Multi-agent Reinforcement Learning-based Decision Making for Twin-vehicles Cooperative Driving in Stochastic Dynamic Highway Environments

Siyuan Chen, Meiling Wang, *Member, IEEE,* Wenjie Song*,
Yi Yang, *Member, IEEE,* Mengyin Fu, *Member, IEEE,*

***Abstract*—In the past decade, reinforcement learning (RL) has achieved encouraging results in autonomous driving, especially in well-structured and regulated highway environments. However, few researches pay attention to RL-based multiple-vehicles cooperative driving, which is much more challenging because of dynamic real-time interactions and transient scenarios. This paper proposes a Multi-agent Reinforcement Learning (MARL) based twin-vehicles cooperative driving decision making method which achieves the generalization adaptation of the RL method in highly dynamic highway environments and enhances the flexibility and effectiveness of collaborative decision making system. The proposed fair cooperative MARL method pays equal attention to the individual intelligence and the cooperative performance, and employs a stable estimation method to reduce the propagation of overestimated joint Q-values between agents. Thus, the twin-vehicles system strikes a balance between maintaining formation and free overtaking in dynamic highway environments, to intelligently adapt to different scenarios, such as heavy traffic, loose traffic, even some emergency. Targeted experiments show that our method has strong cooperative performance, also further increases the possibility of creating a harmonious driving environment.**

***Index Terms*—Multi-agent Reinforcement Learning (MARL), Cooperative driving, Fair cooperation, Overestimation.**

I. Introduction

Autonomous driving has received significant research interests in the past two decades due to its many potential social and economical benefits. Compared to human driving vehicles, autonomous vehicles (AVs) not only promise fewer emissions but are expected to improve safety and efficiency. In current industrial applications, like Baidu Apollo [1], decision making modules for AVs are mostly rule-based, and have made significant progress in urban roads, semi-closed parks and highway scenarios. However, the rule-based methods are computationally intensive and still lack of flexibility and intelligence in practical applications. Reinforcement Learning (RL) is thriving, which has emerged as a promising framework for autonomous driving due to its online adaptation capabilities and the ability to solve complex problems [2]. [3], [4]. At the same time, with the in-depth development of object detection [5] [6] and behavior prediction [7], the results can be used as the input of reinforcement learning, which effectively improves the learning effect. Many researchers have successfully used the RL-based approaches for cruising and lane changing in simple scenarios with an end-to-end way.

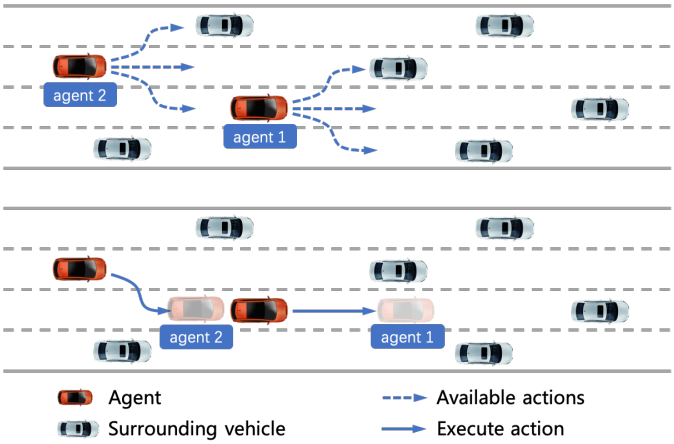


Fig. 1: Collaborative driving scenarios.

Furthermore, on the basis of single-vehicle autonomous driving, the cooperation between AVs also becomes an important research content with the development of the actual intelligent transport system. It can greatly improve the safety and efficiency of the transport system on the premise of making full use of road capacity. Many companies, like Hyundai, used V2V communication, as well as the traditional leader-follower path planning framework for truck formation. But these cooperative tasks based on traditional formation control is not flexible enough in highway scenarios, which is easy to cause interference to the surrounding normal driving vehicles when there are many participants. In order to reduce the impact of formation on the surrounding vehicles and improve the driving efficiency of the fleet, limited priority can be given to obstacle avoidance when necessary to weaken the requirements of formation, which can be gradually restored when the number of surrounding vehicles decreases. Such requirements often put higher demands on the intelligence level of both the individual vehicle and the fleet, which is difficult to formulate in terms of rules.

In the aspect of synchronously improving the individual intelligence and collaborative intelligence, Multi-agent Re-

This work was partly supported by National Natural Science Foundation of China (Grant No. 61903034, U1913203 and 61973034), Program for Changjiang Scholars and Innovative Research Team in University (IRT-16R06, T2014224), Beijing Institute of Technology Research Fund Program for Young Scholars. (Corresponding author: Wenjie Song (email: songwj@bit.edu.cn))

The authors are with School of Automation, Beijing Institute of Technology, Beijing 100081, P.R.China.

inforcement Learning (MARL) has been greatly developed and successfully applied to a variety of complex multi-agent systems such as games [8], traffic light control [9]. In their work, agents can collaborate with each other to achieve common goals. However, few researches focused on AVs collaboration tasks. Although [10] and [11] claimed that they solved the consultation between AVs, they just treated the other AVs as a part of the environment. So it is clear that the consultation was one-sided. According to our analysis, there are several significant challenges for using MARL for collaborative driving tasks. First, it is difficult to extract high-level decision features because the driving scenes including the number of surrounding vehicles, their spatial distribution and driving status are highly stochastic, resulting in extremely large sample space. Second, learning to cooperate is more difficult than learning to compete since the reward function always drives agents to adopt self-interested strategies. Third, the estimation error in single-agent RL methods will be further amplified in MARL methods, which will further affect the performance and learning stability of the model.

In order to solve these problems, we constructs a twin-vehicles cooperative driving task in stochastic dynamic highway environments based on previous work [12], which is modelled as a Markov Game problem. A cooperative MARL algorithm with a fairness measurement factor is introduced to realize the fairness collaboration. Different from the existing methods, its characteristics are as follows: 1) Using distributed decision making mode, each vehicle has independent intelligence, which can realize cooperative formation on the premise of achieving safe obstacle avoidance; 2) When one of the vehicles in the team breaks down, the other vehicle can continue to run normally. Targeted experiments show that our method not only has strong cooperative performance, but also has better stability and environmental adaptability.

II. Related Work

Multi-agent system has been studied and applied to the fields of area exploration, payload transportation and game in prior works [13]–[15], which is seldom applied in the field of autonomous driving. Our work focuses on the cooperative driving of twin-vehicles in complex highway scenarios. In the field of autonomous driving, multi-intelligent collaborative driving can be divided into two categories, rule-based and learning-based collaborative driving methods. These two aspects are discussed in this section.

A. Traditional Cooperative Approaches

In terms of rule-based methods, various control algorithms have been implemented to keep the agents in formation which are typically based on leader-follower [16], virtual structure [17] and behaviour modeling based control architecture [18]. The shortcomings of these methods are evident in the lack of system flexibility and adaptability to stochastic dynamic environments.

B. Learning-based Cooperative Approaches

Learning-based methods, especially RL-based methods, have been widely used in the field of single agent autonomous driving due to their outstanding representation ability for diverse states and the self-adaptive ability for the complex tasks. [19], [20] uses RL for formation control in conjunction with leader-follower architecture, from which the agent can learn how to maintain a specified distance from a designated leader. The RL-based controller alongside a behaviour based controller is used in [21], in which the action was denoted as the average of these two controllers. In these cases, the agents work together for individual goals without considering overall goal of the system. Particularly, for multi-vehicles cooperative driving, collaborative tasks between agents should also be emphasized to achieve the balance between the whole system and the individual flexible driving. Some works focus on the interaction between to cooperate. M. Knopp et al. [19] leverages global information for centralized training to learn to merge into lanes on ramps. [22] and [23] take advantage of the communication between agents to enable information sharing. With this communication assumption, the agents can be treat as individual agents. However, prior knowledge and interaction may play a defining role in these algorithms which are hard to obtain. Up to now, there have been relatively few such studies about MARL-based cooperative autonomous driving.

Actually, MARL algorithms have been successfully applied to collaborative tasks in game fields such as StarCraft [15], which are mainly divided into policy based and value based methods. Lowe et al. [15] proposed a policy based algorithm MADDPG, which provides a general idea to solve the multi-agent problem. Yu et al. [24] proposed an on-policy algorithm multi-agent proximal policy optimization (MAPPO), whose performance is comparable to that of off-policy algorithms. In terms of value-based methods, QMIX [25] is proposed as an improvement over value decomposition network (VDN) [26], which estimates joint action values as a non-linear monotonic function of per-agent values. However, due to the monotonic assumption, it is easy to fall into a local optima. QTRAN [27], Weighted-QMIX [28], QPLEX [29] are proposed to relaxed the monotonic constraint. These methods expand the solution space of joint action-value function. But the performance still need to be improved in many complex scenarios [30].

As we mentioned before, our work aims to solve the problem of twin-vehicles cooperation in stochastic dynamic scenarios without information interaction, which have more randomness, interactions and constraints than game scenarios and require ensuring the generalizability and robustness of the model. Inspired by the above methods, we designed a hybrid framework(Fig. 4), in which the upper decision making module is separated from the lower control module. The upper layer employs a value-based MARL algorithm for the flexible decision making. The concept of fairness collaboration is proposed to cope with collaborative driving tasks and improve the algorithm performance through a stable joint Q estimation method. The lower layer optimizes the speed and orientation respectively through the lane keeping and lane change models to ensure the feasibility and safety. As a whole, our work

significantly extended MARL in the twin-vehicles cooperative driving scenario with the main contributions as follows:

- The cooperative decision making problem in stochastic dynamic scenarios is formulated as a decentralized Markov Game, and normalized input is used to improve feature extraction, so as to improve adaptability to dynamic environments;
- The fairness of collaboration is introduced to measure the performance of the two agents, which promotes the agents learn to cooperate while making decisions independently;
- A stable estimation method is proposed to reduce the propagation of the overestimated Q values between agents, which significantly improves the performance and stability of the algorithm;
- We design and simulate realistic cooperative driving environments to evaluate and compare our approaches against recent MARL baselines, and realize the flexible cooperative driving in different scenarios.

III. PRELIMINARIES

In this section, the algorithms involved in our work including the preliminaries of decentralized partially observable Markov decision processes(Dec-POMDPs), centralized training with decentralized execution(CTDE), and the QMIX algorithm are presented.

A. Decentralized Partially Observable Markov Decision Process

For multi-vehicle driving problem, each agent can only obtain incomplete perception and probability execution. Therefore, we model the cooperation between multi-vehicles as partially observable Markov decision process model(POMDP), particularly modelling it as a distributed POMDPs model in this paper. It can usually be expressed in a tuple,

$$G = \langle S, A, P, r, Z, O, n, \gamma \rangle$$

where $s \in S$ describes the true state of the environment. Specially, Dec-POMDPs consider a partially observable scenario in which each agent has individual, partical observation $z \in \mathcal{Z}$ according to observation function $O(s,a) : S \times A \to [0,1]$. At each time step, each agent $i \in N \equiv \{1, \ldots, n\}$ chooses an action $a_i \in A_i$ according to its policy $\pi_i(a_i \mid \tau_i) : \mathcal{T} \times \mathcal{A} \mapsto [0,1]$, where $\tau_i \in \mathcal{T} := (\mathcal{Z} \times \mathcal{A})^*$ represents the action-observation history. All those actions form a joint action $a \in \mathcal{A} \equiv A'$. And then the next state is produced according to the state trasition function $P(s' \mid s,a) : S \times \mathcal{A} \times S \to [0,1]$. All agents share the same reward function $r(s,a) : S \times \mathcal{A} \to \mathbb{R}$. The action-value function is defined as $Q^{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1:\infty}, a_{t+1:\infty}}[R_t \mid s_t, a_t]$, where $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ is a discounted form.

B. Centralized Training with Decentralized Execution

CTDE [31] has been shown to be a powerful paradigm achieving better performance. With this framework, the agents can use additional information (e.g. the global state, actions or rewards of the environment) during training while make decisions only according to its own action-observation history when execution.

To enable the CTDE paradigm effectively used for multi-agent, it is critical that the joint optimal behavior should be equivalent to the set of individual optimal behavior, which is known as IGM principle. The relationship between individual agent and the whole system is usually measured by value decomposition, which is quite consistent with the cooperative scenarios.

Recent works including VDN and QMIX are also follows the CTDE framework. Agents are trained in a centralized way with access to the overall action-observation history and global state during training, but during execution have access only to their own local action-observation histories. Compared with centralised method, there is no need to optimize the joint policy, just individual policy, which seems more easier. IGM property is a popular concept to realize efficient CTDE as in Eq.(1), where Q_{tot} and Q_a denote the joint-action Q-function and agent-wise utilities respectively.

$$\begin{aligned} &\arg\max_{\boldsymbol{u}} Q_{tot}(s, \boldsymbol{u}) \\ &= \left(\arg\max_{u_1} Q_1(s, u_1), \cdots, \arg\max_{u_n} Q_n(s, u_n) \right) \end{aligned} \tag{1}$$

The IGM property enables efficient decentralized execution, which ensures the consistency between greedy action selection in the local and joint Q-values.

C. QMIX

VDN and QMIX are cooperative multi-agent forms of Q-learning, which estimate the optimal joint action value function Q^* as $Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}) : \mathcal{T}^N \times \mathcal{A}^N \to \mathbb{R}$, where $\boldsymbol{\tau} \in \mathcal{T}^N$ is a joint action-observation history, and $\boldsymbol{a}$ is a joint action, represents the sum of individual action-value functions $Q_i(\tau^i, a^i)$. They both decompose Q_{tot} under the premise of monotony, which can be describes as follows:

$$\underset{\mathbf{a}}{\text{argmax}} Q_{tot}(\boldsymbol{\tau}, \mathbf{a}) = \begin{pmatrix} \text{argmax}_{a^1} & Q_1(\tau^1, a^1) \\ & \vdots \\ \text{argmax}_{a^n} & Q_n(\tau^n, a^n) \end{pmatrix} \tag{2}$$

This allows each agent to participate in a decentralised execution by choosing greedy actions with respect to its Q_a.

VDN factorises Q^{tot} into a sum of each agent utilities: $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=0}^{i=n} Q_i(\boldsymbol{\tau}, a_i)$, while QMIX using a continuous monotonic function to mixing Q_i into Q_{tot}: $Q_{tot}(s, \boldsymbol{a}) = f(Q_1(s, a_1), (s, a_2), ..., (s, a_n))$, where $\frac{\partial f}{\partial Q_i} \geq 0$, $\forall i \in N \equiv \{1, \ldots, n\}$.

QMIX is trained like DQN, to minimise the squared TD error Eq. 3 on a minibatch of samples from the replay buffer, the transition tuples (s, a, o, r, s', o', t), in which the agents take joint action $\boldsymbol{a}$ in state s, receive reward r and transition to s' and t is a boolean indicating if s' is a terminal state, o denotes the agents' own observations. Besides, QMIX uses another hypernetworks to fix the mixing function.

$$\mathcal{L}_{TD}(\theta) = \sum_{i=1}^{n} (Q_{tot}(s, \mathbf{a}; \theta) - y_i)^2 \tag{3}$$

$$y_i = r + \gamma \max_{\mathbf{a}'} Q_{tot}(s', \mathbf{a}'; \theta^-) \tag{4}$$

IV. METHODOLOGY

A. Problem Presentation

In this paper, we focus on high-level behavioral decision making for twin-vehicles cooperative driving. Novel methods and targeted experiments are also carried out around the intelligent decision-making of two vehicles. In order to make the system complete, we use conventional planning and control methods as well, which are briefly introduced in section IV. We formulate the cooperative lane-changing of twin-vehicles as a partially observable Markov decision process. As mentioned before, the tuple is composed of three elements: state space S, action space A, reward function R.

1) State Space: To learn and cooperate more effectively, it is an effective representation to separate the vehicles in the environment according to their roles. As shown in the upper left figure in Fig. 4, observation of the agent is divided into two parts, the cooperative vehicle state and the surrounding vehicle state (occupying grid form). The cooperative vehicle state is described as $(x, y, v_x, v_y, \delta_{heading})$, and the surrounding vehicle state is described by occupation grid.

As we assume that there is no support of Vehicle-to-X (V2X) communication mechanism and each vehicle only relies on on-board sensors to detect the surrounding traffic situations, a two-dimensional occupancy grid that reflects the traffic situation around each vehicle is established. Compared with the kinematic characteristics of fixed number surrounding vehicles, the form of occupancy grid can effectively improve the adaptation of agents to the environment. It is suitable for multiple traffic scenarios, instead of depending on road geometry or the number of surrounding vehicles. Fig. 2 gives an example of the grid map, the yellow one is the host car which is one of the cooperation vehicles. The value is set to the collision time for cells occupied by vehicles and 1 for empty cells. The occupied grid map is represented by a matrix and features are extracted by a convolutional layer network, as shown in Fig.4.

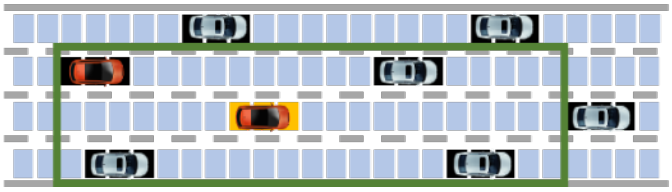
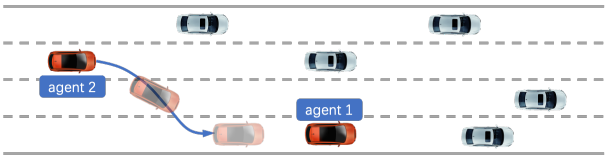


Fig. 2: Local observations in the form of occupied grid map.

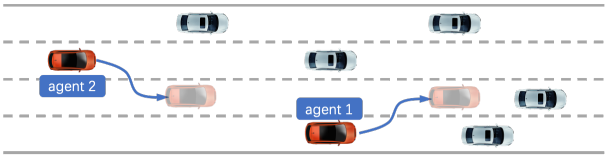
2) Action Space: Action a_i denotes the lane-changing decision of vehicle i and all the actions at each time step form a joint action $a \in \mathcal{A} \equiv A'$. $\mathcal{A}$ is a set defined as belows:

$$a_i \in A = \{\text{left, right, lane-keeping}\}$$

A vehicle can choose to change to the left/right lane or just keep the lane. We use the RL algorithm to learn the proper lane-changing decision, and then the speed is calculated through optimization and control methods which will be described in Section IV.



(a) The unfair cooperative driving scenario.



(b) The fair cooperative driving scenario.

Fig. 3: The representative cooperative driving scenarios.

3) Reward Function: Reward denotes the immediate return after performing an action, and it defines the goal of the learning progress. In this paper, reward is a trade-off among speed, collaboration, and smoothness of the vehicles. We designed the reward function as follows:

$$R = R_v + R_{lc} + R_{cor} \tag{5}$$

$$R_v = \alpha(v_{avg} - v_{ex}) \tag{6}$$

$$R_{lc} = \begin{cases} \beta \frac{v_i}{v_{max}} & \text{if change lane} \\ 0 & \text{others} \end{cases} \tag{7}$$

$$R_{cor} = \sigma(S_{lane} + d_{gap}) \tag{8}$$

In this cooperative task, we use the difference with the expected speed to encourage the improvement of speed, which corresponding to R_v. v_{avg} and v_{ex} denote the average velocity and the expected average velocity, respectively. As for R_{lc}, lane changing behavior is discouraged, the higher the speed, the greater the punishment. v_{max} denotes the max velocity. For R_{cor}, the unexpected formation is also punished. We use S_{lane}, the normalized standard deviation of the lane in which the cooperation vehicles are, and the d_{gap}, the normalized form of the gaps between cooperation vehicles to represent the gap with the expected formation. α, β, σ is the weight factors, which can be changed to appeal to the different tasks. In order to better learn these three features, the three parts need to be clearly differentiated in order of magnitude, so they take the values $(0.8, -0.5, -0.5)$ in this paper.

B. Fairness Measurement

In collaborative scenarios, the agent not only needs to interact with obstacle vehicles, but also needs to consider their teammates, so the dynamic interaction is more difficult. Although the existing methods show outstanding results in some games like StarCraft Multi-Agent Challenge (SMAC), they cannot perform well in some complex cooperative tasks. The performance is even worse than QMIX. Experiments show that it is easy to learn obstacle avoidance and overtaking, while learning to drive cooperatively through the global value function learning is difficult. This is also the reason for low

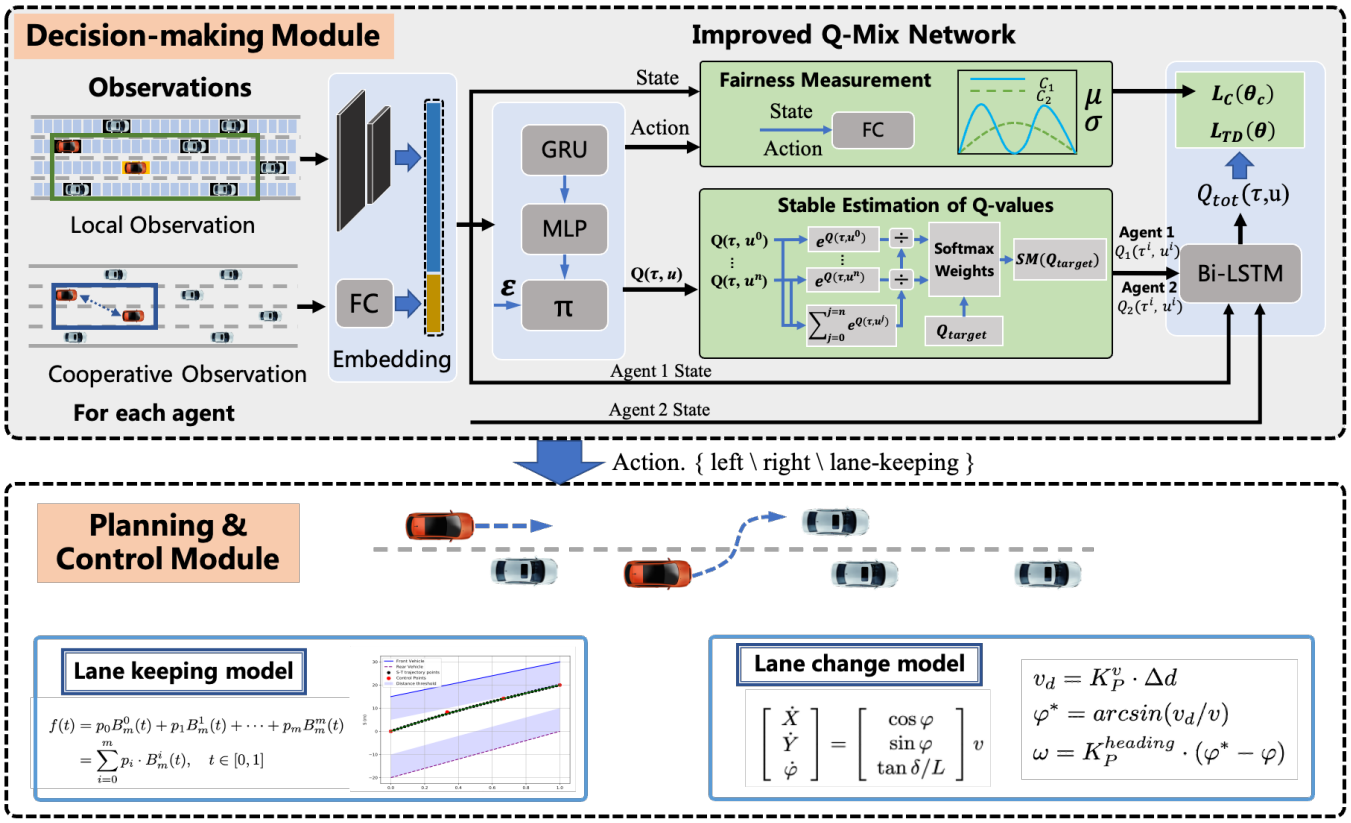


Fig. 4: Dual vehicle cooprative driving decision making and planning system.

Algorithm 1: Fairness Measurement Construction

Input: Action a, Local observation o_{local}, Cooperative observation o_{cor}
Output: Fairness loss $loss_{fair}$

for *episode = 1, ..., N_{eps}* **do**
  for *every agent i* **do**
    (1) Compute the state features :
    parameters are shared with DQN network
    $hidden_{local}$ = ConvLayer(o_{local})
    $hidden_{cor}$ = FC(o_{cor})
    (2) State features are obtained by stitching hidden layer feature vectors. $state_i$ = VectorStack($hidden_{local}$, $hidden_{cor}$)
    m_i, s_i = CalculateGaussianParams(a_i, $state_i$)
    $C_i \sim$ Uniform(m_i, s_i)
  end
  $\mathcal{L}_c = \mathrm{JS}(C_1||C_2)$
end

Algorithm 2: Softmax to Mitigate Overestimation

Input: Q_1, Q_2,..., Q_n; Q_1^-, Q_2^-,..., Q_n^-
Output: Q_{tot}

for *episode = 1, ..., N_{eps}* **do**
  for *every agent i* **do**
    Calculate the softmax factor:
    $w = \frac{e^{\tau Q_i(s,\boldsymbol{a})}}{\sum_{\boldsymbol{a}' \in \boldsymbol{A}} e^{\tau Q_i(s,\boldsymbol{a}')}}$
    Calculate softmax Q_i^-:
    $\mathrm{sm}_{\tau,\boldsymbol{A}}(Q_i^-) = \sum_{\boldsymbol{a} \in \boldsymbol{A}} w \cdot Q_i^-$
  end
  Compute the Q_{tot}:
  $Q_{tot} = f\left(\mathrm{sm}_{\tau,\boldsymbol{A}}(Q_i^-), ..., \mathrm{sm}_{\tau,\boldsymbol{A}}(Q_n^-)\right)$
end

returns. Compared with leader-follower controlling models [32], our collaborative task has no obvious role distinction. Each vehicle has equal status without any "command & obedience" relationship. We hope that their contributions for the whole task are equal, that is, their collaboration is fair.

Some researches solving the problem of social dilemma (such as inequity aversion theory [33]) have inspired us that restricting fairness can promote cooperation effectively. As shown in Fig.3, through fairness constraints, multiple agents can simultaneously contribute to the cooperative goal. The leader-follower model in which the rear vehicle following the front one has low efficiency for coordination. As shown in Fig.3(a), it requires two decision cycles to achieve coordination. By contrast, fairness constraint can promote the decision of lane change of two vehicles at the same time, and the coordination can be achieved within one decision cycle (e.g. Fig.3(b)).

Considering that the agents do not perform explicit communication, we estimate the contribution value distribution of each agent's action and minimize the divergence of the two

distributions to achieve fair cooperation. The KL divergence for calculating the distribution difference between two agents is defined as follows:

$$\begin{aligned} D_{KL}(C_1 \| C_2) &= \mathbb{E}(1) - \mathbb{E}(2) \\ &= \frac{1}{N}\sum_{n=1}^{N} C(x_{1n}) - \frac{1}{N}\sum_{n=1}^{N} C(x_{2n}) \\ &= \frac{1}{N}\sum_{n=1}^{N} \left[-\ln q_1\left(x_n \mid o_1\right) + \ln q_2\left(x_n \mid o_2\right)\right] \end{aligned} \tag{9}$$

where $x_{in}(i = 1, 2)$ denotes the contribution value of the two agents, q_1 and q_2 denote the contribution value distribution of each agent, and C_1, C_2 are abbreviations.

Although KL divergence is a measure of the difference between two probability distributions C_1 and C_2, it is not symmetric, i.e $D_{KL}(C_1 \| C_2) \neq D_{KL}(C_2 \| C_1)$. Therefore, we consider using JS divergence, as shown below.

$$\begin{aligned} \mathrm{JS}(C_1 \| C_2) = &\frac{1}{2} D_{KL}\left(C_1 \| \frac{C_1 + C_2}{2}\right) \\ &+ \frac{1}{2} D_{KL}\left(C_2 \| \frac{C_1 + C_2}{2}\right) \end{aligned} \tag{10}$$

Then, the second objective of the loss function comes up as $\mathcal{L}_c$:

$$\mathcal{L}_c(\theta_c) = \mathrm{JS}(C_1 \| C_2) \tag{11}$$

The procedure of calculating fairness loss is shown in Algorithm 1.

C. Stable Estimation of Q-values

As researches [34] mentioned that Deep Q-Network (DQN) [35] was known to overestimate action values under certain conditions due to the max operation. Since the very basic network of QMIX is DQN, the overestimate problem may even worse in QMIX due to the summarize of each Q value.

Double DQN [34] maintains two deep Q networks to prevent overestimation of Q values in Max operations. However, in the environment of large state space such as autonomous driving with two-vehicle cooperation in this paper, double DQN still does not eliminate the severe overestimation bias in the joint action Q function and the performance degrades in the late period of training, as shown in Fig. 14(d). We propose a stable Q-value estimation method by using the $softmax$ operation to avoid overestimation.

The softmax operator for $q^-(s, \cdot)$ is defined in Eq. 12, where $\tau \geq 0$ is the inverse temperature parameter, $q(s, \boldsymbol{a})$ is action-value function and $q^-(s, \boldsymbol{a})$ is the target action-value function.

$$\mathrm{sm}_{\tau,\boldsymbol{A}}\left(q^-(s, \cdot)\right) = \sum_{\boldsymbol{a} \in \boldsymbol{A}} \frac{e^{\tau q(s,\boldsymbol{a})}}{\sum_{\boldsymbol{a}' \in \boldsymbol{A}} e^{\tau q(s,\boldsymbol{a}')}} q^-(s, \boldsymbol{a}) \tag{12}$$

$$\begin{aligned} &\left|\mathrm{sm}_{\tau,\boldsymbol{A}}\left(q(s, \cdot)\right) - \max_{\boldsymbol{s} \in \boldsymbol{A}} q(s, \boldsymbol{a})\right| \\ &\leq \left|\mathrm{sm}_{\tau,\boldsymbol{A}}\left(q(s, \cdot)\right) - \min_{\boldsymbol{s} \in \boldsymbol{A}} q(s, \boldsymbol{a})\right| \\ &\leq \left|\mathrm{sm}_{\tau,\boldsymbol{A}}\left(q(s, \cdot)\right)\right| + \left|\min_{\boldsymbol{a} \in \boldsymbol{A}} q(s, \boldsymbol{a})\right| \\ &\leq \left|\max_{\boldsymbol{a} \in \boldsymbol{A}} q(s, \boldsymbol{a})\right| + \left|\min_{\boldsymbol{a} \in \boldsymbol{A}} q(s, \boldsymbol{a})\right| \\ &\leq 2 \max_{\boldsymbol{a} \in \boldsymbol{A}} \left|q(s, \boldsymbol{a})\right|. \end{aligned} \tag{13}$$

According to the network structure features $\frac{\partial f}{\partial Q_i} \geq 0$ of QMIX, Q_{tot} is shown in Eq. 14.

$$\begin{aligned} Q_{tot}(s, \boldsymbol{a}) &= f(Q_1(s, a_1), Q_2(s, a_2), ..., Q_n(s, a_n)) \\ &\geq f\left(\mathrm{sm}_{\tau,\boldsymbol{A}}\left(Q_1(s, \cdot)\right), \mathrm{sm}_{\tau,\boldsymbol{A}}\left(Q_2(s, \cdot)\right),\right. \\ &\left. ..., \mathrm{sm}_{\tau,\boldsymbol{A}}\left(Q_n(s, \cdot)\right)\right) \end{aligned} \tag{14}$$

The full algorithm for computing the stable estimation of Q-values is in Algorithm 2.

D. Model and Training Framework

The proposed model is shown in Fig.4, which is composed of three parts: observation encoder, deep Q-network and mixing network.

For more flexible representation, the states of the surrounding obstacle vehicles and the cooperating vehicles are encoded separately. For the surrounding vehicles, the occupancy grid map is input to the convolutional layer. While for the cooperative vehicles, we input the state vector to the fully connected layer. After connecting the results of these two encoding parts, we compute the individual action values based on the deep Q-network. The fairness measurement network is set up to compute the distribution of contributions for each agent in the cooperative task. By calculating the gap between these two distributions, we can obtain the fairness of cooperation.

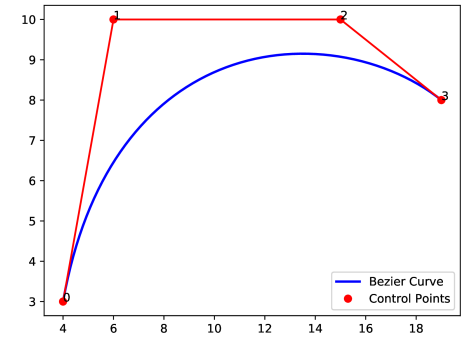
Considering that it is difficult to obtain the global state in a real task, the global state should not be used frequently. Inspired by the vehicle communication field, in which agents need to interact with each other, the Bidirectional LSTM (Bi-LSTM) network [36] was used to make the agents interact with each other so that they can perceive more plentiful information of the surrounding status. In this paper, we also utilize the information interaction to replace the use of global state.

V. Motion Planning

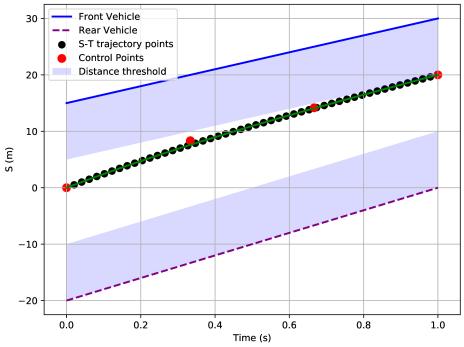
In this section, the details of the low-level motion planning module are presented as follows.

A. Optimization of Bézier Curve in Lane Keeping Scenarios

In lane keeping scenarios, an optimal trajectory under dynamic constraints is generated by Bézier curve optimization method based on ST-graph. As shown in Fig. 5(a), the convex hull property of the Bézier curve ensures that the curve is completely confined to the convex hull supported by control points. Thus, by limiting the control points inside the convex



(a) Illustration of the convex hull property of Bezier curve.



(b) Optimization of Bezier Curves in ST Graph.

Fig. 5: Motion planning model for lane keeping scenarios.

free space, the curve obtained is guaranteed to be collision free.

A degree-m Bézier curve $f(t)$ is defined by $m+1$ control points as follows,

$$\begin{aligned} f(t) &= p_0 B_m^0(t) + p_1 B_m^1(t) + \cdots + p_m B_m^m(t) \\ &= \sum_{i=0}^{m} p_i \cdot B_m^i(t), \quad t \in [0,1] \end{aligned} \tag{15}$$

where p_i denotes the control point and $B_m^i(t) = \binom{m}{i} t^i \cdot (1-t)^{m-i}$ is the Bernstein basis. Denote the set of control points $[p_0, p_1, \cdots, p_m]$ as $\mathbf{p}$.

In this paper, a cubic (m = 4) Bézier curve is adopted as the trajectory parameterization within one second of the lane keeping phase. According to the hodograph property, the k-th derivative of Bézier curve $g^{(k)}(t) = \frac{d^k f(t)}{dt^k}$ is another Bézier curve supported by control points $q_j^{(k)}$, which can be calculated by induction as follows,

$$q_i^{(0)} = p_i, q_i^{(k)} = \frac{m!}{(m-k)!}\left(q_{i+1}^{(k-1)} - q_i^{(k-1)}\right). \tag{16}$$

1) Safety Constraints: To guarantee the generated trajectory is collision-free, we constrain the control points of trajectory in the free space formed by the front and rear vehicles of the ego one, as shown in Fig. 5(b).

2) State Constraints: The generated lane keeping trajectory should start from the given initial state $[s_0^{(0)}, s_0^{(1)}, s_0^{(2)}]$ and terminate at the given goal state $[t_1^{(0)}]$, where $s_\tau^{(k)}$ and $t_\tau^{(k)}$ respectively denote the k-th-order derivative of S and T at time τ. That is to ensure that the initial position, speed, acceleration and terminal position of ego vehicle are fixed. Thus, the first and last control points (p_0, p_m) are determined as $[S_0 = 0, T_0 = 0]$ and $[T_m = 1]$.

3) Dynamical Constraints: Similar to the safety constraints, we can enforce hard constraints on high order derivatives of the entire trajectory. In detail, the velocity and acceleration of ego vehicle are constrained in $[v_{max}^-, v_{max}^+]$ and $[a_{max}^-, a_{max}^+]$ to make the generated trajectory dynamically feasible. According to Eq. 16, the control points are constrained as follows,

$$\begin{aligned} v_{max}^- &\le m \cdot (p_{i+1} - p_i) \le v_{max}^+ \\ a_{max}^- &\le m \cdot (m-1) \cdot (p_{i+2} - 2p_{i+1} + p_i) \le a_{max}^+. \end{aligned} \tag{17}$$

We minimize the cost function given by the time integral of the square of the acceleration, which can be written as,

$$\begin{aligned} J &= \int_0^1 \left(\frac{d^2 f(t)}{dt^2}\right)^2 + \left(v_{max}^+ - \frac{df(t)}{dt}\right)^2 dt \\ &= \mathbf{p}^{\mathrm{T}} \mathbf{Q} \mathbf{p} + \mathbf{C}^{\mathrm{T}} \mathbf{p} \end{aligned} \tag{18}$$

where $\mathbf{Q}$ is the Hessian matrix of the Bézier curve and $\mathbf{C}$ is the coefficient matrix.

Summarizing all the linear equality and inequality constraints, the overall formulation can be written as a quadratic programming (QP) formulation, which can be solved efficiently using off-the-shelf solvers (e.g. CVXOPT [37]).

To cope with the quadratic programming solver timeout and the computed trajectories do not satisfy the constraints, this paper constructed the polynomial lane keeping trajectory while CVXOPT solver solution failed. Due to the unconstrained endpoint position of the longitudinal trajectory, it is represented by a quartic polynomial [38].

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \tag{19}$$

B. Lane Change Model

As shown in Eq. 20, vehicle kinematics model with the rear axle as the origin is adopted in lane change scenes [39].

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \tan\delta / L \end{bmatrix} v \tag{20}$$

where (X, Y) denotes the axle center coordinate of the rear axle of the vehicle, φ is the heading angle, δ is the front wheel steering angle, L is the wheelbase and v is the current speed of ego vehicle.

As the lateral trajectory is not the main factor in our simulation, the speed of ego vehicle is constant while changing the lane. According to the lateral distance deviation (Δd) between the current position of ego vehicle and the target lane, the proportional controller is used to calculate the lateral velocity (v_d) and the yaw angular velocity (ω), respectively.

$$
\begin{aligned}
v_d &= K_P^v \cdot \Delta d \\
\varphi^* &= arcsin(v_d/v) \\
\omega &= K_P^{heading} \cdot (\varphi^* - \varphi)
\end{aligned} \tag{21}
$$

where K_P^v and $K_P^{heading}$ are proportion control parameters, φ^* is the desired heading angle. The front wheel steering angle (δ) can be calculated from Eq. 22, and the state can be updated in the lane changing process according to the kinematics model in Eq. 20.

$$
\begin{aligned}
R &= v/\omega \\
\delta &= \arctan(L/R)
\end{aligned} \tag{22}
$$

VI. Experiments And Results

In this section, the experiments platform and results are introduced. We set up four sets of comparison experiments to verify the effectiveness of our method which are presented in the next four sections. The first one is the overall learning curve, showing the convergence efficiency of the network. The second one is the ability to adapt to the environment, which represents the generalization of the model. Then comes up with the collaborative performance. And the last one is the learning stability. These four parts provide a comprehensive demonstration of the superiority of our method to the collaborative driving task.

A. Experiment Setup

Highway is an important scenario for autonomous driving application because of its clear rules and little social intervention. In this paper, the OpenAI gym-based highway-env simulator [40] is employed to verify the effectiveness of our proposed method. However, the original simulator was for single agent, we improvedit and converted it to the multi-agent module as shown in Fig. 10(a). Specifically, we focused on the dense traffic scene in a four lane one-way highway environment and the lane width is 4 meters. The main task is that two agents cooperate to drive pass through highway without collision. All experiments were implemented in Python 3.8 with pytorch 1.2 on a computer with i9-CPU and NVIDIA GeForce GTX 3090.

TABLE I: Parameters of Q-Fair-MIX

Setting	Item	Value
Training setting	Size of replay buffer	5000 episodes
	Batch size	32 episodes
	Exploration	Anneal noise =50K STEPS
	Discount factor	0.99
Network setting	Optimizer	RMSProp
	Learnning rate	0.001

B. Model Configuration

For better collaboration, a perception network with three parts are set. The first part is a two-linear-layer fully connect network with 32 and 16 units, which is used to extract the features of collaborative agents. The second one is a two-layer convolutional neural network (CNN), which extracts the features of surrounding agents. The last part consists of two hidden layers, which combines the two feature matrices. The combined perception result is referred to hybrid state. The Q-net, mixing-net, and the fairness-net all take the hybrid state as input. The agents share the same Q-net in our work.

To be more realistic, the global states are not used directly in the mixing network, because it is still very unpractical to get the global state in practice because there is no a social scale V2V yet. The hybrid state of each agent is encoded in Bi-LSTM for the input to the mix network as shown in Fig.4.

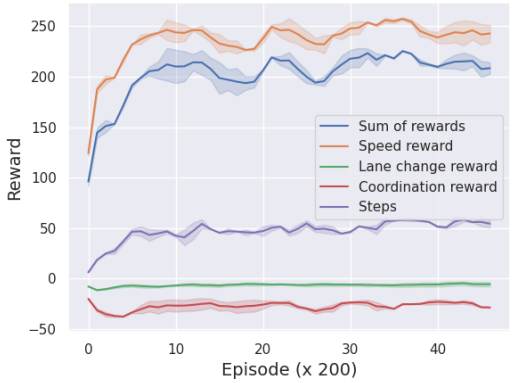
For fairness measurement, two-hidden-layers network is used to calculate the mean and variance of the value distribution to obtain the KL divergence between two agents' values. The hyper-parameters in implementation are set as TABLE I.

Fig. 6: Learning curves of three methods. The model is tested every 400 epochs to calculate the test episode rewards.

C. Study of the Learning Curve

Four algorithms including QMIX, Weighted QMIX, QPLEX and ours are compared in this section. The training curves are represented by a solid line of the mean value and an error band of the 95% confidence interval. Weighted QMIX and QPLEX are the improved methods of QMIX, which perform better than QMIX in the SMAC games. As it can be seen from Fig. 6, our method can reach the highest reward than others. The curve converges faster and the performance is more stable in the later training period. It means the proposed method takes great advantages in searching for optimal solution. According to the reward curves of QMIX and Weighted QMIX in Fig. 6, the curves converge prematurely to a low reward. The two vehicles do not behave cooperatively and even hard to secure themselves in the evaluation experiments. It follows that the agents can hardly make coordinated actions at the same time without the guidelines of fairness mechanism, and the overestimation of Q value may also lead to a local optimum.

For the learning of collaborative tasks, the larger exploration space is not always better. An excessively large search space makes it difficult to find the optimal solution, while the fairness constraint can help reduce the solution space and find the



(a) Performances of three sub-rewards.

(b) Performances of R_{lc} and R_{cor}.

Fig. 7: Impact analysis and learning performance of three sub-rewards.

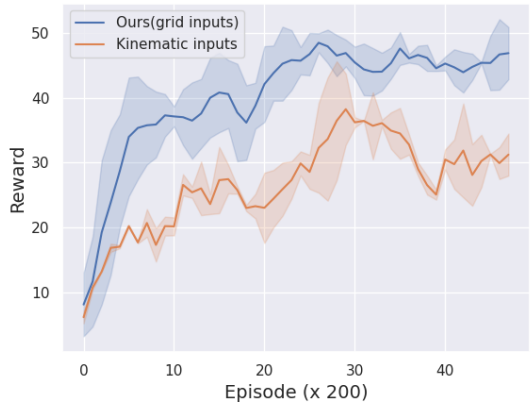


Fig. 8: Learning curves with different kind of observation.

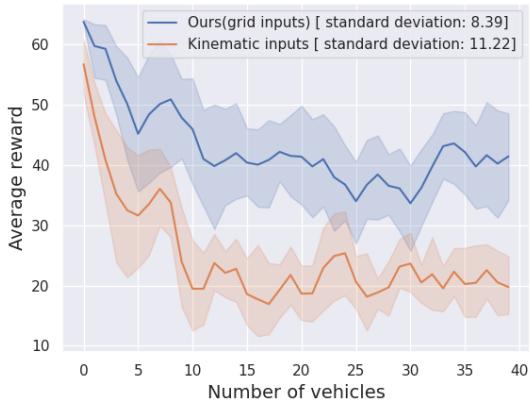


Fig. 9: Variations of driving performance with increasing number of vehicles.

optimal solution faster. $\mathrm{Softmax}$ operation on Q value also has the effect of reducing overestimation, accelerating the model convergence and improving the learning stability.

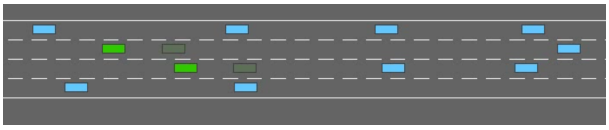
We quantified the learning performance of the three sub-rewards (R_v, R_{lc}, R_{cor}) separately. The raw data statistics for the three components are shown in Fig. 7. Since the experimental tests of an episode may end early due to crashes, the statistical rewards varies with the step length during an episode. To represent the effects of these three components individually, we recalculate them as $R_i = \frac{R_i}{step} * L_{limit}$, where L_{limit} is a hyperparameter representing the step length limit in one episode. This eliminates the effect of step length and reflects the effect of sub-reward's correctly. As can be seen in Fig. 7(a), all three sub-rewards increase during training, and R_v has a more important effect. Fig. 7(b) clearly illustrates the convergence process of R_{lc}, R_{cor} and "Steps". In the early period, since the vehicle did not learn a reasonable lane changing strategy, frequent and unreasonable lane changing led to a low R_{lc} and a drop in R_{cor}. As training continued, the curves of R_{lc} and R_{cor} converged, indicating that the combined effects of lane changing, coordination and speed were balanced as both vehicles gradually learned to make reasonable lane changes for high speed reward.

D. Experiments on Environmental Adaptation

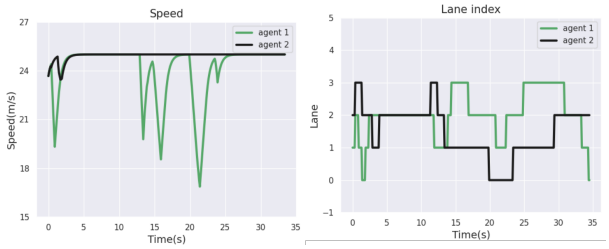
Environmental adaptation is very important for RL-based autonomous driving tasks. In order to demonstrate the robustness of our proposed method to environments with different traffic conditions, the following experiments were carried out.

First we analyzed the effect of the number of vehicles in the environment on the model performance. As shown in Fig. 8 and Fig. 9, the state representation of our method performs better overall than the kinetic representation of fixed number of surrounding vehicles in the presence of the increasing number of surrounding vehicles. And the variance of our method is smaller than that of the kinetic representation, which indicates that our method is able to extract more essential features and thus can cope with different environments more consistently.

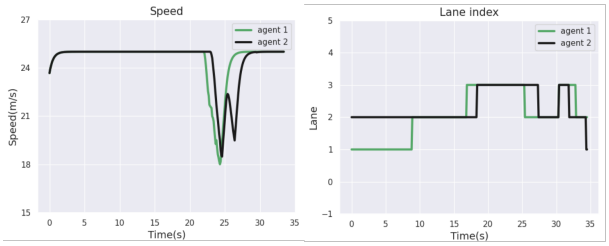
For further analysis, we compared the driving behaviours in loose and congested traffic. The experimental results are shown in Fig. 10 and Fig.12. On the road with heavy traffic, the cooperative vehicles can keep driving smoothly at high speed, and the frequency of lane changes decreased, which makes a more comfortable driving process. To clearly show the lane changes, we sorted the lanes in the scene, and the index of the four lanes from bottom to top is defined as $[0, 1, 2, 3]$. It can



(a) Heavy traffic.

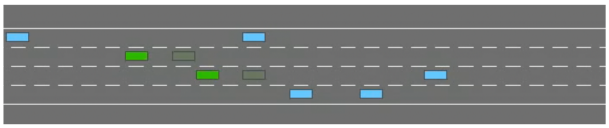


(b) Heavy traffic with Weighted QMIX.

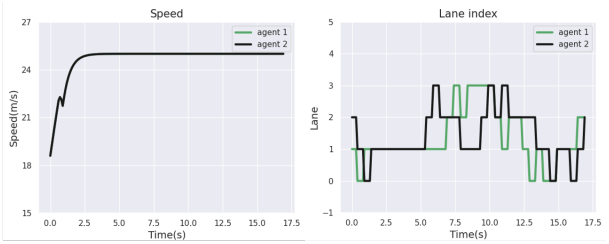


(c) Heavy traffic with our method.

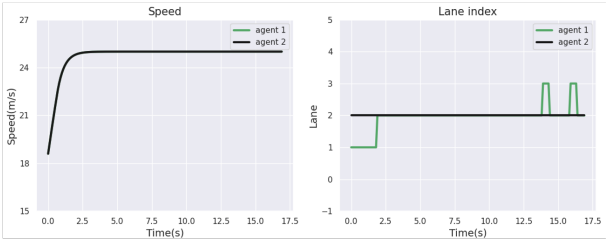
Fig. 10: Driving performance under heavy traffic.



(a) Loose traffic.

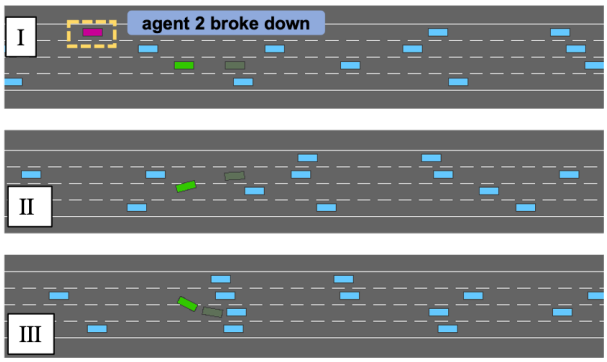


(b) Loose traffic with Weighted QMIX.



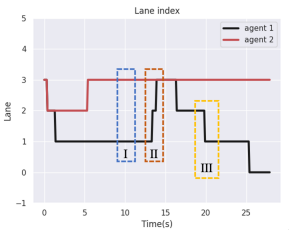
(c) Loose traffic with our method.

Fig. 12: Driving performance under loose traffic.



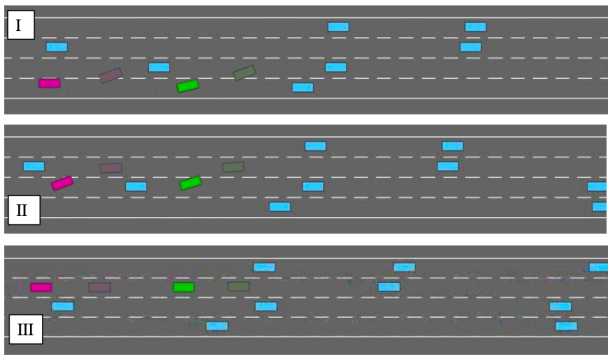
(a) The broken down scene.

(b) Speed curve.

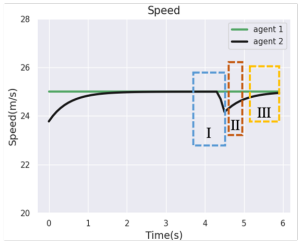


(c) Lane change curve.

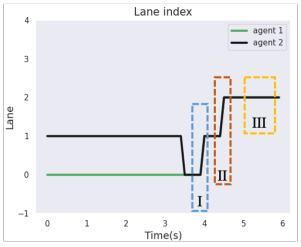
Fig. 11: Driving performance when one vehicle broken down.



(a) Fair collaborative scenario



(b) Speed curve.



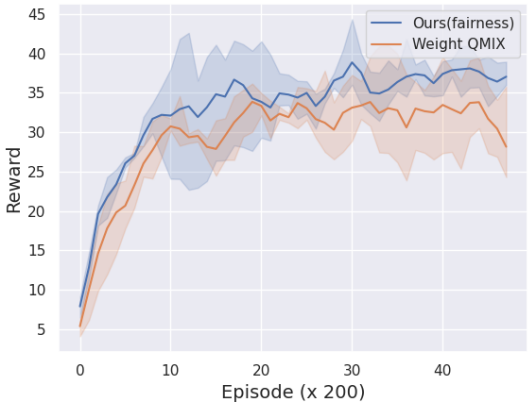
(c) Lane change curve.

Fig. 13: Collaborative performance of our method (QMIX with fairness measurement).

be seen from the Lane index-Time curve, each vehicle passes through the surrounding traffic and then merges. Our proposed method can effectively avoid betrayal, and the vehicles can stay in the same lane or close to each other while overtaking safely.

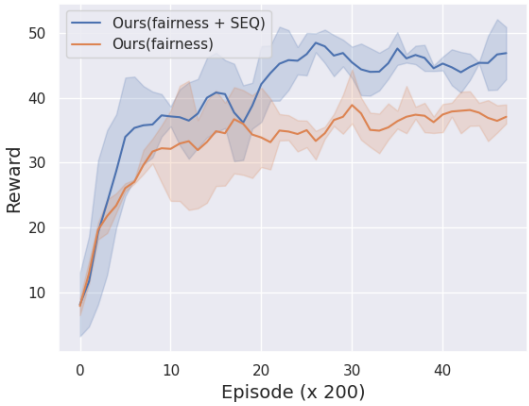
On the road with loose traffic, the goal is to keep high speed while keeping the formation. As shown in Fig. 12(c), unnecessary lane changing is reduced significantly while the unnecessary lane changing still existing with Weighted QMIX. It often appears in Weighted QMIX that high instant rewards are obtained by making and correcting unnecessary lane changing actions without regard to long-term benefits.

In addition, the extreme case is also verified, that is, one

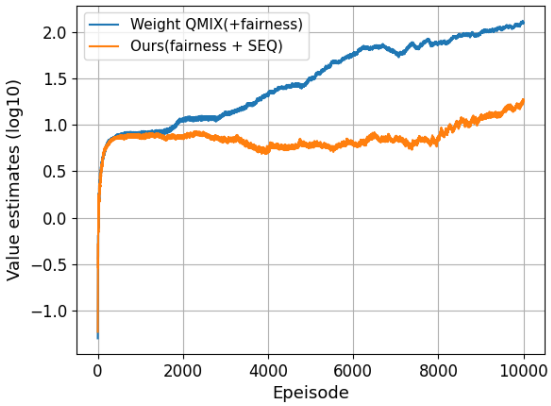


(a) Learning curves of weighted QMIX and QMIX with fairness measurement.

(b) Learning curves of weighted QMIX with and without SEQ.



(c) Learning curves of ours method with and without stable estimation of SEQ.



(d) Value estimation curves for Weighted QMIX (with fairness measurement) and our method (QMIX with fairness measurement and stable estimation of Q-values(SEQ)).

Fig. 14: Benefits of fairness measurement and stable estimation of Q-values(SEQ), and the comparison curves of training effect.

vehicle breaks down during cooperative driving. Since our system follows the CTDE framework and each vehicle has independent intelligence, one vehicle can still drive normally when the other fails thanks to relaxed collaboration constraints, as shown in Fig. 11. After agent 2 has broken down, agent 1 still has the ability to make independent decisions to change lanes to avoid obstacles and increase speed. This is quite important for cooperative tasks, which reflects the robustness and fault tolerance of the system. Since we don't expect a completely collapsed system when just a small part is broken.

TABLE II: Comparison of collaborative performance in different traffic conditions

Traffic conditions	Heavy Traffic		Loose Traffic	
Methods	Our Method	Weighted QMIX	Our Method	Weighted QMIX
Average Speed (m/s)	**24.53**	24.49	**24.73**	24.67
Cooperative Rate (%)	**60.69**	31.79	**82.94**	50.58

E. Experiments on Collaborative Performance

Driven by the fairness measurement, agents can make actions conducive to cooperation at the same time. As shown in Fig.14(a), the proposed method can reach higher rewards with the fairness measurement.

Furthermore, it is clear from Fig. 13 that our method enables two agents to merge into the same lane at the same decision step when conditions permit, so as to achieve the cooperative goal. In stage I, due to the presence of the obstacles ahead, agent 2 (the red vehicle) slows down and changes to the left to avoid collision. In stage II, it continues to change left while accelerating to avoid a collision with the vehicle near behind. Agent 1 (the green vehicle) also changes to the left at the same time. The two vehicles collaborate by lane change and drive safely to a more open road which seems a wisdom decision.

More specific collaboration behaviours can also be illustrated in Fig. 10 and Fig. 12. In order to visualise the benefits of the proposed method, the average speed and cooperative rate of the vehicles in heavy and loose traffic conditions are

calculated separately, as shown in Table II. The cooperative rate represents the percentage of time that both vehicles are in the same lane during the test. Compared to Weighted QMIX, our method is able to achieve a higher cooperative rate with similar average speed.

F. Experiments on Learning Stability

Overestimation in DQN always leads to suboptimal solutions and causes instability during the learning process, which will be further expanded in the multi-intelligence problem. In order to reduce the instability, this paper proposes to use $softmax$ to obtain a more stable estimation of the joint Q-values. Fig. 14(d) evinces the estimating Q_{tot} values with and without $softmax$ operation, while the Fig. 14(c) shows the training reward of them. We also compared the training curves of Weighted QMIX before and after using the SEQ module, as shown in Fig.14(b). The proposed method significantly reduces overestimation of Q values in QMIX, thereby improving model performance and enhancing model stability without catastrophic performance degradation.

VII. CONCLUSIONS

In this paper, a modified MARL algorithm is applied to address the distributed twin-vehicles cooperative driving problem. On the one hand, by introducing the fairness measurement, the proposed method can balance the self-interested and mutual benefit, which enables each agent to learn cooperative driving strategy automatically without unfair behaviors. On the other hand, using $softmax$ operation instead of max operation shows the brilliant ability to avoid overestimating when calculate the Q values. Experimental results in diversiform stochastic dynamic highway environments demonstrate that the proposed method significantly improve the cooperativeness while ensuring the driving efficiency. Further researches will focus on the abstract efficient representation for the relationships of the participate vehicles which may be beneficial to accelerate the searching and training process of RL models.

REFERENCES

[1] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," *arXiv preprint arXiv:1807.08048*, 2018.

[2] C. Xi, T. Shi, Y. Wu, and L. Sun, "Efficient motion planning for automated lane change based on imitation learning and mixed-integer optimization," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[3] S. Wang, D. Jia, and X. Weng, "Deep reinforcement learning for autonomous driving," *arXiv preprint arXiv:1811.11329*, 2018.

[4] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2070–2075.

[5] H. Wang, Z. Chen, Y. Cai, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "Voxel-rcnn-complex: An effective 3-d point cloud object detector for complex traffic conditions," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[6] D. Tian, C. Lin, J. Zhou, X. Duan, Y. Cao, D. Zhao, and D. Cao, "Sa-yolov3: An efficient and accurate object detector using self-attention mechanism for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[8] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[9] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019.

[10] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[11] G. Wang, J. Hu, Z. Li, and L. Li, "Harmonious lane changing via deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[12] S. Chen, M. Wang, W. Song, Y. Yang, and M. Fu, "Multi-agent reinforcement learning-based twin-vehicle fair cooperative driving in dynamic highway scenarios," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 730–736.

[13] A. Baranzadeh and A. V. Savkin, "A distributed control algorithm for area search by a multi-robot team," *Robotica*, vol. 35, no. 6, pp. 1452–1472, 2017.

[14] Y. Sirineni, P. Verma, and K. Karlapalem, "Traffic management strategies for multi-robotic rigid payload transport systems," in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019, pp. 225–227.

[15] 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.

[16] Z. Peng, G. Wen, A. Rahmani, and Y. Yu, "Leader–follower formation control of nonholonomic mobile robots based on a bioinspired neurodynamic based approach," *Robotics and autonomous systems*, vol. 61, no. 9, pp. 988–996, 2013.

[17] C. B. Low, "A flexible virtual structure formation keeping control design for nonholonomic mobile robots with low-level control systems, with experiments," in *2014 IEEE international symposium on intelligent control (ISIC)*. IEEE, 2014, pp. 1576–1582.

[18] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE transactions on robotics and automation*, vol. 14, no. 6, pp. 926–939, 1998.

[19] M. Knopp, C. Aykin, J. Feldmaier, and S. Hao, "Formation control using gq(λ) reinforcement learning," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017.

[20] A. Rawat and K. Karlapalem, "Multi-robot formation control using reinforcement learning," 2020.

[21] C. Da, S. Jian, and S. Wu, "Uavs formation flight control based on behavior and virtual structure," 2012.

[22] M. Turpin, N. Michael, and V. Kumar, "Decentralized formation control with variable shapes for aerial robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 23–30, 2012.

[23] T.-H. Cheng, Z. Kan, J. A. Rosenfeld, and W. E. Dixon, "Decentralized formation control with connectivity maintenance and collision avoidance under limited and intermittent sensing," in *2014 American control conference*. IEEE, 2014, pp. 3201–3206.

[24] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.

[25] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4295–4304.

[26] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018, p. 2085–2087.

[27] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5887–5896.

[28] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, "Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 33, pp. 10 199–10 210, 2020.
[29] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "Qplex: Duplex dueling multi-agent q-learning," *arXiv preprint arXiv:2008.01062*, 2020.
[30] J. Hu, H. Wu, S. A. Harding, S. Jiang, and S.-w. Liao, "Riit: Rethinking the importance of implementation tricks in multi-agent reinforcement learning." *arXiv preprint arXiv:2102.03479*, 2021.
[31] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[32] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.
[33] E. Hughes, J. Z. Leibo, M. Phillips, K. Tuyls, E. Dueñez-Guzman, A. García Castañeda, I. Dunning, T. Zhu, K. McKee, R. Koster *et al.*, "Inequity aversion improves cooperation in intertemporal social dilemmas," *Advances in neural information processing systems*, vol. 31, 2018.
[34] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
[36] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.
[37] M. S. Andersen, J. Dahl, L. Vandenberghe *et al.*, "Cvxopt: A python package for convex optimization," *Available at cvxopt. org*, vol. 54, 2013.
[38] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993.
[39] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
[40] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

Siyuan Chen received the B.S. degree in automation from Beijing Forestry University, China, in 2018. She is currently pursuing the Ph.D. degree at School of Automation, Beijing Institute of Technology, Beijing, China.

Her research interests include the decision and motion planning for autonomous driving, especially in reinforcement learning methods.

Meiling Wang received the B.S. degree in automation from the Beijing Institute of Technology, China, in 1992, and the M.S. and Ph.D. degrees from Beijing Institute of Technology, China, in 1995 and 2007, respectively. She has been teaching in Beijing Institute of Technology since 1995, and worked in University of California San Diego as a visiting scholar in 2004.

She was elected as the Yangtze River scholar Distinguished Professor in 2014. And she is currently the Director of Integrated navigation and intelligent navigation laboratory, Beijing Institute of Technology, China. Her research interests include advanced technology of sensing and detecting and vehicle intelligent navigation.

Wenjie Song received the B.S. degree and the Ph.D. degree from Beijing Institute of Technology, China, in 2013 and 2019, respectively. He studied in Princeton University as a visiting scholar from 2016 to 2017. He is currently a Professor with the School of Automation, Beijing Institute of Technology, Beijing, China. He was selected as the national high-level young talent in 2022 and his research interests include unmanned system autonomous navigation, bionic robot design and control. He has published over 20 papers in the past five years and he was awarded the first Prize of National Science and Technology Progress Award, outstanding doctoral dissertation of China Inertia Technology Society and so on. And he has taken part in 'China Intelligent Vehicle Future Challenge' and other unmanned systems competition for several times as the core member, achieving excellent ranking.

Yi Yang received the Ph.D. degree in automation from Beijing Institute of Technology, Beijing, China, in 2010. He is currently a Professor with the School of Automation, Beijing Institute of Technology, Beijing, China. His research interests include autonomous vehicles, bioinspired robots, intelligent navigation, semantic mapping, scene understanding, motion planning and control, and robot design and development. He is author/co-author of more than 50 conference and journal papers in the area of unmanned ground vehicle.

Mengyin Fu received the B.S. degree from Liaoning University, China, M.S. degree from Beijing Institute of Technology, China, and Ph.D. degree from Chinese Academy of Sciences.

He was elected as a member of the Chinese Academy of Engineering in 2021, the Yangtze River scholar Distinguished Professor in 2009, won the Guanghua Engineering Science and Technology Award for Youth Award in 2010. He has gotten National Science and Technology Progress Award for several times in recent years.

He is the president of Nanjing University of Science and Technology and his research interest covers integrated navigation, intelligent navigation, image processing, learning and recognition as well as their applications.