Cooperative Decision-Making for CAVs at Unsignalized Intersections: A MARL Approach with Attention and Hierarchical Game Priors

Jiaqi Liu, Peng Hang, *Member, IEEE,* Xiaoxiang Na, Chao Huang, *Member, IEEE,* and Jian Sun

***Abstract*—The development of autonomous vehicles has shown great potential to enhance the efficiency and safety of transportation systems. However, the decision-making issue in complex human-machine mixed traffic scenarios, such as unsignalized intersections, remains a challenge for autonomous vehicles. While reinforcement learning (RL) has been used to solve complex decision-making problems, existing RL methods still have limitations in dealing with cooperative decision-making of multiple connected autonomous vehicles (CAVs), ensuring safety during exploration, and simulating realistic human driver behaviors. In this paper, a novel and efficient algorithm, Multi-Agent Game-prior Attention Deep Deterministic Policy Gradient (MA-GA-DDPG), is proposed to address these limitations. Our proposed algorithm formulates the decision-making problem of CAVs at unsignalized intersections as a decentralized multi-agent reinforcement learning problem and incorporates an attention mechanism to capture interaction dependencies between ego CAV and other agents. The attention weights between the ego vehicle and other agents are then used to screen interaction objects and obtain prior hierarchical game relations, based on which a safety inspector module is designed to improve the traffic safety. Moreover, we consider the heterogeneity of human drivers in traffic environments and conduct a series of comprehensive experiments. The proposed approach shows better performance in terms of driving safety, efficiency and comfort than conventional RL algorithms.**

***Index Terms*—Multi-agent reinforcement learning, connected autonomous vehicles, decision-making, attention mechanism, unsignalized intersections**

I. Introduction

Autonomous vehicles (AVs) have undergone remarkable advances in recent years, holding great potential for enhancing transportation efficiency and safety [1]–[3]. Nonetheless, decision-making in complex human-machine mixed traffic scenarios, particularly at unsignalized intersections, remains a considerable challenge for both AVs and human drivers [4]. Game-based and optimization-based approaches have been proposed to address this issue [5]–[7]. However, these methods prove impractical when handling scenarios involving multiple agents and complex interaction behaviors [7]. Reinforcement learning (RL) holds the potential to overcome these limitations, leveraging data-driven methods with its robust learning and efficient reasoning capabilities [8]–[10].

Current RL approaches for unsignalized intersection decision-making problems encounter several challenges. Firstly, most studies consider a single AV at the intersection, modeling the problem as a single-agent RL problem, whereas cooperative decision-making of multiple connected autonomous vehicles (CAVs) is a more challenging and less explored problem. Besides, common RL methods rely on exploration to teach CAVs how to decide and act, which may compromise the learning efficiency and policy safety [1]. Safety is the most critical factor when designing the decision-making algorithm. Moreover, the simulation environments' simplification of human drivers' behaviors may lead to performance gaps between simulation and real-world scenarios.

To cope with these challenges, we propose a novel and efficient algorithm, Multi-Agent Game-prior Attention Deep Deterministic Policy Gradient (MA-GA-DDPG), which formulates the decision-making problem of CAVs at unsignalized, human-machine mixed intersections as a decentralized multi-agent RL problem. In MA-GA-DDPG, each CAV at the intersection is modeled as an agent, enabling it to explore the environment, communicate, and cooperate with other agents. We use Multi-Agent Deep Deterministic Policy Gradient (MADDPG) as the baseline algorithm, where all agents adopt a strategy of centralized training and distributed execution (CTDE). To capture the interaction dependencies between the ego CAV and other agents, we incorporate an attention mechanism [11]. The attention weights between the ego vehicle and other agents are used to screen interaction objects and obtain prior hierarchical game relations. We then develop a safety inspector module to predict and detect potential conflicts with other agents during CAV exploration and make corrections in real-time to improve the algorithm's learning efficiency. We also consider the heterogeneity of human drivers in traffic environments and conduct comprehensive experiments that show our approach performs better in terms of learning efficiency, driving safety, efficiency and comfort.

The contributions of this paper are summarized as follows:

- A novel and efficient algorithm MA-GA-DDPG is pro-

This work was supported in part by the National Natural Science Foundation of China (52232015), the Fundamental Research Funds for the Central Universities, and the Young Elite Scientists Sponsorship Program by CAST (2022QNRC001).

Jiaqi Liu, Peng Hang, and Jian Sun are with the Department of Traffic Engineering and Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, Shanghai 201804, China. (e-mail: {liujiaqi13, hangpeng, sunjian}@tongji.edu.cn)

Xiaoxiang Na is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom (e-mail: xnhn2@cam.ac.uk)

Chao Huang is with Department of Industrial and System Engineering, The Hong Kong Polytechnical University, Hong Kong 999077. (e-mail: hchao.huang@polyu.edu.hk)

Corresponding author: Peng Hang

posed to realize cooperative decision-making of CAVs in complex human-machine mixed traffic scenarios. The heterogeneity of drivers is considered to simulate the real traffic environment and train the algorithm.
- The multi-head attention mechanism is leveraged to capture the complex interactions among CAVs and other vehicles in the mixed driving environment. An interaction object filter based on the attention weights is designed to facilitate the identification of the most relevant agents for interaction.
- A hierarchical game framework is utilized to incorporate the traffic priority priors into the interaction process. This framework enables the modeling of the priority information and its subsequent transmission to the safety inspector module, which effectively supervises and adjusts the actions of CAVs to minimize the risk of collisions.

The rest of the paper is organized as follows: Section II summarizes the recent related works. The decision-making problem at the intersection is formulated in section III. In section IV, the MA-GA-DDPG model is described. In section V, the simulation environment and comprehensive experiments are introduced and the results are analyzed. Finally, this paper is concluded in section VI.

II. Related Works

A. Decision-Making of CAVs at Intersections

Multi-vehicle interaction and decision-making under complex and high-density mixed driving conditions are very challenging for CAVs, and the intersection is one of the most complex and difficult scenarios for interaction [12], [13]. The development of an efficient cooperative decision-making algorithm for CAVs at intersection scenarios is of great significance for improving the efficiency and safety of intersection traffic [14].

The current research on decision-making modeling for autonomous driving at unsignalized intersections mainly includes the following methods:

- Game-theoretical models, such as Level-k games [5], follower-leader games [6], etc. These studies treat each CAV as a rational decision-maker and simulate the reactions and actions of human drivers under rational conditions [15], [16]. However, this purely rational situation cannot fully simulate the real world. Moreover, the efficiency of large-scale game calculations and the scalability of game frameworks are also challenging issues.
- Rule-based methods, such as first-come-first-serve (FCFS) [17], Buffer Coordination [18] etc. These methods are easy to implement and logically clear, but as the traffic demand increases, the efficiency of these methods is poor.
- Optimization-based methods, such as convex optimization methods [7], [19], model predictive control(MPC) [20] etc. The advantage of this method is that it can be solved accurately, with good interpretability and controllability, but the solution efficiency of too large-scale problems often cannot meet the requirements of real-time applications [19].
- Learning-based models, such as Neural Network(NN) [21], [22], Reinforcement Learning(RL) [9], [10], [23]. This method can effectively simulate the dynamics of interactions, with powerful learning and efficient reasoning capabilities [24], [25]. However, the interpretability, convergence, and generalization of the algorithm often require extra attention [26].

B. Multi-Agent Reinforcement Learning and Attention Mechanism

Multi-Agent Reinforcement Learning (MARL) is an emerging research field that focuses on the optimization problem of multiple autonomous intelligent agents making sequential decisions in an environment. In recent years, MARL has been utilized to solve plenty of multi-agent problems, such as traffic control [27], decision-making of autonomous driving [24], [25], [28], games [29], resource allocation [30], etc.

Some works have modeled the traffic system with multi-vehicle scenarios by MARL, which has shown exciting and outstanding performance in lane changing [24], [31], merging [25], intersection [9] scenarios. Nevertheless, these algorithms still fail to guarantee enough security and reliability, which greatly limits further application. To address these challenges, recent studies have proposed various approaches. For example, some works have incorporated interaction priors in the MARL framework [25]. Nevertheless, further research is needed to improve the safety and reliability of MARL algorithms.

Attention mechanism is a cognitive function that is crucial for humans. Recently, this mechanism has been introduced to many fields, including image caption generation, text classification, autonomous driving, and recommendation systems [32]. It is a newly-emerged technique in neural network models and has shown great power in sequence modeling [11]. The attention mechanism enables neural networks to identify correlations and inter-dependencies among variable inputs. It has been applied in the tasks of autonomous driving's decision-making, such as capturing vehicle-to-ego dependencies [12], [33], optimizing interactive behavior strtegies [9], and enhancing the safety of the decision-making algorithm [24], [34].

C. Summary of Related Works

Decision-making of CAVs at intersections is a complex and challenging task. Game-theoretical models, learning-based models, and optimization-based methods have been proposed to model decision-making for autonomous driving at unsignalized intersections. It is worth mentioning that MARL is a promising technique for multi-vehicle interaction and decision-making under complex and high-density mixed driving conditions. However, current MARL algorithms still suffer from the non-stationarity, safety issues, and scalability issues, which restricts their further application [35]. Attention mechanism has been introduced to improve the safety and reliability of decision-making algorithms in autonomous driving. However, the existing research on the application of attention mechanism still has some limitations [32].

To address the cooperative decision-making issue of CAVs in complex human-machine mixed traffic scenarios, we improved the MARL framework considering interaction prior and proposed a safety inspector to resolve the conflicts at the intersection. Besides, the attention mechanism is further explored and utilized to filter interacting objects in complex environments.

III. MARL for Intersection Decision-Making

In this section, the decision-making problem of CAVs is formulated as a MARL problem firstly. To establish a baseline, the MADDPG algorithm for addressing this problem is proposed.

A. Scenario Description and Vehicle Movements

1) Scenario Description: We consider a cooperative decision-making problem for CAVs in an intersection scenario. A single-lane cross-shaped unsignalized intersection is defined, as where there may be a couple of human-driven vehicles (HVs), which may come from different directions and locations and have diverse driving styles. Then a CAV is replaced for each of the entrances in the four directions and the task is not completed until all CAVs safely pass the intersection and reach the destination. CAVs can share the status information with each other for better cooperative decision-making.

2) Vehicle Movements: The actions of CAVs in our problem are decided by the MARL algorithm and will be translated to low-level steering and acceleration signals through a closed-loop PID controller. The vehicle's position is controlled by

$$\begin{aligned} v_{\text{lat},r} &= -K_{p,\text{lat}}\Delta_{\text{lat}}, \\ \Delta\psi_r &= \arcsin\left(\frac{v_{\text{lat},r}}{v}\right) \end{aligned} \tag{1}$$

The vehicle's heading is controlled by

$$\begin{aligned} \psi_r &= \psi_L + \Delta\psi_r, \\ \dot{\psi}_r &= K_{p,\psi}(\psi_r - \psi), \\ \delta &= \arcsin\left(\frac{1}{2}\frac{l}{v}\dot{\psi}_r\right) \end{aligned} \tag{2}$$

where Δ_{lat} is the lateral position of the vehicle with respect to the lane center-line, $v_{\text{lat},r}$ is the lateral velocity command, $\Delta\psi_r$ is a heading variation to apply the lateral velocity command, ψ_L is the lane heading, ψ_r is the target heading to follow the lane heading and position, $\dot{\psi}_r$ is the yaw rate command, δ is the front-wheel angle control, $K_{p,\text{lat}}$ and $K_{p,\psi}$ are the position and heading control gains.

Vehicle motion is determined by a Kinematic Bicycle Model [36]

$$\begin{aligned} \dot{x} &= v\cos(\psi+\beta) \\ \dot{y} &= v\sin(\psi+\beta) \\ \dot{v} &= a \\ \dot{\psi} &= \frac{v}{l}\sin\beta \\ \beta &= \tan^{-1}(1/2\tan\delta) \end{aligned} \tag{3}$$

where (x, y) is the vehicle position, v is forward speed, ψ is heading, a is the acceleration command, β is the slip angle at the center of gravity.

In this paper, HVs' longitudinal behavior is controlled by the Intelligent Driver Model(IDM) from [37]

$$\dot{v} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{d^*}{d}\right)^2\right] \tag{4}$$

$$d^* = d_0 + Tv + \frac{v\Delta v}{2\sqrt{ab}} \tag{5}$$

where v is the vehicle velocity, d is the distance to its front vehicle, v_0 is the desired velocity, T is the desired time headway, d_0 is the jam distance, a, b are the maximum acceleration and deceleration respectively, Δv is the velocity exponent.

HVs' lateral behavior is modeled by the Minimizing Overall Braking Induced by Lane change(MOBIL) model [38].

B. Problem Formulation

The cooperative decision-making problem for multi-CAVs can be formulated as a partially observable Markov decision process (POMDP) [39]. We use the tuple $\mathcal{M}_{\mathcal{G}} = (\mathcal{V}, S, [\mathcal{O}_i], [\mathcal{A}_i], \mathcal{P}, [r_i])$ to define the POMDP, in which $\mathcal{V}$ is a finite set of all controlled agents (CAVs) and S denotes the state space describing all agents. $\mathcal{O}_i$ denotes the set of observation spaces for agent $i \in \mathcal{V}$, $\mathcal{S}_i$ is the set of action space, and r_i represents the reward function. $\mathcal{P}$ denotes the transition distribution. Each agent i at a given time receives an individual observation $o_i : \mathcal{S} \to \mathcal{O}_i$ and takes an action $a_i \in \mathcal{A}_i$ based on a policy $\pi_i : \mathcal{O}_i \times \mathcal{A}_i \to [0, 1]$. Then the agent i transits to a new state s_i' with probability $\mathcal{P}(s'|s, a) : \mathcal{S} \times \mathcal{A}_1 \times ... \times \mathcal{A}_N \to \mathcal{S}$ and obtains a reward $r_i : \mathcal{S} \times \mathcal{A}_i \to \mathbb{R}$. Each agent aims to maximize its expected return

$$R_i = \Sigma_{t=0}^{T}\gamma^t r_i(s_t, a_t^i) \tag{6}$$

where γ represents the discount factor and T represents the time horizon.

1) Observation Space: Due to the limitation of the sensor hardware, the CAV can only detect the status information of surrounding vehicles within a limited distance $\mathcal{L}$. We denote the set of all observable vehicles within the perception range of agent i as $\mathcal{N}_i$. And the observation matrix of agent i, denoted as $\mathcal{O}_i$, is a matrix with dimensions of $|\mathcal{N}_i| \times |\mathcal{F}|$, where $|\mathcal{N}_i|$ is the number of all observable vehicles for agent i, and $|\mathcal{F}|$ is to represent the number of features utilized to portray a vehicle's state. The feature vector of the vehicle k is expressed by

$$\mathcal{F}_k = [x_k, y_k, v_k^x, v_k^y, \cos\phi_k, \sin\phi_k] \tag{7}$$

where x_k, y_k, v_k^x, v_k^y are the longitudinal position, lateral position, longitudinal speed, and lateral speed, $\cos\phi_k, \sin\phi_k$ are the sine and cosine of the vehicle heading angle ϕ_k, respectively. The whole observation space of the system is the combined observation of all CAVs, i.e., $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2 \times \cdots \times \mathcal{O}_{|\mathcal{V}|}$.

2) Action Space: In our research, we pay more attention to the decision-making actions of CAVs rather than the vehicle control level. When passing through the intersection, we pre-set the driving route, and the CAV needs to decide the acceleration and deceleration actions to turn left through the intersection and reach the destination. So we define the action space $\mathcal{A}_i$ for agent i as the set of high-level control decisions that includes $\{slow\ \ down, cruising, speed\ \ up\}$. After selecting a high-level decision, lower-level controllers generate the corresponding steering and throttle control signals to control the CAVs' movement. The overall action space is the combined actions of all CAVs, i.e., $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_{|\mathcal{V}|}$.

3) Reward Function: The reward function has a great effect on the final performance of the algorithm. In order to make agents pass the intersection safely and effectively, the reward of ith agent at the time step t is defined as

$$r_{i,t} = \underbrace{w_c r_c}_{\text{Collision reward}} + \underbrace{w_e r_e}_{\text{Efficiency reward}} + \underbrace{w_a r_a}_{\text{Arrival reward}} \tag{8}$$

where w_c,w_e, and w_a are the weight coefficients of collision reward r_c, efficiency reward r_e, and arrival reward r_a, respectively. These evaluation terms are defined as follows:

- Collision reward r_c: Safety is the most important criterion for a vehicle. In order to make the agent learn to drive safely, we give it a greater penalty when a collision happens.

$$r_c = \begin{cases} -10 & if\ agent_i\ collide \\ 0 & otherwise \end{cases} \tag{9}$$

- Efficiency reward r_e: The speed range $[v_{min}, v_{max}]$ for the agents passing the intersection is set based on real-world traffic rules. The agent is not recommended to drive at a speed outside this range, and we encourage the agent to pass as efficiently as possible within the speed range. And a constant C_e is used to adjust the maximum speed bonus.

$$r_e = C_e \times \min\left\{\frac{v_i - v_{min}}{v_{max} - v_{min}}\right\} \tag{10}$$

- Arrival reward r_a: When any agent completes its ultimate goal to reach the end, they will obtain an arrival reward.

$$r_a = \begin{cases} 5 & if\ agent_i\ reaches\ destination \\ 0 & otherwise \end{cases} \tag{11}$$

where T is the total time steps for one episode.

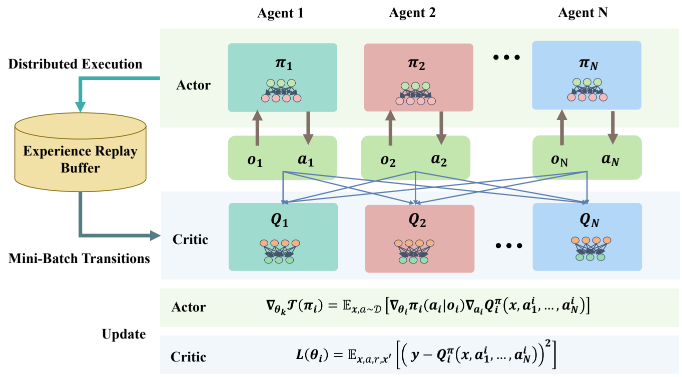

Agent 1
Agent 2
Agent N
Distributed Execution
Actor
π_1
π_2
π_N
Experience Replay Buffer
o_1
a_1
o_2
a_2
o_N
a_N
Critic
Q_1
Q_2
Q_N
Mini-Batch Transitions
Update
Actor
$\nabla_{\theta_k}\mathcal{J}(\pi_i) = \mathbb{E}_{x,a\sim\mathcal{D}}[\nabla_{\theta_i}\pi_i(a_i|o_i)\nabla_{a_i}Q_i^{\pi}(x, a_1^i, \dots, a_N^i)]$
Critic
$L(\theta_i) = \mathbb{E}_{x,a,r,x'}\left[\left(y - Q_i^{\pi}(x, a_1^i, \dots, a_N^i)\right)^2\right]$

Fig. 1. The framework of the Multi-Agent Deep Deterministic Policy Gradient (MADDPG).

C. MADDPG For Decision-Making of CAVs

In our work, a model-free MARL algorithm, MADDPG, is utilized as the baseline algorithm.

In Deep Reinforcement Learning (DRL), a deep neural network (DNN) serves as a non-linear approximator to obtain optimal policies π^* for CAVs (agents). The agents interact with the environment and receive feedback in the form of rewards, which are used to update the agent's policy. Let $\pi = \{\pi_1, \cdots, \pi_N\}$ denote the set of all CAVs' policies and $\theta = \{\theta_1, \cdots, \theta_N\}$ denote the parameter set of corresponding policy, where $N = |\mathcal{V}|$ is the number of CAVs. CAVs update their policies based on the estimation of the Q-function for each possible action using the off-policy actor-critic algorithm MADDPG [40]. The objective function for MADDPG is the expected reward $\mathcal{J}(\theta)$, i.e., $\mathcal{J}(\theta_i) = \mathbb{E}[\Omega_i(t)]$. The optimal policy of each CAV is represented as $\pi^*_{\theta_i} = \arg\max_{\pi_{\theta_i}} \mathcal{J}(\theta_i)$. Then the algorithm calculates the gradient of the objective function with respect to θ_i as

$$\nabla_{\theta_i}\mathcal{J}(\pi_i) = \mathbb{E}_{\mathbf{x},a\sim\mathcal{D}}[\nabla_{\theta_i}\pi_i(a_i|o_i)\nabla_{a_i}Q_i^{\pi}(\mathbf{x}, a_1, \cdots, a_N)], \tag{12}$$

where $\mathbf{x} = \mathcal{O} = (o_1, \cdots, o_N)$, $Q_i^{\pi}(\mathbf{x}, a_1, \cdots, a_N)$ is a centralized action-value funciton, and $\mathcal{D}$ is the replay buffer. $\mathcal{D}$ contains transition tuples $(\mathbf{x}, a, r, \mathbf{x}')$, where $a = (a_1, \cdots, a_N)$ and $r = (r_1, \cdots, r_N)$. To minimize the loss function (13), the centralized action-value function Q_i^{π} is updated for

$$\mathcal{L}(\theta_i) = \mathbb{E}_{\mathbf{x},a,r,\mathbf{x}'}[(y - Q_i^{\pi}(\mathbf{x}, a_1, \cdots, a_N))^2], \tag{13}$$

where $y = r_i + \gamma Q_i^{\pi'}(\mathbf{x}', a_1', \cdots, a_N')|_{a_j'=\pi_j'(o_j)}$. $\pi' = \{\pi_{\theta_1'}, \cdots, \pi_{\theta_N'}\}$ stands for the target policies with delayed parameters θ_i'. The schematic diagram is shown in Fig. 1

IV. Game-prior Attention Model

In this section, the whole framework of our MA-GA-DDPG model is first outlined. Then we introduce an attention mechanism-based policy network and an interaction filter approach. Furthermore, a safety inspector based on the attention weights and the level-k game is applied to enhance the safety performance of the algorithm.

A. Model Overview

The overview of our model is shown in Fig. 2. First, we design a novel policy network based on the attention mechanism for each agent in the MADDPG algorithm. An encoder-decoder framework that contains a multi-head attention layer is used to better capture the interaction relationships between agents. Based on the attention-based policy network, the attention weights of each CAV to all surrounding traffic participants are obtained. Then we introduce a filter rule that applies these attention weights for strongly interacting participants screening.

At the same time, the attention degree of each vehicle in the scenario learned by the policy network is regarded as the priority weight of the vehicle passing through the intersection. Based on the level-k game (hierarchical game), these priority weights are transformed into a level prior and a safety inspector is defined. According to the input game prior,

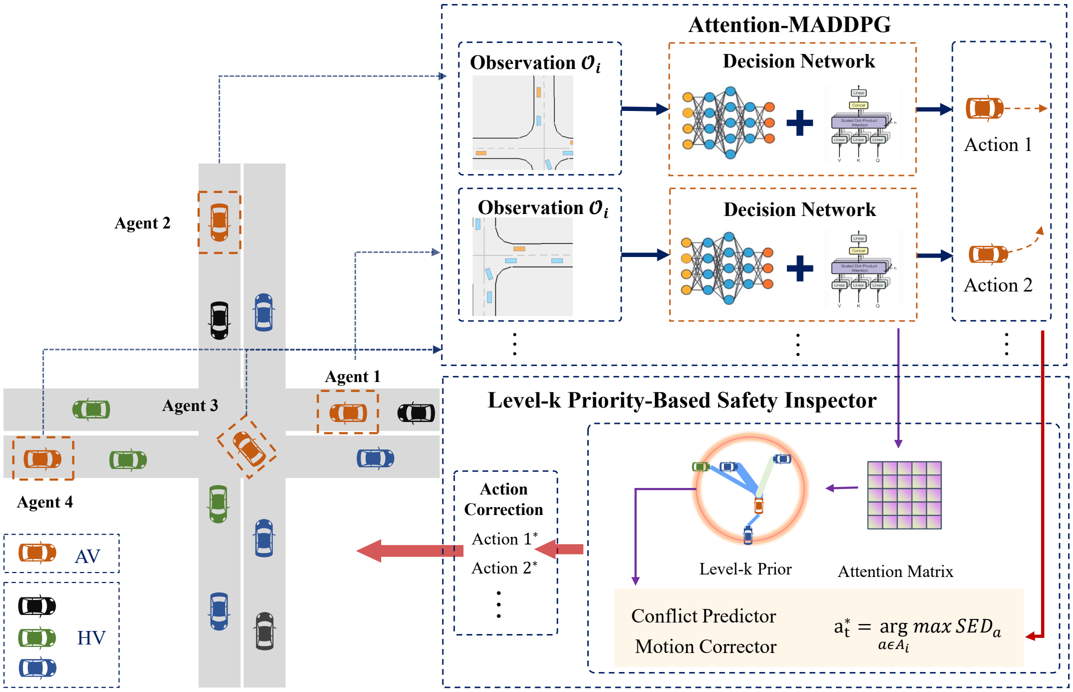


Fig. 2. The overview of our MA-GA-DDPG Model.

the safety inspector can predict, check and correct the high-risk actions in time to alleviate or resolve conflicts for each CAV. The safety inspector is shown later that can effectively improve the safety of the RL algorithm.

B. Attention-Based Policy Network

The attention mechanism has been well-documented to enable neural networks to discover interdependencies among a variable number of inputs and has been applied in the social interaction research of autonomous vehicles [9], [12].

Inspired by [12], we design an attention-based policy network for each agent in a decentralized training process, as shown in Fig. 3. The network contains three modules: encoder block, attention block, and decoder block. In the encoder block, the features $\mathcal{F}_i$ of the agent i and its observation matrix $\mathcal{O}_i$ are encoded as high-dimension vectors by a Multilayer Perceptron (MLP), whose weights are shared between all vehicles.

$$\mathcal{X}_i = MLP(\mathcal{F}_i, \mathcal{O}_i) \tag{14}$$

And then the feature matrix is fed to the attention block, composed of N_{head} attention heads stacked together. Unlike the attention layer in the Transformer model [11], this block produces only the query results (attention weights) of agent i, which indicate how much attention it should pay to the surrounding vehicles.

In the attention block, the ego vehicle emits a single query $Q_i = [q_0] \in \mathbb{R}^{1\times d_k}$ to select a subset of vehicles based on the environment, where d_k is the output dimension of the encoder layer. This query is then projected linearly and compared to a set of keys $K_i = [k_i^0, k_i^1, \cdots, k_i^N] \in R^{(N+1)\times d_k}$ containing descriptive features for each vehicle, using dot product $q_0 k_i^T$ to calculate the similarity. The Q_i, K_i and V_i are calculated as follows.

$$\begin{aligned} Q_i &= W^Q \mathcal{X}_i \\ K_i &= W^K \mathcal{X}_i \\ V_i &= W^V \mathcal{X}_i \end{aligned} \tag{15}$$

where the dimensions of W^Q and W^K are $(d_k \times d_N)$, and W^V's is $(d_v \times d_h)$.

The attention matrix is obtained by scaling the dot product with the inverse-square-root-dimension $\frac{1}{\sqrt{d_k}}$ and normalizing it with a softmax function σ. The attention matrix is then used to gather a set of output values $V_i = [v_i^0, \cdots, v_i^N]$, where each v_i^j is a feature computed using a shared linear projection $L_v \in \mathbb{R}^{d_x \times d_k}$. The attention computation for each head can be written as

$$At_i^m = \sigma\Big(\frac{Q_i K_i^T}{\sqrt{d_k}}\Big) V_i \tag{16}$$

Then the output from all M heads will be combined with a linear layer:

$$At_i = \sum_{m=1}^{M} At_i^m \tag{17}$$

The final attention vector is denoted by $At_i = [At_{i,1}, At_{i,2}, \cdots, At_{i,|\mathcal{N}_i|}]$, where $At_{i,j} (j \in \mathcal{N}_i)$ denotes the weight of the agent i's attention to the surrounding vehicle j, satisfying the cumulative summation relationship: $\sum_{j=1}^{|\mathcal{N}_i|} At_{i,j} = 1$. The vector At_i will be fed to the decoder block along with the encoding result of agent i. Finally, the value of the agent i's action at the next time step will be assessed.

Overall, the attention-based policy network has the following significant advantages: (1) It can handle the variable

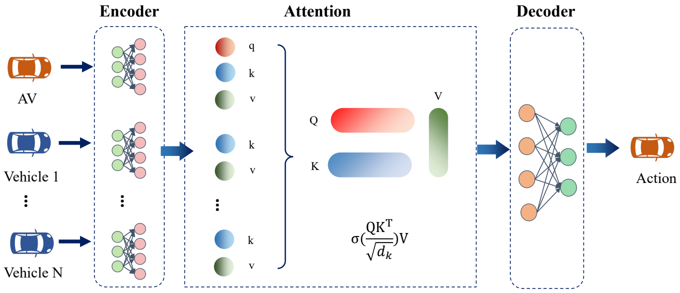


Fig. 3. The attention-based policy network for every single agent.

amount of observation information inputs; (2) It has permutation invariant outputs that are independent of the sequence of surrounding agents; (3) It has good interactive interpretability based on the attention matrix. In the next subsection, we will screen the interactive objects based on the interpretable feature of the attention mechanism and obtain the intersection traffic prior based on the hierarchical game.

C. Hierarchical Game Prior From Agent's Attention

The interpretability of the attention network allows us to further utilize the learned attention weight information. In the real world, in order to pass through intersections, both CAVs and HVs will have competitive or cooperative game relationships with each other. If this game relationship can be learned and sent to the algorithm as prior information, it will greatly help the algorithm explore and learn more efficiently.

To capture the strategic decision-making process of HVs, a game-theoretical concept named level-k reasoning is used. This approach assumes that humans have different levels of reasoning, with level-0 being the lowest. A level-0 agent is a non-strategic agent that makes predetermined moves without considering the possible actions of other agents. On the other hand, a level-1 agent is a strategic agent that assumes all other agents have level-0 reasoning and decides the best response to such actions. Similarly, a level-2 agent assumes all other agents are level-1 and makes decisions based on this assumption. This hierarchy continues for higher levels. However, due to bounded rationality for the agents [41], this assumption may not always hold true. The experiments reveal that humans generally have at most level-3 reasoning [5], but this may vary depending on the game being played.

The level of the agent is obtained based on the attention mechanism and is considered as important prior knowledge. As shown in Fig. 4, our method mainly includes two steps: Interactive Object Selection and Level Prior Determination. The detailed process is summarized in Algorithm 1.

- *Step* 1 : Interactive Object Selection. Similar to human attention, we set an attention radius dis_0 and an attention weight threshold δ_0 for each CAV to select potential interaction objects. At each moment, when a surrounding vehicle j satisfies the condition that the distance dis_{ij} between j and CAV i is less than dis_0 and the attention weight $At_{i,j}$ is greater than δ_0, and the interaction limit $\mathcal{Q}$ has not been reached, it is included in the set of Potential interaction Objects PO_{inter}.
- *Step* 2 : Level Prior Determination. The interaction importance is determined based on the attention weight. In the interaction environment, a vehicle that receives more attention is believed to be more important to the CAV, and vice versa. In an environment with only one CAV, the interaction importance of the environment vehicle with the highest attention weight (i.e., the most attention received) is ranked highest, followed by the others in descending order of attention weight. Finally, we obtain a list of interaction importance $Rank_i$ between the AV and all surrounding vehicles. In a multi-CAVs environment, since each CAV can calculate the attention weight for all vehicles, the global attention weight of vehicle j is the sum of the attention weight given by all CAVs:

$$BAt^j = \sum_{i=1}^{\mathcal{V}} At_{i,j} \tag{18}$$

  The environment vehicle with the highest attention weight and priority is ranked highest, followed by the others in descending order of attention weight. If the interaction object limit $\mathcal{Q}$ of the AV is reached, the top Q objects in the sorted set PO_{inter} are selected.

Algorithm 1: Obtain Hierarchical Level Priors Based On Attention Mechanism

```
Inputs  : $At,\ dis_0,\ \delta_0,\ \mathcal{Q}$
Outputs: $PO_{inter},\ Rank$
Step 1:
for $i = 1$ to $|\mathcal{V}|(i \in \mathcal{V})$ do
    Calculate Attention vector $At_i$ by Eq.16 and Eq.17;
    for $j = 1$ to $|\mathcal{V} + \mathcal{N}_i|$ $(j \in |\mathcal{V}| \cup \mathcal{N}_i)$ do
        Calculate the Euclidean Distance between
         agent $i$ and $j$ : $Dis(i, j)$;
        if $Dis(i, j) > dis_0$ and $At_{i,j} > \delta_0$ then
            $PO^i_{inter} = PO^i_{inter} \cup j$
            if $|PO^i_{inter}| > \mathcal{Q}$ then
                Remove $PO^i_{inter}[-1]$ from $PO^i_{inter}$;
            end
        end
        Sort $PO^i_{inter}$ in descending order based on
         $At_{i,j}$;
    end
end
Step 2:
for $j = 1$ to $|\mathcal{V} + \mathcal{N}_i|(j \in |\mathcal{V}| \cup \mathcal{N}_i)$ do
    $BAt_j = \sum_{i=0}^{\mathcal{V}} \sigma At_{i,j}$;
    $\sigma = \begin{cases} 1\ if\ j\ \in PO^i_{inter} \\ 0\ else \end{cases}$
end
Sort $BAt$ in descending order;
for $j = 1$ to $|\mathcal{V} + \mathcal{N}_i|$ $(j \in |\mathcal{V}| \cup \mathcal{N}_i)$ do
    Update: $Rank_{i,j} = Index(BAt_j)$;
end
```

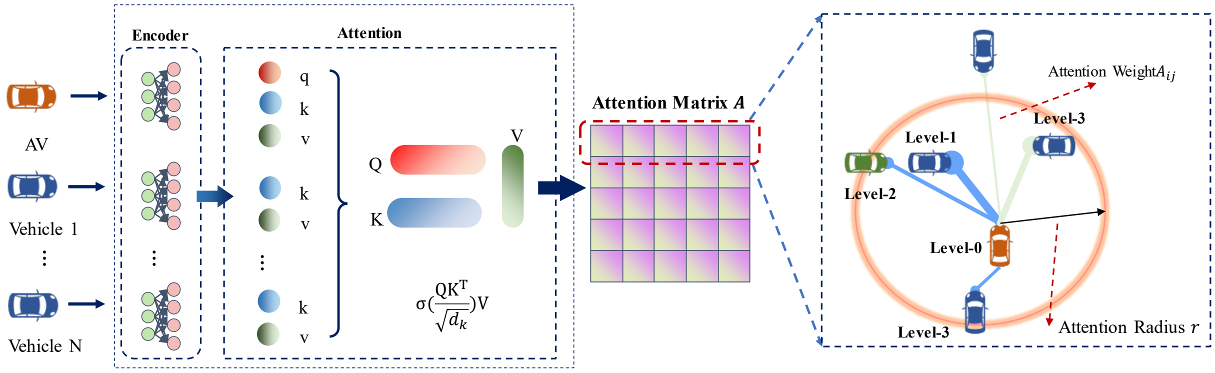


Fig. 4. Attention-based interactive object selection for each CAV.

D. Safety Inspector Based On Hierarchical Game Prior

When passing through an intersection, we always believe that if a vehicle receives more attention, it is more important or dangerous. Therefore, we believe that vehicles with higher attention weights should have a higher right of way at intersections. This right-of-way level can be used as prior information to help the RL algorithm learn action strategies that are consistent with real-world logic more quickly. To this end, we designed a safety inspector module based on the level-k priority, similar to the *Critic* network in the RL algorithm but independent of the policy network. The safety inspector module can be seen as an active conflict regulator, mainly including two parts: Proactive Predictor of Agent Risk and Assisted Corrector of Agent Motion, which are used to assist and correct the agent's inefficient exploration behavior in the early stage and ensure the continuity and stability of the algorithm performance. The level-k priority-based safety supervisor at the intersection is shown in Fig. 5. The specific calculation process of the safety inspector module is as follows:

智能体风险的主动预测和智能体运动的辅助校正

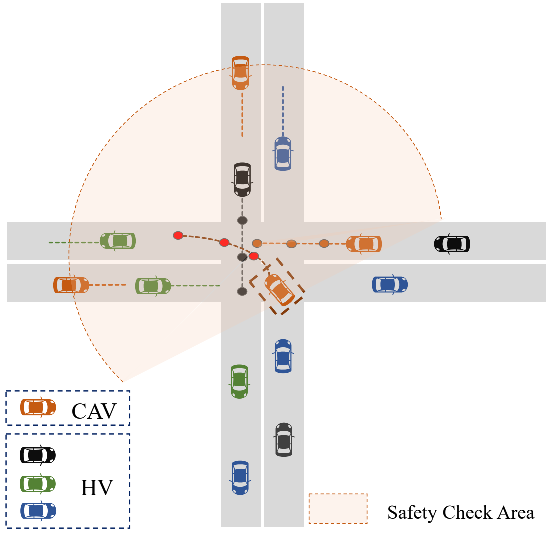


Fig. 5. Trajectory prediction of surrounding agents and conflict checking for CAV i at the intersection

Algorithm 2: Level-k Priority-based Safety Inspector

```
Inputs : $t_0$, $T$, $A_{t_0}$, $O, S$
Output: $A^*_{t_0}$
for $i = 1$ to $|\mathcal{V}| (i \in \mathcal{V})$ do
    Obtain the output action $a_{t_0}$ of policy network and
      level Prior $k_i$ by Alg. 1 ;
    for $t = 1$ to $T$ do
        Sample future trajectories based on $a_{t_0}$ by
          $MDP$:
        $(x^i_{k+1}, y^i_{k+1}) = MDP(x^i_k, y^i_k)$
    end
    $\left(tra^i_{t_0}\right)^{k_i}_{a_{t_0}} =$
      $\left\{\left(x^i_{t_0+1}, y^i_{t_0+1}\right)^{k_i}, \left(x^i_{t_0+2}, y^i_{t_0+2}\right)^{k_i}, \cdots,$
    $\left(\mathrm{x}^i_{t_0+T}, y^i_{t_0+T}\right)^{k_i}\right\}$
    for $j = 1$ to $|\mathcal{V}|$ $(j \in \mathcal{V}$ and $j \neq i)$ do
        Get $\left(tra^j_{t_0}\right)^{k_j}$ by V2V communication;
    end
    for $m = 1$ to $|\mathcal{N}_i|$ $(m \in \mathcal{N}_i)$ do
        Predict future trajectories based on $\mathcal{O}_i$ by
          $IDM$:
        $\left(tra^m_{t_0}\right)^{k_m} = IDM(x^m_k, y^m_k)$
    end
    Calculate the number of conflict points
      $CI(a, Tra)(a \in A_{t_0})$;
    for $a' \in \mathcal{A}_i$ do
        Calculate $CI(a', Tra)$;
        Get $SED(a, a', Tra)$ by Eq. 22;
    end
    Solve the Eq.23 and get $a^{i*}_{t_0}$;
    Update $A^*_{t_0} = A^*_{t_0} \cup a^{i*}_{t_0}$
end
```

- *Step* 1: Agent Trajectory Prediction and Conflict Checking. Through the communication between CAVs, the safety inspector obtains the state information O of all perceived vehicles in the intersection at any time step t_0, and the current action decision set $\mathcal{A}_{t_0}$ of all CAVs. Then based on the level-k prior information, it selects agent i from the CAV set $\mathcal{V}$ in descending order to carry out active prediction of agent risk. For agent i, let k_i denote its prior level, and T denote the forward prediction step size. Obtain the future trajectory sequences of agent i :

$$\left(tra_{t_0}^{i}\right)_{a_{t_0}}^{k_i} = \left\{\left(x_{t_0+1}^{i}, y_{t_0+1}^{i}\right)^{k_i}, \left(x_{t_0+2}^{i}, y_{t_0+2}^{i}\right)^{k_i}, \cdots, \left(x_{t_0+T}^{i}, y_{t_0+T}^{i}\right)^{k_i}\right\}(a_{t_0} \in \mathcal{A}_{t_0}) \tag{19}$$

by sampling T steps through the MDP process, and obtain the future T step trajectories of other CAV j through communication:

$$\left(tra_{t_0}^{j}\right)^{k_j} = \left\{\left(x_{t_0+1}^{j}, y_{t_0+1}^{j}\right)^{k_j}, \left(x_{t_0+2}^{j}, y_{t_0+2}^{j}\right)^{k_j}, \cdots, \left(x_{t_0+T}^{j}, y_{t_0+T}^{j}\right)^{k_j}\right\}(j \in \mathcal{V}) \tag{20}$$

And for all HV m within the observation range of i, use the IDM model to predict their future trajectories :

$$\left(tra_{t_0}^{m}\right)^{k_m} = \left\{\left(x_{t_0+1}^{m}, y_{t_0+1}^{m}\right)^{k_m}, \left(x_{t_0+2}^{m}, y_{t_0+2}^{m}\right)^{k_m}, \cdots, \left(x_{t_0+T}^{m}, y_{t_0+T}^{m}\right)^{k_m}\right\}(m \in \mathcal{N}_i) \tag{21}$$

Then judge whether there is a conflict between the future trajectories of agent i and all other vehicles. If there is a conflict, jump to $Step$ 2, otherwise, the output action of the agent is safe, and the safety check of agent i at time step t ends.

- $Step$ 2: Action Correction For Agent's Decision-making. If the future trajectory of agent i conflicts with other vehicles, the optimal alternative action needs to be found to minimize the conflict in the intersection. We use the number of conflict points of all vehicles in the scenario as the conflicting index CI to measure the danger degree of the scenario at each timestamp. Based on the conflicting index CI, we define the Safety Enhancement Degree SED function to measure the degree of conflict mitigation brought about by agent action correction:

$$SED(a, a', Tra) = CI(a, Tra) - CI(a', Tra) \tag{22}$$

where Tra is the future trajectories set of all vehicles, a and a' are the origin action and corrected action of agent i, respectively, and $Tra^{a'}$ is the future trajectories set of all vehicles based on the corrected action a'.
So our objective function for agent i is derived by

$$a_{t_0}^{i*} = \underset{a' \in \mathcal{A}_i}{\arg} \max SED\Big(a, a', \left(tra_{t_0}^{i}\right)^{k_i}, \left(tra_{t_0}^{j}\right)^{k_j}, \cdots, \left(tra_{t_0}^{m}\right)^{k_m}\Big) \tag{23}$$

- $Step$ 3: Output the optimal actions.
Judging whether all agents have completed the above process, if all have been completed, execute all agent-corrected security actions $\mathcal{A}_{t_0}^{*}$.

The process of safety inspector is described in Algorithm 2.

Same as the MADDPG algorithm, the MA-GA-DDPG algorithm is based on the actor-critic model, where the actor makes decisions over time while the critic evaluates its behavior. Each agent has an actor and a critic; both have behavior and target networks. The actor updates the behavior policy network using gradient ascent with Eq. 12, while the critic updates the behavior Q-function to evaluate actions and updates the target Q-function in a way that minimizes the loss function in Eq. 13. CAVs optimize their own policy to obtain more rewards while updating their critic's Q-function to evaluate actions. Specifically, the objective is to update the target network's θ for CAVs to learn how to act. The actor's neural network remains fixed for a set number of iterations while the behavior network's weights are updated. Our whole improved model is summarized in Algorithm 3.

Algorithm 3: MA-GA-DDPG for CAVs

```
Inputs : $T_{Max}, M, dis_0, \delta_0, \mathcal{Q}$
Output: $\theta$
for $episode = 1$ to $M$ do
    Initialize $\mathcal{D} \leftarrow \emptyset$, a random process $\mathcal{G}$ for action exploration;
    Receive initial state $\mathbf{x}$;
    for $t = 1$ to $T_max$ do
        for $CAV\ i = 1 \in \mathcal{V}$ do
            Get observation $o_{i,t}$;
            Update action $a_i = \pi_{\theta_i}(o_{i,t}) + \mathcal{G}_t$;
        end
        for $i = 1 \in \mathcal{V}$ do
            Get attention weights $At_i$ by Alg. 1;
            Get corrected actions $a_{i,t}^{*}$ by $At_i$ and Alg. 2;
            Execute $a_{i,t}^{*}$ and update $a_{i,t} \leftarrow a_{i,t}^{*}$;
            Observe reward $r_{i,t}$ and new observation $\mathbf{x}'_{i,t}$;
            Update $\mathcal{D}_i \leftarrow (\mathbf{x}_{i,t}, a_{i,t}^{*}, r_{i,t}, \mathbf{x}'_{i,t})$;
        end
        $\mathbf{x} \leftarrow \mathbf{x}'$;
        for $CAV\ i = 1 \in \mathcal{V}$ do
            Sample a random minibatch of $S$ samples
            $(\mathbf{x}_{i,t}, a_{i,t}^{*}, r_{i,t}, \mathbf{x}'_{i,t})$ from $\mathcal{D}_i$;
            Set $y^j = r_i^j + \gamma Q_i^{\pi'}(\mathbf{x}'^j, a'_{1,t}, \ldots, a'_{N,t})$;
            Update critic by minimizing the loss
            $\mathcal{L}(\theta_i) = \frac{1}{S}\sum_j \Big(y^j - Q_i(\mathbf{x}^j, a_1^j, \ldots, a_N^j)\Big)^2$;
            Update actor using the sampled policy gradient:
            $\nabla_{\theta_i}(\pi_i) \approx \frac{1}{S}\sum_j \nabla_{\theta_i}\pi_i(o_i^j)\nabla_{a_i}Q_i^{\pi}(\mathbf{x}^j, a_1^j,$
            $\ldots, a_N^j)$
        end
        Update target network parameters for each CAV $i$:
        $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
    end
end
```

V. Simulation and Performance Evaluation

In this section, the simulation environment we defined is first introduced, and then the training experiments and hyperparameter are described in detail. Finally, the experimental results are compared and a couple of cases are analyzed.

A. Simulation Environment

On the basis of an OpenAI Gym environment [42], we define an RL training simulator for multi-agent centralized training and distributed execution. The simulator is currently able to customize the intersection environment where CAVs and HVs participate according to requirements, and will expand to other scenarios. In the simulator, the actions determined by specific policies are translated to low-level steering and acceleration signals through a closed-loop PID controller. The longitude and lateral decisions of HVs are controlled by the IDM model and MOBIL model respectively, which has been described in the section III-A2.

Meanwhile, in order to simulate the human driver's perception and prediction of the movement of other traffic participants, all HVs are set with the constant-speed motion prediction and collision avoidance functions of the future $T_h s$. For each simulation episode, we randomize the initial states of all agents to prevent the policy network from memorizing a sequence of actions and instead promote learning of generalizable policies.

B. Simulation Settings

1) Training Scenarios Design: We design the human-machine mixed environment to test our algorithm. For more reality, the heterogeneity of drivers is considered. Three kinds of driving styles are set: $Aggressive, Normal$, and $Timid$ [43], whose parameters are shown in Table I:

TABLE I
THE PARAMETERS OF DIFFERENT DRIVING STYLES.

Driving Style	$JamDistance$ $(d_0)(m)$	$DesiredTime$ $Headway(T)(s)$	$Maximum$ $Acceleration(a_0)(m/s^2)$	$Maximum$ $Deceleration(b_0)(m/s^2)$
Aggressive	3.38	0.86	1.35	2.07
Normal	3.67	1.14	1.34	2.06
Timid	3.69	1.27	1.36	1.99

Overall, three typical scenarios with increasing difficulty are set up:

- (a) Just four CAVs and one CAV per entry lane.
- (b) Four CAVs and 3-10 homogeneous HVs, which means the driving style of each HV is the same and the number of HVs is randomly generated, ranging from 3-10.
- (c) Four CAVs and 3-10 heterogeneous HVs. The driving style of each HV is different and randomly generated.

Besides, MADDPG and Attention-MADDPG (MADDPG Algorithm with attention-based policy network) are utilized as our baselines.

2) Implementation Details: The hyperparameter of the model during the training is shown in Table II. The speed range: $[v_{min}, v_{max}]$ is $[3.0, 9.0]$. In the attention-based policy network, the Encoder and Decoder both are MLP, which has two linear layers and the layer size is 64×64. The Attention Layer contains two heads and the feature size is set as 128. When selecting the interaction with attention weights, we set $dis_0 = 40, \delta_0 = 0.5$, and $\mathcal{Q} = 5$. In the safety inspector module, we predict $T = 5$ steps feature trajectories for each vehicle. All the experiments are conducted in a platform with Intel Core i7-12700 CPU, NVIDIA GeForce RTX 3070 Ti GPU, and 32G memory.

TABLE II
THE HYPERPARAMETER OF THE MODEL FOR TRAINING.

Symbol	Definition	Value
N_t	Training Episodes	2000
S_u	Steps Per Update	100
Ψ	Buffer Length	10000
λ	Learning Rate	0.01
B	Batch of Transitions	128
γ	Discount factor	0.95
τ	Target update rate	0.01
w_c	Weight for r_c	1
w_e	Weight for r_e	1
w_a	Weight for r_a	1
dis_0	Maximum interaction distance of CAV	$40m$
δ_0	Attention threshold of CAV	0.05
$\mathcal{Q}$	Maximum number of interactive objects for a CAV	5

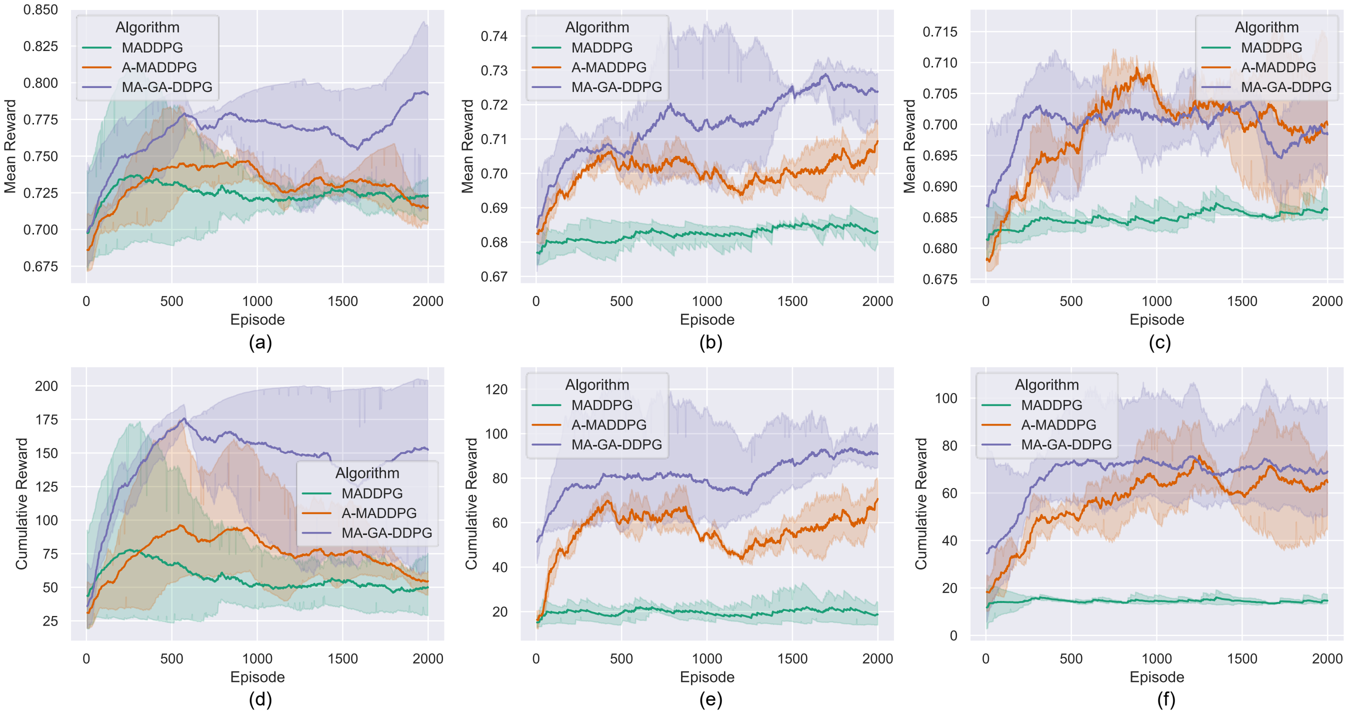
C. Performance Evaluation

1) Overall Performance: The average and cumulative rewards of the agent during training are shown in Fig. 6. In the environment just having CAVs, the performance of Attention-MADDPG and MA-GA-DDPG are both significantly better than that of MADDPG, which proves that the Attention mechanism can effectively improve the performance of the algorithm.

In the CAV-HV mixed driving environment, MA-GA-DDPG shows more obvious advantages. In the mixed driving environment of CAVs and homogeneous HVs, Attention-MADDPG has shown more powerful performance than MADDPG, while the stability and convergence of MA-GA-DDPG are further better than Attention-MADDPG. MA-GA-DDPG obtains the most cumulative rewards during training, indicating that the model agent learns a policy function that can drive safely in traffic for a longer period of time.

In the CAVs and heterogeneous HVs mixed driving environment, HVs with different driving styles will bring more complex and diverse interactive behaviors, which puts forward higher requirements for the learning ability of the model. MA-GA-DDPG still shows the best learning ability, and the average reward and cumulative reward are significantly better than the two baselines. Our proposed model exhibits significant performance improvement in these difficult scenarios.

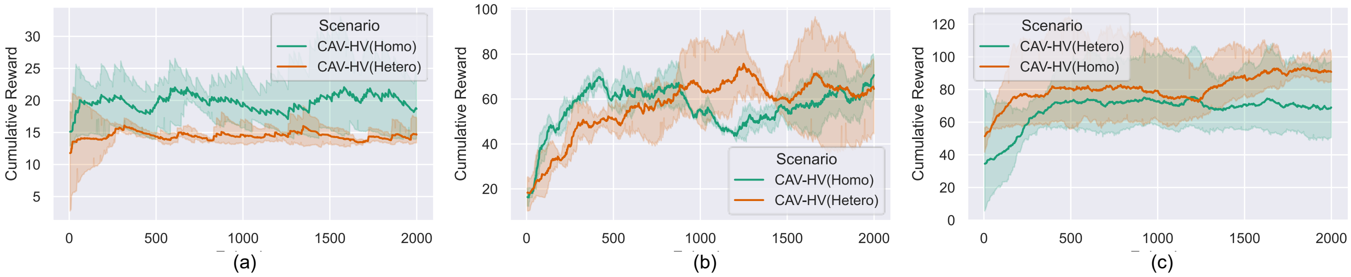
2) Performance in Different Scenarios: At the same time, we compared the performance of the algorithms in different mixed driving environments to analyze the impact and challenges of heterogeneous HVs on model learning. As shown in Fig. 7, in the presence of heterogeneous agents, the performance of the three algorithms varies to different degrees. Specifically, MADDPG displays a certain degree of degradation in performance, whereas Attention-MADDPG shows no significant decline. This indicates that incorporating attention-based policy networks can enhance algorithm performance in complex environments. By contrast, although the performance of MA-GA-DDPG decreases when faced with heterogeneous



(a) (b) (c)

(d) (e) (f)

Fig. 6. The mean reward and cumulative reward of our model and other baselines in different environments, (a) and (d): just CAVs; (b) and (e): CAVs and homogeneous HVs; (c) and (d): CAVs and heterogeneous HVs.



(a) (b) (c)

Fig. 7. The influence of human driving characteristics on performance of different models :(a) MADDPG, (b) Attention-MADDPG, (c) MA-GA-DDPG.

agents, its overall performance still surpasses that of Attention-MADDPG and MADDPG.

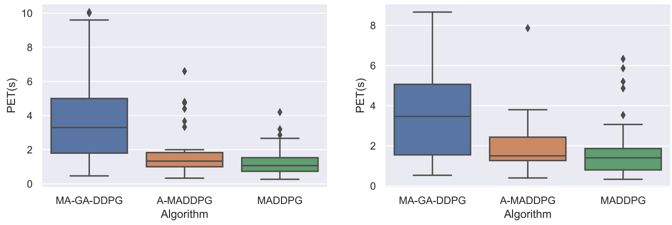
3) Safety Analysis: Safety is the most critical factor to consider when designing AV decision-making algorithms. We first conduct 100 random scenario tests on three algorithms and count the success rate. When all the CAVs in the scenario do not collide and reach the destination smoothly, we record it as a success. After simulation, the success rate of MADDPG, A-MADDPG, and MA-GA-DDPG are 44.0% , 72.0%, 86.0% respectively.

Meanwhile, to evaluate security during interactions, we employ the post-encroachment time (PET) metric [44]. Fig. 8 illustrates the PETs of different algorithms in various scenarios involving interactions between CAVs and other CAVs as well as all HVs. The average PET of MADDPG in CAH-HV (homogeneous) and CAV-HV (heterogeneous) is $1.30s$ and $1.72s$ respectively, and the average PET of A-MADDPG in CAH-HV (homogeneous) and CAV-HV (heterogeneous) is $1.80s$ and $1.96s$ respectively. In contrast, MA-GA-DDPG yields results of $3.61s$ and $3.59s$, representing a 64.0% and 52.1% increase over MADDPG.

Additionally, we found that different algorithms exhibit varying levels of stability when the heterogeneity of the HVs changes. Compared to the homogeneous HV environment, the average PET of MADDPG and A-MADDPG in the heterogeneous HV environment increased by 32.3% and 8.9%, respectively, while the performance of MA-GA-DDPG was more stable, with an average PET fluctuation of only 0.5%.

Overall, our simulation results and analysis indicate that MADDPG is the most aggressive in the interaction process, with the poorest safety performance, while our algorithm is safer and more stable.

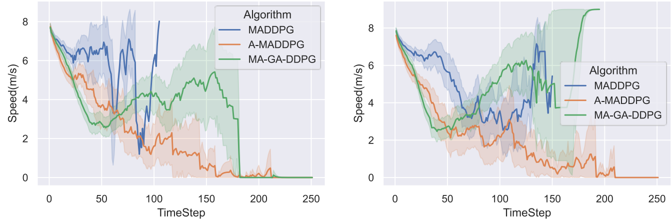
4) Efficiency and Comfort Analysis: We expect that AVs can maintain both efficiency and comfort while prioritizing



(a)

(b)

Fig. 8. CAVs' PET statistics of different algorithms in different scenarios: (a)CAVs and homogeneous HVs; (b) CAVs and heterogeneous HVs.



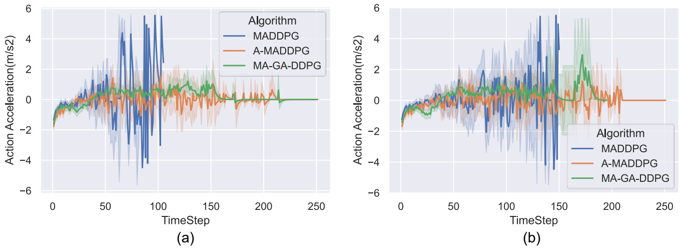
(a)

(b)

Fig. 9. The average speed of CAVs from different algorithms: (a) CAVs and homogeneous HVs; (b) CAVs and heterogeneous HVs.

safety. To gauge the effectiveness of AVs, we examine the speed variation curve of AVs in testing scenarios, while assessing driving comfort by analyzing longitudinal acceleration. As depicted in Fig. 9, the MADDPG algorithm exhibits the highest average speed in both mixed driving environments; however, its aggressive driving behavior leads to dangerous interactions and frequent collisions, as previously analyzed. The A-MADDPG algorithm registers the lowest average speed, but it passes through intersections at a slow pace, resulting in decreased efficiency. On the other hand, our MA-GA-DDPG algorithm can promptly decelerate to ensure interaction safety when approaching the intersection and then accelerate appropriately after passing the conflict point, striking a balance between safety and efficiency.

The acceleration curve, shown in Fig. 10, highlights that MADDPG has an excessive amplitude of acceleration and deceleration when navigating through intersections, causing significant discomfort. In contrast, the A-MADDPG and MA-GA-DDPG algorithms exhibit gentler acceleration and deceleration amplitudes, ensuring improved comfort.



(a)

(b)

Fig. 10. The average action acceleration of CAVs from different algorithms: (a) CAVs and homogeneous HVs; (b) CAVs and heterogeneous HVs.

D. Case Analysis

To investigate the performance of CAVs at the micro level of traffic interactions, three representative interaction cases are selected for analysis. Fig. 11 displays snapshots, velocity curves, and trajectory spatiotemporal plots from these cases. The CAVs approaching from the South Entrance, West Entrance, North Entrance, and East Entrance are labeled as CAV $Nos.1, 2, 3$, and 4, respectively, in all cases. The animated version of three cases can be found at https://drive.google.com/drive/folders/1v8wtHtBeGzpuh3E-qNiGBQKEXMtn2G3K?usp=sharing.

In **Case 1**, in addition to the four CAVs, six HVs from different entrances and with different driving objectives were present. As the CAVs slowed down and yielded, most of the HVs had already left the intersection by the time the four CAVs arrived. At $Timestep = 30$, CAV4 at the South Entrance attempted to turn left and cross the intersection at a speed of $3.40m/s$, but encountered a potential conflict with a straight-going HV at the North Entrance. After assessing the risk, CAV4 opted to slow down and yield. Once the HV passed, the CAV at the North Entrance ($Index = 3$) followed the HV and turned left, crossing the intersection at $Timestep = 50$. The remaining three CAVs crossed the intersection sequentially after negotiation at $Timestep = 90$.

In **Case 2**, four CAVs and six HVs were presented. Upon entering the intersection at $Timestep = 1$, all CAVs began to slow down, and the CAV ($Index = 3$) at the North Entrance approached the center of the intersection and reduced its speed to $3.19m/s$. After determining that there would be no trajectory conflict, it accelerated directly through the intersection at $Timestep = 35$. The remaining three CAVs displayed a more cautious approach, engaging in an obvious trial process. Ultimately, the order of passage was determined through negotiation, and the conflict points at the intersection were crossed safely in sequence at $Timestep = 90$.

However, not all CAVs drove conservatively and yielded blindly. In **Case 3**, which included seven HVs, all CAVs initially slowed down. At $Timestep = 50$, three were still three straight-going HVs and one left-turning HV in the intersection, representing a serious potential conflict. Nevertheless, the CAV at the East Entrance($Index = 4$) chose to accelerate directly after a straight HV at the North Entrance has passed ($Timestep = 63$), with a speed of approximately $5.10m/s$, despite having the potential to conflict with the HVs at the South Entrance, North Entrance, and West Entrance. All HVs were forced to stop and avoid. After CAV4 had completely crossed the conflict point, the remaining HVs passed through the intersection sequentially, while the other CAVs negotiated and crossed the intersection in turn at the end. In **Case 3**, we observed that CAV4 chose an aggressive and dangerous action strategy. This finding suggests that our algorithm has the potential to generate aggressive action strategies. However, based on our observations, such occurrences are relatively rare.

Overall, the CAVs controlled by our MA-GA-DDPG mdoel exhibited a robust and cautious driving style. After entering the intersection, they first slowed down and observed their surrounding environment, predicting their style and future

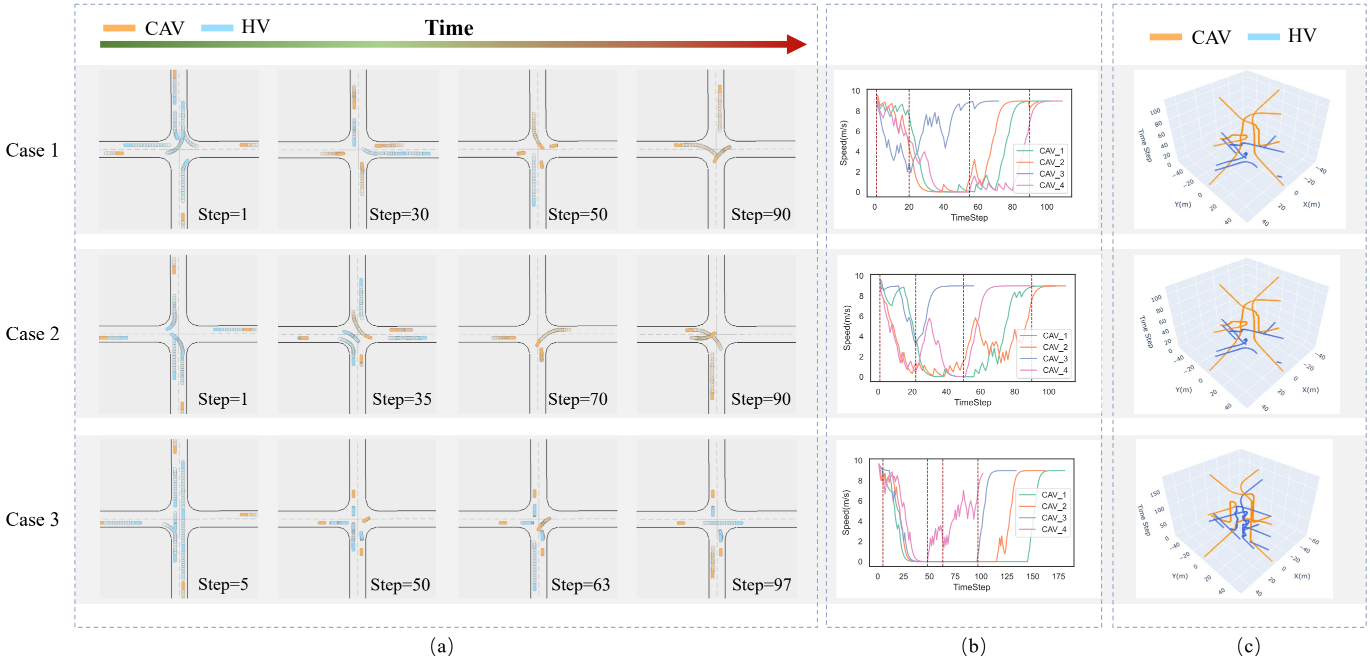


Fig. 11. Details from three interaction cases: (a) snapshots at critical moments, (b) velocity curves during the interaction, and (c) trajectory spatiotemporal plots.

trajectory. They then made different decision-making behaviors, such as accelerating, stopping, or yielding, based on the conflict situation, taking into account safety and efficiency, and displaying excellent interaction capabilities. Finally, the CAVs passed the action test and negotiated among themselves, safely crossing the intersection.

VI. Conclusions

Cooperative decision-making for CAVs in complex human-machine driving environments poses significant challenges for researchers. Despite advancements in autonomous driving technology, current systems still face issues in complex environments, such as unsignalized intersections. This paper formulates the decentralized MARL problem for CAVs at unsignaled intersections and proposes a novel algorithm called MA-GA-DDPG. The algorithm combines an attention mechanism with level-k game priors to enhance the efficiency and safety of CAVs. An attention-based policy network is designed to improve learning efficiency and capture interaction dependencies between the ego CAV and other agents. The attention weights are then used as interaction priors, which are modeled as a level-k game process. Based on the hierarchical game relations, a safety inspector module is constructed to improve the safety performance of CAVs. A series of comprehensive experiments are conducted considering the heterogeneity of human drivers to simulate a more realistic environment. The results demonstrate that our MA-GA-DDPG algorithm provides significant improvements in safety, efficiency, and comfort.

In the future, we plan to extend our algorithm to more complex intersection scenarios and other contexts. We will consider the social interaction between CAVs and human-driven vehicles to ensure that CAVs drive in a manner that resembles human drivers while maintaining safety and efficiency. Additionally, we will design multi-task modules for the algorithm and strive to improve its generalization to ensure stable performance in various scenarios.

References

[1] S. Aradi, “Survey of deep reinforcement learning for motion planning of autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2020.

[2] Q. Rao and J. Frtunikj, “Deep learning for self-driving cars: Chances and challenges,” in *Proceedings of the 1st international workshop on software engineering for AI in autonomous systems*, 2018, pp. 35–38.

[3] R. Chandra, “Towards autonomous driving in dense, heterogeneous, and unstructured traffic,” Ph.D. dissertation, University of Maryland, College Park, 2022.

[4] Y. Rahmati, M. K. Hosseini, and A. Talebpour, “Helping automated vehicles with left-turn maneuvers: a game theory-based decision framework for conflicting maneuvers at intersections,” *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[5] M. Yuan, J. Shan, and K. Mi, “Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 818–825, 2021.

[6] N. Li, Y. Yao, I. Kolmanovsky, E. Atkins, and A. R. Girard, “Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1428–1442, 2020.

[7] X. Pan, B. Chen, S. Timotheou, and S. A. Evangelou, “A convex optimal control framework for autonomous vehicle intersection crossing,” *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[8] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, “Flow: A modular learning framework for mixed autonomy traffic,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1270–1286, 2021.

[9] Z. Dai, T. Zhou, K. Shao, D. H. Mguni, B. Wang, and H. Jianye, “Socially-attentive policy optimization in multi-agent self-driving system,” in *6th Annual Conference on Robot Learning*.

[10] C. Zhang, K. Kacem, G. Hinz, and A. Knoll, “Safe and rule-aware deep reinforcement learning for autonomous driving at intersections,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 2708–2715.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] E. Leurent and J. Mercat, "Social attention for autonomous decision-making in dense traffic," *arXiv preprint arXiv:1911.12250*, 2019.

[13] L. Wei, Z. Li, J. Gong, C. Gong, and J. Li, "Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 44–51.

[14] A. Guillen-Perez and M.-D. Cano, "Raim: Reinforced autonomous intersection management—aim based on madrl," *Proceedings of the NeurIPS*, 2020.

[15] D. Li, A. Liu, H. Pan, and W. Chen, "Safe, efficient and socially-compatible decision of automated vehicles: a case study of unsignalized intersection driving," *arXiv preprint arXiv:2111.02977*, 2021.

[16] P. Hang, C. Huang, Z. Hu, and C. Lv, "Driving conflict resolution of autonomous vehicles at unsignalized intersections: A differential game approach," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5136–5146, 2022.

[17] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Autonomous Agents and Multiagent Systems, International Joint Conference on*, vol. 3. Citeseer, 2004, pp. 530–537.

[18] P. Lin, J. Liu, P. J. Jin, and B. Ran, "Autonomous vehicle-intersection coordination method in a connected vehicle environment," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 37–47, 2017.

[19] 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 Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1136–1147, 2014.

[20] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using robust model predictive control," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 233–238.

[21] S. Hecker, D. Dai, and L. Van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 435–453.

[22] N. Aloufi and A. Chatterjee, "Autonomous vehicle scheduling at intersections based on production line technique," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–5.

[23] J. Xue, B. Li, and R. Zhang, "Multi-agent reinforcement learning-based autonomous intersection management protocol with attention mechanism," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2022, pp. 1132–1137.

[24] J. Wang, Q. Zhang, and D. Zhao, "Highway lane change decision-making via attention-based deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 567–569, 2021.

[25] D. Chen, Z. Li, Y. Wang, L. Jiang, and Y. Wang, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," *arXiv preprint arXiv:2105.05701*, 2021.

[26] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Applied Intelligence*, pp. 1–46, 2022.

[27] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. Li, "Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *The world wide web conference*, 2019, pp. 3620–3624.

[28] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Social coordination and altruism in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 791–24 804, 2022.

[29] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes *et al.*, "The hanabi challenge: A new frontier for ai research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.

[30] C. Zhang, V. Lesser, and P. Shenoy, "A multi-agent learning approach to online distributed resource allocation," in *Twenty-first international joint conference on artificial intelligence*, 2009.

[31] J. Zhang, C. Chang, X. Zeng, and L. Li, "Multi-agent drl-based lane change with right-of-way collaboration awareness," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[32] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[33] Z. Zhang, S. Han, J. Wang, and F. Miao, "Spatial-temporal-aware safe multi-agent reinforcement learning of connected autonomous vehicles in challenging scenarios," *arXiv preprint arXiv:2210.02300*, 2022.

[34] Z. Cao and J. Yun, "Self-awareness safety of deep reinforcement learning in road traffic junction driving," *arXiv preprint arXiv:2201.08116*, 2022.

[35] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[36] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, 2017, pp. 812–818.

[37] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[38] ——, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.

[39] M. T. Spaan, "Partially observable markov decision processes," *Reinforcement learning: State-of-the-art*, pp. 387–414, 2012.

[40] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[41] Y. Wen, Y. Yang, and J. Wang, "Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 414–421.

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

[43] D. Zhang, X. Chen, J. Wang, Y. Wang, and J. Sun, "A comprehensive comparison study of four classical car-following models based on the large-scale naturalistic driving experiment," *Simulation Modelling Practice and Theory*, vol. 113, p. 102383, 2021.

[44] Z. Ma, J. Sun, and Y. Wang, "A two-dimensional simulation model for modelling turning vehicles at mixed-flow intersections," *Transportation Research Part C: Emerging Technologies*, vol. 75, pp. 103–119, 2017.