Connected and automated vehicle control at unsignalized intersection based on deep reinforcement learning in vehicle-to-infrastructure environment

Juan Chen¹, Vijayan Sugumaran² and Peiyan Qu¹

Abstract
In order to reduce the number of vehicle collisions and average travel time when vehicles pass through an unsignalized intersection with connected and automated vehicle, an improved Double Dueling Deep Q Network method with Convolutional Neural Network and Long Short-Term Memory is presented in this article. This method designs a multi-step reward and penalty method to alleviate the sparse reward problem using positive and negative reward experience replay buffer. The proposed method is validated in a simulation environment with different traffic flow and market penetration under the mixed traffic conditions of automated vehicles and human-driving vehicles. The results show that compared with traditional signal control methods, the proposed method can effectively improve the convergence and stability of the algorithm, reduce the number of collisions, and reduce the average travel time under different traffic conditions.

Keywords
Connected and automated vehicle, 3DQN-CNN-LSTM, unsignalized intersection, left-turning, vehicle-to-infrastructure technology

Date received: 24 August 2021; accepted: 29 June 2022
Handling Editor: Yanjiao Chen

Introduction
Traffic congestion and safety have long been the focus of traffic management authorities and drivers. With the continuous increase of car ownership around the world, traffic congestion becomes more frequent, and the number of traffic accidents increases significantly. Therefore, to effectively alleviate congestion and reduce the occurrence of vehicle accidents is an urgent problem to be solved.

Vehicle-to-infrastructure (V2I) technology can effectively improve vehicle safety and reduce unnecessary parking,¹ reduce fuel consumption and exhaust emissions of vehicles at intersections.² 5G network can greatly improve the efficiency of data collection, storage, and processing in vehicle infrastructure integration, making this technology to have a broader application scenario.

An automated vehicle (AV) can achieve safe and efficient driving by receiving control commands from controllers in real time through wireless information

¹SILC Business School, Shanghai University, Shanghai, China
²School of Business Administration, Oakland University, Rochester, MI, USA

Corresponding author:
Juan Chen, SILC Business School, Shanghai University, 505, Shanghai University Wenshang building, No. 20, Chengzhong Road, Jading District, Shanghai 201899, China.
Email: chenjuan82@shu.edu.cn
interaction function, which helps the application of auto-driving technology at intersections. It is expected that by 2035, the AV may occupy about 25% of new car market. Research shows that online controlled transportation can significantly reduce traffic accidents and improve road traffic efficiency. However, as the AV is introduced, there will be a period of mixed traffic flow of AV and human-driving vehicle (HV), then gradually evolve into an environment full of AV. Therefore, it is important to study the intersection control problem in the case of mixed traffic flow of AV and HV.

For the control of unsignalized intersection, the current research mainly focuses on designing vehicle control algorithm to control the vehicle movement process. In the study of micro-control at unsignalized intersections, the method of optimizing the model or setting safety constraints in the simulation model to improve the safety of vehicles is adopted by most of the studies, while there are few studies on the method of directly coordinating vehicle driving behavior in real time to avoid collision.

Reinforcement learning has the advantages of autonomous learning by agents, and is appropriate for solving decision-making problems under the setting of high-dimensional state space and high-dimensional action space. Over the years, deep reinforcement learning (DRL) has evolved into a more mature research framework. In traffic control, automatic driving, and other fields, DRL has obtained some meaningful research results.

AV control at unsignalized intersections refers to the optimization of driving strategies of AV using micro-control methods, so as to achieve the goals of minimizing the collision and delay of vehicles within the control range of unsignalized intersections. Therefore, focusing on the control of unsignalized intersection, this article proposes an improved Double Dueling Deep Q network method, that is, 3DQN method, to reduce the number of vehicle collisions and improve the traffic efficiency.

Thus, considering the mixed situation of AVs and HVs, this article designs a connected and automated vehicle (CAV) control method based on DRL method for unsignalized intersections with V2I technology. The approach proposed in this article can provide a new control method for CAV to drive in an unsignalized intersection. The main contributions of this article are as follows:

1. Aiming at the micro-control problem of left-turning CAV at an unsignalized intersection, a control method based on an improved 3DQN method is presented in this article. The left-turn CAV approaching, inside, and leaving an intersection is integrated into the control method.
2. In the proposed 3DQN method, in which double deep Q network and dueling deep Q network are combined, important features are extracted with convolutional neural network and an advantageous strategy is obtained based on historical and existing features with Long Short-Term Memory (LSTM) network. A new multi-step reward and penalty method is proposed to solve the problem where there is relatively little experience in vehicle collisions and vehicle passing. At the same time, the sparse reward problem is alleviated using positive and negative reward experience replay buffer (PNRERB).
3. To solve the problems studied in this article, a central control method and multi-agent control method of unsignalized intersection are designed based on the improved 3DQN method. When constructing the DRL model, considering the particularity of the problem, the state of the model is processed. In addition, this article uses micro-simulation software VISSIM to build a virtual environment and uses it as an agent-learning environment. The left-turn and straight-line vehicle in the simulation environment are used as the learning agent, so that the agent can learn independently in the virtual environment. Finally, to verify the effectiveness of agent training in different environments, this article analyzes the effect of unsignalized intersections for different traffic flow levels at different market penetration rates of 0, 0.2, 0.4, 0.6, and 0.8.

The remainder of this article is organized as follows. The section “Literature review” summarizes the current state of research on control and DRL for CAVs at unsignalized intersections. The section “Problem formulation” describes the control problem in V2I environments and the assumptions studied in this article, along with the state space, behavior space, and reward function used in this article for DRL. The section “Improved double dueling deep Q network 3DQN” discusses the improved 3DQN method with Convolutional Neural Network and LSTM (3DQN-CNN-LSTM) method proposed in this article, including multi-step reward and penalty methods, positive and negative sampling experience pools, and so on. The section “Experiment” describes the simulation experiments conducted and analyzes the experimental results. Section “Conclusion” concludes the article and identifies future research directions.

**Literature review**

**Unsignalized intersection control of CAV**

The aim of intelligent CAV control at unsignalized intersection is to alleviate vehicle collision and reduce...
delay by optimizing the driving strategy of CAV. To achieve a safe and efficient access of vehicles in unsignalized intersections, researchers mainly design vehicle control algorithm that control the movement of vehicles. A simulation experiment was designed by Shao. In the environment of possible conflict, by optimizing the driving strategy of vehicles, and studying the influence of sight distance conditions and warning conditions on collision avoidance of drivers, a collision avoidance model in no control intersection was designed. The results indicate that the safety of vehicles can be improved by their model, but more complicated situation of straight and left turn hybrid driving is not considered. A distributed and collective intelligence framework is proposed by Kalantari et al.9 to provide navigation for vehicles in intersection. Through the simulation of intersection, it is proved that the proposed framework can reduce collisions and vehicle travel time at the same time. However, the paper did not analyze the optimization effect of control methods under different market penetration. To solve the problem of vehicle collision at the unsignalized intersection, an improved reinforcement learning method was proposed by Isele et al.10 It is verified that the proposed method can effectively ensure the safety of vehicles at unsignalized intersection through simulation experiment. However, the situation of fully autonomous vehicle is only considered through the research.

A two-car collision judgment model based on time to collision (TTC) is established by Duan et al.11 to reduce collisions within the range of unsignalized intersection. The current state of the vehicle can be adjusted by the proposed model through estimating the possible collision area in advance. With the results of simulation, it is shown that the model can reduce the collisions effectively.

Wu et al.12 have proposed a multi-agent learning method decentralized coordination learning of autonomous intersection management (DCL-AIM) to minimize the intersection delay under the condition of no collision constraints, which was verified by simulation experiments. From the experimental results, it is shown that the proposed method can reduce vehicle delay effectively. The safety constraints are considered to avoid the vehicle collision. In the paper by Li et al.,13 two kinds of unsignalized intersection control methods, namely priority-based method and discrete forward rolling optimal control (DFROC) method, are proposed through coordinating the driving behavior of vehicles within the control range of intersection. The safe driving of vehicles was ensured, and the traffic efficiency of vehicles was improved. It is found that two kinds of unsignalized intersection control methods can reduce vehicle delay through simulation.

In addition, to reduce vehicle collision at unsignalized intersections, a lot of methods have been proposed. Xu et al.14 proposed a distributed no collision cooperation method, which coordinated the driving strategies of multiple vehicles at the same time to realize the vehicle no collision passing. Through numerical simulation experiments, it was found that the proposed method can realize vehicles passing through the intersection without any collision. Moreau et al.15 proposed a Bayes curve optimization method. The problem of avoiding collision with obstacles was transformed into an optimization problem. Lagrange method and gradient method were used to solve the problem. It was shown that the method can reduce the occurrence of vehicle unsafe behavior. However, for unsignalized intersections, in most of the studies, avoiding vehicle collision was considered as the optimization objective, and collision avoidance constraints were set as the constraints in the model. In addition, in the current research, the vehicle collision optimization problem was considered in the case of fully CAVs. However, the mixed driving of AV and HV was considered rarely.

In conclusion, the weakness of current research is as follows:

1. Most of the current research in unsignalized intersection control problem has a limitation. In simulation experiments, only the situation that all vehicles are combined with CAV is considered while the situation with mixed AV and HV is not considered often. However, the latter situation might be the main traffic state in the future.

2. Vehicle efficiency, which is described as the total throughput of vehicles in an intersection within a given period of time, can be regarded as a key evaluation indicator for unsignalized intersections control. However, in much of the research, fuel consumption and exhaust emission were considered as the optimization objectives, while the vehicle efficiency is considered relatively less. In addition, setting of safety constraints in the optimization model or simulation model is adopted by most researchers to improve the safety of vehicles, while directly using real-time coordination of vehicle driving behavior to achieve collision avoidance is relatively lacking.

In view of the above problems, aiming at the control problem of unsignalized intersection, this article proposes a reinforcement learning model to study the control problem of unsignalized intersection. Therefore, the methods of DRL will be reviewed below.

**DRL**

Reinforcement learning16 is favored by researchers because of the advantages of agent autonomous learning. To solve practical problems more effectively, the
advantages of deep learning, such as priority in solving the multidimensional state space and multidimensional action space decision problem, are fully utilized. The application of reinforcement learning has been developed to a new stage, which is known as DRL. Through years of study by researchers, a relatively mature framework of DRL has been proposed, with research findings in many fields, such as robotics, traffic control, automatic driving, and so on.

In view of different characteristics of actual problems, various improvements in DRL have been proposed by researchers to solve practical problems more efficiently. In addition, when solving practical problems, it is necessary to consider the specific situation and choose the most appropriate method. DRL methods can be divided into four categories: (1) Value function-based DRL (VF-DRL) method, (2) Policy gradient-based DRL (PG-DRL) method, (3) Value function and policy gradient-based DRL (VFPG-DRL) method, and (4) Multi-agent DRL (MADRL) method. Since this article is mainly based on DQN, a classical value function-based reinforcement learning method, the VF-DRL method will be highlighted among the four methods, while the other situations will be briefly described.

Q-learning method is one of the VF-DRL methods. The agent interacts with the environment, stores the historical data in the Q-table, and updates the Q-table to learn the optimal strategy. In addition, Q-learning can be applied to practical problems. For instance, to optimize the traffic flow of highways, the problem was described as a Markov problem in Walraven et al., and Q-learning was used to explore the optimal strategy, that is, the maximum driving speed on the expressway. Meanwhile, the effect of the method was verified through simulation. The outcomes of simulation show that the strategy learned by Q-learning can greatly reduce traffic congestion in high traffic demand. However, one weakness of Q-learning is that a Q-table related to state and action is necessary to be established, and the size of state and action space depends on the computer memory, which may affect the calculation speed of the algorithm.

To further develop the reinforcement learning method, in 2013, the deep learning method was combined with the reinforcement learning method by Google deep mind team, and the DRL structure was proposed, which is named as deep Q network (DQN). In 2015, the improved DQN algorithm was proposed and successfully published in Nature magazine. DQN has also obtained some achievements in other practical problems. To solve the problem of obstacle avoidance of underactuated unmanned ships under unknown environment interference, the obstacle avoidance algorithm was designed by Cheng and Zhang with DQN structure. The researchers found that the method based on reinforcement learning can avoid obstacles more accurately. Certainly, there are some problems of DQN method in application, such as overestimation and other weaknesses. Therefore, some improved methods for DQN were proposed by Van et al., which can alleviate the overestimation problem of DQN method. The concept of advantage function was introduced by Wang et al. in DQN, and the competitive DQN was proposed. A priority based DQN was designed by Schaul et al. for priority sampling of important experience. After these studies, DQN series methods are gradually applied to practical problems. Based on the actual driving data constructed, the bidirectional DQN method is applied by Zhang et al. to control the vehicle speed. The experimental results are compared with the traditional DQN, which shows that the accuracy of the estimation value and the quality of the strategy are greatly improved. The real-time GPS data is used by Zeng et al. to replace the neural network in DQN with recurrent neural network. A new deep recurrent Q network (DRQN) is designed and applied to solve traffic light control problem at unsignalized intersection. The simulation shows that the effect of DRQN is better than DQN. Based on the above studies, it is clear that there is an excellent performance of VF-DRL method on discrete action space problems. However, many of the practical problems are the problems with continuous action and state space, which are high dimensional space problems. Therefore, VF-DRL has some limitations in solving more extensive practical problems.

In addition to the VF-DRL method, PG-DRL, VFPG-DRL, and MADRL methods also exist. These methods may outperform VF-DRL in some specific scenarios. PG-DRL method can be used to solve the continuous action space problem, and the convergence speed is relatively fast. However, the method is easy to converge to the local optimal solution and difficult to evaluate the policy. VFPG-DRL, a DRL method based on value function and strategy gradient, combines the advantages of VF-DRL and PG-DRL methods. However, there are some problems, such as low sampling efficiency and overestimation of evaluation network. Therefore, in practical applications, it is necessary to regulate and improve the algorithm according to the actual problem. In MADRL method, the agent can obtain the state information and reward information from the environment based on interactions among the agents and environment, and adjust
the strategy of the agents based on the acquired information, thus learning the optimal strategy.

Problem formulation

Through the V2I environment, the information of the vehicles can be obtained by the Road side unit (RSU) within the control range in real time, and the information is transmitted to the control center. The current driving strategy of the CAV is calculated by the control center according to the vehicle information to avoid collision with other vehicles and accelerate the vehicle traffic efficiency. The yellow dot denotes the collision point where the vehicle may have a collision. In addition, in order to describe the state of vehicles within the control range more accurately, this article discretizes the road network of unsignalized intersections at the interval of length $M$, and Figure 1(b) shows the discrete diagram of the road network.

In Figure 1, Agent 1, 2, 3, and 4 were used to obtain optimal driving strategies for vehicles in the straight and left-turn directions, respectively. For example, Agent 1 controls the vehicles going straight from Lane 2 to Lane 17 and Lane 18, and the vehicles turning left from Lane 1 to Lane 20. In Figure 1(a), at the intersection, left turn and through vehicles are more likely to collide, that is, there are more collision points. In order to focus on such complex traffic environment, right turn is not considered in this article. However, the possible collision situations with right turning vehicles in other directions may also be considered. As long as the through vehicle and right turning vehicle are designed in the same lane and the model proposed in this article is trained, the situation including right turning vehicles can be obtained.

This article is based on the following basic assumptions:

1. Within the control range of unsignalized intersection, vehicles are not allowed to overtake or change lanes.
2. In the control range of unsignalized intersection, when receiving the control information, the vehicle drivers are completely subject to the control command of the controller.
3. Within the control range of unsignalized intersection, the delay of information communication between vehicles and road, vehicles and vehicles are acceptable, and there is no packet loss.

Problem description

State space. In this article, the position and speed of vehicles in the control range of unsignalized intersection are considered in the state space, and the state is determined according to the road network discrete diagram in Figure 1

$$s_i = \begin{bmatrix} L_i \\ V_i \end{bmatrix}$$ (1)
The principle of the reward function is as follows. In order to reduce the number of collisions of control vehicles at unsignalized intersections, if the control vehicles collide, a large penalty value will be given, which is related to the speed of collision. When the vehicle passes through the unsignalized intersection successfully, a larger positive reward value is given, which is also related to the speed of the vehicle passing through the detection point. In other cases, in order to make the vehicles stay in the control range of unsignalized intersection, when the control vehicle speed is less than the threshold, a smaller reward can be given. Based on the reward function, it can avoid collision and ensure the vehicles to pass through the intersection as soon as possible.

\[ r_t = \begin{cases} 
-\tau_c \phi_1 & \text{Col} \\
\tau_v/\phi_2 & \text{Pass} \\
\psi_1 \tau_v & \tau_v \geq \delta \\
-\psi_1 \tau_v & \tau_v < \delta
\end{cases} \]
Improved double dueling deep Q network

3DQN-CNN-LSTM

Figure 2 shows the structure diagram of the improved double dueling deep Q network (3DQN-CNN-LSTM) model proposed in this article, which is different from the structure diagram of the double dueling deep Q network (3DQN) in the work by Gong et al.30 In this article, CNN are used to extract important features. Meanwhile, LSTM networks are used to select the best strategy combined with memory information and current information. In addition, the positive and negative reward experience buffer pool method,26 multi-step reward, and penalty method15 are used in this method to speed up convergence.

Multi-step reward and penalty

The less experience of vehicle collision and vehicle passing leads to the decrease of learning efficiency of the algorithm. Therefore, the article proposes a multi-step reward and penalty method based on the error correction method of failure experience proposed by Zhang et al.28 The experience of collision or possible collision, and the experience of vehicle successfully passing or promoting vehicle passing are mainly increased through the method. The main idea is that if the vehicle collides or successfully passes the intersection at time \( t \), adjust the immediate reward value at time \( t - 1 \), time \( t - 2 \), until time \( t - d \), where \( d \) is the first \( d \) step and \( d = \{1, \ldots, D \} \). \( D \) is the predefined maximum step. The specific adjustment method is that if there is a collision at time \( t \), the experience penalty at time \( [t - d \ldots t] \) will be given; if the vehicle passes the intersection successfully at time \( t \), the accumulated reward at time \( [t - d \ldots t] \) will be given. The corresponding reward and penalty are calculated according to equation (6). In equation (6), the immediate reward at \( t - d \) is \( r_{t-d} \) and the discount factor is \( \omega_1 \) and \( \omega_2 \).

\[
r_{t-d} = \begin{cases} 
    \omega_1 \times r_t & \text{if collision in time } t (6a) \\
    \omega_2 \times r_t & \text{if passing in time } t (6b)
\end{cases}
\]

However, in some cases, in order to prevent the possibility of collision, the behavior of deceleration or braking is taken by the agent, which is not supposed to be punished. Therefore, the algorithm is designed to determine whether the agent takes the behavior of deceleration or braking. If the agent takes actions to avoid collision, equation (6) is not used to calculate the reward value from time \( t - d \) to time \( t \). Otherwise, if the vehicle does not take any beneficial action to avoid collision, equation (6) is used to calculate the reward value from time \( t - d \) to time \( t \).

According to equation (6), the reward value at time \( t - d \), that is, \( r_{t-d} \), is calculated according to the reward value at time \( t \), that is, \( r_t \). If the vehicle collides or the vehicle successfully passes the intersection, the reward value at time \( t - d \), \( r_{t-d} \), is calculated according to the reward value calculated by equation (6). Otherwise, the reward value \( r_{t-d} \) is directly calculated according to equation (5).

Positive and negative sampling experience pool

At present, a large number of researchers have applied Deep Deterministic Policy Gradient (DDPG) to solve many problems of continuous action space.31 However, in the application of DDPG algorithm, there are still some problems. For example, when sampling in the experience playback buffer pool, the selected historical experience is randomly selected. Therefore, it is difficult to balance the proportion of positive reward and negative reward experience, which leads to the poor stability of the algorithm.

In the original DDPG,32 the experience replays buffer (ERB) is mixed with positive and negative experiences. In order to solve the problem of vehicle delay caused by parking within the control range of unsignalized intersection, the historical experience is divided into positive reward experience and negative reward
experience, the positive and negative experience are stored in the PNRERB, respectively, in this article. Similar to the original DDPG method, the sizes of the two cache pools are initialized first in this article, and the historical experience are replaced with the new one after the cache pool is full of experience.

In addition, small batch learning method is used to train DRL network in this article. Therefore, let the agent interact with the environment, collect a certain amount of historical experience, and then extract experience from the cache pool to train the neural network. In order to improve the sampling performance of the algorithm, the experience pool is divided into two experience pools, which are the experience pool with reward value greater than 0, that is, positive reward experience pool, and the experience pool with reward value less than or equal to 0, that is, negative reward experience pool, respectively. The sampling ratio from positive and negative experience pool is set as 3:1.

Central control method and multi-agent control method based on DRL

At present, intersection control can be divided into central control and distributed control. Based on the idea of these two control methods, this article proposes a central control method and a multi-agent control method for unsignalized intersection based on improved DRL. In these methods, the implementation of multi-agent control method is based on the distributed control of multi-agent reinforcement learning method. The structure diagram and pseudo code of these two control methods are introduced in the following section.

Central control method. The central control method is to set up an agent in the central control system, and the agent controls all the CAV in the intersection control area, as shown in Figure 3. Table 1 gives the pseudo code of the central control method.

Multi-agent control method. Multi-agent control method is mainly to set up multiple control agents in the central control system, which will control all the CAV in their respective control areas of the intersection.

In this article, the CAV of the four entrances in southeast and northwest directions are regarded as the control vehicles of the four agents. Each agent controls the whole process from the CAV entering the control area to the vehicles leaving the control area.

Figure 4 shows the interaction structure diagram of multi-agent control with the environment. From Figure 4, it can be seen that multiple agents interact with the VISSIM environment separately. Taking the four agents in this article as an example, each agent obtains the state information of its own control direction within the control range of the intersection from the VISSIM environment, and unites with the states of the other three directions to constitute the combined state. Each agent makes the action of the current control direction of the CAV according to the joint state. Finally, by sending its action to the VISSIM environment, each agent gets the immediate reward under the current state-action and the state of the next moment.

Table 2 shows the pseudo code for the multi-agent control method.

In this article, the main difference between Algorithms 1 and 2 mainly lies in the Lines 14 to 16 of the pseudo code. It can be seen from Tables 1 and 2 that the central control method in the pseudo code Line 14 mainly uses the central control agent to select the corresponding action according to the state in time \( t \). Meanwhile, in the multi-agent control method, in Line 14 of the pseudo code, N agents select the corresponding action according to the state at time \( t \). In the multi-agent control method, multiple agents work in a distributed way. For agent \( i \), if there is a connected vehicle in its control area, agent \( i \) will be activated and the action at time \( t \) will be selected according to the state at time \( t \). Otherwise, the input of agent \( i \) will be empty.

Experiment

The optimization effect of different DRL methods has been verified through the simulation experiment of vehicle control at unsignalized intersection, which is presented in this section. First, the simulation platform...
and parameters of the unsignalized intersection are described. Second, the control methods based on DRL are discussed. Specifically, the structure and parameter setting of each method is explained. Third, the scheme in this article is outlined and finally, the result based on the DRL control method is analyzed.

Simulation platform and parameter setting
A virtual road network is built in the experiment under the VISSIM simulation environment. The road network structure is shown in Figure 5. The cycle length in the signal control intersection is 80 s, the green time of east–west and north–south straight line is 37 s, the yellow time is 3 s, the green time of east–west and north–south straight line is 22 s and the green time of left-turning is 8 s, and the yellow time is 3 s.

The differences between AV and HV are mainly explained from the following two aspects: (1) In terms of the concept defined, AV refers to the intelligent connected vehicle. HV refers to traditional manned vehicle. AV mainly simulates the actual intelligent vehicle behavior, and HV mainly simulates the actual human driving vehicle behavior; (2) In terms of the driving strategy, the optimal driving strategy of AV are obtained according to the DRL algorithm proposed in this article, while the driving strategy of HV are obtained according to the built-in model of VISSIM simulation system. HVs are used to simulate the actual driving behavior of human beings. It receives the driving strategy of the built-in model of VISSIM simulation system and does not receive the control of DRL agent. DRL agent is mainly used to control AVs driving behavior.

The simulation interval in VISSIM is 1 s, that is, 1 frame/1 simulation step. This article mainly adopts VISSIM and Python interactive simulation. The DRL algorithm is realized in Python environment. At time $t$, the DRL algorithm obtains the simulation data from VISSIM, calculates the optimal driving strategy, and then provides the optimal driving strategy to VISSIM. The connected vehicle (CV) in VISSIM obtains the driving strategy and continues the simulation at the next time.

The results of the two classes of methods are compared in this section. One class is mainly based on DQN methods, which contains Deep Q Network method (DQN-NN),26 Deep Q Network method based on CNN (DQN-CNN),27 Double Deep Q Network

| Table 1. The central control method. |
|-------------------------------------|
| Algorithm 1 Central Control Method  |
| 1: Initialize the positive reward experience playback buffer pool $R^+$, the negative reward experience playback buffer pool $R^-$, the number of episodes $M$. |
| 2: Initialization action value function $Q$, parameter $\theta$. |
| 3: Initialization target action value function $Q'$, parameter $\theta'$, initialization threshold $\sigma$. |
| 4: for episodes $= 1, M$, do |
| 5: while True |
| 6: if CAV does not exist in the current road network |
| 7: if the current time reaches the maximum cycle time $T$, then end if |
| 8: if the entrance detector detects the entry of a vehicle, then add the vehicle number |
| 9: if the exit detector detects a vehicle leaving, then remove the vehicle number |
| 10: Else if CAV existed in the road network, then go to Line 11 |
| 11: End, run VISSIM simulation for single step, then go to Line 5, |
| 12: Store the current status $S_t$ of each CAV |
| 13: while True |
| 14: Execute $a_t$, calculate $r_t$ and $S_{t+1}$ based on equation (5) |
| 15: If the historical experience of the controlled vehicles is $d$, |
| 16: if vehicle collision happens or pass the intersection at time step $t$, calculate $r_{t,d}$ according to equation (5), store $r_{t,d}$ in $R^+$ and $R^-$. |
| 17: else store the $r_{t,d}$ in experience pool |
| 18: Select batch of historical memory from memory pool randomly |
| 19: Set $y_t = r_t + \gamma Q'(s', \text{arg max}_Q Q(s', a))$ not the end of episode |
| 20: Set $L = (r + \gamma Q'(s', \text{arg max}_Q Q(s', a)) - Q(s, a))^2$ |
| 21: Update gradient $\nabla L = E_{a,s,r,s'}[(r + \gamma Q'(s', \text{arg max}_Q Q(s', a)) - Q(s, a))\nabla Q(s, a)]$ |
| 22: end for |


method based on CNN (Double-DQN-CNN), 29 and Dueling Deep Q Network method based on CNN (Dueling-DQN-CNN). 27 The other class is mainly based on 3DQN method, which contains Double Dueling Deep Q Network method based on CNN (3DQN-CNN), 14 Multi Agent of Double Dueling Deep Q Network method based on CNN (multi-3DQN-CNN), 14 and the two methods proposed in this article, that is, Double Dueling Deep Q Network method based on CNN and LSTM (3DQN-CNN-LSTM), and Multi Agent of Double Dueling Deep Q Network method based on CNN and LSTM (multi-3DQN-CNN-LSTM). The network structure of multi-3DQN-CNN-LSTM method used in the article is the same as 3DQN-CNN-LSTM. In addition, multi-agent method is also adopted in multi-3DQN-CNN-LSTM.

In order to verify the optimization result of the vehicle performance index of the proposed method for the actual intersection, the mixed driving of straight vehicles, and left turning vehicles in the unsignalized intersection is considered in this article. The experimental scheme is given in Table 3.

In the scheme, there are straight vehicles and left turning vehicles driving in the unsignalized intersection. Table 3 shows the distribution of traffic flow at each entrance lane of the scheme. When the vehicles in the unsignalized intersection are straight vehicles and left turning vehicles, the total flow is 800 vehicles/h, 1680 vehicles/h, and 2560 vehicles/h, respectively. The first column in Table 3 is the total flow. The second to ninth columns show the flow of each inlet. According to Figure 1, Inlets 1, 4, 7, and 10 are left turn lanes, and Inlets 2, 5, 8, and 11 are straight lanes.

According to Figure 1, Inlets 3, 6, 9, and 12 are reserved lanes for these right-turning vehicles. Meanwhile, Lanes 13 to 20 are outlets, and these lanes are controlled by the agents in the corresponding directions. In the simulated environment of the intersection, there are three inlets but only two outlets in each direction.

**Table 2.** Multi-agent control method.

**Algorithm 2 multi-agent control method**

1: Initialize the positive reward experience playback buffer pool $R^+_t$, the negative reward experience playback buffer pool $R^-_t$, set i as the number of agents, set the number of episodes as $M$, set the number of agents as $N$.
2: Initialization action value function $Q_t$, parameter $\theta_t$.
3: Initialization target action value function $Q^{-}_t$, parameter $\theta^{-}_t$, initialization threshold $\sigma$.
4: for episodes = 1, M, do
5:     while True do
6:         If no CAV existed in the current road network then end if
7:         if the current time reaches the maximum cycle time T then end if
8:         if the entrance detector detects the entry of a vehicle, then add the vehicle number
9:         if the exit detector detects a vehicle leaving, then remove the vehicle number
10:        if CAV exist in the network, go to Line 11, else run VISSIM simulation for single step, and go to Line 5.
11:       if CAV exist in the network.
12:       Obtain the current status $s_i$, if $\sigma$.
13:       if random probability less than $\sigma$, choose action $a_i$ randomly; else select action with $a_i = \max_a Q^i(s_i, a; \theta^i)$
14:       Execute action $[a_i]_i$, obtain the reward $r_i$ and the states $[s_i+1]_i$. if the historical experience of the control agent is $d$
15:       if the CAV collide or pass the intersection safely in time $t$.
16:       Calculate the reward value based on equation (6): $r_{t-d} = \omega_1 r_{t-d}$ if collison in time $t$
17:       if passing in time $t$.
18:       Store the positive reward and the negative reward into experience pool $s_i$, $a_i$, $Q_i$, $s_{i+1}$, $r_i$, $\max_{a_i} Q_i(s_i, a; \theta_i)$.
19:       for agent = 1, N, do
20:       calculate $y_i = \begin{cases} r_i + \gamma Q_{i+1}(s_{i+1}, \max_a Q(s_{i+1}, a; \theta_i)) & \text{not the end of episode} \\ y_{i+1} + \gamma Q_1(s_{i+1}, \max_a Q(s_{i+1}, a; \theta_i)) & \text{end of episode} \end{cases}$
21:       calculate loss value $L' = (r_i + \gamma Q_{i+1}(s_{i+1}, \max_a Q(s_{i+1}, a; \theta_i)) - Q(s_i, a_i))^2$
22:       calculate the gradient $\nabla L' = E_{s, a, r, s'} (r_i + \gamma Q_{i+1}(s_{i+1}, \max_a Q(s_{i+1}, a; \theta_i)) - Q(s_i, a_i)) \nabla \theta_i Q(s_i, a_i)$
23:       end for
24:       end for
Experimental result

In this article, the success rate of vehicles successfully passing through the intersection\(^{12}\) is taken as the evaluation index to test the effect of the method based on DRL. The calculation formula of vehicle success rate is given in equation (7)

\[
R = 1 - \frac{CV}{TF} \tag{7}
\]

where \(R\) denotes the success rate, \(CV\) denotes the number of vehicle collisions within a certain simulation time, and \(TF\) denotes the total traffic issued within a certain time.

The calculation formula of average travel time (ATT) is

\[
B_{\text{Average Travel Time}} = \frac{ATT_s - ATT_{RL}}{ATT_s} \times 100\% \tag{8}
\]

where \(B_{\text{Average Travel Time}}\) denotes the ATT gain, \(ATT_s\) denotes the ATT of the vehicle under the signal control method, \(ATT_{RL}\) denotes the ATT of the vehicle under the DRL method.

The two key evaluation metrics in this article are the average vehicle travel time and the total throughput of the vehicle. Equation (8) shows the correlation between these two metrics

\[
k_{ATT} = \frac{1}{n} \sum_{k=0}^{n} TT_k \tag{9}
\]

Equation (9) represents the ATT of all the vehicles, from entering the control area to leaving the control area of the intersection. Here, \(k_{ATT}\) represents ATT, \(TT_k\) represents the travel time of the \(k_{th}\) vehicle, and \(n\) indicates the total throughput of the intersection control area during the total simulation time.
In the following section, the success rate, ATT, and vehicle trajectory experimental results of vehicles passing through the intersection under this scheme will be analyzed, respectively.

The success rate. Figure 6 shows the success rate of vehicles passing through the intersection when the total flow of the intersection is 800 vehicles/h based on DQN method and 3DQN method. It can be seen from Figure 6 that under the five permeability levels (20%, 40%, 60%, 80%, and 100%), the success rate based on DQN method and 3DQN method is greater than that when the permeability is 0%, and when the permeability is 100%, the success rate based on 3DQN method can be higher than that based on DQN method. This is also shown at the intersection of left turn and straight traffic. In addition, in the 3DQN-based method, under the same permeability, the success rates of 3DQN-CNN-LSTM method and 3DQN-CNN method are higher than those of multi-3DQN-CNN-LSTM method and multi-3DQN-CNN method, respectively, indicating that the optimization effect of central control method is better than that of multi-agent control method.

The percentage of success rate under two methods when traffic flow is equal to 800 vehicles/h, 1680 vehicles/h, and 2560 vehicles/h is shown as Table 4. The values highlighted in gray in Table 4 indicate the optimal values under the same penetration rate. In Table 4, M1 is DQN-NN method, M2 is DQN-CNN method, M3 is Double-DQN-CNN method, M4 is Dueling-DQN-CNN method, M5 is 3DQN-CNN method, M6 is multi-3DQN-CNN method, M7 is 3DQN-CNN-LSTM method, M8 is multi-3DQN-LSTM method, and MPR is market penetration rate. According to Table 4, under three different traffic flow, the percentage of success is more than 70% based on DQN method and 3DQN method. However, under the same traffic flow, the percentage of success based on the two methods is rising gradually with the increase of penetration rate. However, under the same penetration rate, the percentage of success is decreasing with the increase of traffic flow. Therefore, comprehensively, under three different traffic flows, the percentage of success can be optimized effectively based on DQN method and 3DQN method. Within the same traffic flow, the higher the penetration rate, the better the optimization effect of two methods, and within the same penetration rate, the greater the traffic flow, the worse the optimization effect of the two methods. Within the same traffic flow and penetration rate, the percentage of success of 3DQN method is higher than DQN method. Based on the DQN method, the percentage of success of DQN-NN method, DQN-CNN method, Double-DQN-CNN method, and DQN-LSTM method is lower than 70%.
method, and Dueling-DQN-CNN is similar. Within the method based on DQN, there is a little difference between 3DQN-CNN method, multi-3DQN-CNN method, 3DQN-CNN-LSTM method, and multi-3DQN-LSTM method when the traffic flow is 800 vehicles/h and 1680 vehicles/h, respectively. Under the same penetration rate, the percentage of success of 3DQN-CNN-LSTM and multi-3DQN-LSTM are higher than 3DQN-CNN and multi-3DQN-CNN, respectively, when the traffic flow is equal to 2560 vehicles/h. In addition, under three different traffic flow and the same penetration, the optimization effect of 3DQN-CNN-LSTM is the best (within the method based on 3DQN, the highest percentage of success is obtained by 3DQN-CNN-LSTM method). However, under the same penetration rate and traffic flow, within the method based on DQN, the percentage of success of 3DQN-CNN-LSTM method and 3DQN-CNN method are higher than multi-3DQN-LSTM method and multi-3DQN-CNN method, respectively. In a summary, the percentage of success of 3DQN-CNN-LSTM is the highest, which means there is a better performance of central control method than the multi-agent control method in the optimization of percentage of success. Therefore, the 3DQN-CNN-LSTM method proposed in this article is the best for optimizing.

**ATT.** Table 5 shows the ATT of vehicles and the number of vehicles successfully passing through the intersection obtained based on DQN-based method and 3DQN-based method when the total flow is 2560 vehicles/h. The values in bold font in Table 5 represent the optimal values under the same permeability. It can be seen from Table 5 that the ATT obtained by DQN-based method is less than that obtained by 3DQN-based method, but the number of vehicles successfully passing through the intersection obtained by 3DQN-based method is greater than that obtained by DQN-based method. The signal control method has the highest total throughput at 0% penetration. However, its ATT is about 3 times that of the other methods. It can be considered that the signal control method has the highest security, but its combined efficiency is not as good as the other reinforcement learning methods. When the penetration rate is less than or equal to 100%, the number of vehicles passing through the intersection successfully obtained by DQN-NN method increases very little, perhaps because
DQN-NN method cannot correctly judge the driving strategies of other vehicles, resulting in wrong driving decisions. When the penetration rate is 100%, the number of vehicles successfully passing through the intersection under 3DQN-NN-LSTM method is more than that of DQN method, 3DQN-CNN method, multi-3DQN-CNN method, and multi-3DQN-CNN-LSTM method. This shows that even when the traffic flow is large, 3DQN-NN-LSTM method can still ensure the safe and rapid passage of vehicles. At the same time, it also shows that the optimization effect of central control method is better than that of multi-agent control method.

Figure 7 shows the ATT of vehicles obtained by DQN-based method and 3DQN-based method with a total flow of 2560 vehicles/h. It can be seen from Figure 7 that (1) The ATT under signal control method is nearly 3 times larger compared to other DQN-based method and 3DQN-based method when the permeability is larger than 0%; (2) The ATT under 3DQN-CNN method and multi-3DQN-CNN method is higher than that of DQN method, 3DQN-CNN-LSTM method, and multi-3DQN-CNN-LSTM method.

To further verify the experimental results, different permeability environments with traffic flow of 800 vehicles/h and 1680 vehicles/h are also investigated in this article, and the experimental results are shown in Tables 6 and 7, respectively. According to Table 6, the total throughput under the 3DQN-CNN-LSTM method gradually increases with the increase of penetration rate, but is always lower than the total throughput under signal control method. According to Table 7, the total throughput of the 3DQN-CNN-LSTM method is higher than the total throughput under signal control at a penetration rate of 100%. In addition, the ATT required for the model with DRL method is consistently smaller than that of the signal-controlled approach. Therefore, it can be concluded that the 3DQN-CNN-LSTM method proposed in this article will outperform the signal control method under each evaluation indicator in the case of high penetration rate and high throughput.

Table 5. Number of vehicles successfully passing through the intersection when the flow is 2560 vehicles/h.

| Penetration rate | Method          | Average travel time (s) | Total throughput |
|-----------------|-----------------|-------------------------|-----------------|
| 0%              | Signal control  | 36.6                    | 2542            |
| 20%             | DQN-NN          | 10.6                    | 2033            |
|                 | DQN-CNN         | 10.6                    | 2073            |
|                 | Double-DQN-CNN  | 10.5                    | 2071            |
|                 | Dueling-DQN-CNN | 10.5                    | 2066            |
|                 | 3DQN-CNN        | 10.8                    | 2104            |
|                 | Multi-3DQN-CNN  | 10.7                    | 2086            |
|                 | 3DQN-CNN-LSTM   | 10.7                    | 2139            |
| 40%             | DQN-NN          | 10.6                    | 2099            |
|                 | DQN-CNN         | 10.7                    | 2132            |
|                 | Double-DQN-CNN  | 10.6                    | 2122            |
|                 | Dueling-DQN-CNN | 10.7                    | 2133            |
|                 | 3DQN-CNN        | 11.9                    | 2152            |
|                 | Multi-3DQN-CNN  | 11.7                    | 2135            |
|                 | 3DQN-CNN-LSTM   | 11.0                    | 2235            |
|                 | Multi-3DQN-LSTM | 11.1                    | 2198            |
| 60%             | DQN-NN          | 11.3                    | 2111            |
|                 | DQN-CNN         | 11.5                    | 2135            |
|                 | Double-DQN-CNN  | 11.7                    | 2155            |
|                 | Dueling-DQN-CNN | 11.8                    | 2159            |
|                 | 3DQN-CNN        | 12.9                    | 2205            |
|                 | Multi-3DQN-CNN  | 12.7                    | 2198            |
|                 | 3DQN-CNN-LSTM   | 11.3                    | 2342            |
|                 | Multi-3DQN-LSTM | 11.5                    | 2299            |
| 80%             | DQN-NN          | 11.5                    | 2109            |
|                 | DQN-CNN         | 11.6                    | 2200            |
|                 | Double-DQN-CNN  | 11.8                    | 2229            |
|                 | Dueling-DQN-CNN | 11.9                    | 2267            |
|                 | 3DQN-CNN        | 13.6                    | 2331            |
|                 | Multi-3DQN-CNN  | 12.9                    | 2298            |
|                 | 3DQN-CNN-LSTM   | 11.6                    | 2465            |
|                 | Multi-3DQN-LSTM | 11.6                    | 2344            |
| 100%            | DQN-NN          | 11.8                    | 2169            |
|                 | DQN-CNN         | 11.8                    | 2268            |
|                 | Double-DQN-CNN  | 11.9                    | 2297            |
|                 | Dueling-DQN-CNN | 11.8                    | 2286            |
|                 | 3DQN-CNN        | 14.0                    | 2390            |
|                 | Multi-3DQN-CNN  | 13.8                    | 2386            |
|                 | 3DQN-CNN-LSTM   | 11.8                    | 2551            |
|                 | Multi-3DQN-LSTM | 11.6                    | 2446            |

DQN-NN: Deep Q Network method; CNN: convolutional neural networks; 3DQN: double dueling deep Q network; LSTM: long short-term memory.
success rate of 3DQN-CNN-LSTM method can reach 99%; (2) Compared with the signal control method, the ATT based on DQN method and 3DQN method is shorter. When the permeability is less than or equal to 80%, the number of vehicles successfully passing through the intersection based on DQN method and 3DQN method is less than that of the signal control method. When the permeability is 100%, the number of vehicles successfully passing through the intersection based on 3DQN method is close to that of signal control method, and the number of vehicles under 3DQN-CNN-LSTM method is higher than that of signal control method under some total flow (such as 2560 vehicles/h); (3) No matter the convergence of the algorithm or the reduction of the number of vehicle collisions, the optimization effects of 3DQN-CNN and 3DQN-CNN-LSTM methods are better than multi-3DQN-CNN and multi-3DQN-CNN-LSTM methods, respectively, which shows that the central control method proposed in this article is better than multi-agent control method.

In summary, the collision times and travel time of vehicles can be effectively reduced based on DQN method and 3DQN method. The 3DQN-CNN-LSTM method proposed in this article is the most excellent method among DQN-NN, DQN-CNN, double-DQN-CNN, dueling-DQN-CNN, 3DQN-CNN, multi-3DQN-CNN, 3DQN-CNN-LSTM, and multi-3DQN-CNN-LSTM methods, which can reduce the number of vehicle collisions effectively, and reduce the ATT of vehicles between 47% and 71% no matter under low traffic flow or high traffic flow.

The evaluation indicators in this article are mainly the ATT and the total throughput. In this article, the strengths and weaknesses of the model are analyzed mainly based on these two indicators. The signal control method has the highest total throughput when the penetration rate is 0%. However, its ATT is about 3 times higher than the other methods. Therefore, we believe that although the signal control method has the highest security, its overall efficiency is inferior to the 3DQN-CNN-LSTM method proposed in this article.

The goal of this article is to optimize the traffic efficiency on the premise of avoiding the collision of vehicles at the intersection. Therefore, if the vehicle collides, a greater penalty will be given. Therefore, in order to ensure that there is no collision, the vehicle may reduce the speed, thereby reducing the traffic efficiency. When the penetration rate is lower than 100%, in order not to collide with other vehicles, HV may not deliberately reduce the speed on the simulation road; however, CAV may accelerate or decelerate to reduce the collision. Therefore, in order to ensure safety, some traffic efficiency may be lost. However, when the penetration rate is set as 100%, all vehicles are connected vehicles, and all vehicles themselves will take the behavior of avoiding collision, so the traffic efficiency is further guaranteed in a relatively safe environment.

The success rate and ATT under complex traffic environment. The reason why the right turn movement is not considered in above discussion is that the purpose of this article is to focus on the unsignalized intersection control under the objective of reducing vehicle collision and delay as much as possible and considering the mixed traffic of HV and AV. In the two-way four lane unsignalized intersection environment tested in this article, the setting of collision points in the road network diagram of unsignalized intersection is shown in Figure 1. It can be seen from Figure 1 that the collision between motor vehicles mainly exists in the left turn and through traffic flow, and the right turn traffic flow has little impact on the traffic flow. In addition, this article does not consider the mixed traffic flow in the presence of both pedestrians and non-motor vehicles. Therefore, right turning vehicles may be ignored. However, if the mixed traffic flow with both pedestrians and non-motor vehicles is considered in the research, the right turn traffic flow must be considered.
We also analyzed and discussed the experimental results of the success rate of vehicles passing through the intersection and the ATT of vehicles under different traffic flows, comprehensively considering the through traffic flow, left turn traffic flow, and right turn traffic flow in the next part.

Figure 8 shows the success rate of vehicles passing through the intersection under the comprehensive consideration of straight, left turn, and right turn traffic flows. In Figure 8, the total flow is 2000 vehicles/h.

Table 6. Number of vehicles successfully passing through the intersection when the flow is 800 vehicles/h.

| Penetration rate | Method       | Average travel time (s) | Total throughput |
|-----------------|--------------|-------------------------|-----------------|
| 0%              | Signal control | 33.3                   | 768             |
| 20%             | DQN-NN       | 9.5                     | 739             |
|                 | DQN-CNN      | 9.6                     | 746             |
|                 | Double-DQN-CNN | 9.5                   | 744             |
|                 | Dueling-DQN-CNN | 9.7                   | 746             |
|                 | 3DQN-CNN     | 9.5                     | 746             |
|                 | Multi-3DQN-CNN | 9.6                   | 747             |
|                 | 3DQN-CNN-LSTM | 9.5                   | 756             |
|                 | Multi-3DQN-LSTM | 9.6           | 750             |
| 40%             | DQN-NN       | 9.6                     | 741             |
|                 | DQN-CNN      | 9.6                     | 739             |
|                 | Double-DQN-CNN | 9.6                   | 746             |
|                 | Dueling-DQN-CNN | 9.4                   | 741             |
|                 | 3DQN-CNN     | 9.6                     | 750             |
|                 | Multi-3DQN-CNN | 9.8                   | 751             |
|                 | 3DQN-CNN-LSTM | 9.7                   | 748             |
|                 | Multi-3DQN-LSTM | 9.8           | 751             |
| 60%             | DQN-NN       | 9.6                     | 734             |
|                 | DQN-CNN      | 9.7                     | 739             |
|                 | Double-DQN-CNN | 9.7                   | 746             |
|                 | Dueling-DQN-CNN | 9.8                   | 743             |
|                 | 3DQN-CNN     | 9.6                     | 754             |
|                 | Multi-3DQN-CNN | 9.8                   | 749             |
|                 | 3DQN-CNN-LSTM | 9.7                   | 760             |
|                 | Multi-3DQN-LSTM | 9.8           | 756             |
| 80%             | DQN-NN       | 9.7                     | 740             |
|                 | DQN-CNN      | 9.9                     | 746             |
|                 | Double-DQN-CNN | 9.8                   | 744             |
|                 | Dueling-DQN-CNN | 9.8                   | 740             |
|                 | 3DQN-CNN     | 9.7                     | 760             |
|                 | Multi-3DQN-CNN | 9.8                   | 759             |
|                 | 3DQN-CNN-LSTM | 9.9                   | 762             |
|                 | Multi-3DQN-LSTM | 9.8           | 761             |
| 100%            | DQN-NN       | 9.8                     | 742             |
|                 | DQN-CNN      | 9.9                     | 749             |
|                 | Double-DQN-CNN | 9.9                   | 754             |
|                 | Dueling-DQN-CNN | 10.0                  | 755             |
|                 | 3DQN-CNN     | 9.7                     | 764             |
|                 | Multi-3DQN-CNN | 9.8                   | 762             |
|                 | 3DQN-CNN-LSTM | 10.0                  | 764             |
|                 | Multi-3DQN-LSTM | 9.9             | 765             |

Table 7. Number of vehicles successfully passing through the intersection when the flow is 1680 vehicles/h.

| Penetration rate | Method       | Average travel time (s) | Total throughput |
|-----------------|--------------|-------------------------|-----------------|
| 0%              | Signal control | 34.4                   | 1695            |
| 20%             | DQN-NN       | 9.6                     | 1510            |
|                 | DQN-CNN      | 9.6                     | 1515            |
|                 | Double-DQN-CNN | 9.5                   | 1513            |
|                 | Dueling-DQN-CNN | 9.5                   | 1512            |
|                 | 3DQN-CNN     | 9.6                     | 1521            |
|                 | Multi-3DQN-CNN | 9.6                   | 1522            |
|                 | 3DQN-CNN-LSTM | 9.7                   | 1546            |
|                 | Multi-3DQN-LSTM | 9.6           | 1542            |
| 40%             | DQN-NN       | 9.7                     | 1522            |
|                 | DQN-CNN      | 9.6                     | 1530            |
|                 | Double-DQN-CNN | 9.5                   | 1525            |
|                 | Dueling-DQN-CNN | 9.6                   | 1527            |
|                 | 3DQN-CNN     | 9.8                     | 1554            |
|                 | Multi-3DQN-CNN | 9.8                   | 1550            |
|                 | 3DQN-CNN-LSTM | 10.2                  | 1586            |
|                 | Multi-3DQN-LSTM | 9.9           | 1578            |
| 60%             | DQN-NN       | 9.7                     | 1534            |
|                 | DQN-CNN      | 9.7                     | 1544            |
|                 | Double-DQN-CNN | 9.6                   | 1540            |
|                 | Dueling-DQN-CNN | 9.8                   | 1552            |
|                 | 3DQN-CNN     | 10.0                    | 1563            |
|                 | Multi-3DQN-CNN | 9.9                   | 1559            |
|                 | 3DQN-CNN-LSTM | 10.2                  | 1586            |
|                 | Multi-3DQN-LSTM | 10.0           | 1578            |
| 80%             | DQN-NN       | 9.9                     | 1554            |
|                 | DQN-CNN      | 9.8                     | 1591            |
|                 | Double-DQN-CNN | 9.7                   | 1589            |
|                 | Dueling-DQN-CNN | 9.6                   | 1588            |
|                 | 3DQN-CNN     | 9.6                     | 1587            |
|                 | Multi-3DQN-CNN | 10.1                  | 1589            |
|                 | 3DQN-CNN-LSTM | 10.0                  | 1659            |
|                 | Multi-3DQN-LSTM | 10.5           | 1613            |
| 100%            | DQN-NN       | 9.9                     | 1567            |
|                 | DQN-CNN      | 9.7                     | 1587            |
|                 | Double-DQN-CNN | 9.8                   | 1597            |
|                 | Dueling-DQN-CNN | 9.9                   | 1606            |
|                 | 3DQN-CNN     | 10.0                    | 1643            |
|                 | Multi-3DQN-CNN | 10.5                  | 1644            |
|                 | 3DQN-CNN-LSTM | 10.8                  | 1699            |
|                 | Multi-3DQN-LSTM | 10.8           | 1684            |

DQN-NN: Deep Q Network method; CNN: convolutional neural networks; 3DQN: double dueling deep Q network; LSTM: long short-term memory.

We also analyzed and discussed the experimental results of the success rate of vehicles passing through the intersection and the ATT of vehicles under different traffic flows, comprehensively considering the through traffic flow, left turn traffic flow, and right turn traffic flow in the next part.

Figure 8 shows the success rate of vehicles passing through the intersection under the comprehensive consideration of straight, left turn, and right turn traffic flows. In Figure 8, the total flow is 2000 vehicles/h.
the best results. When the penetration rate is 100%, the success rate can reach 99%, while 3DQN-CNN and multi-CNN-LSTM methods and 3DQN-CNN and multi-3DQN-LSTM are better than DQN-NN, DQN-CNN, double DQN-CNN, and dueling DQN-CNN, that is, they have a higher success rate.

Figure 9 shows the ATT of vehicles under the comprehensive consideration of straight, left turn, and right turn traffic flows. Table 8 shows the ATT of vehicles and the number of vehicles successfully passing through the intersection under the comprehensive consideration of straight, left turn, and right turn traffic flows. In Figure 9 and Table 3, the total flow is 2000 vehicles/h. Different methods, such as DQN-NN, DQN-CNN, double-DQN-CNN, dueling-DQN-CNN, 3DQN-CNN, multi-3DQN-CNN, 3DQN-CNN-LSTM, and multi-3DQN-CNN-LSTM are tested. It can be seen from Figure 9 and Table 8 that the ATT optimized by 3DQN-CNN and multi-3DQN-CNN is longer, indicating that although 3DQN-CNN and multi-3DQN-CNN can increase vehicle safety, they also increase vehicle travel time. Compared with the 3DQN-CNN-LSTM method and the multi-3DQN-CNN-LSTM method, the 3DQN-CNN method and the multi-3DQN-CNN method not only have a longer ATT but also have a relatively small number of vehicles successfully passing through the intersection.

**Vehicle trajectory.** The 3DQN-CNN-LSTM method is adopted as control method. Through the above analysis, it is clear that there is a better performance of 3DQN-CNN-LSTM method, thus the vehicle trajectory of 3DQN-CNN-LSTM method is mainly analyzed in this section. When the penetration is 0%, the trajectory is the vehicle trajectory obtained under the signal control method, in which the network CAV trajectory is represented by red line and the HV trajectory is represented by blue line. The trajectories in this section are those of vehicles passing through the intersection.
The trajectories of vehicles in collision are removed in this section.

The spatiotemporal trajectory of vehicles in Lane 5 at a total traffic flow of 1680 vehicles/h is shown as Figure 10. According to Figure 10(b)–(f), there is an increment in the vehicle trajectory with the increase of penetration. Compared with Figure 10(a), the vehicle trajectory in Figure 10(b)–(f) is smoother, which means that the optimization of the vehicle trajectory based on Table 8.

| Total flow (vehicles/h) | Penetration rate of CV | Method           | Average travel time (s) | Number of vehicles |
|-------------------------|-----------------------|------------------|-------------------------|--------------------|
|                         |                       |                  |                         |                    |
| 2000                    | 0%                    | Signal control   | 19.9                    | 2020               |
|                         | 20%                   | DQN-NN           | 9.3                     | 1526               |
|                         |                       | DQN-CNN          | 10.1                    | 1540               |
|                         |                       | Double-DQN-CNN   | 10.0                    | 1541               |
|                         |                       | Dueling-DQN-CNN  | 9.9                     | 1537               |
|                         |                       | 3DQN-CNN         | 12.4                    | 1633               |
|                         |                       | Multi-3DQN-CNN   | 12.6                    | 1629               |
|                         |                       | 3DQN-CNN-LSTM    | 9.7                     | 1652               |
|                         |                       | Multi-3DQN-LSTM  | 9.7                     | 1650               |
|                         | 40%                   | DQN-NN           | 9.3                     | 1591               |
|                         |                       | DQN-CNN          | 10.7                    | 1614               |
|                         |                       | Double-DQN-CNN   | 10.9                    | 1616               |
|                         |                       | Dueling-DQN-CNN  | 10.6                    | 1610               |
|                         |                       | 3DQN-CNN         | 15.2                    | 1749               |
|                         |                       | Multi-3DQN-CNN   | 14.9                    | 1742               |
|                         |                       | 3DQN-CNN-LSTM    | 10.3                    | 1764               |
|                         |                       | Multi-3DQN-LSTM  | 10.2                    | 1762               |
|                         | 60%                   | DQN-NN           | 9.0                     | 1552               |
|                         |                       | DQN-CNN          | 11.4                    | 1578               |
|                         |                       | Double-DQN-CNN   | 11.5                    | 1590               |
|                         |                       | Dueling-DQN-CNN  | 11.5                    | 1586               |
|                         |                       | 3DQN-CNN         | 16.2                    | 1794               |
|                         |                       | Multi-3DQN-CNN   | 16.1                    | 1795               |
|                         |                       | 3DQN-CNN-LSTM    | 10.5                    | 1900               |
|                         |                       | Multi-3DQN-LSTM  | 10.3                    | 1896               |
|                         | 80%                   | DQN-NN           | 8.9                     | 1538               |
|                         |                       | DQN-CNN          | 12.4                    | 1660               |
|                         |                       | Double-DQN-CNN   | 12.2                    | 1662               |
|                         |                       | Dueling-DQN-CNN  | 12.4                    | 1659               |
|                         |                       | 3DQN-CNN         | 15.5                    | 1810               |
|                         |                       | Multi-3DQN-CNN   | 15.4                    | 1805               |
|                         |                       | 3DQN-CNN-LSTM    | 10.0                    | 1972               |
|                         |                       | Multi-3DQN-LSTM  | 10.2                    | 1969               |
|                         | 100%                  | DQN-NN           | 8.6                     | 1561               |
|                         |                       | DQN-CNN          | 13.4                    | 1650               |
|                         |                       | Double-DQN-CNN   | 13.2                    | 1652               |
|                         |                       | Dueling-DQN-CNN  | 13.6                    | 1653               |
|                         |                       | 3DQN-CNN         | 14.5                    | 1833               |
|                         |                       | Multi-3DQN-CNN   | 14.7                    | 1830               |
|                         |                       | 3DQN-CNN-LSTM    | 10.3                    | 2020               |
|                         |                       | Multi-3DQN-LSTM  | 10.4                    | 2018               |

DQN-NN: Deep Q Network method; CNN: convolutional neural networks; 3DQN: double dueling deep Q network; LSTM: long short-term memory.

Figure 10. Vehicle trajectory of Lane 5 when the total flow at the intersection is 1680 vehicles/h: (a) penetration rate = 0%, (b) penetration rate = 20%, (c) penetration rate = 40%, (d) penetration rate = 60%, (e) penetration rate = 80%, and (f) penetration rate = 100%.
3DQN-CNN-LSTM method is excellent at the unsignalized intersection with through and left-turning vehicles.

As can be seen from Figure 10, compared with the signal control method, the vehicle trajectory optimized by 3DQN-CNN-LSTM method has less parking waiting, and most of them are relatively smoother. Therefore, 3DQN-CNN-LSTM method can effectively alleviate the phenomenon of vehicle parking queue. Under the same traffic flow, with the increase of the penetration of CAV, the vehicle trajectory gradually increases, indicating that with the increase of the penetration of CAV, the number of vehicle collisions is less.

Conclusion

In order to reduce vehicle collisions and improve traffic efficiency, the unsignalized intersection is researched and the methods based on DQN and 3DQN are designed to solve this problem. The results show that the percentage of success based on DQN method and 3DQN method increases with the increase of the penetration rate of CAV under the same traffic flow. In addition, under the same traffic flow and penetration rate, the optimization effect based on 3DQN method is better than that of the DQN method, and the success rate is up to 99%. At the same time, compared with the signal control method, the ATT of vehicles passing through the intersection based on DQN method and 3DQN method has been greatly reduced, and the ATT is in the range of 18%–72%. In addition, the optimization effects of 3DQN-CNN and 3DQN-CNN-LSTM methods perform better than those of multi-3DQN-CNN and multi-3DQN-CNN-LSTM methods in terms of vehicle collisions. It shows that the optimization effect of central control method is better than that of multi-agent control method.

An improved DRL method for vehicle control at urban road intersections has been proposed in this article. Although the methods proposed have some results for vehicle control problems, there is still room for improvement. For example, the vehicle type studied in this article is car, and the possibility of other vehicle types coexisting has not been considered. In the future, more vehicle types and effective MADRL will be considered.

Acknowledgements

The authors thank the reviewers for useful suggestions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors thank the National Natural Science Foundation of China (61104166).
14. Xu B, Li S, Bian Y, et al. Distributed conflict-free cooperation for multiple connected and automated vehicle CAV at unsignalized intersections. *Transp Res Part C Emerg Technol* 2018; 93: 322–334.

15. Moreau J, Melchior P, Victor S, et al. Reactive path planning in intersection for autonomous vehicle. *IFAC PapersOnLine* 2019; 52(5): 109–114.

16. Sutton RS, McAllester DA, Singh SP, et al. Policy gradient methods for reinforcement learning with function approximation. In: Solla S, Leen T and Müller K (eds) Advances in neural information processing systems. Cambridge, MA: MIT Press, 2000, pp.1057–1063.

17. Tenor M, Madrigal F, Antonio J, et al. Towards a common implementation of reinforcement learning for multiple robotic tasks. *Expert Syst Appl* 2018; 100: 246–259.

18. Wu Y, Yu Z, Li CY, et al. Reinforcement learning in dual-arm trajectory planning for a free-floating space robot. *Aerosp Sci Technol* 2020; 98: 105657.

19. Aslani M, Mesgari MS and Wiering M. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transp Res Part C Emerg Technol* 2017; 85: 732–752.

20. Guo S, Zhang X, Zheng Y, et al. An autonomous path planning model for unmanned ships based on deep reinforcement learning. *Sensors* 2020; 20(2): 426.

21. Walraven E, Spaan MTJ and Bakker B. Traffic flow optimization: a reinforcement learning approach. *Eng Appl Artif Intell* 2016; 52: 203–212.

22. Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning. *arXiv Preprint arXiv: 1312.5602*, 2013, https://arxiv.org/abs/1312.5602

23. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015; 518(7540): 529–533.

24. Cheng Y and Zhang W. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* 2018; 272: 63–73.

25. Van H, Guez A and Silver D. Deep reinforcement learning with double Q-learning. In: *Third AAAI conference on artificial intelligence*, Phoenix, AZ, 12–17 February 2016, pp.1813–1819. Palo Alto, CA: AAAI Press.

26. Wang Z, Freitas N and Lanctot M. Dueling network architectures for deep reinforcement learning. *PMLR* 2016; 48: 1995–2003.

27. Schaul T, Quan J, Antonogiou I, et al. Prioritized experience replay. In: *4th International conference on learning representations*, San Juan, Puerto Rico, 2–4 May 2016. La Jolla, CA: ICLR.

28. Zhang Y, Sun P, Yin Y, et al. Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning. In: *2018 IEEE intelligent vehicles symposium (IV)*, Changshu, China, 26–30 June 2018, pp.1251–1256. New York: IEEE.

29. Zeng J, Hu J and Zhang Y. Adaptive traffic signal control with deep recurrent Q-learning. In: *2018 IEEE intelligent vehicles symposium (IV)*, Changshu, China, 26–30 June 2018, pp.1215–1220. New York: IEEE.

30. Gong Y, Abdel-Aty M, Cai Q, et al. Decentralized network level adaptive signal control by multi-agent deep reinforcement learning. *Transp Res Interdiscip Perspect*, 2019; 1: 100020.

31. Zhu M, Wang X and Wang Y. Human-like autonomous car-following model with deep reinforcement learning. *Transp Res Part C Emerg Technol* 2018; 97: 348–368.

32. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. *Comput Sci 2015; 8(6): A187.*