2,#], Yang Lu[2], Xin Xu[2], Xinglong Zhang[2], Qingwen Ma[2]

[1]School of Mechanical Engineering, Sichuan University of Science and Engineering, Zigong 643000, Sichuan, China.
[2]College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, Hunan, China.
[#]Authors contributed equally.

Correspondence to: Dr. Qingwen Ma, College of Intelligence Science and Technology, National University of Defense Technology, No.109, Deya Road, Kaifu District, Changsha 410073, Hunan, China. E-mail: maqingwen@mail.nwpu.edu.cn

How to cite this article: Zhang, R.; Xu, X.; Lu, Y.; Xu, X.; Zhang, X.; Ma, Q. Learning-based cooperative decision-making and control for multiple autonomous vehicles in unsignalized intersections. *Intell. Robot.* **2025**, *5*(3), 695-716. http://dx.doi.org/10.20517/ir.2025.36

Received: 15 Jun 2025 **First Decision:** 16 Jul 2025 **Revised:** 5 Aug 2025 **Accepted:** 13 Aug 2025 **Published:** 29 Aug 2025

Academic Editor: Chaomin Luo **Copy Editor:** Pei-Yun Wang **Production Editor:** Pei-Yun Wang

Abstract

Cooperative navigation of multiple autonomous vehicles (MAVs) at unsignalized intersections remains a core challenge in intelligent transportation systems. This paper proposes a learning-based cooperative decision-making and control (LCDMC) method for MAVs, which improves policy learning efficiency and ensures safe and efficient cooperative navigation. In the proposed LCDMC algorithm, the global value function is decomposed into two components: a local utility function and a joint-action utility function among vehicles, which incorporates both the offline policy learning phase and the online deployment phase. During the offline phase, the kernel-based least-squares policy iteration method is employed to learn localized decision-making policies from high-dimensional samples. In the online deployment phase, a coordination graph for MAVs is developed, and a collaborative utility function characterizing joint action performance is designed. To solve optimized decision actions, the local utility function is integrated with a message propagation mechanism, and then the decision actions are converted into velocity commands. Furthermore, a receding-horizon reinforcement learning approach is designed to achieve trajectory tracking control of the autonomous vehicles in MAVs. Finally, to verify the effectiveness of the proposed method, numerical simulations of MAVs are performed, and the results demonstrate that the proposed LCDMC method exhibits superior performance in both traffic efficiency and safety for cooperative navigation of MAVs at unsignalized intersections.

Keywords: Multiple autonomous vehicles, unsignalized intersection, decision and control, reinforcement learning, coordination graphs

1. INTRODUCTION

Autonomous driving technology is advancing at an unprecedented pace and demonstrating significant application potential in real-world traffic scenarios[1,2]. Although autonomous vehicles (AVs) perform well in structured environments such as highways, their deployment in urban road settings still faces numerous challenges. Urban road environments are intricate and complex, containing a multitude of uncertainties. This uncertainty significantly increases the difficulty of decision-making for AVs[3]. Among these challenges, one of the most demanding scenarios is the unsignalized urban intersection. Relevant research indicates that even experienced human drivers often experience considerable pressure at such intersections, where momentary lapses can potentially lead to traffic accidents[4]. Therefore, enhancing the capability of autonomous driving systems to respond effectively in such high-risk scenarios is a critical research direction.

The cooperative navigation of multiple vehicles in unsignalized intersections poses key challenges, including high-dimensional nonlinear dynamics, incomplete-information Nash equilibrium, and uncertainty in neighboring vehicle behaviors, thus attracting extensive research attention[5,6]. In recent years, the cooperative navigation problem between AVs and multiple human-driven vehicles at intersection scenarios has been widely investigated, primarily focusing on AVs decision-making mechanisms[7,8], path planning[9], and social behavior modeling[10]. However, the aforementioned methods are inadequate for fully autonomous and cooperative intersection scenarios, which require further research on distributed negotiation mechanisms, multi-agent collaborative decision-making frameworks, and global traffic efficiency optimization strategies. Cooperative decision-making and control schemes for multi-vehicle systems have demonstrated potential to enhance traffic throughput efficiency and safety[11,12], particularly under dynamic and uncertain environments. The common coordination methods for multi-vehicle systems can be systematically categorized into the following classes: rule-based decision-making approaches, convex optimization-based methods[13], game-theoretic models[14,15], and learning-based decision-making frameworks[16].

Rule-based decision-making methods involve creating a rule database for vehicle behaviors in operational scenarios based on experience and predefined rules. These rules enable a deterministic mapping from vehicle states to behaviors, with typical approaches such as finite state machine (FSM)[17], first come first served (FCFS)[18], and longest queue first (LQF)[19]. Rule-based methods exhibit clear logic, operational stability, and strong practicality. However, their finite state representation may inadequately capture all operational conditions faced by AVs, and limited scenario coverage degrades decision-making performance in complex dynamic environments. Furthermore, Liu *et al.* designed a multi-agent game-theoretic attention-based deep deterministic policy gradient algorithm[20], which improves traffic flow efficiency at unsignalized intersections. Although game-theoretic methods offer theoretical advantages for multi-agent intersection coordination, they face scalability issues due to computational inefficiency and reduced accuracy in modeling vehicle dynamics in complex traffic scenarios.

Learning-based decision-making methods primarily include Bayesian inference[21,22], decision trees[23], and Markov decision processes[24]. In multi-agent systems, reinforcement learning (RL), particularly deep reinforcement learning (DRL), has become the primary approach for solving complex sequential decision-making problems[25]. Current research in multi-agent deep reinforcement learning (MDRL) is rapidly expanding into real-world traffic scenarios. The focus has shifted from fundamental tasks such as lane changing[26–28] and overtaking[29,30] to more complex collaborative decision-making on road systems. Chen *et al.* developed a curriculum learning-based decision-making framework for highway on-ramp merging to improve traffic flow efficiency and safety under mixed traffic conditions[31]. Guo *et al.* proposed a heuristic MDRL algorithm by combining the advantages of heuristic strategies and deep learning[32], which enhances the smoothness and safety of traffic flow at intersections. Zhao *et al.* presented a safety RL method based on multi-agent projection-constrained policy optimization[33], demonstrating superior coordination performance for AVs at unsignalized intersections. However, although the aforementioned methods have achieved considerable per-

formance, they generally suffer from high algorithmic complexity and computational costs. Moreover, most of these methods are designed for specific scenarios and rely on idealized assumptions, which significantly limit their practicality. Additionally, non-stationarity and poor interpretability also make these methods difficult to apply in real-world scenarios. To tackle the issues of excessive parameter size and slow inference speed in AV decision-making models, a lightweight optimization approach based on Video Swin Transformer and DRL is proposed in Ref[34]. This method effectively reduced model parameters and memory requirements while improving inference efficiency for long-sequence tasks through parallel spatiotemporal feature extraction with risk assessment mechanisms and the adoption of double-softmax linear self-attention. However, this approach still exhibits limitations including difficulties in hyperparameter tuning and performance degradation on short-sequence tasks.

To address the above problem, the paper proposes a learning-based cooperative decision-making and control (LCDMC) method for multiple AVs (MAVs), improving policy learning efficiency while ensuring safe and efficient cooperative navigation. The LCDMC method decomposes the global value function of the system into two components: a local utility function and a collaborative utility function representing the joint actions among vehicles, thereby enhancing the representational capacity of value functions in multi-agent RL. Numerical simulations of MAVs demonstrate that the LCDMC method exhibits superior performance in both traffic efficiency and safety for cooperative navigation of MAVs at unsignalized intersections. The main contributions are listed as follows.

(1) A coordination graph-based cooperative decision-making for MAVs is proposed to ensure the safe and efficient cooperative navigation. Unlike independent learning methods that focus solely on individual performance, the policies of the proposed decision-making method can achieve fast convergence to the optimal values through iterative message passing, which improves the value function's representational capacity.
(2) A lightweight network is constructed to solve the local utility functions via the kernel-based least-squares policy iteration (KLSPI) approach, which overcomes the drawbacks of low computational efficiency and difficulties in policy convergence existing in DRL.
(3) A receding-horizon optimization mechanism is designed to enhance the online learning efficiency, and further an efficient tracking control scheme is developed based on the mechanism, which improves the real-time control performance of MAVs.
(4) Extensive numerical simulations of MAVs are performed to verify the effectiveness of the proposed cooperative decision-making and control method, and the results demonstrate the superior performance in both traffic efficiency and safety for cooperative navigation of MAVs at unsignalized intersections.

The subsequent sections of the paper are structured as follows. Section 2 analyzes the foundational concepts of Markov decision process modeling for MAVs and coordination graphs. In Section 3, a LCDMC approach for MAVs is proposed. Section 4 carries out simulation verification, while Section 5 draws conclusions.

2. PRELIMINARIES AND PROBLEM FORMULATION

2.1. MAVs Markov decision process modeling

The MAVs' cooperative decision-making at intersections is modeled as a decentralized partially observable Markov decision process (Dec-POMDP), which is given by the following tuple:

$$\langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^{N}, P, \{R_i\}_{i=1}^{N}, \{O_i\}_{i=1}^{N}, N, \gamma \rangle \tag{1}$$

As shown in Figure 1, a T-junction without traffic signals involves six AVs requiring cooperative decision-making, including left-turning vehicles, right-going straight vehicles, and left-going straight vehicles. The state space of the N AVs at the intersection is denoted by $\mathcal{S}$. Taking the left-turning AV as an example to illustrate the sample collection process, the AVs in the straight-going lane are controlled using the car-following model[35], providing state information for the left-turning AV. When the left-turning AV enters the decision-making zone,

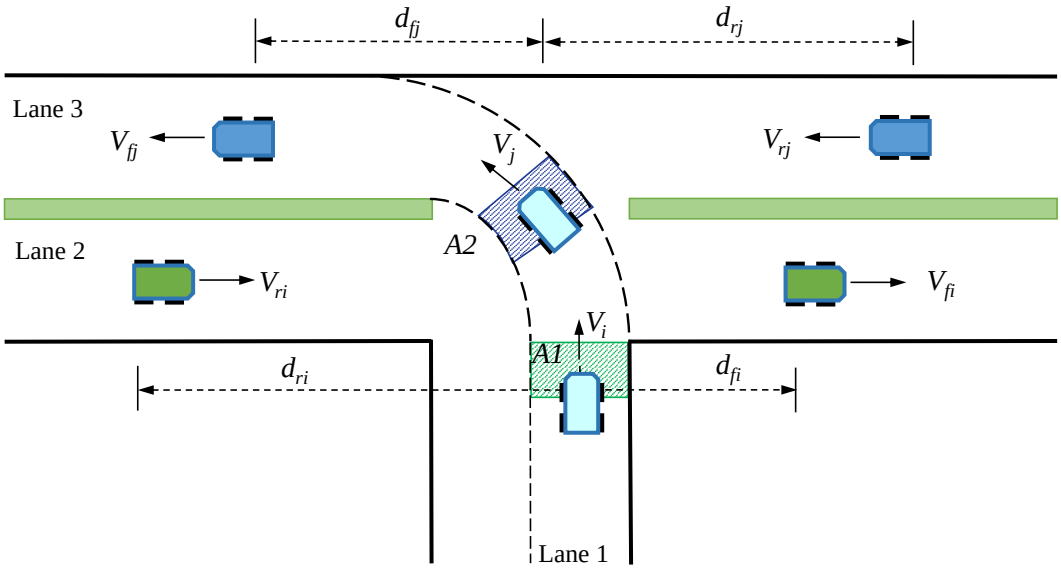

Figure 1. Illustration of MAVs decision-making modeling. MAVs: Multiple autonomous vehicles.

its state at discrete time k is defined as $s_i(k) = [v_i, v_{ri}, d_{ri}, v_{fi}, d_{fi}] \in \mathcal{S}$, including its own velocity v_i, velocities of the closest front vehicle v_{fi} and rear vehicle v_{ri}, and distances to those vehicles d_{fi} (front) and d_{ri} (rear). The action space for vehicle i is denoted as $\mathcal{A}_i$. At time step k, vehicle i selects an action $a_i(k) \in \mathcal{A}_i$ corresponding to one of the three behaviors: decelerating, maintaining current speed, or accelerating. The state transition probability P is given by $s_i(k+1) \sim P(\cdot \mid s_i(k), a_i(k))$, indicating the probability-based transition from state $s_i(k)$ and action $a_i(k)$ at time step k to the next state $s_i(k+1)$. The reward function R_i reflects the traffic conditions of MAVs at the intersection.

2.2. Coordination graphs

Probabilistic graphical model (PGM) integrates probability theory with graph theory and constitutes a pivotal research direction in machine learning[36]. PGM primarily focuses on the representation, inference, and learning of directed graphical models and undirected graphical models. A core objective in artificial intelligence is to extract implicit, fundamental knowledge from observed data, and PGM serves as a powerful methodology to achieve this goal. In MAVs systems, nodes of the probabilistic graph represent vehicles, while edges with connecting relationships describe the local performance of joint actions taken by adjacent vehicles, inferring the posterior distribution of joint actions based on current vehicle information. Probabilistic graph-based approaches can effectively model coupling relationships among MAVs, which facilitate the construction of large-scale artificial intelligence systems to address complex real-world problems.

The coordinated RL method introduces a coordination graph to establish connecting edges among agents with cooperative relationships[37]. The messages $m_{ij}(a_j)$ transmitted along adjacent edges $(i, j) \in E$ can fully describe the coordination among multiple agents, as illustrated in Figure 2. By incorporating coordination graphs into multi-agent systems, the representational capacity of the value function is enhanced, which effectively addresses decision-making problems. In this case, the global state-action value function is expressed as

$$Q(s,a) = \sum_{i=1}^{N} p_i(s_i, a_i) + \sum_{(i,j)\in E} p_{ij}(a_i, a_j) \tag{2}$$

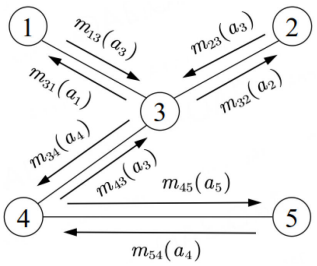

Figure 2. Message passing graph.

where $p_i(s_i, a_i)$ represents the local utility function of agent i after taking action a_i, and p_{ij} represents the collaborative utility function of neighboring agents i and j after they take actions a_i and a_j, respectively.

The Ref[36] demonstrates a direct duality between computing the maximum posteriori probability in PGM and solving the optimal joint action in coordination graphs. Consequently, the belief propagation algorithm, used for inference in PGM, is also applicable to multi-agent collaborative decision-making using coordination graphs. In Figure 2, agent i sends a message $m_{ij}(a_j)$ to neighbor j, which represents the maximum reward that agent i can obtain after agent j takes action a_j [38]. The message value is expressed as

$$m_{ij}(a_j) = \max_{a_i} \left\{ p_i(a_i) + p_{ij}(a_i, a_j) + \sum_{\bar{k} \in \Gamma(i) \backslash j} m_{ki}(a_i) \right\} + c_{ij} \tag{3}$$

where $\Gamma(i)$ denotes the set of neighbors for the i-th agent, $\bar{k} \in \Gamma(i) \backslash j$ indicates that $\bar{k}$ belongs to the set of neighbors, excluding j. When there are cycles in the graph, $c_{ij} = \frac{1}{|A_k|} \sum_k m_{ik}(a_k)$, otherwise, $c_{ij} = 0$. Before convergence, agent i aggregates the messages received from its neighbors, but it does not require enumerating the joint action spaces of neighboring agents. After several iterations of message passing and updates among neighboring agents, the messages $m_{ij}(a_j)$ converge to a fixed point. In each iteration, the state-action value function of agent i is represented as

$$Q_i(a_i) = p_i(a_i) + \sum_{j \in \Gamma(i)} m_{ji}(a_i) \tag{4}$$

which indicates that the state-action value function of agent i consists of the local utility function $p_i(a_i)$ and the message values from different subtrees with neighbor agent j as the root. The optimal decision action for agent i can be derived as

$$a_i^* = \arg\max_{a_i} Q_i(a_i) \tag{5}$$

Building upon the aforementioned mechanism, this paper proposes a LCDMC method for MAVs. The local utility function $p_i(a_i)$ of vehicle i is constructed via a KLSPI RL algorithm, while the local collaborative utility function p_{ij} characterizes the performance of joint actions. A coordination graph-based message passing mechanism is employed to resolve the cooperative decision-making control problem for MAVs at unsignalized intersections.

3. LCDMC FOR MAVS

This section presents the LCDMC method for MAVs. The method employs KLSPI to solve the local utility function and integrates a message-passing mechanism via a cooperation graph to compute the optimized de-

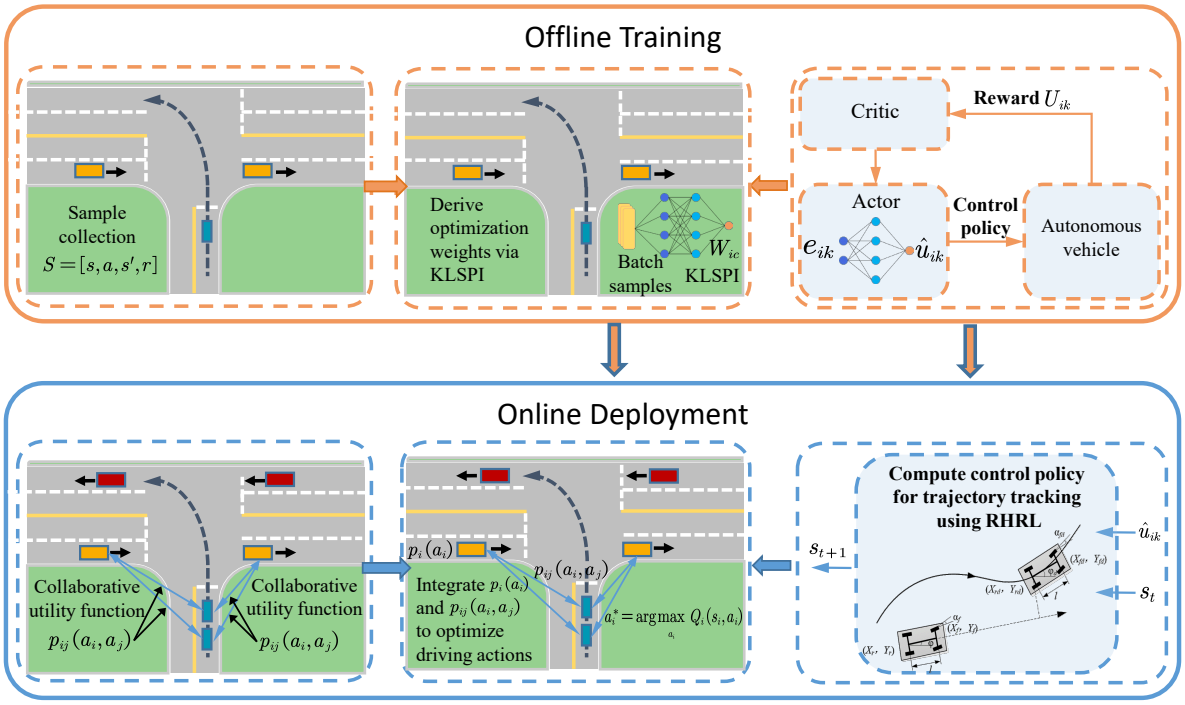

Figure 3. The block diagram of the proposed LCDMC algorithm. LCDMC: Learning-based cooperative decision-making and control.

cision actions for AVs, which are then used as reference signals. Furthermore, a receding-horizon RL (RHRL) method is designed for trajectory tracking control of AVs.

3.1. The overall framework of the proposed method

The proposed algorithm primarily consists of two stages: offline policy learning and online deployment implementation. The algorithmic framework is depicted in Figure 3. In the offline training phase, samples are collected using a random policy. An approximate linear dependency (ALD) analysis is employed to learn model features and construct the kernel dictionary from the high-dimensional state space, thereby reducing the dimensionality of the samples. Then, a sparse kernel-based least-squares RL method is adopted to learn the policy. During the online deployment phase, when AV i enters the decision-making zone, it forms interactions with neighboring AVs j, $j \in \Gamma(i)$. A collaborative utility function $p_{ij}(a_i, a_j)$ is defined to evaluate the joint actions between AVs i and j. Based on its current state s_i, the AV applies the offline-learned policy to compute the local utility function $p_i(a_i)$ for each candidate action. Following the message-passing Equation (3), the action of AV i is derived by the local utility $p_i(a_i)$ and collaborative utility function $p_{ij}(a_i, a_j)$.

Based on coordination graph theory, the global state-action value function is given by Equation (2). This formulation demonstrates that optimal decisions for MAVs can be computed in a distributed manner by iteratively exchanging messages along the coordination graph edges, thus avoiding exhaustive searches over the joint-action space. The proposed method contrasts with AV i directly computing actions via KLSPI. The decision module's output speed serves as the reference input, and an online RHRL method computes an optimal control law to ensure tracking of a reference trajectory. Gradient-based iterative updates of the critic and actor networks improve the real-time performance of the system.

3.2. Kernel-based efficient learning for local utility function

This section elaborates on the proposed LCDMC method for MAVs, which learns the local utility function via KLSPI. During the offline phase, extensive samples are pre-collected at unsignalized intersections. An

Table 1. Sampling parameters of MAVs corresponding to the scenario displayed in Figure 1

Lane of autonomous vehicles	Decision zone (m)	Action space	Front/rear vehicle lane	Front/rear vehicle speed (m/s)	Front/rear vehicle distance (m)
Lane 1	$y \in [-4, -2]$	sd, kv, acc	Lane 2	[1, 15]	[0, 50]
Lane 2	$x \in [-5.8, -3]$	sd, kv, acc	Lane 1	[1, 15]	[0, 30]
Lane 3	$x \in [-2, 1]$	sd, kv, acc	Lane 1	[1, 15]	[0, 30]

The scenario depicts an unsignalized T-junction where six AVs require cooperative decision-making, including left-turning vehicles in Lane 1, right-going vehicles in Lane 2, and left-going vehicles in Lane 3. MAVs: Multiple autonomous vehicles; AVs: autonomous vehicles.

optimized local decision-making policy is derived in a data-driven manner and subsequently deployed online for real-time decision-making.

In the coordinated traversal of multiple vehicles at the unsignalized intersection shown in Figure 1, the tuple for multi-vehicle cooperative decision-making modeling is defined in Equation (1). Under the current state $s_i(k)$, the AV updates its state upon executing action $a_i(k)$, while the reward function is designed to reflect the traffic dynamics of MAVs at the intersection. If a left-turning AV safely traverses the straight lane within 5 s, sampling terminates, and the reward is assigned as $R_i = -\varepsilon t$, where ε is an adjustable positive constant. If the AV collides with neighboring vehicles on the straight lane, the reward function incurs a significant penalty $R_i = -1,000$. If the vehicle remains stationary at the initial starting point without collision, the reward is set to $R_i = -400$. The reward function is given in Equation (6), and the sample $[s_i(k), a_i(k), R_i(k), s_i(k+1)]$ is collected. AV samples in adjacent lanes are collected similarly. Table 1 lists the sampling parameters, where **sd**, **kv**, and **acc** represent deceleration, constant-speed maintenance, and acceleration, respectively.

$$R_i(k) = \begin{cases} -\varepsilon t & \text{(Safely traverses the lane)} \\ -400 & \text{(Remains at the initial point)} \\ -1000 & \text{(Collision occurs)} \end{cases} \tag{6}$$

Upon acquiring a substantial set of samples, the KLSPI algorithm is employed for policy learning. This section extends sparse kernel-based feature representation methods[39] to cooperative decision-making in MAVs. In the proposed LCDMC framework, the local utility function of the AV is represented via basis functions constructed from feature vectors. A kernel sparsification technique is applied to extract features from the high-dimensional samples, yielding representative subsamples through ALD analysis. Assuming the kernel function $\kappa(\cdot, \cdot)$ satisfies Mercer's condition, there exists a mapping $\phi(\cdot)$ from the sample space to the Hilbert space, that is,

$$\kappa(s_i(t), s_i(q)) = \langle \phi(s_i(t)), \phi(s_i(q)) \rangle \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in the Hilbert space. This implies that all inner products can be computed via the kernel function, without explicit knowledge of the mapping $\phi(\cdot)$.

Then, the representative subsample is obtained using the ALD analysis technique. Sample sparsification is implemented via ALD in two steps. Firstly, the distance between sample $s_i(t)$ and feature point $s_i(q)$ is calculated using

$$\zeta_i(t) = \min_c \left\| \sum_{s_i(q) \in D_i(t)} c_q \phi(s_i(q)) - \phi(s_i(t)) \right\|^2 \tag{8}$$

where $c = [c_1, c_2, \cdots, c_{t-1}]^\top \in \mathbb{R}^{t-1}$, $D_i(t) \in \mathbb{R}^{t-1}$ denotes the current kernel dictionary. Secondly, the dictionary is updated. μ_i is set to quantify the degree of sample sparsification. If the distance between $s_i(t)$ and $s_i(q)$ is not less than μ_i, it implies that insufficient representational capacity of the current kernel dictionary for

the given sample. Consequently, $s_i(t)$ is incorporated into the dictionary, $D_i(t) = D_i(t-1) \cup s_i(t)$. Otherwise, the dictionary $D_i(t)$ remains unchanged.

The feature construction module utilizes the subsampled points generated previously to construct the basis function for each original sample as $K(s_i(k)) = [\kappa(s_i(k), s_i(t_1)), \kappa(s_i(k), s_i(t_2)), \cdots, \kappa(s_i(k), s_i(t_n))]^\top \in \mathbb{R}^n$, where n denotes the dimensionality of the feature space. $\kappa(x_1, x_2) = \exp(-\frac{\|x-y\|^2}{2\iota^2})$, ι is the kernel width. The local utility function is approximated using

$$\hat{p}_i(s_i(k)) = K^\top(s_i(k))W_{ic} = \sum_{l=1}^{n} \kappa(s_i(k), s_i(t_l))w_{ic} \tag{9}$$

where $W_{ic} \in \mathbb{R}^n$ is the weight vector of the network. The temporal difference (TD) error of RL is given as $\eta_i(k) = \hat{p}_i(s_i(k)) - \bar{p}_i(s_i(k))$, where $\bar{p}_i(s_i(k)) = s_i^\top(k)\mathcal{Q}s_i(k) + u_i^\top(k)\mathcal{R}u_i(k) + \gamma_i\hat{p}_i(k+1)$, $\mathcal{Q}$ and $\mathcal{R}$ are positive definite weight matrices, with γ_i denoting the discount factor.

By combining Equation (9) and rewriting TD error, we have:

$$[K^\top(s_i(k)) - \gamma_i K^\top(s_i(k+1))]W_{ic} = s_i^\top(k)\mathcal{Q}s_i(k) + u_i^\top(k)\mathcal{R}u_i(k) + \eta_i(k) \tag{10}$$

By multiplying both sides of Equation (10) by $K(s_i(k))$ and defining

$$\begin{aligned} \bar{A} &= \sum_{k=1}^{N_{\text{sam}}} K(s_i(k))(K^\top(s_i(k)) - \gamma_i K^\top(s_i(k+1))) \\ \bar{B} &= \sum_{k=1}^{N_{\text{sam}}} K(s_i(k))(s_i^\top(k)\mathcal{Q}s_i(k) + u_i^\top(k)\mathcal{R}u_i(k)) \end{aligned} \tag{11}$$

where $\bar{A}$ is a full-rank matrix. By minimizing the TD error, the optimal network weights can be obtained through the least-squares method, which is:

$$W_{ic} = \bar{A}^{-1}\bar{B} \tag{12}$$

Based on the above analysis, the KLSPI method is employed to learn local policies for AVs from collected high-dimensional samples. This method enables real-time derivation of the local utility function p_i during the online deployment phase, which constitutes a component of the i-th vehicle's local state-action value function. Then, the optimized decision action can be computed online based on the derived function.

3.3. Online decision-making using coordination graphs

In the proposed LCDMC method for MAVs, a cooperative relationship is established among vehicles across different lanes. The Max-plus message passing mechanism is utilized to iteratively propagate local joint-action rewards through cooperative communication edges, enabling optimized decision-making for MAVs.

The cooperative interactions among AVs are inherently dynamic due to the multi-vehicle traffic flow dynamics. At unsignalized intersections, an undirected edge between AVs i and j is established if their paths may intersect. The potential path intersection is determined based on the distance and speed of the preceding and following vehicles once an AV enters the decision zone[40]. As shown in Figure 4, when a left-turning AV enters the decision zone, it observes the speed and distance of the nearest leading and following AVs in the straight lane. If no AV is detectable within the perception range, the distance and speed of neighboring vehicles are set to a predefined large constant. The position coordinate of AV i is represented as $z_i = [x_i, y_i]^\top$. Based on potential path intersections, the neighbors of vehicle i are defined as follows

$$\mathcal{N}_i = \{j : j \neq i, (z_i \in A_i \vee z_j \in A_j)\} \tag{13}$$

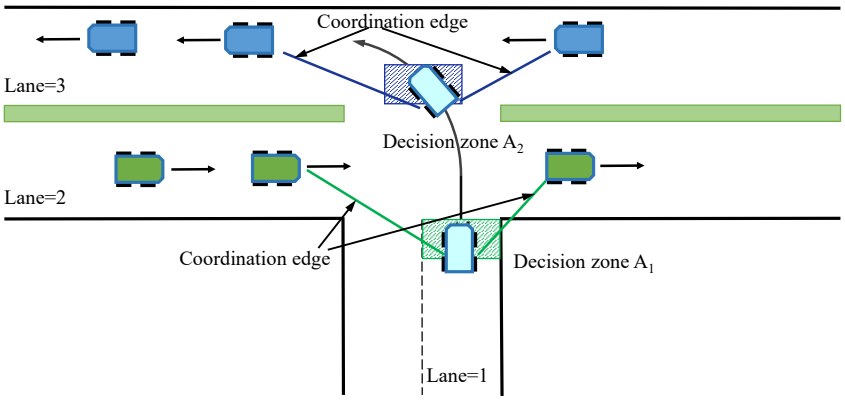

Figure 4. Cooperative relationships in MAV systems. MAV: Multiple autonomous vehicle.

where A_i and A_j denote the decision zones of vehicles i and j in their respective lanes, as shown by the shaded regions in Figure 4.

After AV i acquires coordination edges and neighboring vehicles, the local collaborative state is represented as $\Psi_i(t) = [z_i^\top, z_j^\top]^\top$, where $j \in \mathcal{N}_i$. Upon executing the joint action, the collaborative state transitions to $\Psi_i(k+1)$. The collaborative utility function between adjacent AVs in the current state is defined as

$$p_{ij}(a_i, a_j) = |\bar{z}_i(k+1) - \bar{z}_j(k+1)| \tag{14}$$

where $\bar{z}_h(k+1)$, $(h = i, j)$ represents the predicted distance of vehicle h to the target position $z_m = [x_m, y_m]^\top$, and

$$\bar{z}_h(k+1) = \sqrt{(dx_h(k+1))^2 + (dy_h(k+1))^2} \tag{15}$$

where $dx_h(k+1) = x_h(k+1) - x_m$, and $dy_h(k+1) = y_h(k+1) - y_m$, with $h = i, j$.

The term $p_{ij}(a_i, a_j)$ characterizes the influence of joint actions between AVs in cooperative settings. Utilizing dynamic coordination graphs, the Max-plus message propagation mechanism enables AVs to determine optimal joint actions that maximize the predicted distance. Specifically, the mechanism prioritizes vehicles closer to their target positions for intersection crossing, while guaranteeing collision avoidance. This approach facilitates safe and coordinated multi-vehicle navigation at unsignalized intersections.

The computational complexity of the coordination graph primarily stems from two components: coordination edge updates and message propagation. Coordination edge updates refer to the distance computation between vehicles during each path intersection detection, while message propagation denotes the iterative process of the Max-plus algorithm along each edge until convergence. The complexity is $O(km(n+m)|\epsilon|)$, where k denotes the number of message-passing iterations, m is the action space size of a single agent, N and $|\epsilon|$ represent the number of agents and coordination edges[36], respectively.

The forming and dissolving of edges in the coordination graph are primarily determined by the vehicles' traversal tasks and spatial correlations. The specific rules are as follows. Edge forming triggered by paths intersect: each lane at an intersection has a designated decision zone. When a vehicle enters this zone, its onboard sensors detect the speed and position of leading and following vehicles in adjacent lanes. If the detected vehicles' trajectories spatially intersect, a coordination edge is established, indicating the need for cooperative decision-making. Edge dissolving triggered by complete traversal or distance threshold: the system removes the corresponding coordination edge when an AV either completes cooperative traversal or exits the decision zone, thereby reducing computational complexity.

3.4. RHRL controller for real-time trajectory tracking

A RHRL approach is designed to achieve trajectory tracking control of the AVs in this section, with the reference velocity profile determined through multi-vehicle cooperative decision-making. The kinematic model of an Ackermann-steering AV is given as

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\varphi}_i \end{bmatrix} = \begin{bmatrix} \cos\varphi_i \\ \sin\varphi_i \\ \tan\delta_i / L \end{bmatrix} v_i \tag{16}$$

where x_i, y_i, φ_i represent the vehicle's x-coordinate, y-coordinate, and yaw angle in the global coordinate system, respectively; and δ_i,v_i, L denote the front wheel steering angle, longitudinal velocity, and wheelbase, respectively. The tracking error is defined as $\bar{e}(k) = z_i(k) - r_i(k)$, where $z_i = [x_i, y_i, \varphi_i]^\top$, $r_i = [x_{ir}, y_{ir}, \varphi_{ir}]^\top$ represents the actual and reference trajectory. Furthermore, transforming the tracking error into the local coordinate system yields

$$e_i(k) = \begin{bmatrix} e_{ix} \\ e_{iy} \\ e_{i\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi_i & \sin\varphi_i & 0 \\ -\sin\varphi_i & \cos\varphi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{e}_i(k) \tag{17}$$

By combining Equations (16) and (17), the tracking error dynamics can be discretized as follows:

$$\begin{cases} e_{ix}(k+1) = e_{ix}(k) + \Delta t(v_i(k)\dfrac{\tan\delta_i(k)}{L} e_{iy}(k) - v_i(k) + v_{ir}(k)\cos e_{i\varphi}(k)) \\ e_{iy}(k+1) = e_{iy}(k) + \Delta t(-v_i(k)\dfrac{\tan\delta_i(k)}{L} e_{ix}(k) + v_{ir}(k)\sin e_{i\varphi}(k)) \\ e_{i\varphi}(k+1) = e_{i\varphi}(k) + \Delta t(-v_i(k)\dfrac{\tan\delta_i(k)}{L} + v_{ir}(k)\dfrac{\tan\delta_{ir}}{L}) \end{cases} \tag{18}$$

where v_{ir} denotes the reference velocity corresponding to the decision action, Δt represents the sampling interval, and δ_{ir} indicates the front wheel steering angle of the reference trajectory. For the tracking control problem of the nonlinear discrete-time system (16), a finite horizon performance index is defined as

$$J_i(e_i(k)) = \sum_{\tau=k}^{k+H-1} U_i(e_i(\tau), u_i(\tau)) + e_i^\top(k+H) P_i e_i(k+H) \tag{19}$$

where $U_i(e_i(\tau), u_i(\tau))$ represents the reward function for vehicle i, $U_i(e_i(\tau), u_i(\tau)) = e_i^\top(\tau)\hat{Q}_i e_i(\tau) + u_i^\top(\tau)\hat{\mathcal{R}}_i u_i(\tau)$, where $\hat{Q}$ and $\hat{\mathcal{R}}$ are predefined positive definite weighting matrices. The prediction horizon length H is a positive integer. The terminal penalty function is given as $e_i^\top(k+H)P_i e_i(k+H)$, and the terminal penalty matrix P_i can be obtained by solving the Riccati equation for the linearized discrete-time system [41].

At the sampling instant k, the learning strategy is implemented over the prediction horizon $[k, k+H-1]$, and the corresponding value function is expressed as

$$\begin{aligned} &V_i^*(e_i(\tau)) = U_i(e_i(\tau), u_i(\tau)) + V_i^*(e_i(\tau+1)), \tau \in [k, k+H-1] \\ &V_i^*(e_i(\tau)) = e_i(\tau)^\top P_i e_i(\tau), \tau = k+H \end{aligned} \tag{20}$$

The costate of the value function is defined as $\lambda_i(\tau) = \partial V_i(\tau)/\partial e_i(\tau)$, and the optimal costate is given as

$$\begin{aligned} &\lambda_i^*(\tau) = 2\hat{Q}_i e_i(\tau) + \left[\frac{\partial e_i(\tau+1)}{\partial e_i(\tau)}\right]^\top \lambda_i^*(\tau+1), \tau \in [k, k+H-1] \\ &\lambda_i^*(k+H) = 2P_i e_i(k+H) \end{aligned} \tag{21}$$

Then, the optimal control output is defined as

$$\begin{aligned} u_i^*(e_i(\tau)) &= \arg\min_{u_i}(U_i(e_i(\tau), u_i(\tau)) + V_i^*(e_i(\tau+1))) \\ &= -\frac{1}{2}\hat{\mathcal{R}}_i^{-1}\left[\frac{\partial(e_i(\tau+1))}{\partial u_i^*(\tau)}\right]^\top \lambda_i^*(\tau+1) \end{aligned} \tag{22}$$

Due to the difficulty in obtaining analytical expressions of the control input for nonlinear systems, an actor-critic RL framework is adopted to learn the policy within the prediction horizon. This framework utilizes a value iteration approach, in which neural networks are employed to approximate the costate function and the optimal control input. The structure of the critic network is expressed as:

$$\hat{\lambda}_i(\tau) = W_{ci}^{\top}\sigma_c(e_i(\tau)) \tag{23}$$

where W_{ci} and $\sigma_c(\cdot)$ represent the weight matrices from the hidden layer to the output layer, and the activation function of the critic network, respectively. The TD error of the critic network is defined as $\eta_{ci}(\tau) = \hat{\lambda}_i(e_i(\tau)) - \bar{\lambda}_i(e_i(\tau))$, where

$$\begin{aligned} &\bar{\lambda}_i(\tau) = 2\hat{Q}_i e_i(\tau) + \left[\frac{\partial(e_i(\tau+1))}{\partial e_i(\tau)}\right]^{\top}\hat{\lambda}_i(\tau+1), \tau \in [k, k+H-1] \\ &\bar{\lambda}_i(k+H) = 2P_i e_i(k+H) \end{aligned} \tag{24}$$

The critic network approximates the optimal costate by minimizing the error function $E_{ci}(\tau) = \frac{1}{2}(\eta_{ci}(\tau))^2$. Using the gradient descent method, the update rule for the network weights is formulated as

$$W_{ci}(\tau+1) = W_{ci}(\tau) - \alpha_c \eta_{ci}(\tau)\frac{\partial \hat{\lambda}_i(\tau)}{\partial W_{ci}(\tau)} \tag{25}$$

where $0 < \alpha_c < 1$ is the learning rate of the critic network.

The actor network of the AV is a three-layer neural network that takes the tracking error e_i as input and approximates the control input u_i. Its structure is as follows

$$\hat{u}_i(\tau) = W_{ai}^{\top}\sigma_a(e_i(\tau)) \tag{26}$$

where W_{ai} and $\sigma_a(\cdot)$ represent the weight matrices from the hidden layer to the output layer, and the activation function of the actor network, respectively. The approximation error of the actor network is defined as $\eta_{ai}(\tau) = \hat{u}_i(\tau) - \bar{u}_i(\tau)$, with $\bar{u}_i(\tau) = -\frac{1}{2}\hat{\mathcal{R}}_i^{-1}\left[\frac{\partial(e_i(\tau+1))}{\partial u_i(\tau)}\right]^{\top}\hat{\lambda}_i(\tau+1)$. The actor network approximates the optimal control input by minimizing the error function $E_{ai}(\tau) = \frac{1}{2}(\eta_{ai}(\tau))^2$. Based on the gradient descent method, the update rule for the network weights is given as

$$W_{ai}(\tau+1) = W_{ai}(\tau) - \alpha_a \eta_{ai}(\tau)\frac{\partial \hat{u}_i(\tau)}{\partial W_{ai}(\tau)} \tag{27}$$

where $0 < \alpha_a < 1$ is the learning rate of the actor network.

Employing the RHRL approach, an optimized control sequence is learned within the prediction horizon at time step k, and the first element is applied to the system. The actor-critic networks maintain continuity between adjacent prediction horizons by leveraging prior experience during learning, thereby accelerating policy convergence and improving real-time computational efficiency.

The online learning computational complexity of the RHRL approach stems from Equations (25) and (27), which is approximately $O(H(n_{ci}+n_{ui}+n_i)n_i)$, where H, n_i, n_{ci}, and n_{ui} denote the prediction horizon length, the dimension of input layer, the number of hidden layer neurons in the critic and the actor network, respectively[41]. Additionally, the proposed approach obtains optimized control policies with an explicit structure, enabling direct deployment of the learned policy with reduced complexity $O(n_{ui}n_i)$, thus meeting the real-time control requirements of AVs.

Based on the above analysis, the offline learning phase of the proposed method employs KLSPI to learn local policies from high-dimensional collected samples, which improve the computational efficiency and policy

Algorithm 1 The LCDMC Algorithm

Require: maximum simulation step $k_{\max}$, maximum number of iterations $l_{\max}$ for policy iteration, convergence threshold ϱ;

Ensure: Decision action a_i for i-th vehicle;

Offline Phase

// Collect samples

for $k = 1$ to $k_{\max}$ **do**

  Randomly initialize the vehicle's distance within the range $[0, d_{\max}]$ and speed within $[0, v_{\max}]$, then record $s_i(k) \leftarrow [v_i, v_{fi}, d_{fi}, v_{ri}, d_{ri}]$;

  Randomly select an action $a_i(k)$, and compute the control law $u_i(k) = \pi(s_i(k))$;

  Determine the reward R_i via Equation (6), then store the sample $\mathcal{S} \cup [s_i(k), a_i(k), s_i(k+1), R_i]$;

end for

// Train policy

Extract the sample features using Equation (7);

for $l = 1$ to $l_{\max}$ **do**

  Construct the approximation structure of the local utility function via Equation (9);

  Update the policy W_{ic} using Equation (12);

  if $\left\|W_{ic}^{l} - W_{ic}^{l-1}\right\| < \varrho$ **then**

    $W_{ic}^{*} = W_{ic}^{l}$;

    break;

  end if

end for

// Online Deployment

Compute the local collaborative utility function via Equation (14);

Determine the message value using Equation (3);

Calculate the state-action value function Q_i of the i-th vehicle via Equation (4).

Obtain the optimal decision action for the vehicle: $a_i^{*} = \arg\max_{a_i} \; Q_i(a_i)$;

Update the state of the i-th vehicle using the control policy in Equation (26).

convergence. During the online deployment phase, when AV i enters the decision zone, communication edges with neighboring vehicles j on adjacent lanes are established. A collaborative utility function $p_{ij}(a_i, a_j)$ is introduced, which characterizes the performance of joint actions between vehicles i and j. Simultaneously, vehicle i deploys the learned policy and obtains the local utility function $p_i(a_i)$ for different decisions. The final decisions are determined by combining the local utility function $p_i(a_i)$ with the collaborative utility function $p_{ij}(a_i, a_j)$ through the iterative message passing Equation (3). The optimized decision output serves as the reference signal for RHRL-based trajectory tracking control, ensuring safe and efficient intersection traversal. The proposed algorithm is summarized in Algorithm 1.

4. SIMULATION VERIFICATIONS

This section validates the efficacy and superiority of the proposed LCDMC method for MAVs at unsignalized intersections. In these scenarios, AVs in each lane must monitor the trajectories of neighboring vehicles with potential conflicts and employ appropriate coordination mechanisms to ensure safe and efficient traversal to their target lanes. In the simulation, the lane width, vehicle length, and vehicle width are set as 3.75, 4, and 1.54 m, respectively; the sampling interval is $\Delta t = 0.1$ s, and the sample set size per vehicle is $N_{\text{sam}} = 30,000$.

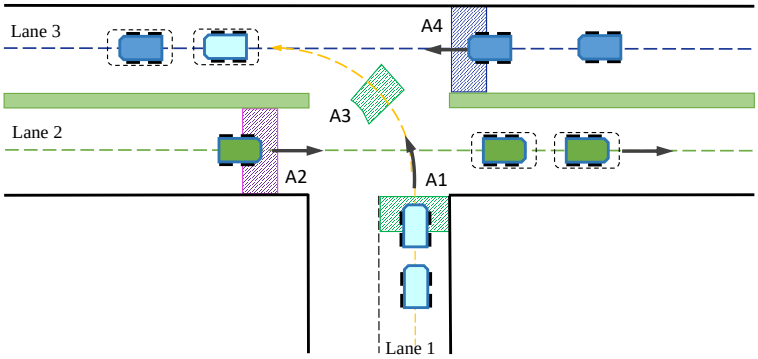

Figure 5. Illustration of MAVs intersection traversal scenario 1. MAVs: Multiple autonomous vehicles.

Table 2. The parameters for local policy learning

Lane	γ_i	Kernel width	μ_i	Iteration error threshold
Lane 1	0.95	20	0.6	10^{-5}
Lane 2	0.95	30	0.5	10^{-5}
Lane 3	0.95	5	0.5	10^{-5}

4.1. Scenario 1: MAVs coordination at an unsignalized T-shaped intersection

In scenario 1, AVs in Lane 1 perform left turns, while AVs in Lanes 2 and 3 continue straight, as depicted in Figure 5. During navigation, vehicles estimate the motion states of neighboring AVs. By establishing coordination edges with neighbors, they compute optimal actions for safe intersection traversal. The state definitions for AVs in scenario 1 are specified as follows: When an AV in Lane 1 enters decision zone $A1$, it establishes coordination edges with AVs in Lane 2. By observing the velocity and relative distances of vehicles in Lane 2, the state vector $s_1(k) = [v_1, v_{f2}, d_{f2}, v_{r2}, d_{r2}]$ is recorded. Upon entering decision zone $A2$, the AV in Lane 2 establishes coordination edges with AVs in Lane 1, acquiring its state vector $s_2(k) = [v_2, v_{f1}, d_{f1}, v_{r1}, d_{r1}]$. When an AV in Lane 1 enters decision zone $A3$, it establishes coordination edges with AVs in Lane 3 and acquires its state vector $s_1(k) = [v_1, v_{f3}, d_{f3}, v_{r3}, d_{r3}]$. Similarly, when an AV in Lane 3 enters decision zone $A4$, it observes the states of vehicles in the left-turn lane and establishes coordination. The dashed bounding boxes mark AVs that have completed their traversal tasks.

During the sampling phase for scenario 1, vehicle states are sampled according to the parameters in Table 1. Each vehicle randomly selects an action (acceleration, deceleration, or maintaining speed) and executes it for 5 s. The updated state vector $s_i(k+1)$ is then recorded, including the ego vehicle's velocity, neighboring vehicles' velocities, and their relative distances. The reward $R_i(k)$ is computed based on the state transition via Equation (6), yielding the experience tuple $[s_i(k), a_i(k), s_i(k+1), R_i(k)]$ for policy learning. The minimum safe distance is set to 3 m, with collisions detected when longitudinal inter-vehicle distances in adjacent lanes fall below this threshold. Upon collecting N_{sam} samples per vehicle, the ALD method performs kernel sparsification to extract model features, with training parameters listed in Table 2. The kernel dictionary dimensions obtained via ALD are 193, 240, and 323 for the three lanes, respectively. The policy converges after 7 offline training iterations, with the weight convergence shown in Figure 6. The learned policy is subsequently deployed at unsignalized intersections to evaluate the performance of the proposed LCDMC algorithm. Figure 7 shows the steering angle responses for vehicles across different lanes, where red and green curves represent leading and following AVs, respectively.

Figure 8 demonstrates the cooperative decision-making and motion trajectories at typical time instants in scenario 1, where **sd** and **acc** represent deceleration and acceleration, respectively. Initially, vehicles in each lane

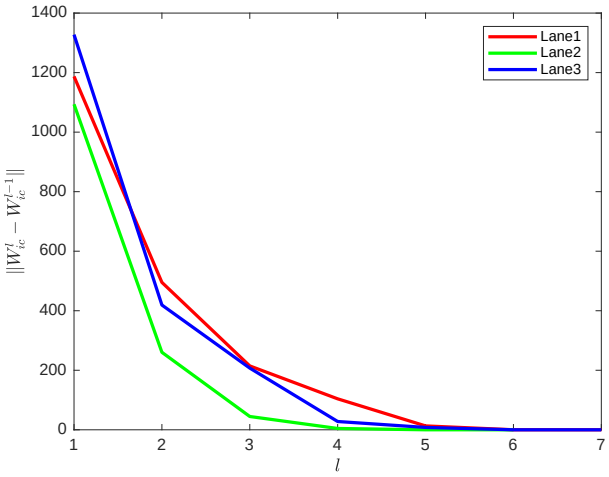

Figure 6. Convergence process of local policy learning via KLSPI. KLSPI: Kernel-based least-squares policy iteration.

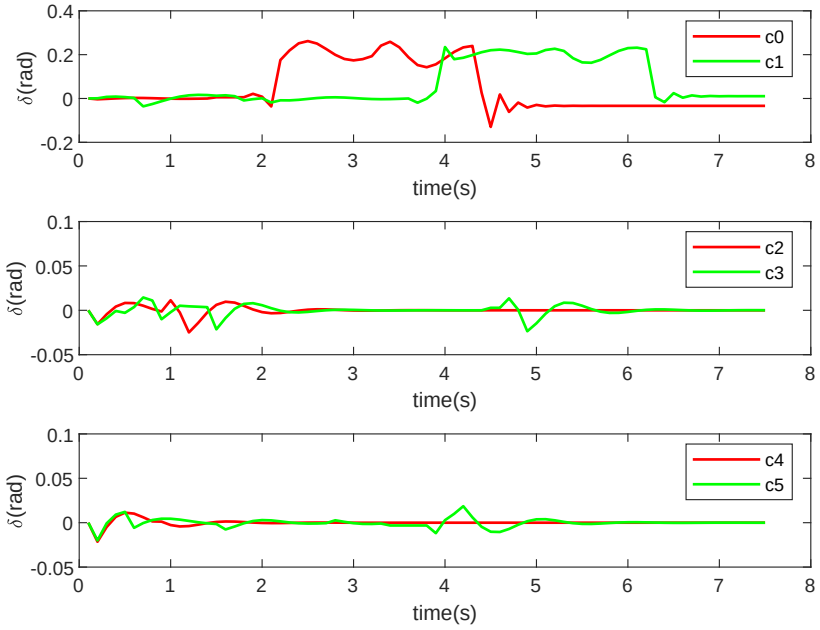

Figure 7. The steering angles of the leading and following AVs on different lanes in scenario 1. AVs: Autonomous vehicles.

are randomly positioned within $[0, 50]$m of the intersection, with randomly generated velocities in $[0, 10]$m/s. Before entering the decision zones, all vehicles maintain tracking control with a constant speed of 8 m/s. At t = 1.3 s, the red-marked vehicle c0 on Lane 1 enters the decision zone $A1$ and detects the blue-marked vehicle c2 on Lane 2 near the convergence point. By integrating p_i with the collaborative utility function p_{ij}, the Max-plus algorithm derives optimal actions:c0 decelerates while c2 accelerates through the zone. At t = 3.3 s, vehicle c0 enters decision zone A_3 and observes a preceding vehicle on Lane 3. Given that c0 is closer to the convergence point than the trailing vehicle c5 on Lane 3, the control strategy determines that c0 should accelerate while c5 should decelerate. At t = 4.4 s, black-marked vehicle c3 on Lane 1 enters the decision zone. Detecting that its preceding vehicle is still traversing the zone and no trailing vehicles are within sight, the LCDMC assigns deceleration to c3 and acceleration to green-marked vehicle c1. At t = 4.9 s, c3 observes that the preceding vehicle c1 has cleared the intersection, while c1 detects vehicle c5 ahead with no trailing vehicles.

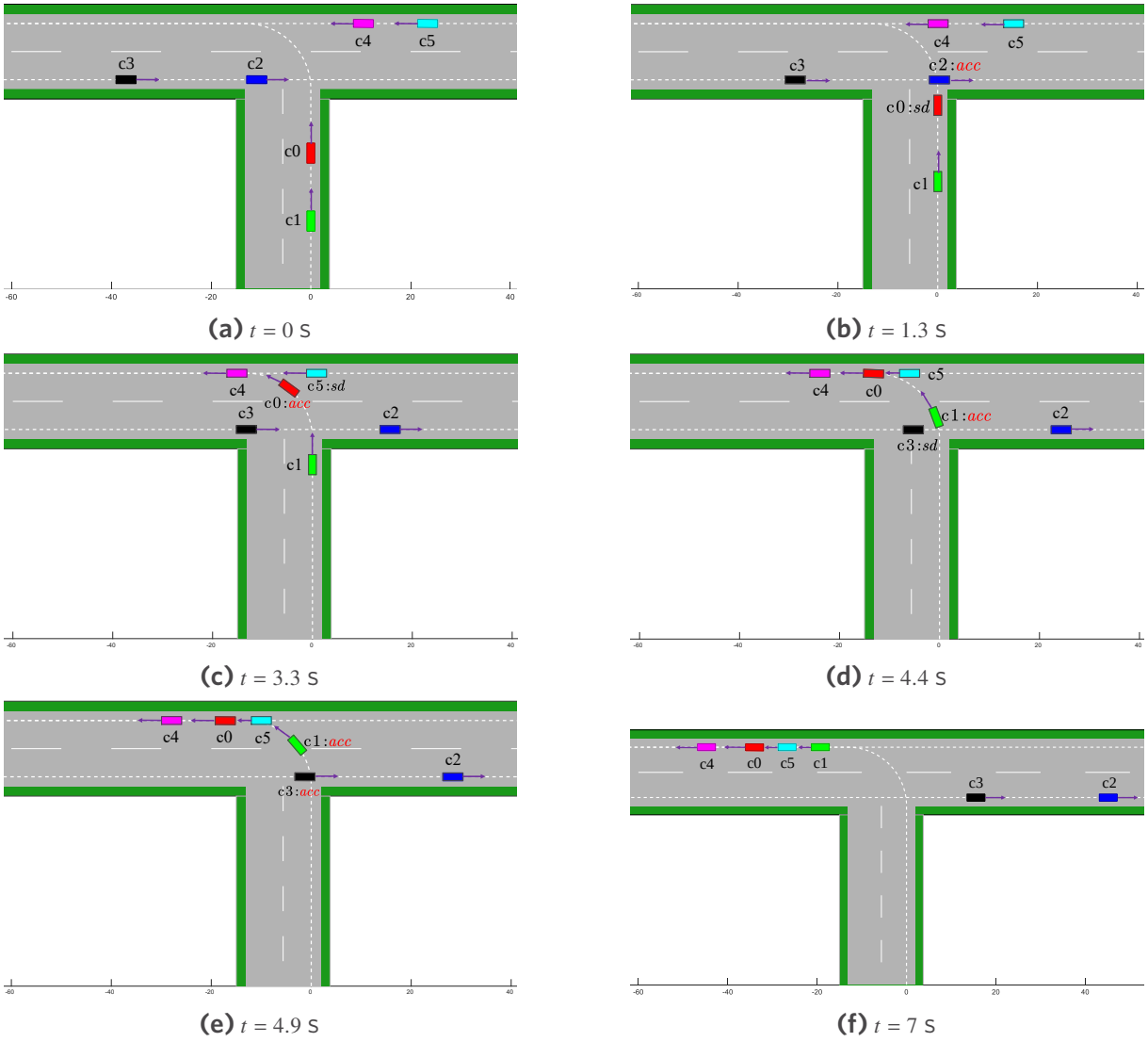

Figure 8. The cooperative decision-making and motion trajectories of MAVs at typical time instants in scenario 1. MAVs: Multiple autonomous vehicles.

Both c3 and c1 execute acceleration. By t = 7 s, all AVs have successfully negotiated the intersection.

From Figures 7 and 8, it can be observed that the following vehicles dynamically adjust their acceleration based on the velocity and distance of the leading vehicles, while considering their own speed. This behavior is consistent with the car-following model[35] and effectively prevents collisions.

The performance metrics of the algorithms are statistically analyzed to demonstrate the superiority of the proposed approach. Each algorithm is evaluated over 1,000 test trials. The performance metrics, including average execution time and its standard deviation, collision rate, failure rate, and comfort level, are presented in Table 3. Traversal time denotes the duration for all vehicles to completely pass the intersection and reach their target lanes. Collision rate counts the percentage of collision occurrences across 1,000 trials. Failure rate represents the percentage of trials in which vehicles remain within the intersection after 10 s, failing to complete traversal. Comfort is quantified as the average root mean square (RMS) of vehicle accelerations, where lower values indicate higher comfort. Independent learning is a decentralized RL approach in which multiple agents

Table 3. Performance metrics statistics for scenario 1

Method	Travel time		Collision rate (%)	Failure rate (%)	Comfort
	Avg. Time (s)	Std. Dev.			
LCDMC	**7.3976**	**0.3340**	**0**	**0**	**0.4877**
Independent learning	8.8453	0.8298	22.8	0.91	0.5435

LCDMC: Learning-based cooperative decision-making and control.

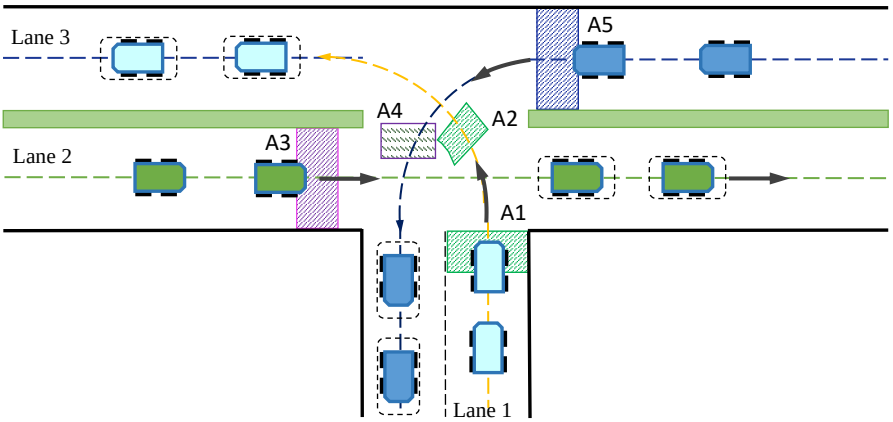

Figure 9. Illustration of MAVs intersection traversal scenario 2. MAVs: Multiple autonomous vehicles.

independently learn their state-action value functions without coordination or communication[38], treating others as part of the environment and updating their functions based on individual observations and rewards. The independent learning method[38] deploys identical policies to those in the proposed LCDMC approach, it does not consider the performance of joint actions.

As demonstrated in Table 3, the proposed LCDMC method achieves higher traffic throughput efficiency compared to independent learning approaches at unsignalized intersections. This improvement is attributed to LCDMC's cooperative decision-making mechanism, which evaluates joint-action performance through the Max-plus message propagation mechanism, facilitating timely and rational decisions that reduce traversal time. In contrast, in independent learning approaches, vehicles prioritize self-interest by maximizing individual p_i values, leading to aggressive traversal attempts and frequent collisions. Conversely, the collaborative utility function in LCDMC is designed to prevent collisions while improving overall efficiency. Across 1,000 trials, independent learning exhibited larger standard deviations in intersection traversal time and more frequent exceedances of the 10 s timeout threshold. The LCDMC algorithm achieved smoother velocity profiles and superior comfort metrics. These results demonstrate that LCDMC outperforms independent learning in multi-vehicle intersection scenarios in terms of traffic throughput efficiency, collision avoidance and motion stability.

4.2. Scenario 2: MAVs coordination at a complex unsignalized T-shaped intersection

Scenario 2 extends scenario 1 by introducing more complexity in the intersection traffic conditions. In this scenario, AVs in Lane 2 proceed straight, while AVs in Lane 1 and Lane 3 perform left turns, as illustrated in Figure 9. During traversal, AVs in different lanes must consider the motion states of neighboring vehicles to establish cooperative relationships and make rational decisions for safe intersection crossing. The learned policy is deployed in the unsignalized T-junction scenario to evaluate the performance of the LCDMC algorithm. Figure 10 shows the coordinated decision-making and motion trajectories at typical timestamps in scenario 2.

Similar to scenario 1, we conducted 1,000 test trials for cooperative decision-making for MAVs in scenario 2 to

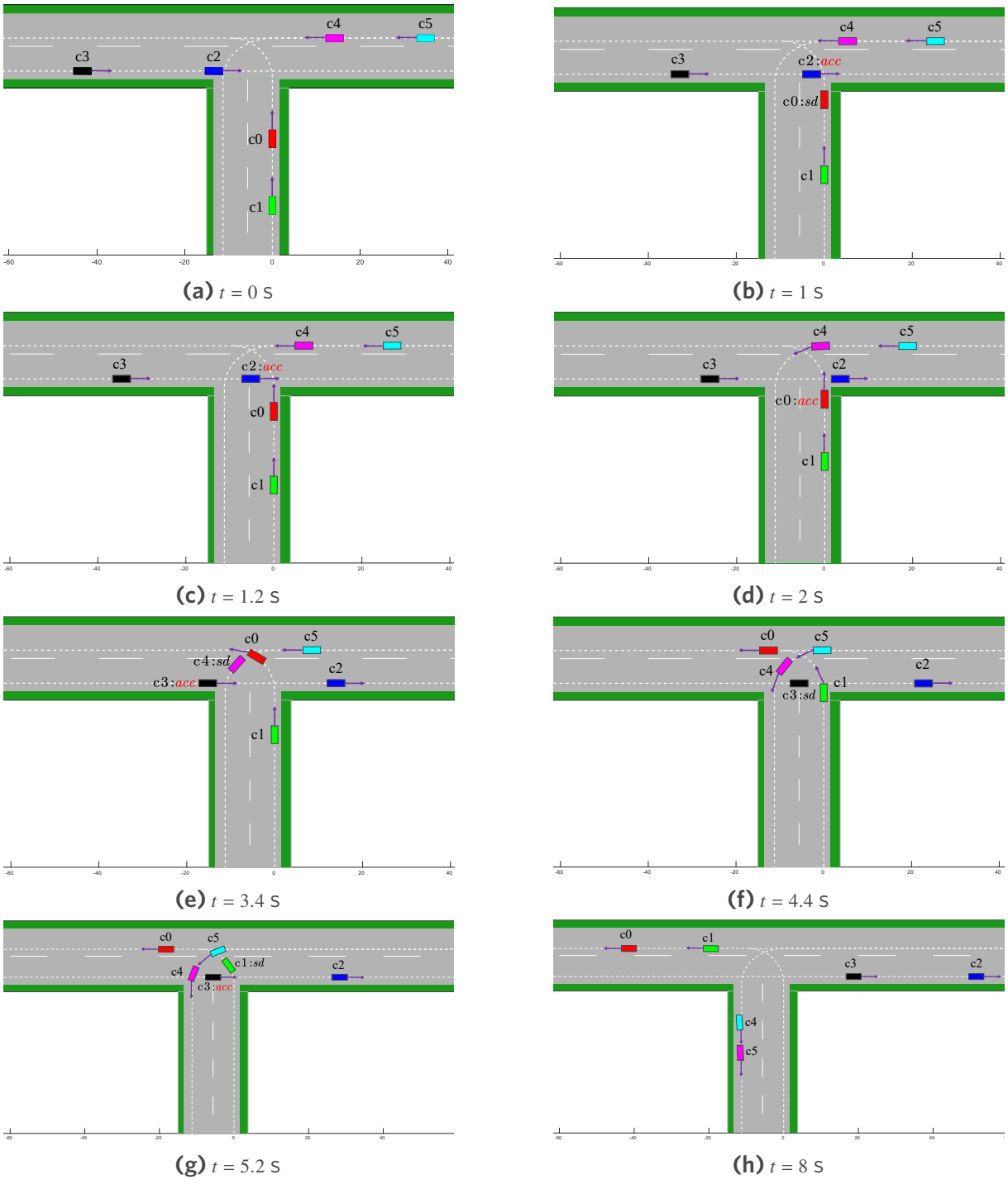

Figure 10. The cooperative decision-making and motion trajectories of MAVs at typical time instants in scenario 2. MAVs: Multiple autonomous vehicles.

validate the algorithm's superiority. Statistical results of performance metrics are listed in Table 4. The results demonstrate that LCDMC achieves higher traffic throughput efficiency than independent learning at unsignalized intersections. This improvement stems from LCDMC's coordinated decision-making mechanism, which evaluates joint-action performance during policy deployment and leverages message propagation to achieve time-optimal decisions, which reduce traversal time. In contrast, the vehicle action policy in independent

Table 4. Performance metrics statistics for scenario 2

Method	Travel time		Collision rate (%)	Failure rate (%)	Comfort
	Avg. Time (s)	Std. Dev.			
LCDMC	**8.1132**	0.4733	**0**	**0.1**	0.8572
Independent learning	8.9317	**0.4618**	27.2	4.1	**0.6110**

LCDMC: Learning-based cooperative decision-making and control.

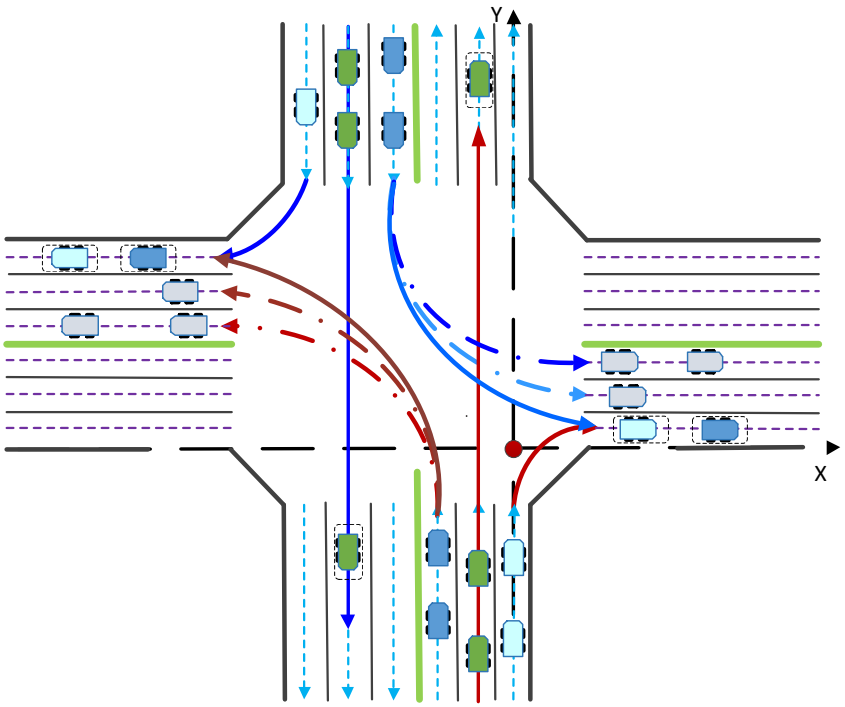

Figure 11. Illustration of MAVs intersection traversal scenario 3. MAVs: Multiple autonomous vehicles.

learning prioritizes individual rewards, leading to aggressive maneuvers and frequent collisions. LCDMC mitigates this issue through a joint-action utility function that reflects the joint-action performance. Across 1,000 trials, LCDMC achieved a lower collision rate compared to independent learning. It should be noted that LCDMC's collision avoidance strategy occasionally requires deceleration commands, resulting in slightly worse comfort metrics than independent learning. Overall, the results demonstrate LCDMC's superiority in multi-vehicle coordination efficiency and safety for complex intersection navigation.

4.3. Scenario 3: MAVs coordination at an unsignalized cross intersection

This subsection evaluates the proposed LCDMC algorithm for unsignalized cross intersections. The scenario is illustrated in Figure 11, where AVs perform straight-through, right-turn, and left-turn maneuvers in mixed traffic. Effective navigation requires cross-lane coordination mechanisms to enable optimized decision-making, ensuring safe and efficient crossing. Figure 12 depicts the coordinated decision-making process and resulting motion trajectories at typical time instants in scenario 3.

Subsequently, the performance metrics of different algorithms are calculated to validate the superiority of the proposed approach. Each method undergoes 100 trials, and the performance metrics are recorded in Table 5. The results show that independent learning exhibits a higher standard deviation in traversal time compared to the proposed LCDMC algorithm. Independent learning yields a collision rate of 26% and a failure rate of 14.5%, where failure is defined as exceeding the 13 s timeout threshold. LCDMC achieves

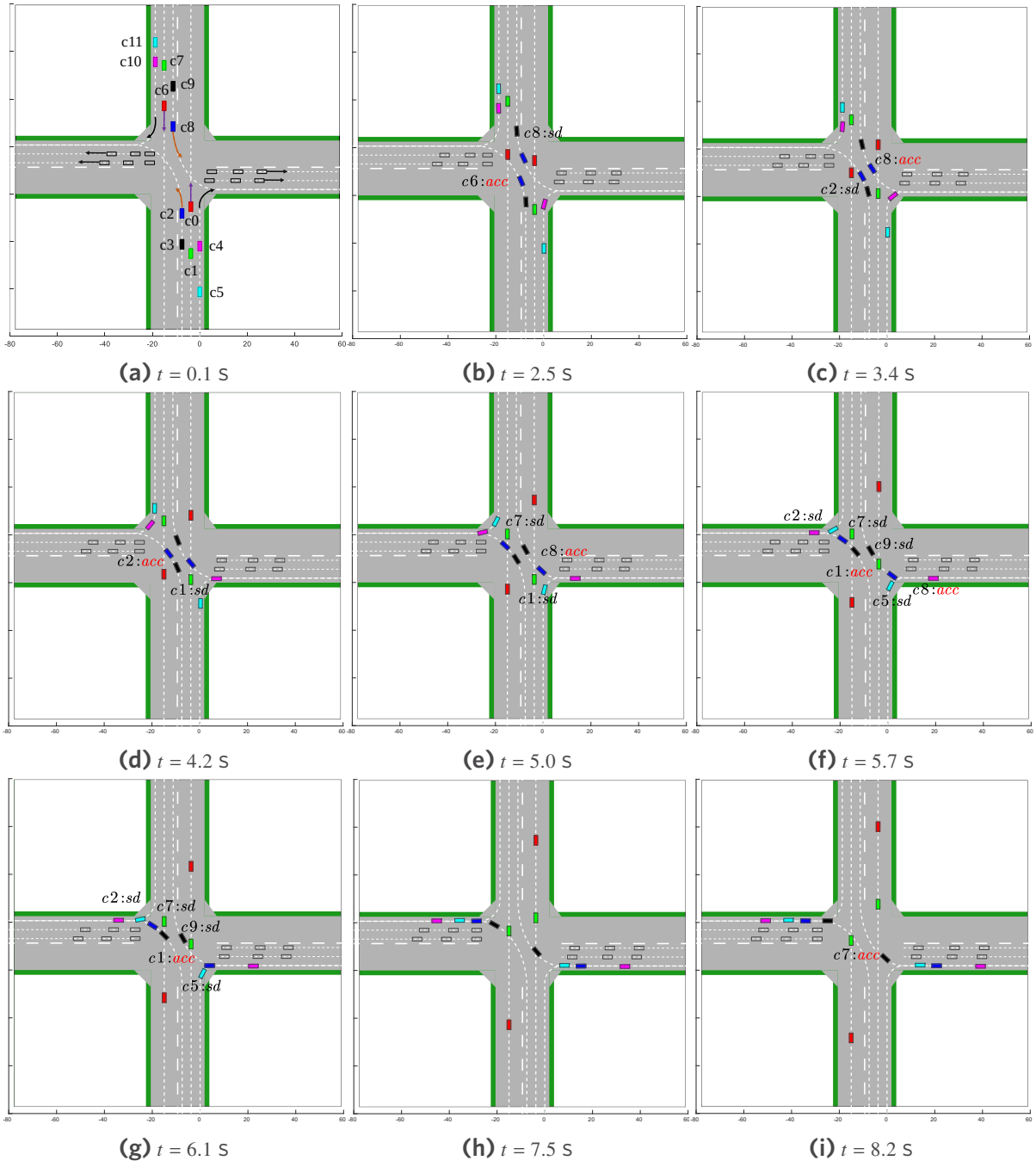

Figure 12. The cooperative decision-making and motion trajectories of MAVs at typical time instants in scenario 3. MAVs: Multiple autonomous vehicles.

better comfort performance compared to independent learning. Overall, LCDMC outperforms independent learning for MAVs in intersection scenarios in terms of traffic throughput efficiency, collision avoidance and driving smoothness. These results align with scenarios 1 and 2.

The LCDMC algorithm integrates KLSPI to compute the local utility function p_i. Meanwhile, the collaborative utility function $p_{ij}(a_i, a_j)$ predicts the joint-action performance of AVs in adjacent lanes, enabling coordinated decision-making. Message propagation across coordination graphs drives an optimized decision policy. The resulting decision is converted into a reference velocity for AV trajectory tracking control via a RHRL approach.

Table 5. Performance metrics statistics for scenario 3

Method	Travel time		Collision rate (%)	Failure rate (%)	Comfort
	Avg. Time (s)	Std. Dev.			
LCDMC	**11.0440**	**0.3063**	**0**	**0**	**0.7136**
Independent learning	11.8831	0.9657	26	14.5	0.7260

LCDMC: Learning-based cooperative decision-making and control.

Multiple simulation scenarios demonstrate that the proposed LCDMC method improves policy learning efficiency and ensures safe and efficient cooperative navigation of MAVs at unsignalized intersections.

5. CONCLUSIONS

Aimed at the low traffic efficiency of multi-vehicle navigation at unsignalized intersections, a LCDMC method for MAVs is proposed. LCDMC approach enhances the representational capacity of value functions in multi-agent RL, thereby improving the coordination performance in decision-making and control. The LCDMC algorithm comprises an offline policy learning phase and an online deployment phase. It decomposes the system's global value function into two components: a local state-action value function and a utility function for the joint actions of vehicles. Unlike independent learning methods that solely focus on individual performance, the LCDMC method introduces a coordination graph during the policy deployment phase to effectively characterize the performance of joint actions among MAVs. The optimized policy is derived through iterative message passing among neighboring AVs with cooperative relationships. Moreover, a receding-horizon-based controller is designed, which can enhance the system's real-time performance. Finally, various simulation results demonstrate that the LCDMC method exhibits superior performance in both traffic efficiency and safety for cooperative navigation of MAVs at unsignalized intersections.

DECLARATIONS

Authors' contributions

Made equal contributions to conception and design of the study and simulation and interpretation: Zhang, R.; Xu, X.

Technical and material support, draft improvement: Lu, Y.

Conceptualization, supervision: Xu, X.; Zhang, X.

Supervision, draft improvement: Ma, Q.

Availability of data and materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Financial support and sponsorship

None.

Conflicts of interest

Xu, X. (Xin Xu) is Associate Editor of the journal *Intelligence & Robotics*. Xu, X. (Xin Xu) was not involved in any steps of editorial processing, notably including reviewers' selection, manuscript handling and decision making. The other authors declare that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Zhao, J.; Zhao, W,; Deng, B.; et al. Autonomous driving system: a comprehensive survey. *Expert. Syst. Appl.* **2024**, *242*, 122836. DOI
2. Tampuu, A.; Matiisen, T.; Semikin, M.; Fishman, D.; Muhammad, N. A survey of end-to-end driving: architectures and training methods. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1364–84. DOI
3. Zhang, R.; Sun, C.; Valiollahimehrizi, R.; Czarnecki, K.; Khajepour, A. An uncertainty-aware, dual-tiered decision-making method for safe autonomous driving. *IEEE Trans. Intell. Transport. Syst.* **2025**, *26*, 691-702. DOI
4. Zhao, J.; Knoop, V. L.; Wang, M. Microscopic traffic modeling inside intersections: interactions between drivers. *Transp. Sci.* **2022**, *57*, 135–55. DOI
5. Spatharis, C.; Blekas, K. Multiagent reinforcement learning for autonomous driving in traffic zones with unsignalized intersections. *J. Intell. Transp. Syst.* **2024**, *28*, 103-19. DOI
6. Reda, M.; Onsy, A.; Haikal, A. Y.; Ghanbari, A. Path planning algorithms in the autonomous driving system: a comprehensive review. *Robot. Auton. Syst.* **2024**, *174*, 104630. DOI
7. Li, S.; Peng, K.; Hui, F.; Li, Z.; Wei, C.; Wang, W. A decision-making approach for complex unsignalized intersections by deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 16134–47. DOI
8. Al-Sharman, M.; Dempster, R.; Daoud, M. A.; Nasr, M.; Rayside, D.; Melek, W. Self-learned autonomous driving at unsignalized intersections: a hierarchical reinforced learning approach for feasible decision-making. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 12345–56. DOI
9. Xu, Y,; Bao, R.; Zhang, L.; Wang, J.; Wang, S. Embodied intelligence in RO/RO logistic terminal: autonomous intelligent transportation robot architecture. *Sci. China Inform. Sci.* **2025**, *68*, 150210. DOI
10. Li, X.; Liu, K.; Tseng, H. E.; Girard, A.; Kolmanovsky, I. Decision-making for autonomous vehicles with interaction-aware behavioral prediction and social-attention neural network. *IEEE Trans. Control Syst. Technol.* **2025**, *33*, 1285-300. DOI
11. Guan, Y.; Ren, Y.; Sun, Q.; et al. Integrated decision and control: toward interpretable and computationally efficient driving intelligence. *IEEE Trans. Cybern.* **2023**, *53*, 859–73. DOI
12. Peng, Z.; Wang, Y.; Zheng, L.; Ma, J. Bilevel multi-armed bandit-based hierarchical reinforcement learning for interaction-aware self-driving at unsignalized intersections. *IEEE Trans. Veh. Technol.* **2025**, *74*, 8824–38. DOI
13. Pan, X.; Chen, B.; Timotheou, S.; Evangelou, S. A. A convex optimal control framework for autonomous vehicle intersection crossing. *IEEE Trans. Intell. Transport. Syst.* **2023**, *24*, 163–77.DOI
14. Yuan, M.; Shan, J.; Mi, K. Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles. *IEEE Robot. Autom. Lett.* **2022**, *7*, 818–25. DOI
15. Li, N.; Yao, Y.; Kolmanovsky, I.; Atkins, E.; Girard, A. R. Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 1428–42. DOI
16. Rizk, Y.; Awad, M.; Tunstel, E. W. Decision making in multiagent systems: a survey. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *10*, 514-29. DOI
17. Mozaffari, S.; Al-Jarrah, O. Y.; Dianati, M.; Jennings, P.; Mouzakitis, A. Deep learning-based vehicle behavior prediction for autonomous driving applications: a review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 33–47. DOI
18. Wu, Y.; Chen, H.; Zhu, F. DCL-AIM: decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2019**, *103*, 246–60. DOI
19. Qian, X.; Altché, F.; Grégoire, J.; de La Fortelle, A. Autonomous intersection management systems: criteria, implementation and evaluation. *IET Intell. Transp. Syst.* **2017**, *11*, 182–89. DOI
20. Liu, J.; Hang, P.; Na, X.; Huang, C.; Sun, J. Cooperative decision-making for CAVs at unsignalized intersections: a MARL approach with attention and hierarchical game priors. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 443–56. DOI
21. Noh, S. Decision-making framework for autonomous driving at road intersections: safeguarding against collision, overly conservative behavior, and violation vehicles. *IEEE Trans. Ind. Electron.* **2019**, *66*, 3275-86. DOI
22. Ebert, J. T.; Gauci, M.; Mallmann-Trenn, F.; Nagpal, R. Bayes bots: collective Bayesian decision-making in decentralized robot swarms. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 31–Aug 31, 2020. IEEE, 2020. pp 7186–92. DOI
23. EL Bourakadi, D.; Yahyaouy, A.; Boumhidi, J. Multi-agent system based sequential energy management strategy for micro-grid using optimal weighted regularized extreme learning machine and decision tree. *Intell. Decis. Technol.* **2019**, *13*, 479–94. https://www.researchgate.net/publication/339188003_Multi-agent_system_based_sequential_energy_management_strategy_for_Micro-Grid_using_optimal_weighted_regularized_extreme_learning_machine_and_decision_tree. (accessed 21 Aug 2025)
24. Gronauer, S.; Dieopold, K. Multi-agent deep reinforcement learning: a survey. *Artif. Intell. Rev.* **2022**, *55*, 895-943. DOI
25. Tang, C.; Abbatematteo, B.; Hu, J.; Chandra, R.; Martín-Martín, R.; Stone, P. Deep reinforcement learning for robotics: a survey of

real-world successes. *Annu. Rev. Control Robot. Auton. Syst.* **2025**, *8*, 153-88. DOI

26. Wang, S.; Wang, Z.; Jiang, R.; Zhu, F.; Yan, R.; Shang, Y. A multi-agent reinforcement learning-based longitudinal and lateral control of CAVs to improve traffic efficiency in a mandatory lane change scenario. *Transp. Res. Part C Emerg. Technol.* **2024**, *158*, 104445. DOI
27. Zhang, J.; Chang, C.; Zeng, X.; Li, L. Multi-agent DRL-based lane change with right-of-way collaboration awareness. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 854–69. DOI
28. Wang, T.; Ma, M.; Liang, S.; Yang, J.; Wang, Y. Robust lane change decision for autonomous vehicles in mixed traffic: a safety-aware multi-agent adversarial reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* **2025**, *172*, 105005. DOI
29. Hu, X.; Liu, Y.; Tang, B.; Yan, J.; Chen, L. Learning dynamic graph for overtaking strategy in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11921–33. DOI
30. Chen, S.; Wang, M.; Song, W.; Yang, Y.; Fu, M. Multi-agent reinforcement learning-based decision making for twin-vehicles cooperative driving in stochastic dynamic highway environments. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12615–27. DOI
31. Chen, D.; Hajidavalloo, M. R.; Li, Z.; et al. Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11623–11638. DOI
32. Guo, Z.; Wu, Y.; Wang, L.; Zhang, J. Heuristic-based multi-agent deep reinforcement learning approach for coordinating connected and automated vehicles at non-signalized intersection. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 16235–48. DOI
33. Zhao, R.; Wang, K.; Li, Y.; Fan, Y.; Gao, F.; Gao, Z. Centralized cooperative control for autonomous vehicles at unsignalized all-directional intersections: a multi-agent projection-based constrained policy optimization approach. *Expert Syst. Appl.* **2025**, *267*, 126153. DOI
34. Li, G.; Yan, J.; Qiu, Y.; et al. Lightweight strategies for decision-making of autonomous vehicles in lane change scenarios based on deep reinforcement learning. *IEEE Trans. Intell. Transport. Syst.* **2025**, *26*, 7245-61. DOI
35. Treiber, M.; Kesting, A. Traffic flow dynamics: data, models and simulation. Berlin: Springer; 2013. DOI
36. Kok, J. R.; Vlassis, N. Collaborative multiagent reinforcement learning by payoff propagation. *J. Mach. Learn. Res.* **2006**, *7*, 1789–828. https://jmlr.org/papers/v7/kok06a.html. (accessed 21 Aug 2025)
37. Guestrin, C.; Lagoudakis, M.; Parr, R. Coordinated reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, San Francisco, USA. 2002. pp. 227–34. https://cdn.aaai.org/Symposia/Spring/2002/SS-02-02/SS02-02-014.pdf. (accessed 21 Aug 2025)
38. Böhmer, W.; Kurin. V.; Whiteson. S. Deep coordination graphs. *arXiv* **2019**, arXiv:1910.00091. https://doi.org/10.48550/arXiv.1910.00091. (accessed 21 Aug 2025)
39. Liu, J.; Huang, Z.; Xu, X.; Zhang, X.; Sun, S.; Li, D. Multi-kernel online reinforcement learning for path tracking control of intelligent vehicles. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 6962–75. DOI
40. Troullinos, D.; Chalkiadakis, G.; Papamichail, I.; Papageorgiou, M. Collaborative multiagent decision making for lane-free autonomous driving. In *AAMAS '21: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, Richland, USA. 2021. pp. 1335–43. https://dl.acm.org/doi/10.5555/3463952.3464106. (accessed 21 Aug 2025)
41. Zhang, X.; Pan, W.; Li, C.; et al. Toward scalable multirobot control: fast policy learning in distributed MPC. *IEEE Trans. Robot.* **2025**, *41*, 1491-512. DOI