A Study on Collaborative Lane Change Decision Making of Multi-automated Vehicles Based on Deep Graph Reinforcement Learning

Xiang Li[1(✉)], Jianxun Cui[1,2], and Haozhe Ji[3]

[1] Harbin Institute of Technology, Harbin City 150000, Heilongjiang Province, China
23s132082@stu.hit.edu.cn
[2] Chongqing Research Institute of HIT, 618 Liangjiang Avenue, Longxing Town, Yubei District,Chongqing, China
[3] Jiamusi University, Jiamusi City 154000, Heilongjiang Province, China

Abstract. The lane change decision making module plays a crucial role in autonomous driving systems, facing the challenge of balancing collaborative traffic operation. Modeling complex interactions among multiple autonomous vehicles in coexisting environments poses significant challenges. This study focuses on collaborative lane change decision making for multiple autonomous vehicles by employing deep graph convolutional neural networks. These networks effectively model the interaction and collaboration among vehicles, while reinforcement learning facilitates the iterative evolution of decision-making. To evaluate the performance of the proposed Graph Reinforcement Learning (GRL) method, an interactive driving scenario with two ramps on a highway was developed. Simulation experiments were conducted on the SUMO platform to compare different GRL methods. Results were analyzed from multiple perspectives and dimensions to compare the characteristics of different GRL methods in the scenario of highway merging traffic. The findings demonstrate that the utilization of deep graph convolutional neural network can effectively model the complex interactions among vehicles and the combination of graph convolution and reinforcement learning can significantly improve the performance of lane-changing behaviors in terms of both efficiency and safety.

Keywords: Autonomous Driving · Collaborative Decision Making · Deep Graph Reinforcement Learning

1 Introduction

Due to economic development and improved quality of life, traffic frequency has significantly increased,resulting in more severe issues related to traffic safety and

X. Chen et al. (Eds.): IoTaaS 2023, LNICST 585, pp. 163–182, 2025.
https://doi.org/10.1007/978-3-031-70507-6_14

congestion. Global road traffic fatalities reach 1.35 million annually since 2000, with injuries totaling 5,000. Additionally, economic losses surpass $500 billion each year. According to a 2015 report by the U.S. Department of Transportation, 94% of traffic accidents were caused by driver errors, including recognition errors (41%) and decision-making errors (33%)[1].

In the field of intelligent networked vehicles research, driving behaviors including lane keeping, merging into traffic, lane changing, and overtaking play a crucial role in ensuring driving safety. Making reasonable decisions based on different traffic environments is especially important, particularly in high-complexity and high-dynamics traffic scenarios, where irrational decision-making behaviors can lead to serious consequences. These traffic environments are characterized by high dynamics and uncertainty, making behavioral decision-making a challenging task in self-driving vehicle research. The problem of behavioral decision-making in intelligent networked vehicles is a high-dimensional and continuous sequential decision-making problem. Traditional dynamic planning methods and rule-based traffic models are insufficient to meet the requirements of dynamic decision-making in complex environments. Recently, reinforcement learning methods have gained attention as a research hotspot in the field of autonomous driving. By utilizing reinforcement learning, more accurate decision-making of single automated vehicle can be achieved, effectively avoiding traffic accidents and reducing economic losses. Furthermore, studying multi-vehicle automatic cooperative lane-changing control for intelligent networked vehicles in mixed traffic flow can enhance lane-changing efficiency, particularly in complex environments (e.g. highway on/off ramping scenarios). But how to effectively capture the complex interactions among the decisions made by different automated vehicles and finally improve the overall driving efficiency and safety of this multi-vehicle system is essentially very challenging. In this research, we try to use deep graph convolutional neural network combined with reinforcement learning, i.e. GCN-DRL, to model the information sharing and collaborative decision making among multiple automated vehicles.

2 Literature Review

Current research on lane changing decision making for intelligent networked vehicles mainly focuses on the following aspects: controlling vehicles for decision making by setting up scenarios and rules in advance, lane changing decision making based on deep reinforcement learning, and applying Vehicle-to-Vehicle (V2V) communication technology for multi-vehicle collaboration.

2.1 Rule-Based Decision Modeling for Lane Changes

In 1986, Gipps proposed a model that takes into account the necessity, propensity, and safety aspects of lane-changing behavior, and this approach prioritizes different influences on lane-changing choices, but does not take into account subsequent differences in driver behavioral characteristics [2]. Yang et al. improved

the flexibility of the model by considering lane change probability [3]. Ahmed designed a probabilistic model by stepwise decomposition of lane changing behavior, mainly based on utility selection theory [4].Toledo et al. continued Ahmed et al.'s work on probabilistic models of utility selection and designed a model that can integrate forced and active lane changing, but the model, although very theoretically rigorous, is overly complex and extremely difficult to calibrate the parameters [5]. In 2007, Kesting et al. proposed the MOBIL model [6]. In a paper published in 2014, Jakob Erdmann proposed the LC2013 model applied to the open-source simulation platform SUMO, which sets up four lane-changing scenarios, and decides whether to change lanes or not by calculating whether a threshold is reached in each scenario through a formula that facilitates the successful execution of the desired lane-changing maneuver [7] .

2.2 Deep Reinforcement Learning Based Lane Change Decision Making

In addition to rule-based lane changing methods, the most mainstream application of deep reinforcement learning for decision making is currently available. In 2017, Sallab et al. applied DQN for the learning of lane-merging strategies, focusing on vehicle ramp merging [8] . In this strategy, Sallab et al. took into account both the short-term goal of successful merging and the long-term goal of smoothing the merging trajectory. To ensure that the learned strategy balances safety and strategy, the state perception of the surrounding environment is sensed through a long and short-term memory network. For the same ramp merging scenario, Huegle et al. of the University of Freiburg, in 2019, analyzed the performance of existing classical perception methods utilized during reinforcement learning strategy optimization, and tested a comparison of ramp merging scenarios via the DQN algorithm [9] . Although deep reinforcement learning has more advantages, it also has certain problems, mainly including the problem of slow learning rate due to frequent interaction with the environment for data collection. To address these problems, Ye et al. proposed a lane-changing strategy based on the optimization of the proximal policy, which effectively solves the problem of slow learning efficiency of reinforcement learning without destroying the more stable performance of lane-changing [10]. After that, Mukadam et al. proposed a Q-masking technique to improve the learning rate again . Mukadam et al. proposed a Q-masking technique to improve the learning rate again [11]. Mukadam et al. Most of the above studies are single-vehicle automated driving, in order to improve the efficiency of the passage, multi-vehicle collaboration is a direction that must be considered, but this involves the balance between the overall efficiency and the individual efficiency, so multi-intelligent body reinforcement learning is applied in the lane changing process [12] . For example, Dong et al. had collected data from surrounding vehicles through sensors for cooperative strategy training [13] .

3 Methodology

The objective of this paper is to generate cooperative lane change decisions for self-driving cars in an interactive traffic scenario. To achieve this goal, a framework (named as GCN-DRL) is proposed (see Fig. 1), which consists of three key components: an interactive traffic scene, a GCN algorithm and several DRL algorithms, and a simulation platform. The vehicles in the scenario are modeled as a graph, where nodes represent different vehicles and edges represent interactions between every two vehicles. The graph is then processed into node features, adjacency matrices and RL filters, which will be used to get the state description of the whole vehicle system and will be detailed in Sects. 3.1-3.2 separately. The GCN-DRL algorithm is developed to train lane changing strategies, which takes the graphical representation as an input and generates the Q-values for different lane changing actions. The simulation part takes the Q-values as input and generates the lane change decision of each vehicle to update the interactive traffic scene for continuous training of the GCN-DRL network.

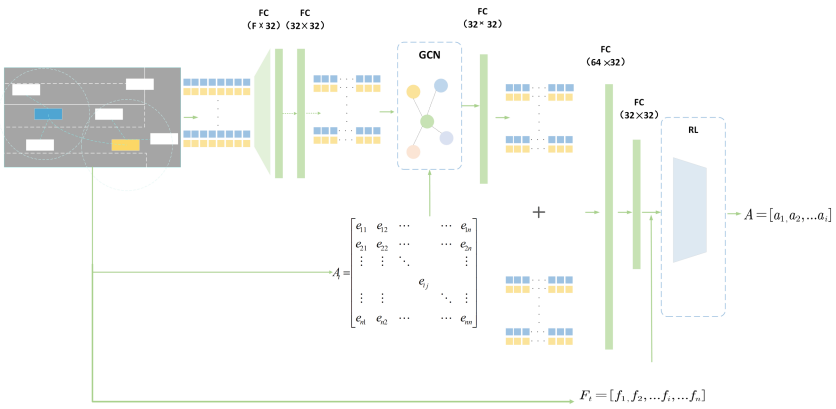

Fig. 1. Collaborative control decision model based on graph neural network and deep reinforcement learning

The scenario studied in this paper is based on a mixed traffic flow scenario on a freeway. The scenario consists of a 3-lane freeway with two ramp exits. In this scenario, two types of vehicles operate: human-driven vehicles and self-driven vehicles. Different vehicles have different driving tasks and need to cooperate in order to accomplish the intended driving tasks more efficiently. White vehicles represent HVs, entering from the left side of the freeway and exiting from the right side. Colored vehicles represent AVs entering from the left side of the freeway, specifically yellow vehicles exiting from Ramp 1 and blue vehicles exiting from Ramp 2 (Fig. 2).

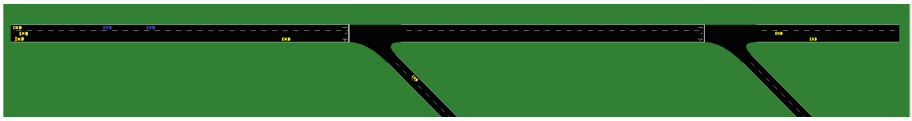

Fig. 2. Collaborative control decision model based on graph neural network and deep reinforcement learning

The self-driving vehicle interacts with the traffic environment by taking lane changing actions in discrete time steps in a constructed traffic scenario. After the action is taken, a state of the traffic environment changes and the self-driving vehicle receives a reward. Moreover, the environment information is fully observable within the observation range of the self-driving vehicle.

The scene is modeled as an undirected graph. Each car in the scene is considered as a node of the graph, and the interactions between cars are considered as edges of the graph. More precisely, the constructed graph is described as $G = \{N, E\}$,where $N = \{n_i, i \in \{1, 2, ...N\}\}$ is a set of node attributes and $E = \{e_{ij}, i, j \in \{1, 2, ...N\}\}$ is a set of edge attributes. Specifically, n denotes the number of nodes in the constructed graph, which is equal to the total number of vehicles. The properties and interactions of the vehicles are then modeled using the graph representation. The state space is discrete and can be described as follows:

$$S_t = [N_t, A_t, F_t] \tag{1}$$

where:$N_t$denotes the node feature matrix, $A_t$denotes the adjacency matrix, and $F_t$denotes the RL filter matrix. Together, the above three matrices form the state space matrix, which is the initial input to the entire model.

Node Characterization Matrix. The node feature matrix represents the features of the constructed scene. It consists of a feature matrix for each vehicle and can be described as follows:

$$N_t = \begin{bmatrix} [V_1 & X_1 & L_1 & I_1] \\ [V_2 & X_2 & L_2 & I_2] \\ & \cdots & & \\ [V_i & X_i & L_i & I_i] \\ & \cdots & & \\ [V_n & X_n & L_n & I_n] \end{bmatrix} \tag{2}$$

where $[V_i, X_i, L_i, I_i]$ denotes the normalized state matrix of each vehicle. Specifically, in the lane changing task, $V_i = V_{i^{practice}} \setminus V_{max}$ denotes the normalized longitudinal velocity of the vehicle with respect to the maximum longitudinal velocity; $X_i = X_{i^{vertical}} \setminus L_{highway}$ denotes the normalized longitudinal coordinates of the vehicle with respect to the length of the roadway; L_i denotes a one-heat-code matrix of the current channel representing the vehicle's position (left lane, right lane, and middle lane); and Ii is a one-heat-code matrix denoting the vehicle's current intention (change to left lane, change to right lane, and straight ahead) of a one-heat code matrix.

Adjacency Matrix. The adjacency matrix represents the interactions between vehicles and reflects the information sharing between vehicles in the constructed scenario. The adjacency matrix is computed based on the following assumptions:

- All vehicles can share information with themselves.
- All AVs can share information.
- All AVs can get information from HVs, which are within the sensing range of the AVs.

The adjacency matrix is derived as follows:

$$A_t = \begin{bmatrix} e_{11} & e_{12} & \cdots\cdots & e_{1n} \\ e_{21} & e_{22} & \cdots\cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ & & e_{ij} & \\ \vdots & \vdots & & \ddots \\ e_{n1} & e_{n2} & \cdots\cdots & e_{nn} \end{bmatrix} \tag{3}$$

where e_{ij} indicates whether or not $vehicle_i$ and $vehicle_j$ are within their respective observation ranges. When $vehicle_i$ and $vehicle_j$ share information, i.e., can be observed or information can be transferred, $e_{ij} = 1$; while when $vehicle_i$ and $vehicle_j$ are not in the observation range or cannot transfer information $e_{ij} = 0$.

RL Filtering Matrix. The RL filtering matrix is constructed to filter out the corresponding terms of the HVs used for the output of the algorithm.The RL filtering matrix is derived as follows (Fig. 3):

$$F_t = \left[f_1, f_2, \cdots f_i, \cdots f_n\right] \tag{4}$$

where $f_i = 0$ or 1, and $f_i = 1$ if $vehicle_i$ belongs to an artificial vehicle controlled by the GCN-DRL algorithm; otherwise $f_i = 0$.

3.1 Graph Convolutional Neural Network Module

Autonomous vehicles can transmit information to each other through the vehicular network, but artificial vehicles cannot transmit information to each other. And each self-driving vehicle can only observe the information of artificial vehicles within a certain range of itself, therefore, the communication function of self-driving vehicles can be utilized to share the information of each self-driving vehicle through the vehicular network, so that the self-driving vehicles within the road section have a more comprehensive control of the information in the road section, in order to realize the function of information aggregation, we designed the graph neural network module, i.e. GNN module.

The GNN module consists of a fully connected layer and a graph convolutional network,where the fully connected network is two layers and the GCN network is one layer.

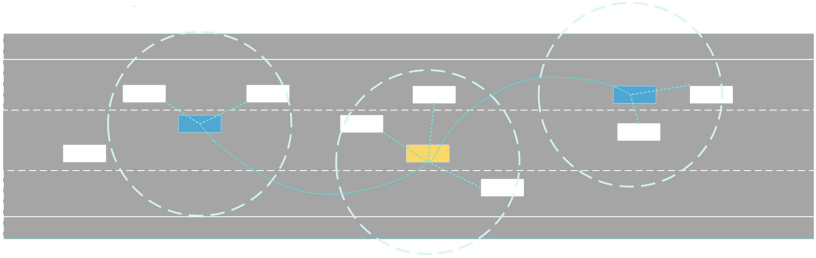

Fig. 3. Schematic diagram of graphical neural network information aggregation

Before implementing the graph convolution process, the node feature matrix Nt is first input into the fully connected layer to map the node features into the sample labeling space as described below:

$$N_{FC} = \phi_{FC}(N_t) \tag{5}$$

Where: N_{FC} denotes the node feature matrix output from the fully connected layer; ϕ_{FC} denotes the neural network with full connectivity. Then, $N_{FC}$and the adjacency matrix A_t are firstly input into the GCN to generate graph convolution features; and the graph convolution process can be expressed as the following equation:

$$G_t = \phi_{GCN}(N_{FC}, A_t) = \sigma(D_t^{1/2} A_t D^{-1/2} N_{FC} W_t + b) \tag{6}$$

Where: g_t denotes the graph convolution feature starting from GCN; ϕ^{GCN} denotes the graph convolution operator; matrix D_t is calculated based on A_t ; W_t denotes the layer-specific trainable weight matrix; b denotes the offset and b=0 is defined in this study; σ denotes the activation function and ReLU is chosen in this study. The output of the graph neural network module is fed into the DRL module and then filtered through the RL filter matrix to generate the Q-value of the AV's lane changing maneuver, the Q-value is derived as follows:

$$Q(s,a) = F_t - \left[\phi^{DRL}(G_t)\right] \tag{7}$$

where $Q(s,a)$ denotes the Q-value of the AVs' lane-changing action; and ϕ^{DRL} denotes the policy neural network.

3.2 Deep Reinforcement Learning Module for Lane Change Decision Making Problems

The deep reinforcement learning module for the lane change decision problem selects DQN , Double DQN , Dueling DQN , Dueling Double DQN (D3QN) for the comparison, and the state space is the information of the self-driving vehicle that has been processed by the graph neural network module. This section defines the behavior space and reward function of the self-driving vehicle in the reinforcement learning module.

Behavioral Space. For each time step, each CAV has a discrete action space representing the potential actions to be taken, as follows:

$$a_i = \{turn \quad left, straight, turn \quad right\} \tag{8}$$

Where: "straight" is to keep the current state of the vehicle. In the lateral decision, "turn left" or "turn right" generates the lane number of the target lane. The control module adjusts the lateral speed of the vehicle according to the target lane number. The overall action space of the deep reinforcement learning module is a combinatorial space that aggregates all possible combinations of a single CAV: $A = \{a_i\}_{i=1}^{n}$, i is the number of CAVs at the current time step. It is important to note that the simulation process restricts vehicles from leaving the simulated path, but does not prevent them from colliding during lane changes.

Reward Function. The reward function consists of: 2 reward types and 2 penalty types. They are intentional reward, speed reward, lane change penalty and collision penalty (Table 1). The intent reward is to ensure that all CAVs merge out of the prescribed ramp. Speed rewards encourage actions that improve system efficiency. Specifically, to balance with other reward sources and encourage cooperation, we define "soft and smooth" intentional rewards at each time step, rather than rewarding vehicles immediately upon successful divergence. We denote CAVs exiting the first ramp as "$merge\_1$" CAVs and CAVs merging from the second ramp as "$merge\_2$" CAVs. Intentional rewards are computed only on the first 2 segments: seg_1 and seg_2, where p_{ij} and $r_{ij}$are the penalties and rewards for vehicles exiting from different ramps on two different segments, and x_{ij} is the perpendicular distance of a vehicle exiting from a different ramp from the baseline defining the different segments, and thus reverts to 0 when the vehicle enters a new segment (Fig. 4).

Table 1. Reward and Penalize

Sections	Norm	Define	Calculation
Seg_1	p_{11}	Penalty for "$merge\_1$" CAV in the top lane increases as approaching Ramp 1.	$p_{11} = \frac{x_{11}}{L_1}$
	r_{11}	Reward for "$merge\_1$" CAV in the bottom lane, increased from approach to ramp 1.	$r_{11} = 1 + \frac{x_{11}}{L_1}$
	p_{21}	Penalty for "$merge\_2$" CAV in the top lane, increase on approach to ramp 1.	$p_{21} = \frac{x_{21}}{L_1}$
Seg_2	p_{22}	Penalty for the "$merge\_2$" CAV in the top lane increases as you approach ramp 2.	$p_{22} = \frac{x_{22}}{L_2}$
	r_{11}	The bottom lane on the "$merge\_2$" CAV is rewarded with an increase from approaching ramp 2.	$r_{22} = 1 + \frac{x_{22}}{L_2}$

The total intent reward is defined as the individual reward or penalty for each CAV on different sections of the road:

$$R_i = \sum_{k=1}^{n} \sum_{i,j} (r_{ij} - p_{ij}) \delta_{ij}^{n} \tag{9}$$

Among them:δ_{ij}^{n} is used to determine whether a vehicle has entered the area where it can get penalties and rewards, p_{ij} and r_{ij} are the penalties and rewards

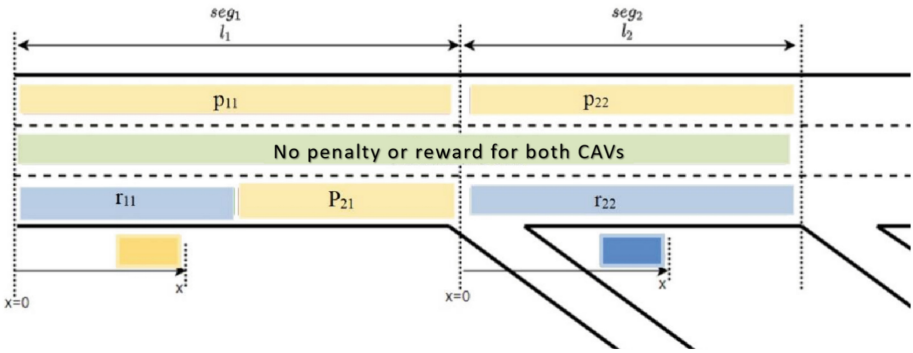

Fig. 4. Schematic diagram of penalties and rewards

for different types of vehicles in different segments, and we have set a "soft" reward to encourage "$merge\_2$".

CAVs yield the bottom lane to "$merge\_1$" CAVs before the first on-ramp and merge into the bottom lane as soon as they enter the second segment. In addition, the absolute value of the reward and penalty is a function of vehicle position to encourage CAVs to make early lane changing decisions.

In addition to the intent bonus, other performance metrics were considered, including efficiency, safety, and comfort, which were expressed using the speed bonus, collision penalty, and lane change penalty. Speed bonus is defined as R_V:

$$R_V = 1/n \sum_{i=1}^{n} \frac{v_{cav}^i}{v_{max}} \tag{10}$$

Where: v_{cav}^idenotes the speed of the i^{th} CAV on the road;v_{max} is the maximum speed limit of the vehicle on the roadway. The collision penalty $P_c$is a fixed large positive value used to constrain behaviors that may produce collisions. Lane Change Penalty $P_{LC}$is defined as a constant penalty added to each lane change decision and is used to encourage CAVs to stay in their lanes instead of changing lanes frequently.

The overall reward function is defined as:

$$R_{total} = \omega_1 R_l + \omega_2 R_V + \omega_3 P_C + \omega_4 P_{LC} \tag{11}$$

Where:ω_1 ...ω_4 are weights that can be adjusted according to the model, it should be noted that the ratios between "Intent Satisfaction", "Driving Speed", "Safety", and It should be noted that when ω_1 and ω_2 are dominant compared to ω_3 and ω_4 . This setting will encourage vehicles to change lanes to meet intentions and travel fast to reach their destinations, but there is a risk that decision-making will be error-prone and overly aggressive. On the other hand, making ω_3 and ω_4 larger than ω_1 and ω_2 means higher penalties for unsafe and frequent lane changing behavior. In this case, the model learns to adopt very conservative and safety-conscious behaviors even at the risk of failing to achieve the merger intent.

4 Experiments

4.1 Settings

Barrier Vehicle Settings. For all artificial vehicles, the longitudinal control is done using an intelligent driver model, i.e., the IDM model.The principle of the IDM model is that after inputting the speed of the vehicle in front, the distance from the vehicle in front, and the speed of the self vehicle variables, the final output is an acceleration that is used for the vehicle to perform the longitudinal control, and the equations of acceleration are obtained as follows:

$$\dot{v} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*(v,\Delta v)}{s}\right)^2\right] \tag{12}$$

Where: a is the maximum acceleration of the modeled vehicle; v is the current speed of the modeled vehicle; $v_0$is the desired speed of the modeled vehicle; $\Delta v$is the relative speed between the modeled vehicle and the vehicle in front of it; $\delta$is the acceleration index, with the increase of δ, the acceleration and deceleration process of the vehicle will be more intense; s is the distance between the modeled vehicle and the vehicle in front of it;$s^*(v, \Delta v)$ is the desired following distance of the modeled vehicle.$\left(\frac{v}{v_0}\right)^{\delta}$ is used to measure the difference between the current speed and the desired speed to promote vehicle acceleration.$\left(\frac{s^*(v,\Delta v)}{s}\right)^2$is used to measure the difference between the current distance and the desired distance to facilitate vehicle braking.The desired vehicle spacing is obtained from the following equation:

$$s^*(v, \Delta v) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}} \tag{13}$$

Where:b for the comfortable deceleration;s_0 for the vehicle minimum distance, this parameter is related to the assumption of the IDM model, even if the vehicle is at a standstill, the minimum distance to be maintained, if the distance between the two workshops is less than s_0 , it can be assumed that a collision of the two vehicles;T for the safety of the headway when the distance.The expectation distance equation contains a balancing term $s_0 + vT$, and a power term $\frac{v\Delta v}{2\sqrt{ab}}$ to realize an intelligent braking strategy. The settings of the model parameters for the IDM model in this experiment are shown in the table below:

Table 2. IDM model parameters

Parameters	**Value**	**Unit**
maximum acceleration	6	m/s^2
Maximum (comfortable) deceleration	-5	m/s^2
Desired speed of the vehiclev_0	30	m/s^2
acceleration index (math.)δ	4	–
Minimum distance between cars s_0	2	s
For safe headway T_0	1	s

The experimental vehicle lateral control algorithms all use SUMO's built-in rule-based LC2013 algorithm.LC2013 calculates the vehicle's decision to change lanes in a single simulation step, based on the vehicle's travel route and the current and historical traffic conditions around the vehicle. In addition, it calculates the vehicle's own and the obstacle vehicle's speed change to facilitate the successful execution of the desired lane change maneuver.The LC2013 model explicitly distinguishes between four different motivations for changing lanes: Strategic change, Cooperative change, Tactical change, and Obligatory change. Each of the four motivations is calculated independently, and a lane change is triggered when the value of a certain motivation reaches a certain threshold.LC2013 has one drawback, namely, it considers that a lane change occurs instantaneously and does not take into account factors such as the angle of deflection of the vehicle during the lane change, as shown in Fig. 5:

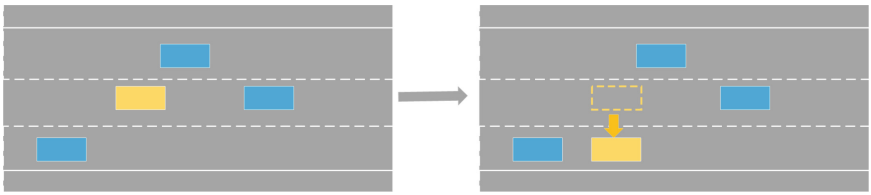

Fig. 5. Schematic diagram of LC2013 lane change model

Experimental Scenarios and Parameter Settings. This experiment uses the SUMO platform to design a 500m long highway. The highway is a three-lane road with exit ramps at 200m and 400m from the entrance. The vehicles involved in the experiment include artificial vehicles and self-driving vehicles. The artificial vehicles are white and their longitudinal control algorithm is selected from the IDM model. Their transverse algorithms are based on the built-in LC2013 model in SUMO. The self-driving vehicles are categorized into two types based on the exit ramps. Vehicles exiting from the first ramp are yellow, while vehicles exiting from the second ramp are blue. The longitudinal control algorithm of the self-driving vehicles is the same as that of the manual vehicles. However, the transverse control algorithm (i.e., lane changing process) adopts the GCN-DRL model proposed in this paper. The self-driving vehicle fuses the information from both the manually driven vehicle and the self-driving vehicle through the GCN algorithm. The fused information is used as input to the deep reinforcement learning module, which in turn outputs the lane changing behavior of each self-driving vehicle. In this paper, different deep reinforcement learning algorithms are selected for training for three times, and each training is set to 200 rounds.

After training, each algorithm is tested three times with each test set to 20 rounds. The following table shows the parameters of the experimental settings. The following table shows the parameters of the experimental settings.

Table 3. Parameter settings for simulation experiments

Parameters	Value
Number of training rounds	200
Followed by exploration phase step	20000
Number of small batches processed	32
Experience pool capacity	50000
Discount factor γ	0.9
Adam Learning Rate	10^{-4}
randomized action probability	0.3
Frequency of online web updates	10
Frequency of target network updates	1000

As shown in Table 3, the number of manually driven vehicles is 20. The number of self-driving vehicles from the first ramp is 10, and the number of self-driving vehicles from the second ramp is 10. The traffic flow of various vehicles will be changed according to the purpose of the experiment. The maximum speed of the manual vehicles is 30 m/s, and the maximum speed of the self-driving vehicles is 25 m/s. The units of speed are all m/s, which is consistent with the default units of speed in SUMO. The observation radius of the self-driving vehicle is set to 25m, which can be changed according to the specific experimental objectives. A Warmup session was performed for the first 2000 steps, i.e., a smaller learning rate was used for about 15 rounds to prevent overlearning in the early stages when the simulation data and vehicle behavior were poor, resulting in the need for more training rounds in the later stages in order to correct the training results. In reinforcement learning, the intelligence is trained through each state transfer process. In general simulation, after learning a state transfer process, this state transfer process is discarded, which results in a large amount of wasted experience. Therefore, this paper adopts the method of experience playback: set up an experience pool which can store 50000 processes. When the experience pool is full, for each step of simulation, one old data in the experience pool is eliminated, and then an updated state transfer process is deposited. In order to improve the efficiency, we set the minimum batch as 32, when the experience pool data is larger than 32 that is, we adopt the minimum batch method and randomly select 32 data in the experience pool for training. In addition, in order to increase the chance of exploration, a simple greedy strategy with a random action probability of 0.3 is set in the training process.

Table 4. Parameters of the training process

Parameters	Value
umber of manual vehicles	20
Number of self-driving vehicles	20
Number of self-driving vehicles exiting the first ramp	20
Number of self-driving vehicles exiting the second ramp	10
Length of highways	500
First ramp location	200
Second ramp location	400
Maximum speed for manual vehicles	$30m/s$
Maximum speed of self-driving vehicles	$25m/s$
Traffic flow of manual vehicles	$0.1 - 0.5vel/s$
Volume of Autonomous Vehicles Exiting Ramp 1	$0.1 - 0.5vel/s$
Traffic flow of self-driving vehicles exiting the second ramp	$0.1 - 0.5vel/s$
Radius of observation for self-driving vehicles	$25m$

Test Task Setting and Evaluation Indicators. In order to obtain more generalized and accurate results, we set up a variety of scenarios during the testing process:

- The reward loss etc. obtained during training is analyzed for the rule-based approach as well as for four different deep reinforcement learning algorithms.
- Changing the proportions of the components of the reward function in the deep reinforcement learning module makes the vehicle more efficiency-focused or more robust as it travels to its destination.
- Changing the traffic flow of manual and self-driving vehicles and comparing the difference between the metrics of different deep reinforcement learning algorithms and rule-based methods, respectively.
  In order to better judge the experimental results, the following evaluation indexes are selected in this paper:
- The final total reward value of each round. The reward value is one of the important indexes for evaluating the effect of reinforcement learning module, which directly reflects the effect of lane changing, and because the speed, the number of lane changing, and the intention to change lanes have been taken into account when setting the reward function, the reward is also an index with comprehensive evaluation, which is of great significance.
- Loss per round in training. The speed of loss decrease reflects the efficiency of the training process.
- Average speed of the self-driving vehicle. Although this paper designs only the lane changing decision model, the lane changing behavior of the self-driving vehicle still affects the longitudinal following behavior. Therefore the lane changing decision model should not only realize the task of driving out of the ramp through the lane changing behavior, but also need to be beneficial to

enhance the vehicle speed and improve the driving efficiency. So the average speed is also an important evaluation index.

4.2 Experiments Results

Experimental Results of Different Reinforcement Learning Lane-changing Strategies. The longitudinal control algorithms of the vehicles all use the IDM model, and the lateral control method of the manual vehicle uses LC201.The lateral control of the self-driving vehicle uses five methods, namely LC2013, DQN, DoubleDQN, Dueling DQN, and D3QN, respectively. It should be noted that when performing the per-round reward statistics, the reward function calculation is also performed when the self-driving vehicle uses LC2013, a rule-based lane-changing model, and the reward function calculation method is the same as that of the four reinforcement learning algorithms. However, no training in terms of reinforcement learning is performed on the vehicle. This is the explanation for setting the Rule-based curve in the reward image. In the simulation process, it can be found that in the early stage of training, due to the setting of collision penalties, the vehicle's lane changing is very frequent, and even shows oscillation phenomenon. With the increase in the number of iterations and the constraint of frequent lane changing penalty, the vehicle can gradually avoid unnecessary lane changing behavior.

Below is a line graph of the reward values for each round in training for the five lane changing methods LC2013, DQN, DoubleDQN, Dueling DQN, and D3QN. Three training sessions were conducted for each method in training, and the results of the three tests were eventually averaged.

As shown in Fig. 6 The total reward value of the four reinforcement learning algorithms is much higher than the rule-based LC2013 method. It proves that GCN can well fuse the information between individual vehicles and allow vehicles to interact. And the use of various types of reinforcement learning algorithms in combination with GCN can improve the performance of lane changing behavior generation. In the Warmup phase, the reward value is low and volatile due to the use of a smaller learning rate and the random generation of the behavior of the self-driving vehicle. After the Warmup phase, the reward value increases rapidly and the fluctuation becomes smaller. However, the reward curves of these four types of reinforcement learning do not show significant differences. From the table, it can be concluded that the average reward of Double DQN is higher than that of DQN; the average rewards of Dueling DQN and D3QN are much higher than that of DQN.This is because the establishment of Dueling Network can better optimize the generation of behavioral decisions of AV.

As shown in Fig. 7 The loss of the four algorithms during the training process decreases as the number of training times increases and converges to a stable value.DQN and Double DQN do not have significant differences in the loss convergence process, but are significantly higher than Dueling DQN. and the convergence speed is relatively slow. After the loss convergence process, the loss curves of the four algorithms are not significantly different. From the table, it can be seen that Dueling DQN has the lowest average loss and there is no

significant difference in the average loss between DQN and Double DQN. This result can prove that using Dueling Network optimization strategy can effectively reduce the loss during the training process.

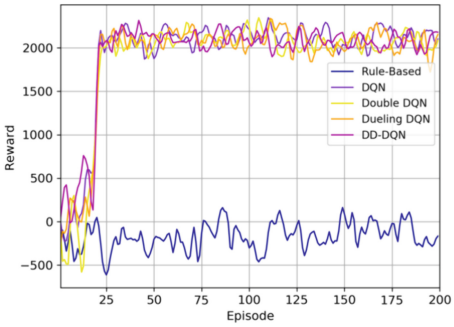

Fig. 6. Reward for five lane changing methods

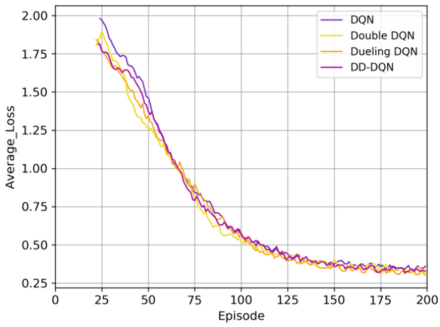

Fig. 7. Losses for four lane changing methods

Table 5. Average loss for different channelization methods

Loss	Value
DQN	0.712
Double DQN	0.707
Dueling DQN	0.694
Double Dueling DQN	0.692

As can be seen in Fig. 8, the average Q-value increases as the number of trainings increases. However, the four algorithms have different trends in the average Q. The curve of DQN is higher in the later stages, but the average return is relatively low, which is due to the overestimation problem caused by the maximal operator in calculating the target Q. The relatively smooth change of the curve of D3QN suggests that the implementation of Double and Dueling operations can effectively stabilize the Q evaluation.The average Q of the Dueling DQN is is lower than that of DQN, but higher than that of Double DQN and D3QN, indicating that the establishment of Dueling network is also favorable for Q-value evaluation.

After training, we conducted twenty tests of the four GRL algorithms with the same experimental parameter settings. In the simulation process, even in the first round, the test process did not have the same phenomenon of frequent lane changing of vehicles as in rounds 1 and 2 of the training, which proves that the frequent lane changing penalty set in the reward effectively curbed the unnecessary lane changing behavior.

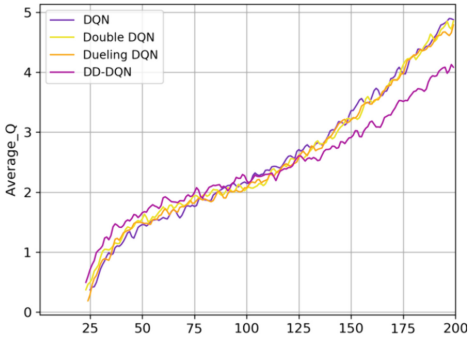

Fig. 8. Average Q of different channelization methods

Table 6. Test reward

Reward	Value
DQN	2053.4779
Double DQN	2055.7757
Dueling DQN	2102.6616
Double Dueling DQN	2165.0412

During the test, the average reward was calculated and the results are shown in Table 6 and the rewards for each round are shown in Fig. 9.
In Table 6, the average test rewards of the four GRL algorithms are the highest for D3QN, and the improved D3QN method gives better results in the constructed traffic scenarios. During the test, the average speed was calculated and the results are shown in the Table and Figure (Fig. 10).

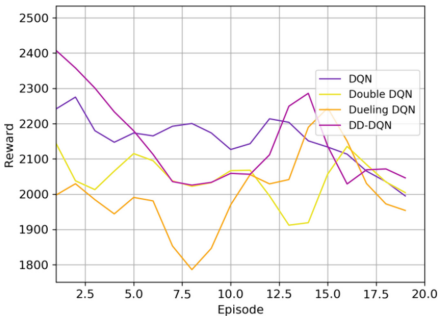

Fig. 9. Testing Reward

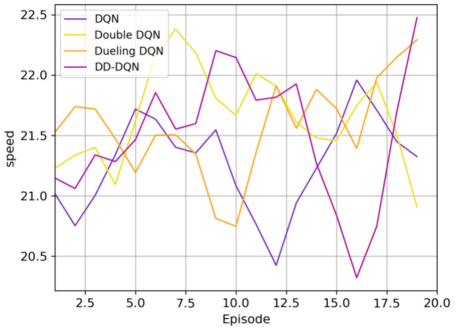

Fig. 10. Testing Speed

Table 7. Testing Speed

Speed	Value
DQN	21.2783
Double DQN	21.6193
Dueling DQN	21.5666
Double Dueling DQN	21.5114

In Table 7, Double DQN is the fastest, Dueling DQN and Double Dueling DQN are the next fastest, and DQN is the worst realized, but no gap is created in the average speed of the four GRL algorithms.

Test Results for Different Reward Function Weights. The reward function in this paper (see Equation 17), contains four components: 2 reward types and 2 penalty types. These are intent reward, speed reward, lane change penalty and collision penalty, where:$\omega_1 ... \omega_4$are weights that can be adjusted according to the model.When ω_1 andω_2 dominate compared to ω_3 and ω_4 . This setting will encourage vehicles to change lanes to meet the intent, but carries the risk of error-prone and overly aggressive decision making. Whenω_3 and ω_4 are greater than ω_1 andω_2, this means higher penalties for unsafe and frequent lane changing behavior. The model will adopt very conservative and safety-conscious behavior. In order to explore the impact of two different reward function weights, this paper sets up two experiments using DQN as the lane changing decision algorithm, the data of the two experiments are shown in the table, and the experimental results are shown in Figs. 11 and 12

Table 8. First experiment weights

Parameters	First experiment		Second experiment	
	Weights	Value	Weights	Value
R_I	ω_1	3	ω_1	8
R_V	ω_2	0.8	ω_2	3
P_C	ω_3	0.05	ω_3	0.05
P_{LC}	ω_4	0.8	ω_4	0.8

In order to facilitate the comparison of the size of the weights and change, the weights of the two experiments are not normalized, and the size of the reward function of each part of the order of magnitude is not the same, so after changing the weights, the significance of the rewards, losses, and Q-value invalidated, but the average speed of the vehicle and the number of collisions during the training process is still valid and important indicators.

The results exhibited in Figs. 11 and 12 coincide with those previously conjectured.

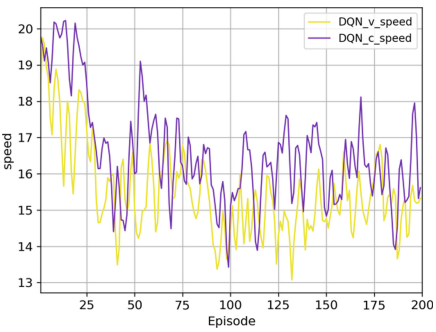

Fig. 11. Velocity of two experiments

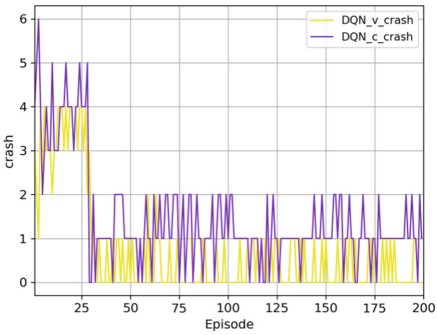

Fig. 12. Number of collisions in two experiments

Different Manual Traffic Flow Test Results. The combined model using GCN and DQN was tested in mixed traffic with different traffic densities. The training in the previous section was performed at a density of $0.3veh/s$ for manual vehicles , $0.2veh/s$ for self-driving vehicles. In this section, the model is tested with different flow rates, with the flow rate of manual vehicles set to $0.1veh/s$, $0.2veh/s$, $0.3veh/s$, $0.4veh/s$, $0.5veh/s$, for a total of 200 rounds of training, and the rest of the settings are the same as in the previous section. Finally, the respective reward values as well as the average speeds for different flows were counted as follows.

Table 9 shows the reward values for different flow rates, and the reward values basically show a climb as the manual vehicle flow rate increases. After the artificial traffic flow changes from $0.3veh/s$ to $0.4veh/s$, there is a relatively large increase in the reward value, presumably because of the fixed number of artificial vehicles set, if the artificial driving vehicle traffic is too large, it will all drive out when the self-driving vehicles are not all present in the environment, resulting in only self-driving vehicles in the later simulation phase. Due to fewer vehicles, there is an increase in speed, which in turn increases the speed bonus, while the collision penalty is almost zero.The average speed gradually increases as the flow rate of manual vehicles increases.

Table 9. First experiment weights

Reward	Value	Speed	Value
DQN_Reward_01	1763.8816	DQN_speed_01	21.4422
DQN_Reward_02	1819.7418	DQN_speed_02	22.0929
DQN_Reward_03	1819.3983	DQN_speed_03	22.1618
DQN_Reward_04	1943.0998	DQN_speed_04	22.6169
DQN_Reward_05	1954.3864	DQN_speed_05	22.6594

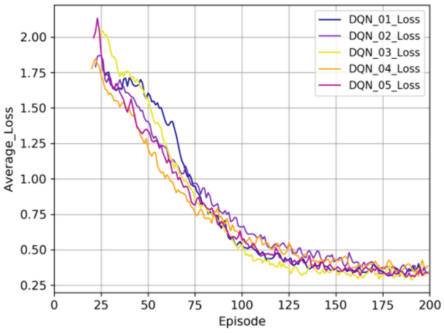

Fig. 13. Average loss

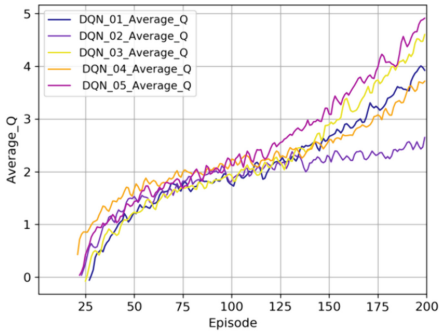

Fig. 14. Average Q

5 Conclusion

In this research, collaborative lane change decision making of multiple automated vehicles in a complex, mixed and highly interactive traffic scene is studied by deep graph neural network combined with reinforcement learning, i.e. GRL. An innovative modular framework, named as GCN-DRL, is proposed which supports different types of combinations of GNN and DRL methods and can be validated in various types of interactive traffic scenarios. Besides, this research proposes a new fusion method based on graph convolutional neural network, which solves the dynamic input size problem by aggregating information from multiple sources with graphical representations. A centralized multi-intelligence body controller is then constructed based on the fused information to make collaborative lane change decisions for a dynamic number of CAVs in a CAV network. Subsequently, the proposed framework is programmed in Ubuntu 20.04 system using python and simulated through SUMO platform to verify the functionality of the proposed framework. Through a series of experiments, the differences between the four different deep reinforcement learning networks were compared by validating them one by one and counting the information of their respective reward value, loss value and Q value. The final results show that the decision-making effect

of the method using graph neural network with reinforcement learning is much better than the rule-based LC2013 model, and among the four kinds of reinforcement learning, the best comprehensive effect is Dueling Double DQN,and the worst one is DQN, but the difference between the four is not big.

References

1. Singh, S.: Critical reasons for crashes investigated in the national motor vehicle crash causationsurvey (2019)
2. Gipps, P.G.: A model for the structure of lane-changing decisions. Trans. Res. Part B: Method. **20**(5), 403–414 (1986)
3. Yang, Q.I., Koutsopoulos, H.N.: A microscopic traffic simulator for evaluation of dynamic traffic management systems. Trans. Res. Part C: Emerging Technol. **4**(3), 113–130 (1996)
4. Ahmed, K.I.: Modeling drivers' acceleration and lane changing behavior. Massachusetts Institute of Technology, Cambridge (1999)
5. Toledo, T., Koutsopoulos, H.N., Ben-Akiva, M.: Integrated driving behavior modeling. Trans. Res. Part C Emerging Technol. **15**(2), 96–112 (2007)
6. Schubert, R., Schulze, K., Wanielik, G.: (2010) Situation assessment for automatic lane-change maneuvers. IEEE Trans. Int. Trans. Syst. **11**(3), 607–616 (2010)
7. Erdmann, J.: Lane-changing model in SUMO, Proceedings of the SUMO2014 modeling mobility with open data, **24**, 77-88 (2014)
8. Sallab, A.E., Abdou, M., Perot, E., et al.: Deep reinforcement learning framework for autonomousdriving. Electron. Imaging **26**(19), 70–76 (2017)
9. Huegle,M., Kalweit,G., Mirchevska,B., et al.: Dynamic input for deep reinforcement learningin autonomous driving, In: 32nd IEEE/RSJ International Conference on Intelligent Robotsand Systems (IROS),pp. 7566-7573.IEEE (2019)
10. Ye ,F., Cheng,X., Wang, P., et al.: Automated lane change strategy using proximal policy104 optimization-based deep reinforcement learning, In: 31st IEEE Intelligent Vehicles Sym-posium (IV),pp. 1746-1752. IEEE (2020)
11. Mukadam ,M., Cosgun, A. ,Nakhaei, A., et al.: Tactical decision making for lane changing withdeep reinforcement learning . In:6th International Conference on Learning Representations,(2017)
12. Wang, G., Hu, J., Li, Z., et al.: Harmonious lane changing via deep reinforcement learning. IEEE Trans. Int. Trans. Syst. **23**(5), 4642–4650 (2022)
13. Dong, J., Chen, S., Li, Y., et al.: Space-weighted information fusion using deep reinforcementlearning: the context of tactical control of lane-changing autonomous vehicles and connec-tivity range assessment. Trans. Res. Part C: Emerging Technol. **128**, 103192 (2021)