Research Article

Dynamic Coordination-Based Reinforcement Learning for Driving Policy

Huaiwei Si, Guozhen Tan, Yanfei Peng, and Jianping Li

Dalian University of Technology, No. 2 Linggong Road, Ganjingzi District, Dalian City, Liaoning Province, China

Correspondence should be addressed to Huaiwei Si; sihuaiwei@mail.dlut.edu.cn

Received 16 February 2022; Accepted 28 March 2022; Published 16 April 2022

Academic Editor: Kuruva Lakshman

Copyright © 2022 Huaiwei Si et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of communication technology and artificial intelligence technology, intelligent vehicle has become a very important part of Internet of Things technology. At present, the single-vehicle intelligence is gradually improved, and more and more unmanned vehicles appear on the road. In the future, these individual intelligence applications need to be transformed into collective intelligence to give full play to the greater advantages of unmanned driving. For example, individual intelligence is self-interest. If there is no collective cooperation, it may affect the whole traffic flow for its own speed. Although the vehicle ad hoc network technology provides a guarantee for the communication between vehicles and makes cooperation between vehicles possible, there are still challenges in how to adapt to coordination learning. Coordination reinforcement learning is one of the most promising methods to solve the multiagent coordination optimization problems. However, existing coordinative learning approaches that usually rely on static topologies cannot be easily adopted to solve the vehicle coordination problems in the dynamic environment. We propose a dynamic coordination reinforcement learning to help vehicles make their driving decisions. First, we apply driving safety field theory to construct the dynamic coordination graph (DCG), representing the dynamic coordination behaviors among vehicles. Second, we design reinforcement learning techniques on our DCG model to implement the joint optimal action reasoning for the multivehicle system and eventually derive the optimal driving policy for each vehicle. Finally, compared with other multiagent learning methods, our method has a significant improvement in security and speed, which is about 1% higher than other multiagent learning methods, but its training speed is also significantly improved about 8%.

1. Introduction

Vehicle ad hoc network (VANET) [1, 2] is an emerging paradigm that enables communication between vehicles in an ad hoc manner [3, 4]. The communication between vehicles makes it possible to exchange more information between vehicles. These technologies have led to the rapid development of automatic driving technology in recent years; it attracts widespread interest from technology companies and research institutions [5–9], such as the Google self-driving car (WAYMO) [5], DARPA [7], the Baidu Apollo Project [8], the Urban Challenge [9]. Especially with the rapid development of artificial intelligence (AI) and vehicle ad hoc network technology, an increasing number of researchers attempt to use various AI algorithms to solve the problem of autonomous driving, using the Internet of Vehicles and other vehicle ad hoc network technologies to obtain information about vehicle and road, and then use the AI algorithms to analyze information and generate solutions to problems in autonomous driving.

In automatic driving, driving policy generates specific driving actions; it plays an important role in driving safety and efficiency. A reasonable driving policy can improve local and even global traffic efficiency and fully reflect the advantages of automatic driving, with the information perceived by the vehicle network [10]. Traditional driving policies are generated on the basis of expert rules [11]. However, these methods are simple and cannot be adapted to different traffic environments [12]; for example, two vehicles may be driving in the same direction. The vehicle running ahead affects the speed of the vehicle behind. According to the expert rules, the vehicle in front cannot take any driving policy, the
vehicle behind passively waits for the vehicle in front to make an acceleration policy, and then, the vehicle behind reaches a higher speed. Therefore, designing a set of driving policy generation algorithm can prove the corresponding coordination driving policy according to the information about the vehicle and its surroundings research topic is very meaningful.

From a theoretical point of view, a vehicle is usually considered an agent and we can use intelligent reinforcement learning (RL) to solve the problem of driving policy. RL is a policy learning method, which uses interactive information between the agent and the environment to achieve maximum return or a specific goal. It adjusts in accordance with the characteristics of real-time feedback, especially suitable for solving the problem of autonomous driving policy. As early as 1990, Hulse et al. used RL to control wheeled robots [12]. In recent years, autonomous driving policy learning methods based on RL are cropping up [10, 11, 13–15]. According to [9, 16], the tasks of an intelligent vehicle can be divided into three categories, namely, perception, planning, and control. Furthermore, according to [17, 18], the planning can be divided into three main levels of route planning, behavior planning, and motion planning. Behavior planning describes the tactical behavior of the vehicle at the maneuver layer. The tactical decisions in such scenarios include lane keeping or lane changing to the left or to the right. These works take the intelligent vehicle as an agent, simply abstract the environmental change into Markov decision-making, and do not consider the interaction between intelligent vehicles.

Although some studies use RL methods and VANET to train autonomous driving policy [10–13, 15, 17, 18], most methods simply extend the single-agent learning method to a multivehicle environment or use simple static topology combined with RL as multivehicle coordination reinforcement learning (CRL) method. However, due to vehicle mobility, the multi-intelligent vehicle system is a dynamic traffic environment, and the interaction among intelligent vehicles is also dynamic [19]. With VANET technology becoming more and more advanced, an extension of simple RL or static topology of CRL is applied to a multivehicle environment, which cannot capture the dynamic characteristic of vehicles, thereby continuously changing the topology and leading to inefficient, even uncoordinated driving decisions. Introducing the representation of dynamic relation into CRL to generate coordination multivehicle automatic driving policy is the focus and solution of this study. Although these works consider the interaction between intelligent vehicles and use the interactive information between vehicles to make single vehicle decisions, they do not consider the cooperation between multiple vehicles and do not give full play to the advantages of intelligent vehicles.

At present, more and more self-driving intelligent vehicles appear in our life, and these intelligent vehicles only consider the most selfish behavior. In the future, if we need to give full play to the advantages of intelligent vehicles, we must make them cooperate like humans. On the basis of these studies and motivations, to adapt to the dynamic relationship between vehicles and give full play to the advanced nature of VANET technology, we propose a coordination reinforcement learning based on a dynamic coordination graph.

Our RL method can be used to learn the multivehicle coordination driving policy, which adapts to the dynamic relationship between vehicles. To realize the representation of the dynamic relationship among vehicles, we use the driving safety field (DFS) method to construct a dynamic coordination graph in real time to realize the processing of dynamic relationships between vehicles. More specifically, we initially apply driving safety field theory to construct the dynamic coordination graph (DCG), representing the dynamic coordination behaviors among vehicles. We then design reinforcement learning techniques on our DCG model to implement the joint optimal action reasoning for the multivehicle system (coordination graph-based formalization allows reasoning about the joint action based on the structure of interactions,) and eventually derive the optimal driving policy for each vehicle. Moreover, we use pretraining techniques and parameter sharing mechanisms to accelerate the learning process. To verify the effectiveness of this method, the highway simulation environment is used to verify the five scenarios with vehicles, eight vehicles, and 11 vehicles. The results show that the autonomous driving policy trained by our method can effectively improve the safety of multivehicle coordination driving and the driving speed, and it can be expanded.

Our main contributions are as follows:

1. Using safety force field as the basis for the construction of explicit coordination relationship, the explicit coordination graph model between agents is dynamic and automatic (irregular)

2. Taking the explicit graph model as the cooperative guidance of agents, the global utility of agents is explicitly decomposed into agents and the combination of local utility between agents, and the belief propagation algorithm is used to solve the overall maximum utility to guide agent learning

3. We regard the local utility between agents as a kind of prior knowledge and use the methods of pretraining and knowledge sharing to transfer between agents to complete the sharing of prior knowledge

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the intelligent vehicle MDP model, RL, CG, and DCG models, which are the bases of our method. Section 4 describes our coordination reinforcement learning algorithm based on DCG. Section 5 shows the verification of our method in the multi-intelligent vehicle environment and the comparison with other methods. Finally, Section 6 summarizes this study and the possible future research fields.

2. Related Work

Driving policy is an important research field in autonomous driving planning; it ensures safe and efficient driving of the
automatic driving vehicle combined with VANET. Among many research methods, deep reinforcement learning (DRL) is widely used to adjust the characteristics of strategy actions according to real-time feedback. This type of method abstracts the intelligent vehicle as an agent and perceives the surrounding environment with IoT technology, which learns the autonomous driving policy by interacting with the environment; it also abstracts the interaction process as Markov Decision Process (MDP). RL has been widely used in the field of driving policies. Loiacono et al. used tabular RL to train autonomous driving policies in an autonomous driving simulation environment [20]. Guo and Wu applied approximate functions combined with a policy gradient to achieve good results in the racing game environment [21]. The combination of deep learning and reinforcement learning has greatly promoted the application of RL in more complex driving environments.

For example, Wang et al. [11] used DRL with rule-based constraints to learn driving policy. Other works used DRL to develop driving policy directly from real-world scenarios [22]. Chae et al. used DRL to train an autonomous driving brake system [23]. Belletti et al. studied a multiobjective autopilot merge policy based on DRL [24]. Makantasis et al. applied a DRL architecture based on Q-masking to make highway driving policy decisions [25]. In [26], the author proposed an RL method based on the surrounding environment perceived by the surrounding environment with IoT technology, which is a reference to training autonomous vehicles to learn to abide by traffic rules and drive safely in various scenarios. Ref. [27] proposed a layered DRL framework to assist vehicles in focusing on surrounding vehicles and learn smoother driving policy. The work in [28] applied proximal policy optimization (PPO) to automatic driving control learning and to actual vehicles. Ref. [29–31] also presented significant work in driving policy.

Recently, many studies on tactical behavior planning also use tools from reinforcement learning combined with VANET. Alizadeh et al. developed a novel simulation environment that emulates autonomous lane changing and trains a DRL agent that yields consistent performance in various dynamic and uncertain traffic scenarios [32]. Chen et al. designed a hierarchical DRL algorithm to learn lane change behavior in dense traffic environment [27]. Wang et al. proposed a deep Q-learning for automated lane changing in highway environment [33]. Yuan et al. used various incentive mechanisms to learn different lane changing policies in the highway environment [34]. Wang et al. proposed the former prospective Q-learning highway lane change method, which is based on intensive microscopic simulations [35]. Bey et al. learned tactical behavior planning for intelligent vehicles by predicting other vehicle features [18]. Xing et al. proposed an RL method to learn the tactical behavior planning under uncertainties for the intelligent vehicle in urban scenarios, considering the intentions of the surrounding road users [17].

Although these methods achieved good performance in their respective applications, most of them did not consider simulating the interaction between multiple vehicles. These studies mostly focused on the driving policies of a single vehicle and personal driving style [36], without considering the interactions and coordination between multiple vehicles, ignoring the benefits of vehicle networking. Therefore, the benefits of applying these models directly to a networked intelligent vehicle environment are limited. Some studies [37, 38] have proposed methods to simulate vehicle interaction and formation using graph theory, but these methods focus on formation and communication. Although Yu et al. proposed two methods to build an autonomous vehicle CG, the methods were based only on the relative position between the vehicles or the initialization sequence number and did not consider the security interaction [38]. In accordance with the dynamics among multi-intelligent vehicles, we realize coordinated policies in multiautonomous vehicles through dynamic CG and RL, which can provide safer and more accurate decisions for autonomous driving. The related works are summarized in Table 1.

### 3. Preliminaries

The proposed driving decision processes in a two-lane highway scenario and the learning process of intelligent vehicles are modeled as MDP, which is the basis of RL. With the vehicle network, we can cooperate in the case of multivehicle interconnection. We will make full use of the advantages of vehicle network for multivehicle coordination learning. We introduce the traffic scenario, the single-vehicle RL problem, and MDP model. On this basis, we introduce the DCG model, which describes the relationship between vehicles. These terminologies are the foundations of this study.

#### 3.1. MDP and RL

MDP is the mathematical framework of RL and the theoretical model foundation of solving RL problems [39]. MDP can be expressed as four tuples of $M = <S, A, P, R>$, where $S$ is a set of finite state space (the agent can perceive all states) and $s \in S$, $A$ is a limited action space (the set of all actions that an agent can perform) and $a \in A$, $P$ is an MDP probability transformation function, $P(s, a, s') : S \times A \times S \rightarrow [0, 1]$ (the probability of an agent executing the action $a$ to next state $s' \in S$ in current state $s$), and $R$ is a reward function (the reward value obtained from the environment when the agent performs action $a$ to next state $s'$ under current state $s$, $R : S \times A \rightarrow [0, 1]$). In the process of RL, $\pi$ is the policy of an agent, which is the probability distribution from state $s \in S$ to action $a \in A$, $\pi : S \times A \rightarrow [0, 1]$. The value function $V^\pi(s)$ represents the expectation of accumulating rewards according to policy $\pi$ in state $s$ ($s \in S$). The goal of RL is to learn policy $\pi$ and maximize the value function $V^\pi(s)$ in each state. The definition of value function $V^\pi(s)$ is given by

$$V^\pi(s) = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) | s_0 = s \right],$$

where $E$ represents the expectation of long-term cumulative rewards under policy $\pi$, $s_t$ represents the state of the agent at time $t$, and $\gamma \in [0, 1]$ is the discount factor. RL can find an optimal policy $\pi^*$ for any finite MDP, such as the case in...
all states $V^\pi(s) \geq V^\pi(s)$, If an agent fully knows the reward function and state transition function of the environment, then it can use linear programming or dynamic programming to calculate the optimal policy $\pi^*$. If the environment model (reward function and state transition probability function) is not completely known to the agent, then the RL methods can be used to solve the MDP problem, and an optimal policy can be found. Among them, DQN [40] is one of the most important and widely used DRL methods. DQN is a nonstrategic time difference algorithm, and its update rules are as follows:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \left[ R(s, a) + \gamma \max_{a'} Q_t(s', a') - Q_t(s, a) \right],$$

(2)

where $Q_t(s, a)$ indicates any subscript represents state-action value, $\alpha \in (0, 1)$ is the learning rate, and other variables have the same meaning as above. Equation (2) is called the $Q$ value function and represents the utility value of action $a$ in state $s$. In the DRL method, the depth network is used to approximate the $Q$ value function.

3.2. Single-Intelligent Vehicle MDP and RL. A behavior planning decision, such as a car-following or overtaking decision, of an intelligent vehicle can be modeled as MDP $<S, A, P, R>$. Here, vehicle 0 is used to explain the state of each intelligent vehicle, as illustrated in Figure 1. The state includes all of the factors affecting the decision-making of the intelligent vehicles. The ten-dimensional states entailed in $(l, l_0, l_1, l_2, l_3, l_4)$ $(i = 1, 2, 3, 4)$ are shown in Figure 1.

$l$ is the lane where the vehicle is located. Subsequently, $l = 1$ means that the vehicle is driving on the driving lane, and $l = 2$ means that the vehicle is driving on the overtaking lane.

- $v_0$ : $v_0$ is the speed of the vehicle
- $v_1$ : $v_1$ is the speed of the leading vehicle that is closest to the vehicle on the driving lane
- $v_2$ : $v_2$ is the speed of the lagging vehicle that is closest to the vehicle on the driving lane
- $v_3$ : $v_3$ is the speed of the leading vehicle that is closest to the vehicle on the overtaking lane
- $v_4$ : $v_4$ is the speed of the lagging vehicle that is closest to the vehicle on the overtaking lane
- $d_1$ : $d_1$ is the distance between the leading vehicle and the vehicle on the driving lane that is closest to the vehicle
- $d_2$ : $d_2$ is the distance between the lagging vehicle and the vehicle on the driving lane that is closest to the vehicle
- $d_3$ : $d_3$ is the distance between the leading vehicle and the vehicle on the overtaking lane that is closest to the vehicle
- $d_4$ : $d_4$ is the distance between the lagging vehicle and the vehicle on the overtaking lane that is closest to the vehicle

The abovementioned state definition regards the position and speed information of neighboring vehicles as the two dimensions of the state. In fact, the position and speed of the neighboring vehicle can be combined with the remaining reaction time (RRT) to reduce the redundancy of information, thereby further reducing the dimension of the state. As such, the dimension of the autonomous vehicle’s state is reduced from ten to five dimensions. After the dimensionality reduction process, the state of the vehicle can be expressed as $(l, t, t_0, t_1, t_2, t_3, t_4)$. $d_{s}^m$ means the shortest safe distance between two vehicles when the current vehicle is driving at an acceleration of $-6 \text{m/s}^2$, and the following vehicle is driving at an acceleration of $-4 \text{m/s}^2$. For example, the minimum safe distance between a vehicle and the leading vehicle on the driving lane is $d_{s}^m = v_0^2/(2 \cdot a_0) - v_2^2/(2 \cdot a_2) + 10$, where $a_0 = -6 \text{m/s}^2$, $a_1 = -4 \text{m/s}^2$, and 10 is the predetermined distance that considers the reaction time of the driver of the leading vehicle and the length of the vehicle (the derivation of the shortest safe distance is in the appendix).

- $t_1 : t_1 = (d_1 - d_{s}^m)/v_0$ indicates the remaining reaction time between the vehicle and the leading vehicle on the driving lane
- $t_2 : t_2 = (d_2 - d_{s}^m)/v_2$ indicates the remaining reaction time between the vehicle and the lagging vehicle on the driving lane
- $t_3 : t_3 = (d_3 - d_{s}^m)/v_0$ indicates the remaining reaction time between the vehicle and the leading vehicle on the overtaking lane

<table>
<thead>
<tr>
<th>Method</th>
<th>Adaptation property</th>
<th>Some relevant work</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL</td>
<td>Complex end-to-end environment</td>
<td>[21–30], [31]</td>
</tr>
<tr>
<td>Deep reinforcement learning combined with</td>
<td>Single-vehicle policies considering other intelligent vehicle behaviors</td>
<td>[17, 18, 27], [32–35]</td>
</tr>
<tr>
<td>VANET</td>
<td>Multivehicle coordination decision</td>
<td>[37, 38]</td>
</tr>
</tbody>
</table>

Table 1: The summary of related works.
As this study focuses on high-level driving decisions during vehicle driving, the action set mainly includes the following two actions:

1. Driving on the driving lane: the vehicle follows the leading vehicle. For the planned speed $v_p$ on the driving lane, the speed limit on this driving lane is $v_d = 35 \text{ m/s}$.

2. Driving on the overtaking lane: the vehicle follows the leading vehicle. For the planned speed $v_p$ on the overtaking lane, the speed limit on this overtaking lane is $v_o = 40 \text{ m/s}$.

The driving planned speed $v_p$ means that the vehicle follows the leading vehicle at the planned speed $v_p$. When the planned speed $v_p$ is greater than the speed limit ($v_d$ or $v_o$) of the lane, the vehicle follows the leading vehicle at a certain speed ($v_d$ or $v_o$) where it is driving on. The planned speed of a vehicle can be calculated using various car-following models [41, 42] based on the relative position and relative speed of the vehicle in front. We use the heuristic method to implement the car-following model and calculate the planned speed $v_p$ as follows the paper [41].

The last important element of MDP is the reward function, which is an index used to evaluate the learning performance of an agent. In autonomous driving, safety is the ultimate goal and the most important criterion for evaluating the decision-making of an agent. By considering the safety of driving, the definition of the reward function is given by

$$
\begin{align*}
    r_{\text{safe}} &= \begin{cases} 
        -40 & \text{if collision} \\
        \min(t_1, t_2) & \text{if } l = 1 \text{ and } d_1 > 3 \text{ and } d_2 > 3 \\
        \min(t_3, t_4) & \text{if } l = 2 \text{ and } d_1 > 3 \text{ and } d_4 > 3 \\
        -5 & \text{else}
    \end{cases}, \\
    r_{\text{speed}} &= \begin{cases} 
        v_p - v_t & \text{if } v_p > v_t, \\
        0 & \text{else}
    \end{cases}
\end{align*}
$$

where $v_t$ is the task speed of the last decision. When the vehicle runs on the driving lane and the distance between the vehicles (leading vehicle and lagging vehicle) is not too close ($d_1, d_2 > 3 \text{ m}$), the reward value is the minimum of the remaining reaction time on the driving lane ($\min(t_1, t_2)$). Thus, the greater are the values of $t_1$ and $t_2$, the greater is the safety reward of the vehicle on the driving lane. When the vehicle runs on the overtaking lane, and the distance between the vehicles (leading vehicle and lagging vehicle) is not too close ($d_3, d_4 > 3 \text{ m}$), the reward value is the minimum of the remaining reaction time on the overtaking lane. Given ($\min(t_3, t_4)$), the larger are the values of $t_3$ and $t_4$, the greater is the reward to be obtained by the vehicle on the overtaking lane. If the distance between the vehicles and any vehicle in the same lane is less than 3, then the vehicle will be apprehended; that is, the reward value is $-5$.

3.3. Multi-Intelligent Vehicle Coordination Graph. When multiple agents coexist in the same environment and learning scenario, the problem is a MARL problem. There are instability problems caused by multiple agents learning in the same environment. How to make the agent learn a coordinative policy in a nonstationary environment is extremely challenging. CG can represent relationships among agents and combine with RL, which is used to guide coordination learning. In this section, we introduce CG.

CG [43] is an effective method to represent the relationship between agents. In CG, the relations can be described by an undirected graph $G = (V, E)$. $V$ is the node of the graph, and each node represents an agent $i \in V$, $E$ is the edge of the node, and the edge $(i, j) \in E$ represents that the corresponding agent must make coordinative decisions. Therefore, a new local value function must be defined between each pair of agents to ensure that the agents can make coordinative decisions. The global value function $Q(S, A)$ can be decomposed into a value function on each agent $q_i$ and a linear combination of the coordinate value function $Q_{ij}$ between agents with a coordinative relationship:

$$
Q(S, A) = \sum_{i \in V} q_i(s_i, a_i) + \sum_{(i, j) \in E} Q_{ij}(s_{ij}, a_i, a_j),
$$

where $S = \{s_1, \ldots, s_n\}$ and $A = \{a_1, \ldots, a_n\}$ are states and actions of all agents and $s_{ij}$ is the joint states of agents $i$ and $j$. The main goal of CG is to find a coordination action.
policy that maximizes \( Q(S, A) \) in the joint states \( s_{ij} \). In CG, we used variable elimination (VE) to find the optimal joint action. VE is a message passing algorithm used to solve coordination problems. In the VE algorithm, an agent (node) must collect all local payoff functions directly related to it (edges) and then calculate its maximum joint utility and correspondingly optimal action. The agent sends the calculated utility function to the neighbors’ agent and then is eliminated from the CG. The neighboring agent continues to calculate the maximum joint utility based on the received information and then is eliminated from the graph. When there is only one agent in CG, the agent can select the optimal action through the collected utility function. Finally, according to the reverse order of agent node elimination, each agent can calculate the joint action according to its maximum joint utility [44].

The above coordination graph represents the relationship among vehicles in a predefined way. The relationship between vehicles is dynamic, which needs to build a coordination graph dynamically according to the dynamic interaction among vehicles. As shown in Figure 2, the CG (a) becomes CG (b), which is according to the interaction among agents. The interaction structure of the agent and the object is dynamic instead of fixed.

4. Method

With advanced Internet of Vehicles technology, it is possible to communicate and interconnect our vehicles. This is the basic premise for the entire multismart car system to realize collaborative learning. The traditional CG is a static topological state graph, which is a fixed structure in a predefined way. Although our previous work has proposed a variety of self-driving vehicle modeling graph structure methods [38], effectively realizing the coordination decision-making between vehicles by calculating the coordination utility between vehicle pairs, however, the methods not only need to train a neural network to calculate the coordination utility between each pair of agents but must also satisfy the integrity of the five basic vehicle coordination structures, which seriously limits the learning efficiency and generalization ability of the methods. Therefore, our objective has been to search for a more effective method that can resolve the abovementioned issue. With respect to the previous work, the contributions of our present in-depth study are as follows: (1) the driving safety field (DSF) is used as a basis to construct the explicit coordination relationship and dynamically and automatically (irregularly) build the explicit coordination graph (CG) model between agents. (2) The explicit graph model is used as the coordinative guidance of agents. The global utility of agents is explicitly decomposed into the combination of agents and the local utility between agents, and the belief propagation algorithm is used to solve the global maximum utility to guide agent learning. (3) We regard the local utility between agents as a kind of prior knowledge, and use pretraining and knowledge sharing methods to transfer between vehicles to complete the sharing of prior knowledge. The relationship between the methods is shown in Figure 3.

4.1. Generating Dynamic Coordination Graph for Multiple Vehicles. In a multiagent cooperation environment, the decision-making of a single agent must depend on the behavior of other agents. Especially, in the rapidly changing environment, the relationship between agents also changes with the change of environment. We apply DSF theory to calculate the field among vehicles as the relationship between intelligent vehicles and then build the dynamic coordination graph.

DSF modeling [45] is a type of “physical field” that characterizes the impact degree of each factor on the driving risk in the driving environment. Figure 4 is a plot of the field strength distribution of the corresponding DSF in the lane changing scenario on a two-lane highway. The field is constructed by two components, namely, the motion field formed by moving vehicles and the behavior field of vehicles. The field strength distribution of the DSF can be used to judge the degree of interaction between vehicles.

Figure 4 shows the use of DSF to build a coordination relationship between vehicles, which is the general vehicle driving scenario on a highway. Take vehicle 1 (No. 1) and 2 (No. 2) in Figure 4 as an example; the speed of No. 1 and No. 2 is \( v_1 \) and \( v_2 \), respectively, and the distance between No. 1 and No. 2 is \( d \). According to the equation of the DSF model,

\[
F_i = \frac{G R_1 R_2 M_1 M_2}{d^4} (1 + D_{11})(1 + D_{12}) \exp \left[-k_2(v_1 - v_2)\right],
\]

where \( F_i \) is the field force on No. 1. Because the main research focus here is the problem of coordinative lane change decision-making, for the convenience of calculation, we assume that all intelligent vehicles have the same physical quantity. Here, we choose the following set of parameters according to the paper [46]. We set the DSF of parameters as follows: \( G = 0.001 \), which is the adjustment factor that represents the field gravitational constant. \( R_1 = R_2 = 1 \) represents the road condition factors of vehicle 1 and vehicle 2. \( M_1 = M_2 = 5000 \text{ kg} \) indicates the equivalent masses of
judge whether there is interaction between the two intelligent vehicles according to the force field size. If an interaction exists, then the two intelligent vehicles as vertex and relation as edge add in the coordination graph $G$ and regarded as edges of $G$.

4.2. Pretraining of Local Coordinative Value Function. Although the multiagent environment is very complex, we find that the interaction between local agents is similar, so we can get this local interaction model through pretraining. Our DCG is constructed based on DSF to analyze the interaction between vehicles. Although the interaction between vehicles can be obtained according to the force field between them, which is represented by the edge of the coordination graph, so we can use a utility function to generally represent the payoff between vehicles.

In the CG, we use the edge to represent the coordinative relationship between vehicles; it is usually represented by rules or defined functions in traditional methods. Local relations with the same state can be represented by the same rule or function. RL is used to learn and reuse the local coordinative utility function in high-dimensional state space. The global utility $Q_{\text{global}}$ in coordination graph can be composed of local utility function $Q_{ij}$ on each edge $(i, j \in E)$. Thus, we train $Q_{ij}$, which is used to represent locally similar utility functions, without traditional custom rules or functions. $Q_{ij}$ is a local coordinative value function (LCVF), which is input into traffic flow for pretraining, and it can be used as a local general utility function on all edges of the complete DCG. This approach effectively reduces the exponential complexity of collaborative computation in the traditional graph model.

The local coordinative value function trains a couple of vehicles in the multivehicle environment to learn coordinated decision-making. In the two-lane environment, autonomous vehicles are controlled by independent neural networks. We use DSF to calculate the relationship between intelligent vehicles according to the DSF and determine the intelligent vehicles that need coordination to form a coordination graph. Each local utility function in the CG is represented by LCVF. We use a deep network as the LCVF model. Each output represents the $Q$ value of each action combination of the two vehicles. The vehicles with a low driving risk
are controlled by the individual neural network, the neural network parameter is $\omega_i$, and the update rule is as follows:

$$\text{MSE}(\omega_i) = \left[ r_i + \gamma \max q_i(s'_{ij}, a'_{ij}; \omega_i) - Q_i(s_{ij}, a_{ij}; \omega_i) \right],$$

(6)

where $s_i$ is the state of vehicle $i$, $a_i$ is the action of vehicle $i$, and $r_i$ is the immediate reward when vehicle $i$ takes $a_i$. The vehicles $(i, j)$ with large field forces and greater driving risk must be controlled by the LCVF, which is also updating as deep Q-learning:

$$\text{MSE}(\omega_j) = \left[ r_{ij} + \gamma \max Q_{ij}(s'_{ij}, a'_{ij}; \omega_j) - Q_{ij}(s_{ij}, a_{ij}; \omega_j) \right],$$

(7)

where $r_{ij}$ is the joint reward of two vehicles $(i, j)$ under the joint states $s_{ij}$ for taking joint actions $a_{ij}$ and transferring to the next state $s'_{ij}$. We obtain the local utility function by pretraining in multivehicle environment, and the final pre-trained network is a deep network, which has the ability to autonomously generalize the benefits of different states. We use LCVF to measure the interaction in the coordination graph. Then, we verify that the LCVF model can effectively represent the local cooperative utility.

4.3. Coordination Reinforcement Learning. A problem in which multiple agents coexist in the same environment and learn at the same time is a MARL problem. MARL has been widely used in various virtual and real fields. We apply DCG to solve the MARL problems. The main goal of CG is to build the relation between agents which indirectly achieve the purpose of decomposition of the multiagent system. The VE algorithm can be used to maximize the global payoff function value and find the optimal action.

In our method, the local utility represents the local benefits of the agent, and the global utility is composed of these local utilities. Local utility measures the benefit (utility) brought by the decision-making action of the agent. Therefore, we can adjust the local utility by adjusting the agent decision-making action. And the actions between different agents will produce different local utilities. We use the collaboration graph to construct the relationship representation between agents and use the belief propagation algorithm to adjust the decision-making actions of different agents according to the mutual collaboration relationship, so that the decisions made by agents meet the maximum global utility rather than the maximum local utility.

When using a MARL technology, each vehicle is designed to achieve the same goal, that is, fast driving by the group while ensuring driving safety. Therefore, in this paper, we assume that all vehicles have the same settings and willingness to achieve coordination among vehicles. The purpose of our coordination reinforcement learning is to calculate a policy $\pi$ to maximize the global value function, as follows:

$$Q^\pi(js, ja) = E_{\pi} \left[ \sum_{t=0}^{\infty} y^t R(js_t, ja_t) | js_0 = js \right],$$

(8)

where $js$ and $ja$ are the joint states and actions of all vehicles, respectively, and $R(js, ja)$ is the reward function. When using DCG combined with MARL, the global payoff function can be decomposed into the following:

$$Q^\pi(s, a) = \sum_{i \in V} q_i(s_i, a_i) + \sum_{(i, j) \in E} Q_{ij}(s_{ij}, a_{ij}, a_j).$$

(9)

Equation (9) is a representation of the decomposition of global functions according to the constructed coordination graph. $V$ indicates that the nodes in the coordination graph correspond to the intelligent vehicle, and $E$ represents the edges in the coordination graph, corresponding to the interaction between intelligent vehicles. Each vehicle has its own action value function $q_i(s_i, a_i)$ and has joint utility $Q_{ij}(s_{ij}, a_{ij}, a_j)$ with vehicles with interaction relationship.

Although many studies on the combination of coordination graph and RL have been conducted to train agents [47], applying coordination graph directly to RL, especially in multi-intelligent vehicle, which has fast velocity, is infeasible. The proposed DGC solves the dynamic coordinative relationship in multiagent system. On the basis of CG, we train each intelligent vehicle by using the global maximum joint utility and action. The maximization of joint utility is shown in equation (7), in which each intelligent vehicle maintains the individual action value function and the joint utility function. We maximize it to obtain the maximum utility and corresponding actions.

$$\arg \max_{ja} Q_{\text{global}} = \arg \max_{ja} \left[ \sum_{i \in V} q_i(s_i, a_i) + \sum_{(i, j) \in E} Q_{ij}(s_{ij}, a_{ij}, a_j) \right].$$

(10)

We use the VE algorithm to solve equation (10), which is a message passing algorithm between agents. We use the global maximum utility obtained by the VE algorithm to guide the learning of each intelligent vehicle. The difference with individual Q-learning update is that each intelligent vehicle learning is no longer only considering individual utility but also updating its learned action value function according to the global utility value. We use the deep
network to represent the action value function of each agent; the learning update equation is as follows:

$$\text{MSE}(w_i) = \left[ r_i + \gamma \max_{ja} q_i'(s', a'_{ja} ; w_j) - q_i(s, a_{ja} ; w_i) \right]^2,$$

(11)

where $r_i$ is the obtained reward of vehicle $i$ when it takes an action, $q_i(s, a_{ja} ; w_i)$ is local value function, and $ja$ comes from (7). The pseudocode of Algorithm 2 provides the detailed calculation process. In each time step, a vehicle with a potentially conflict relationship is built into a DCG. Each vehicle uses the VE algorithm on DCG to select its best action (following or changing lanes) and a predefined exploration strategy to execute this action. After the vehicle moves to a new location, the CG will be rebuilt.

4.4. Sharing Parameter Mechanism. In the multiagent environment, there is often a problem of knowledge sharing between agents. Agents can improve learning efficiency through knowledge interaction. The existing MARL methods often transfer knowledge through the method of parameter sharing between agents. For a long time, parameter sharing among agents has been a main aspect of RL research of distributed multiagents. This approach is helpful when expanded to the environmental tasks of a large action space and state space of modern benchmarks. The most advanced value function decomposition methods, VDN [48] and QMIX [49], also share parameters among the neural networks of approximate value functions. In the traffic environment, vehicles always have the same general rules. Drivers want to achieve faster speed and higher safety through the control of vehicles on the premise of obeying the traffic laws. Autonomous vehicles should also follow this driving principle. Thus, the driving strategies learned in the traffic environment will have certain similarity, which can accelerate the learning process of strategies for autonomous vehicles through parameter sharing.

Therefore, we add an improved multiagent parameter sharing learning method to the proposed coordinative learning method. Our knowledge sharing is that a single intelligent vehicle learns and shares parameters. To a certain bottleneck, all agents use dynamic collaborative graph for collaborative reinforcement learning. In the stage of multivehicle centralized training, all autonomous vehicles share network parameters and jointly reduce the joint gradient to the minimum. After reaching the learning bottleneck, the centralized training neural network is distributed to all autonomous vehicles, and the decentralized learning is continued on each individual. At the same time, in view of the overall convergence of the learned autonomous driving policies, we can directly extend the basic network trained in a few individual environments to a larger-scale multiagent environment to improve the learning efficiency and scalability of our coordinative learning algorithm. All agents use the centralized training method of parameter sharing; thus, only one neural network control is needed for different autonomous vehicles to save some storage space. At the same time, because there is no need for independent learning for each autonomous vehicle, only one centralized sampling is needed for learning, which greatly reduces the number of learning iterations and saves time. Moreover, each autonomous vehicle is in a different state of the environment, and each step of centralized training can collect different directions of driving environment exploration, thereby breaking the correlation of data used in each training and increasing the adaptability and convergence efficiency of the network. We add this process to Algorithm 2 and finally form PS-DCG.

In Algorithm 2, in the first to third lines, we initialize a vehicle set for each episode of RL and the driving parameters for each vehicle. Each training termination condition indicates that a collision occurs between the vehicles or training to maximum step size (400 steps). On the fourth line, the dynamic relationship is constructed according to the DSF. The fifth to sixth lines collect the independent state information of each vehicle. The seventh to eighth lines collect the joint state information of the vehicles, which interact with each other. The ninth to 10th lines use the pretrained model and variable elimination algorithm to calculate the optimal joint action and execute the joint optimal action in the environment. The 11th to 16th lines rebuild the coordination graph according to the relationship between vehicles and collect the transfer states, immediate reward of each vehicle, and joint states, which interact with each other. The lines 17th to 18th lines calculate the optimal joint action according to the state and reward parameters and update each agent according to formula (6); each updated parameter is shared among agents.

5. Experimental Evaluation

Driving policy is an important research field in autonomous driving planning; it ensures safe and efficient driving of the automatic driving vehicle combined with VANET [50, 51]. Our algorithm is applicable to the high-speed intelligent vehicle driving environment, where all vehicles are intelligent vehicles. We use the open-source project highway-env to analyze and test multivehicle autonomous driving. We have learned and tested multivehicle cooperative learning in this environment. The project address is https://github.com/eleurent/highway-env.

The simulation driving environment is a highway segment with a two-lane highway scenario on one direction. The segment length is 1000 m, and each lane width is 3.75 m. A scene of the simulation scenario is shown in Figure 5. We only simulate the highway environment of straight-line driving and simulate the decision-making learning process of multivehicle autonomous driving behavior. The essential maneuvers at this level are lane changing and lane keeping. Traffic on the highway can be customized as required. All vehicles can perform practical car-following behavior with the proposed expert rule [41]. The intelligent vehicle is controlled by an RL-based intelligent agent. The agent learns making behavior decisions, including lane changing and lane keeping. We tested our dynamic
collaborative algorithm in five, eight, and 11 highway environments.

Our lane changing scenarios is that the vehicle changes lane to the driving lane immediately when the vehicle satisfies the changing conditions on the overtaking lane. As shown in Figure 6, the yellow vehicle drove on the overtaking lane. When the vehicles on the driving lane and the yellow vehicle satisfy the condition of the lane changing, the yellow vehicle changes lane immediately. If the lane changing condition of yellow vehicle is not satisfied, then the yellow vehicle continues to drive on the overtaking lane until the condition of lane change is met.

The MDP of multiple autonomous driving is described in the third part. We use different methods to train vehicle to learn coordinative driving. The average reward and other performance indicators of the training process are shown in Figure 7. The horizontal axis represents each episode, and the vertical axis represents the corresponding index value of each episode. In our experiment, we choose two traditional methods to compare with RL, which is the expert rule-based approach (Expert) [41] and the famous mobility model MOBIL [42]. DCG denotes the dynamic coordinated graph learning approach, PS-DCG denotes our proposed parameter sharing dynamic coordination graph learning, and LCVF denotes the local coordinative value function. The specific experimental parameters are defined in Table 2.

During the training process, each episode is terminated until either a collision occurred or 400 time steps were run. In the individual learning approach, each vehicle maintaining an independent Q-value deep network and each vehicle learn the driving policy individually without coordination. Because the individual learning does not consider other vehicles, the resulting performance may not be optimal. In our pretraining method, we train a LCVF to measure the local utility of each edge in DCG.
Figure 7: Indicators under different methods.
nation learning in di
cate that LCVF can solve the local con
independent method (individual learning). The results indi-
DCG) and the identi
position-based dynamic coordination graph algorithm (P-
DCG. We also compare the proposed method with the
utility function, and can measure the utility of interaction in
DCG and I-DCG, thereby con
get more reward, and convergence speed is better than the P-
tion sequence number [17]. More evidently, the PS-DCG can
on the relative position between the vehicles and the initializa-
coordinated lane changing, thereby achieving safe decisions
by the low
ffi
safety, they
ffi
expert rules usually end up in a straight line on the driving
ffi
the lane-changing behavior of the vehicle. Vehicles based on
usually realize a higher-speed driving policy by coordinating
each episode. Compared with the independent learning
average cumulative speed change in each decision of
index after various algorithms converge and the training time
ing that our experimental data is stable.

Table 3 shows the average value of each driving microin-
dex after various algorithms converge and the training time
required for convergence. “Stability” in Table 3 represents
the average cumulative speed change in each decision of
each episode. Compared with the independent learning
method and the expert rules, the coordination method can
usually realize a higher-speed driving policy by coordinating
the lane-changing behavior of the vehicle. Vehicles based on
expert rules usually end up in a straight line on the driving
way and drive at a relatively low speed. The policy learned
by the independent learning method is usually more aggres-
which makes the vehicle distribute in two lanes and
drive at a relatively high speed. However, due to the lack of

<table>
<thead>
<tr>
<th>Vehicles</th>
<th>Method</th>
<th>Reward</th>
<th>Lane change</th>
<th>Speed</th>
<th>Stability</th>
<th>Min. Distance</th>
<th>RRT</th>
<th>Times (hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 vehicles</td>
<td>Individual</td>
<td>169.34 (7.13)</td>
<td>2.73 (0.12)</td>
<td>33.98 (1.39)</td>
<td>0.95 (0.04)</td>
<td>155.09 (7.13)</td>
<td>130.96 (6.02)</td>
<td>2 (0.09)</td>
</tr>
<tr>
<td></td>
<td>DCG</td>
<td>183.76 (8.83)</td>
<td>38.65 (1.55)</td>
<td>33.08 (1.36)</td>
<td>1.28 (0.05)</td>
<td>158.65 (7.30)</td>
<td>154.21 (6.93)</td>
<td>6 (0.20)</td>
</tr>
<tr>
<td></td>
<td>PS-DCG</td>
<td>181.88 (7.46)</td>
<td>21.32 (1.02)</td>
<td>32.70 (1.60)</td>
<td>0.79 (0.03)</td>
<td>157.93 (7.26)</td>
<td>156.27 (6.41)</td>
<td>5.5 (0.23)</td>
</tr>
<tr>
<td>8 vehicles</td>
<td>Individual</td>
<td>237.50 (11.16)</td>
<td>3.45 (0.15)</td>
<td>34.45 (1.37)</td>
<td>1.65 (0.06)</td>
<td>151.70 (6.82)</td>
<td>166.15 (7.31)</td>
<td>2.5 (0.10)</td>
</tr>
<tr>
<td></td>
<td>DCG</td>
<td>288.36 (12.98)</td>
<td>43.17 (1.88)</td>
<td>33.07 (1.36)</td>
<td>1.60 (0.76)</td>
<td>157.44 (0.63)</td>
<td>232.64 (10.46)</td>
<td>7 (0.28)</td>
</tr>
<tr>
<td></td>
<td>PS-DCG</td>
<td>280.68 (11.5)</td>
<td>12.87 (0.59)</td>
<td>31.64 (1.55)</td>
<td>0.97 (0.03)</td>
<td>157.72 (6.15)</td>
<td>258.72 (10.09)</td>
<td>6 (0.23)</td>
</tr>
<tr>
<td>11 vehicles</td>
<td>Individual</td>
<td>276.54 (11.33)</td>
<td>2.21 (0.08)</td>
<td>34.05 (1.49)</td>
<td>2.05 (0.08)</td>
<td>148.00 (7.74)</td>
<td>177.98 (7.29)</td>
<td>3.5 (0.14)</td>
</tr>
<tr>
<td></td>
<td>DCG</td>
<td>369.37 (17.36)</td>
<td>38.41 (1.38)</td>
<td>32.50 (1.3)</td>
<td>1.80 (0.06)</td>
<td>156.60 (7.83)</td>
<td>317.18 (14.65)</td>
<td>11 (0.45)</td>
</tr>
<tr>
<td></td>
<td>PS-DCG</td>
<td>387.55 (17.43)</td>
<td>12.05 (0.54)</td>
<td>31.37 (1.22)</td>
<td>1.14 (0.05)</td>
<td>157.41 (6.20)</td>
<td>349.35 (17.46)</td>
<td>9 (0.30)</td>
</tr>
</tbody>
</table>

Figure 7(a) shows the reward value of five-vehicle coordi-
nation learning in different methods. Figure 7 illustrates that
the learning methods are better than rules. Figure 7 shows that
LCVF can obtain better cumulative reward than a completely
independent method (individual learning). The results indi-
cate that LCVF can solve the local conflict problem, as a local
utility function, and can measure the utility of interaction in
DCG. We also compare the proposed method with the
position-based dynamic coordination graph algorithm (P-
DCG) and the identification-based coordination graph algo-
ithm (I-DCG). These two methods build a CG based only
on the relative position between the vehicles and the initializa-
tion sequence number [17]. More evidently, the PS-DCG can
generate more reward, and convergence speed is better than the P-
DCG and I-DCG, thereby confirming proofs that the pro-
posed method can achieve better coordination among vehi-
cles. The reason is that the coordinative structure
constructed by our method is more reasonable.

Through the analysis of different evaluation indicators in the
experiment, the proposed approach is found to increase
RRT and minimum forward distance (Figures 7(c) and 7(f))
to ensure safety by sacrificing speed (Figure 7(b)). In addition,
the driving policy becomes more stable, as reflected by the low
speed change (Figure 7(e)) and low frequency of lane change
Figure 7(d). Vehicles controlled by expert rule-based approach
are too conservative. Although they ensure traffic safety, they
rarely take an overtaking action. Vehicles that use individual
learning are usually more aggressive and will take the action
of overtaking to speed up the vehicle. However, coordination
was not considered, and the minimum forward distance
between vehicles using individual learning is very small. Our
approach provides a safer and faster environment through
coordinated lane changing, thereby achieving safe decisions
as shown in the final reward. Although P-DCG can obtain
higher speeds, this method leads to frequent lane changing
of unmanned vehicles, which is very unstable.

Further verifying the scalability of the methods, we
applied the individual learning, Expert, MOBIL, DCG, and
PS-DCG approaches to environments with different vehicle
densities. The density of vehicles increased from 5 to 8 and
11, respectively. Figure 8 shows the reward and convergence
rate of each method under the three types of densities.
Figure 8 shows that the effect of individual learning becomes
increasingly gradual, and the performance gap between
MOBIL and the individual learning decreases. The PS-
DCG approach always obtains higher average rewards in dif-
ferent vehicle densities, indicating that it has better scalabil-
ity and can be extended to more complex situations. In
summary, through the analysis of vehicle motion character-
istics and experiments with different vehicle densities, our
approach is proven to effectively solve the latent conflict risk
between vehicles by coordinated lane changing. This finding
further verifies the effectiveness of our approach. Our exper-
imental data is the average of 50 experiments. We indicate
the variance of the data in the brackets of the statistical table.
Our experimental data is the average of 50 experiments. We
indicate the variance of the data in the brackets of the statis-
tical table. The variance of each index is below 0.5%, indicat-
ing that our experimental data is stable.

Table 3 shows the average value of each driving microin-
dex after various algorithms converge and the training time
required for convergence. “Stability” in Table 3 represents
the average cumulative speed change in each decision of
each episode. Compared with the independent learning
method and the expert rules, the coordination method can
usually realize a higher-speed driving policy by coordinating
the lane-changing behavior of the vehicle. Vehicles based on
expert rules usually end up in a straight line on the driving
way and drive at a relatively low speed. The policy learned
by the independent learning method is usually more aggres-
sive, which makes the vehicle distribute in two lanes and
drive at a relatively high speed. However, due to the lack of
coordination mechanism between vehicles, the minimum distance between vehicles is relatively small and the safety is low. The autonomous vehicle trained by the DCG method changes lanes relatively frequently to maintain a relatively high speed. Our PS-DCG method trains the autonomous vehicles to change lanes less, which makes the whole team drive at a relatively high speed, and the training time is less, which proves that our coordination graph construction method and coordination learning method are more reasonable.

In addition, we verified the adaptability of the parameter sharing method in environments with different vehicle densities. We take the policy network trained in a five-vehicle environment as the learning basis neural network (trained network) and extend it to an eight-vehicle environment. The experimental results are shown in Figure 9(a), where DCG and PS-DCG represent the method of learning from scratch, and DCG (retrained) and PS-DCG (retrained) represent the method of continued training from the basic neural network. In the eight-vehicle environment, using only the basic neural network to control the autonomous vehicle achieves a low reward value. The method of continuous training based on the basic neural network has a high starting point, which enables it to quickly surpass the method of training from zero in the initial stage of learning and converge to the state with the highest reward value with the fastest speed. With increasing number of vehicles, the complexity of the environment increases, the method of learning from scratch easily falls into the local optimum, leading to premature convergence to a sub-optimal solution. The method of parameter sharing has some knowledge of automatic driving; it can avoid these problems and finally learn a better coordinative driving policy. The comparison between DCG (untrained) and PS-DCG (trained) that without sharing parameters, the DCG (untrained) method has

Figure 8: Reward and convergence of each method in different vehicle densities.
a learning bottleneck due to the vehicles’ lazy behavior. The PS-DCG (trained) method can overcome this problem and converge to higher reward value.

We also extended the trained network in a five-vehicle environment to different vehicle densities from three vehicles to 11 vehicles. Figure 9(b) shows the average reward value obtained by different methods after 5000 episodes of training. Among them, the reward value of the untrained basis neural network is relatively low, and with increasing number of vehicles, the reward values of the zero start learning method and the parameter-sharing expansion method converge to reveal evident differences. Figure 9(c) shows the number of training episodes required for the zero-based learning method and parameter sharing extension method to achieve the final convergence effect. The figure shows that under different vehicle densities, the parameter sharing extension method can achieve a convergence effect in less training episodes, thereby effectively improving the learning efficiency of the coordination learning algorithm in the autonomous driving environment.

6. Conclusions

Because of the advanced Internet of Vehicles technology, vehicle-to-vehicle communication is possible, making the autonomous vehicles be regarded as a multiagent system. Although the vehicle ad hoc network technology provides a guarantee for the communication between vehicles and makes cooperation between vehicles possible, there are still...
challenges in how to adapt to coordination learning. A challenging problem in this field is that vehicles make their driving decisions via coordinating with each other.

Coordination reinforcement learning is one of the most promising methods to solve the multiagent coordination optimization problems. However, due to vehicles’ mobility, existing coordination learning approaches that usually rely on static topologies cannot be easily adopted to solve the vehicle coordination problems in the dynamic environment. We propose a dynamic coordination reinforcement learning to help vehicles make their driving decisions. First, we apply driving safety field theory to construct the dynamic coordination graph (DCG), representing the dynamic coordination behaviors among vehicles. Second, we design reinforcement learning techniques on our DCG model to implement the joint optimal action reasoning for the multivehicle system and eventually derive the optimal driving policy for each vehicle. Finally, experiments verify that the proposed approach can achieve accuracy and safety of lane driving decision-making, and with increasing number of vehicles, the approach has good scalability. Compared with other intelligent vehicle deep reinforcement learning methods, this paper not only considers the behavior of other vehicles but also learns the cooperative behavior of multiple vehicles, rather than only selfish behavior. At the same time, driving field knowledge is used to more reasonably express the relationship between intelligent vehicles, and the relationship between intelligent vehicles is abstracted as a coordination graph. Combined with reinforcement learning, intelligent vehicles learn a multivehicle coordination policy. In the learning process, use some driving experience to accelerate learning.

Our main contributions are as follows. Using safety force field as the basis for the construction of explicit coordination relationship, the explicit coordination graph model between agents is dynamic and automatic (irregular). Taking the explicit graph model as the cooperative guidance of agents, the global utility of agents is explicitly decomposed into agents and the combination of local utility between agents, and the belief propagation algorithm is used to solve the overall maximum utility to guide agent learning. We regard the local utility between agents as a kind of prior knowledge and use the methods of pretraining and knowledge sharing to transfer between agents to complete the sharing of prior knowledge.

Our research is actually carried out in a dynamic environment, open environment, and a multivehicle environment. A scenario is provided for future intelligent vehicle research because with the development of autonomous driving technology, considering the coordinative driving of multiple intelligent vehicle is necessary in the future. We demonstrate the effectiveness and scalability of our method in simulating high-altitude highway environments. All intelligent vehicles are homogeneous agents. In fact, every vehicle has a different structure and function. Transforming these heterogeneous intelligent vehicles into vehicles is coordinately a new challenge.

In the future, we will further study the multivehicle cooperation mode under different scenarios based on this paper. In this paper, multivehicle coordination learning is carried out in the fully cooperative highway scene, and the reality is facing the challenges of man-machine mixing and imperfect driving information. We will conduct in-depth research from these aspects. In the multivehicle cooperative driving of man-machine hybrid, we should consider the behavior of human driving in automatic driving. How to avoid the dilemma of zero sum game with human driving is a major challenge we face. For imperfect information, how to use limited state information to reach an agreement with surrounding vehicles in the process of collaborative learning is another major challenge.

### Appendix

Assuming that the vehicle runs as shown in Figure 10, we use Figure 11 as the calculation explanation of the shortest safe distance. The shortest safety distance is the distance difference between the two vehicles when they decelerate from the current velocity to 0 at the same time. In more detail, before the two vehicles start to decelerate, if the distance between two vehicles is greater than the shortest safe distance, there will be no collision; if it is less than the shortest safe distance, there will be a collision. In the latest manuscript, we use $d_{ij}$ to replace $d_{ij}$ and $d_{si}$ to replace $d_{si}$.
We use \( s_i \) (the distance of intelligent vehicle when it decelerates from current speed to 0) and \( s_j \) (the distance of lead intelligent vehicle when it decelerates from current speed to 0) in Figure 2 to calculate \( d_{sij} \) (the shortest safe distance between intelligent vehicle and lead vehicle). The derivation process is as follows:

\[
d_{sij} = s_i - s_j,
\]

\[
s_i = v_i^2 \Delta t_0 + \frac{1}{2} a_0 \Delta t_0^2,
\]

\[
\Delta t_0 = \frac{v_i' - v_i}{a_0}.
\]

Combining (2) and (3),

\[
s_i = \frac{v_i^2 - v_i'^2}{2 \times a_0},
\]

where \( v_i \) is current velocity of the intelligent vehicle. The shortest safe distance requires the distance difference between two vehicles when they decelerate from current speed to 0 at the same time; therefore, the velocity \( v_i' \) value is 0. \( a_0 \) is the deceleration of intelligent vehicle in deceleration motion, and the value is 6 m/s² [25]. \( \Delta t_0 \) is the time required for the velocity from \( v_i \) to \( v_i' \). Since \( v_i' \) is 0, it can be eliminated in (4). It is necessary to substitute a negative sign in the calculation of distance by using deceleration \( a_j \); therefore, the negative sign can cancel in (4), and finally, we get the distance \( s_i \):

\[
s_i = \frac{v_i^2}{2 \times a_0}.
\]

In the same way, we can get the distance \( s_j \):

\[
s_j = \frac{v_j^2}{2 \times a_1}.
\]

\( v_j \) is the velocity of the lead vehicle. According to paper [25], the deceleration of the lead vehicle is \( a_1 = 4 \text{ m/s}^2 \) and uses a predefined distance 10 metres to represent the reaction distance and the lead vehicle’s length. Finally, we get the shortest safe distance: \( d_{sij} = v_i^2/(2 \times a_0) - v_i'^2/(2 \times a_1) + 10 \). Similarly, we can get \( d_{sij} = v_i^2/(2 \times a_0) - v_i'^2/(2 \times a_1) + 10 \), where \( v_k \) is the velocity of the lag vehicle. The specific derivation process is the same as \( d_{sij} \), and the other parameters are the same as above.

**Data Availability**

The data used to support the findings of this study are included within the article.

**Conflicts of Interest**

The authors declare that they have no competing interest.

**Acknowledgments**

This work was supported in part by the National Natural Science Foundation of China under Grant No. U1808206.

**References**


