electronics

MDPI

Article

Deep Reinforcement Learning for Intersection Signal Control Considering Pedestrian Behavior

Guangjie Han [1,2], Qi Zheng [1,*], Lyuchao Liao [1], Penghao Tang [1], Zhengrong Li [1] and Yintian Zhu [1]

[1] School of Transportation, Fujian University of Technology, Fuzhou 350118, China
[2] Department of Information and Communication System, Hohai University, Changzhou 213022, China
* Correspondence: zqi.hhxx@gmail.com

Abstract: Using deep reinforcement learning to solve traffic signal control problems is a research hotspot in the intelligent transportation field. Researchers have recently proposed various solutions based on deep reinforcement learning methods for intelligent transportation problems. However, most signal control optimization takes the maximization of traffic capacity as the optimization goal, ignoring the concerns of pedestrians at intersections. To address this issue, we propose a pedestrian-considered deep reinforcement learning traffic signal control method. The method combines a reinforcement learning network and traffic signal control strategy with traffic efficiency and safety aspects. At the same time, the waiting time of pedestrians and vehicles passing through the intersection is considered, and the Discrete Traffic State Encoding (DTSE) method is applied and improved to define the more comprehensive states and rewards. In the training of the neural network, the multi-process operation method is adopted, and multiple environments are run for training simultaneously to improve the model's training efficiency. Finally, extensive simulation experiments are conducted on actual intersection scenarios using the simulation software Simulation of Urban Mobility (SUMO). The results show that compared to Dueling DQN, the waiting time due to our method decreased by 58.76% and the number of people waiting decreased by 51.54%. The proposed method can reduce both the number of people waiting and the waiting time at intersections.

Keywords: traffic signal timing; deep reinforcement learning; pedestrian behavior

check for updates

Citation: Han, G.; Zheng, Q.; Liao, L.; Tang, P.; Li, Z.; Zhu, Y. Deep Reinforcement Learning for Intersection Signal Control Considering Pedestrian Behavior. *Electronics* **2022**, *11*, 3519. https://doi.org/10.3390/electronics11213519

Academic Editor: Amir Mosavi

Received: 7 October 2022
Accepted: 26 October 2022
Published: 29 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Urban traffic affects and propels the development of the entire functional urban layout. With the improvement in economic levels and the continuous acceleration of urbanization, urban road traffic congestion has become a problem that restricts the development of modern cities. The intersection has become the critical node and main bottleneck of the road traffic network so the intelligent control of the intersection signal plays a crucial role in alleviating traffic congestion [1]. Previous traffic signal control research was mostly based on vehicles. However, in recent years, researchers have increasingly focused on intelligent traffic signal control for pedestrians. It is necessary because when the delay of pedestrians exceeds their tolerance time, they will become impatient, which can result in violations of traffic rules [2] and even traffic accidents, endangering personal safety. According to the World Health Organization's Global Status Report on Road Safety 2018, more than half of all road traffic fatalities are among vulnerable road users: pedestrians, cyclists, and motorcyclists. Pedestrians and cyclists account for 26% of all deaths [3]. Therefore, the right design, appropriate setting, and effective use of traffic control signals are of great significance to improving the traffic capacity of vehicles at intersections, reducing the waiting time of pedestrians crossing the street and ensuring the safety of pedestrians and vehicles using the street.

The current crossing control methods include unsignalized, timing signal control, adaptive signal control, and intelligent traffic signal control (ITSC) [4]. The timing signal

control method operates according to a set timing scheme. The adaptive method is a more comprehensive scheme, which is formulated by comprehensively considering traffic flow, parking time, and other factors. The ITSC method is more flexible than the timing control method and can improve the traffic capacity of the intersection. The intelligent traffic signal control method collects the traffic information of vehicles and pedestrians in all directions at the intersection in real time. It selects the signal light phase according to the real-time signal situation of the intersection, reducing the number of vehicles stopping and the waiting time of pedestrians and avoiding the waste of green-light time.

Most of the ITSC systems are only aimed at general and special vehicles on the road and research on pedestrians is scarce. In addition, "pedestrian priority" should become the basic principle of urban road traffic management, therefore the setting of an ITSC system should also fully consider the needs of pedestrians. Only in this way can the problem of urban road traffic congestion be relieved efficiently [5]. Most existing studies consider pedestrians on the basis of vehicles and do not consider vehicles and pedestrians as a whole or say that the two are separate.

To solve the problem of pedestrian waiting time caused by vehicle-based signal control systems, we propose a real-time optimization model to coordinate the signal control between pedestrians and vehicles at intersections. The pedestrian and vehicle factors are also considered in the state and reward design. We modify the discrete traffic state encoding method proposed by Gender [6] et al. to add sidewalks to obtain pedestrian location status information. Furthermore, simulation experiments using actual intersection information demonstrate the validity of the method.

The main contributions of this work are as follows:

1. To improve the sampling efficiency of the model, we combine a Dueling Double Deep Q Network (3DQN) with a multi-process operation method. The method can be trained in multiple environments at the same time to generate multiple experiences, improve the sampling efficiency of the experience pool, and accelerate the convergence;
2. We modify the discrete traffic state encoding method and add sidewalks to the range of detection lanes to extract the location information of pedestrians. The factors of pedestrians and vehicles are jointly considered, and more comprehensive state and reward designs are defined;
3. To evaluate the model, we employ the SUMO simulation to conduct experiments based on real intersections.

This work is organized as follows: Section 2 introduces the related work on traffic signal control. Section 3 discusses the traffic signal model, which presents the algorithm architecture of the proposed model, the reinforcement learning model, and the design of the deep reinforcement learning algorithm. In Section 4, we conduct simulation experiments and discuss the results. Finally, the work concludes with a conclusion.

2. Literature Review

The traditional traffic signal control method adopts a fixed timing scheme. With an increase in the traffic flow, the fixed signal timing scheme finds it difficult to deal with complex traffic scenarios; therefore, researchers have proposed a variable signal control scheme. For example, the British Institute of Transport and Road Research proposed the TRANSYT system [7]; PB et al. [8] proposed the SCOOT system, which was an improvement on the TRANSYT system; and Luk et al. [9] proposed the SCATS system, which can select a signal scheme in real time.

With the development of variable signal technology, researchers have employed reinforcement learning (RL) [10] technology in traffic signal control [11,12]. MIKAMI [13] was the first to employ RL technology in traffic signal control. However, the method led to a sharp increase in the complexity of the algorithm and affected model training. Therefore, researchers have combined deep learning and reinforcement learning to solve the problem of high algorithm complexity, namely deep reinforcement learning (DRL) [14,15].

DRL has good performance in traffic signal control and therefore is widely used in traffic signals [16]. In order to improve the efficiency of the traffic system, Genders [6] et al. were the first to apply a DQN in the field of traffic signal control. The optimal traffic signal control scheme was obtained through the training of agents. Subsequently, Zeng [17] et al. used the memory-based deep reinforcement learning method and combined the recurrent neural network to solve the signal control problem. The proposed DRQN algorithm achieved better results than the DQN. Liang [18] et al. proposed a D3QN algorithm by combining Dueling DQN and Double DQN. Through simulation experiments, it was found that their method could reduce the average waiting time of vehicles and had a better learning speed. Gu [19] et al. proposed the double DQN algorithm of double agents to improve the stability of system training. The experiments showed that their algorithm could effectively improve the capacity of intersections. Although the above models improved the traffic efficiency of vehicles at intersections, they ignored the relationship between pedestrians and vehicles at the intersections.

In terms of pedestrian crossings, some researchers have used traditional control methods. For example, Zhuang [20] proposed to implement a pedestrian button-type activation method when crossing the street, where the phase of the signal lights is changed by the pedestrians. This method did not consider pedestrians and vehicles as a whole, which is not conducive to signal light control. MA [21] used the Pedestrian-Specific Phase (EPP), which eliminated all interactions between vehicles and pedestrians, helping to improve safety but affecting the efficiency of traffic. The other researchers used different algorithms to optimize the signal control system. Liu et al. [22] proposed an intelligent traffic light control method based on distributed multi-agent Q-learning. The method minimized the risks of vehicle delays and pedestrian crossings. Zhang et al. [23] proposed a traffic signal method considering pedestrian safety and the delay of vehicles. At the end of each pedestrian Flashing Green (FG), they introduced an additional Dynamic All Red (DAR) stage to reduce the rate of pedestrian violations at crosswalks. Wu et al. [24] proposed a multi-agent recurrent deep deterministic policy gradient (MARDDPG) method, which considered the number of pedestrians in the state space. This method alleviated the traffic congestion at intersections. Zhu [25] et al. proposed a new context-aware multi-agent generalized reinforcement learning method and applied a semi-synthetic data set to traffic signal control, which solved the problem of missing pedestrian data in the open data set. Xu [26] et al. modified the max-pressure signal control strategy to consider pedestrian factors. The activation function based on pedestrian waiting time and tolerance time were designed to effectively limit pedestrian waiting time. Although these studies considered pedestrian factors, they did not consider pedestrian location information. As a result, the demands of pedestrians could not be determined in time and the relationship between pedestrians and vehicles could not be better balanced.

Most studies only used discrete traffic state encoding [6] to extract the vehicle location information when setting the reinforcement learning states, ignoring the pedestrian traffic conditions, signal phases, and waiting times. Based on this, under the premise of considering the safety performance and traffic efficiency of pedestrians and vehicles at intersections, we propose an optimization method for signal control that coordinates pedestrians and vehicles. Pedestrian location information, pedestrian waiting time, and other information are taken into account in the state space. In the reward function, the maximum tolerance time of vehicles and pedestrians is considered on the basis of vehicle traffic and pedestrian waiting time, taking into account pedestrian crossing safety and reducing the average traffic delay of vehicles and pedestrians. On the basis of the 3DQN method, multiple processes are used to train to improve the training efficiency of the system.

3. Traffic Signal Control Model

3.1. Algorithmic Framework

We propose a signal control system in a single intersection to reduce the waiting times of pedestrians and vehicles. The system uses the deep reinforcement learning method,

which is primarily split into three parts: the main network, the target network, and the experience pool. The model framework is shown in Figure 1.

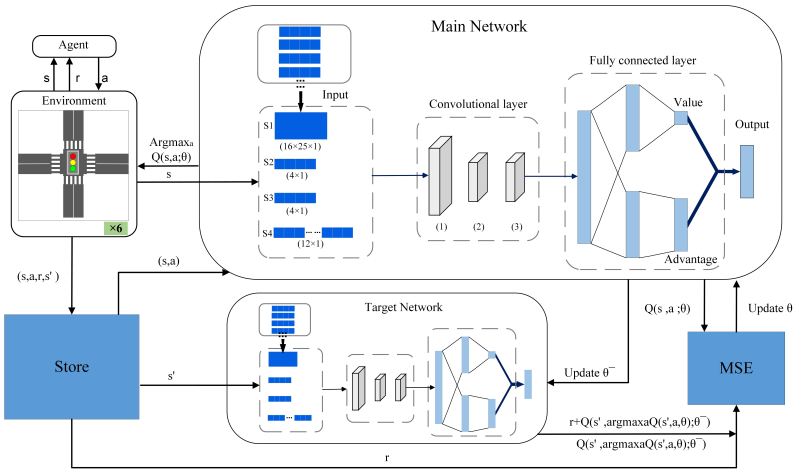


Figure 1. Algorithm framework.

The main network calculates the Q value and selects actions through a convolutional neural network. We integrate the position status of pedestrians and vehicles, input the integrated results into the convolution layer, together with the signal phase and traffic light duration status, and input the results into the full connection layer after flattening. We adopt Dueling DQN to divide the fully connected layer into two parts: the value V and advantage function A, and add the two parts to obtain the final Q value.

The target network is a mechanism to shuffle the correlation and the target network and the main network are identical in structure. The main network uses the latest experience data for model training and the target network adopts the older data to predict the real Q value. The value output by the main network is used to evaluate the current state and the target network derives the Q value and updates the parameters of the main network according to the loss function. After iterations, the model copies the parameters of the main network to the target network. Double DQN employs the main network to determine the action and the target network determines the action's value and changes the target value's calculation method.

When the experience data reaches the preset value, the model randomly extracts a certain amount of experience data for neural network training. Such an extraction method can disrupt the correlation between data, making learning more efficient and stable.

3.2. Reinforcement Learning Model

This work combines the idea of single-intersection signal control to design the relevant elements of deep reinforcement learning. We design an agent's states, actions, and rewards according to the traffic control standards and signal light configuration requirements.

3.2.1. States

In reinforcement learning, the agent decides what action to take based on the state of the environment. The state space is the input part of the training and its selection determines the quality of the control strategy so the selection of the state is particularly critical. Due to the differences in an actual intersection situation, a traffic signal control system based on deep reinforcement learning can use many different state representations. In order to meet the needs of vehicles and pedestrians passing through the intersection, we set four states in this module: vehicle and pedestrian status information, signal phase scheme information, pedestrian red light duration, and vehicle green light duration.

As shown in Figure 2, we set the position state information of pedestrians and vehicles as the first state S_1. We use and improve the discrete traffic state encoding (DTSE) to reflect

the real-time position state of pedestrians and predict the next state. Unlike the original DTSE, which only considers the position of vehicles, we also consider pedestrians. The road is gridded according to the actual situation. Several lanes and sidewalks are divided into several departments. Vehicle and pedestrian information is stored in different departments and the position information of the traffic flow is formed as a state input in the form of a vector.

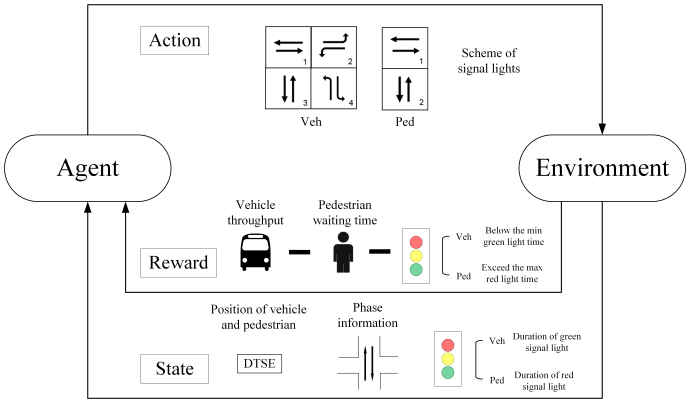


Figure 2. Reinforcement learning model diagram.

Figure 3 shows the west entrance of a standard intersection as an example. Based on the original lane, we add a sidewalk on the right side of the lane to simultaneously detect the positions of pedestrians and vehicles. Assuming that the length of the westward entrance road is l and the length of the vehicle is c, the lane is divided into l/c grids. Each grid stores information about the presence of vehicles and pedestrians in its lane. If there are no vehicles or pedestrians in the grid cell, the value is set to 0; if the presence of vehicles is detected in the grid, it is recorded as 1; and if there are many pedestrians in the same grid, we superimpose them. As the figure shows, two pedestrians are detected in the grid so the value is set to 2. The first state S_1 (Equation (1)) we set is obtained by integrating the position state information of vehicles and pedestrians in the four directions of the intersection:

$$S_1 = [s_W, s_E, s_S, s_N]^T. \tag{1}$$

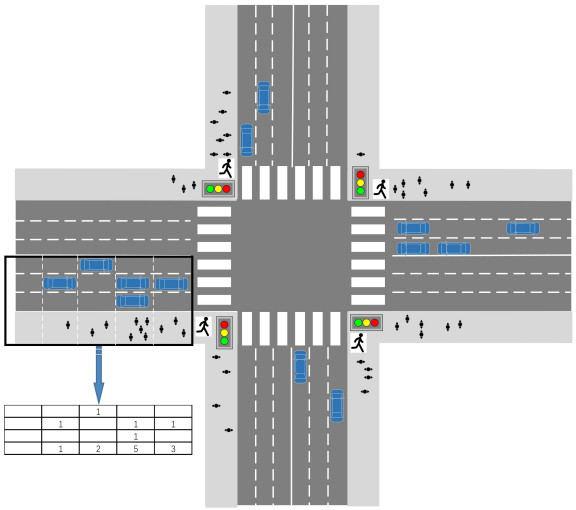


Figure 3. Schematic diagram of the status information of people and vehicles.

To reduce the waiting times of pedestrians crossing the street, we capture the position of vehicles and pedestrians in each lane of the intersection. Because of the effect caused by the current state of the signal lights, we also need to collect the current state of the signal

lights so we set the signal phase scheme information as the second state S_2 (Equation (2)). According to the driving characteristics of typical intersections, we design a four-phase basic scheme and the phase of the signal light is represented by a four-dimensional vector with one-hot encoding.

$$S_2 = (x_1, x_2, x_3, x_4), \tag{2}$$

where x_1, x_2, x_3, and x_4 represent four phases. If the current phase is the first phase, we set the value of x_1 as 1 and the remaining values are 0. The second state S_2 is represented as $S_2 = (1, 0, 0, 0)$.

To increase pedestrians' rights of way and reduce their waiting times, we consider pedestrian safety factors. Therefore, the third state S_3 is the duration of the red light for pedestrians, which is used to record the total time of the red light for pedestrians for each sidewalk. At the same time, to prevent the problem of too short a duration of a green light due to the high priority of pedestrians, the fourth state S_4 is set as the duration of the green light for vehicles. The third and fourth states are both in the form of vectors and we calculate the duration of the red light and green light for each lane or sidewalk, respectively, finally forming the states S_3 and S_4 through integration. This work takes a standard three-lane intersection as an example and the specific formulas are shown in Equations (3) and (4):

$$S_3 = (l_{r0}, l_{r1}, l_{r2}, l_{r3}), \tag{3}$$

$$S_4 = (l_{g0}, l_{g1}, l_{g2}, l_{g3}, l_{g4}, l_{g5}, l_{g6}, l_{g7}, l_{g8} l_{g9}, l_{g10}, l_{g11}). \tag{4}$$

3.2.2. Agent Actions

After the agent obtains the current state, it selects the action from the action space and observes the reward for the action. The agent selects the action phase every second so that the signal light's execution phase and time are not fixed. The agent can choose the optimal action based on the actual position state of pedestrians and vehicles. Hence, the flexibility of the signal light control system is increased.

The action phase in the first and next seconds can be the same or different. If the action phase is different, we insert a yellow light for 3 s to ensure traffic safety. Based on the above situation, the action phase is shown in Table 1.

Table 1. Action Phase Settings.

Action Number	Traffic Direction	Signal Status
a_1	EW	GrrGGrGrrGGrGrGr
a_2	EWL	GrrGrGGrrGrGrrrr
a_3	SN	GGrGrrGGrGrrrGrG
a_4	SNL	GrGGrrGrGGrrrrrr

As shown in Table 1, the action space contains four actions. The traffic direction corresponding to the first action a_1 is EW, which means that the vehicle in the east–west direction is allowed to go straight and the pedestrian in the east–west direction passes through the intersection when the action is executed. EWL in a_2 indicates that left-turn vehicles in the east–west direction are allowed to pass and pedestrians in any direction are not allowed to pass through the intersection, regardless of the control of the right-turn signal lights. The signal light statuses are listed in Table 1 (where 'G' represents green light and 'r' represents red light).

3.2.3. Reward Function

The reward function in the reinforcement learning model is the feedback on the previous action. The reward function should take into account the correlation between the control objectives [27]. In this work, we design a reward function that includes vehicle throughput, pedestrian waiting time, exceeding pedestrian maximum red time, and less

than vehicle minimum green time. This method aims to reduce the waiting time of pedestrians crossing the street without affecting the efficiency of vehicle traffic. The observation time of the agent in this work is only 1 s and the interval time is too short to calculate accurately. We use a time period Δt to calculate the reward value. The first part of the reward is the pedestrian waiting time:

$$r_{waiting} = \sum_{i=t-\Delta t}^{t} d_i, \tag{5}$$

where d_i represents the waiting time of all pedestrians during i time period at the intersection, Δt is a constant, and we set the period constant at 20 s. $r_{waiting}$ represents the waiting time of all pedestrians within a period Δt and $r_{waiting}$ is set to a negative value to reduce the waiting time of pedestrians.

To avoid our proposed model only considering the passing rights of pedestrians, we also calculate the throughput at the same moment. The second part of the reward is vehicle throughput, as shown in Equation (6):

$$r_{throughput} = \sum_{i=t-\Delta t}^{t} T_i, \tag{6}$$

where T_i represents the throughput of the intersection at time period i. $r_{throughput}$ is the throughput, except for right-turn vehicles.

To ensure pedestrians' rights of way, the red light time for pedestrians should be less than the maximum waiting time that pedestrians can tolerate. In the third part, we penalize the part that exceeds the maximum red light time, as shown in Equation (7):

$$r_{maxred} = \sum_{n} max(R_{nt} - R_{max}, 0), \tag{7}$$

where R_{nt} represents the duration of the red light at the entrance n at time t and R_{max} is the maximum red-light duration. After reviewing the relevant research literature, the maximum waiting time for pedestrians to cross the street is 70 s so we take this value as 70 s [28,29].

Regarding rationality and safety, setting a minimum green light duration is necessary. When the green light duration is less than the minimum green light duration, the model should give a penalization. Hence, we define the vehicle minimum green time penalization as $r_{mingreen}$, as shown in Equation (8):

$$r_{mingreen} = \sum_{n} max(G_{min} - G_{nt}, 0), \tag{8}$$

where G_{nt} represents the duration of the green light for entrance n at time t and G_{min} is the minimum green light duration. According to the literature [30], the minimum green light time for vehicles is 10 s so we take it as 10 s.

After integrating the four parts of the reward function, we can obtain the final reward function, as shown in Equation (9):

$$r_t = r_{throughput} - r_{waiting} - r_{mingreen} - r_{maxred}. \tag{9}$$

3.3. Algorithm Design and Model Training

Our network adopts the 3DQN (Dueling Double Deep Q Network) algorithm. 3DQN is an algorithm that combines Double DQN and Dueling DQN. The Double DQN model uses the optimal action value to obtain the target Q value through the target network, which reduces the deviation of the final value. In addition, Dueling DQN uses two fully connected layers to estimate the Q value.

The dueling network inputs the state into two fully connected layers after passing through the convolution layer; one generates a value function $V(s;\theta,\beta)$ and the other generates an advantage function $A(s,A;\theta,\alpha)$. Finally, the value function V and the advantage function A are added to obtain the final Q value, where θ is the parameter of the convolutional layer and α and β are the parameters of the fully connected layer. The formula for calculating the Q value is shown in Equation (10):

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + (A(s,a;\theta,\alpha) - \frac{1}{|A|}\sum_{a'} A(s,a';\theta)), \tag{10}$$

where $V(s;\theta,\beta)$ is the estimated value of the state function. The second part of the formula $A(s,a;\theta,\alpha) - \frac{1}{|A|}\sum_{a'} A(s,a';\theta,\alpha)$ uses the mean value to estimate the advantage function. The Q value is estimated by comparing the advantages of the different actions in the advantage function. Therefore, we can select the more valuable states without knowing the effect of each state.

Double DQN uses two action-value functions to reduce the maximizing bias. One of two action-value functions is to estimate the action and the other is to estimate the value of the action. The main network is adopted to determine the action and the target network estimates the action value. The loss function formula is shown in Equations (11) and (12):

$$TargetQ = r + \gamma Q(s', argmax_{a'} Q(s',a';\theta);\theta^{-}), \tag{11}$$

$$L(\theta) = E[(TargetQ - Q(s,a;\theta))^2], \tag{12}$$

where θ is the network parameter and the loss function is the mean squared error.

The design of the neural network is shown in Figure 1. Because of the high correlation of the position information between vehicles and pedestrians in state S_1, firstly, we put the position information of the four directions in state S_1 into a separate convolutional neural network. Then, we put the output of the previous convolutional neural network into the next convolutional neural network, together with the information of the other three states.

In addition, to improve the efficiency of the neural network training [31], we adopt the method of collecting experience in multiple processes. In each training of DQN, only one sample data can be obtained in the experience pool, which makes the number of samples grow slowly and reduces the training efficiency of the model. Therefore, we adopt a multi-process approach, running multiple environments simultaneously for training. This method can generate multiple sample data at one time, which helps the neural network to quickly extract empirical data for training and improves the training efficiency of the model. The neural network training and experience generation process in this work is as follows:

In Algorithm 1, the model simultaneously uses multiple traffic environments for training and selects an action corresponding to the maximum value or randomly selects an action according to the probability factor ϵ. We execute the chosen action in the current environment, then, we obtain the reward r_t and the next state s_{t+1}. Finally, the model stores the gained experience (s_t, a_t, r_t, s_{t+1}) in the experience pool.

Algorithm 1 Experience Pool Experience Generation

Input: the maximum capacity M of the experience pool D, the total simulation step S, the total simulation time T, the random probability factor ϵ
Output: experience matrix (s_t, a_t, r_t, s_{t+1})

Initialize simulation step size i = 0
while $i < S$ **do**
    Launch multiple simulation environments $e_1, e_2, \ldots, e_m$
    if simulation time $< T$ **then**
        if random number in (0, 1) $< \epsilon$ **then**
            Randomly choose an action a_t
        else
            $a_t = argmaxQ(s, a; \theta)$
        end if
        Action acts on the road network to obtain a new traffic state s_{t+1} and reward r_t
        Update state $s_t \leftarrow s_{t+1}$
        Store the experience matrix (s_t, a_t, r_t, s_{t+1}) in the experience pool
    end if
    i←i+1
end while

As shown in Algorithm 2, the model randomly selects several samples from the experience pool to form small batches and then feeds them into the neural network. The neural network parameters are updated by gradient descent. With the increase in training, the empirical data generated in the early stage have little reference value for the current action selection. Therefore, when the experience reaches a certain amount, the model deletes the early experience data to reduce the operating burden of the system. The agent decides the optimal action at the current stage according to the corresponding action of the optimal Q value obtained by training. After iteration, the agent chooses the best action based on the maximum Q value to maximize the reward value.

Algorithm 2 Neural Network Training

Input: the maximum capacity of the experience pool D is M and the sample batch size is B
Output: neural network parameters θ

if the size of memory$> M$ **then**
    Randomly draw B samples from experience pool D environments
    Feed the neural network and compute the estimated Q value
    $Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + (A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha))$
    and the target Q value: $TargetQ = r + \gamma Q(s', argmax_{a'} Q(s', a'; \theta); \theta^-)$
    Calculate the loss function: $L(\theta) = E[(TargetQ - Q(s, a; \theta))^2]$
    Update the neural network parameters by gradient descent according to loss function θ
    Eliminate the initially generated experience from the experience pool
end if

4. Simulation Experiment and Results Analysis

4.1. Simulation Experiment Setup

We used SUMO version 1.8.0 as the traffic simulation platform and the Traci interface provided by SUMO was coded with Python 3.7.3. The deep learning version used to build the neural network was Tensorflow-GPU 2.5.0, and the parameters are shown in Table 2.

Table 2. Parameter Settings.

Parameter	Initial Value
Action size	4
Batch-size	64
Discount factor	0.99
Random probability	0.01
Greedy factor	0.01
Episode	300
Number of environments	6
Experience pool capacity	20,000

To evaluate the proposed model comprehensively, we used the average waiting time of pedestrians and vehicles, the average number of waiting people, and throughput as our evaluation metrics. The average waiting time for pedestrians was the average waiting time for all pedestrians and vehicles at the entrances of intersections in a period. The average number of waiting people was the number of waiting pedestrians in a period. The smaller the value of these three evaluation indicators, the shorter the waiting time for people, and the better the model's performance. Throughput refers to the number of vehicles passing through an intersection in a period. The more vehicles that pass through, the better the traffic performance of the intersection.

4.2. Standard Intersection Experiment Results

Intersection setting: As shown in Figure 4, the intersection setting is a crossroad that consisted of four roads, each with a length of 100 m. There are three lanes for vehicles at all four entrances of the intersection. There is a right-turn lane on the right of each direction; only right-turn vehicles are allowed to pass and the signal light is always green. A pedestrian waiting area and eight sidewalks are set up on the edge of the road and four pedestrian crossings are created.

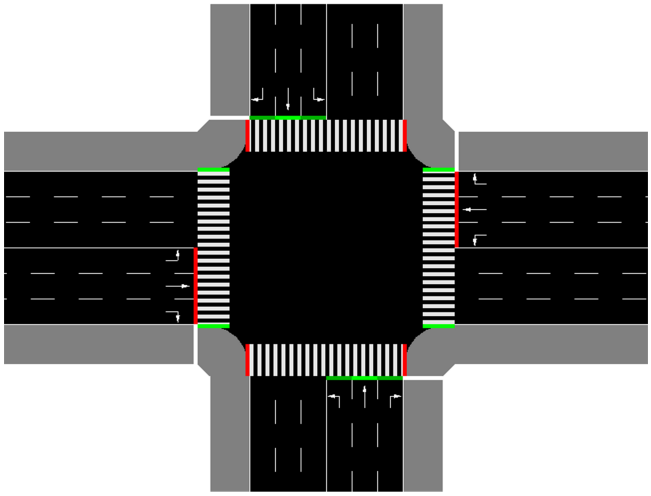


Figure 4. Schematic diagram of standard intersection.

Three methods were used as comparative experiments to explore the effectiveness of our method. The first was the fixed-time traffic control method, which is a traditional signal light control method. The cycle runs according to a set duration, regardless of other factors on the road. The fixed durations in this work were set to 30 s and 40 s. The second method was the Webster timing method [32]. The Webster method needs to confirm the traffic volume of the road. According to the different traffic flows of roads, the signal timing is calculated with the goal of the minimum vehicle delay. Finally, the third method was Dueling DQN, an improved version of the DQN that was developed in recent years, which effectively improved the model's performance.

We collected the data from the 300 epochs of the simulation experiments and we used the average value of each epoch as the result for each epoch. Different control strategies were compared with the pedestrian waiting time and the number of waiting people. The simulation results of the pedestrian waiting time are shown in Figure 5. It can be seen that our model outperformed the other models. The training results of the fixed-duration and Webster methods were almost straight lines, whereas Dueling DQN and our method both showed better results than the fixed-duration model, although they fluctuated greatly in the early stages of training. Compared with Dueling DQN, the time required for pedestrians to pass through the intersection was significantly reduced under the training of our model and the model fluctuation was smaller. This demonstrates the superiority of our method. Specifically, in terms of waiting time, the pedestrian waiting time using the Webster method was about 16 s, which was shorter than the waiting times for the fixed times of 40 s and 30 s, indicating that the Webster method achieved better results in this scenario. The waiting time of Dueling DQN was about 13 s after 250 epochs, and the pedestrian waiting time dropped to about 8 s after 200 epochs of training with our method. In terms of the number of people waiting, the numbers of waiting people for the fixed times of 40 s and 30 s were about 66 and 44, about 35 for Webster, and about 26 for Dueling DQN after 250 epochs, although the number of waiting pedestrians in our method stabilized at 10 after 150 epochs. In addition, as shown in Table 3, after training using our proposed traffic signal control method considering pedestrians, compared with the Webster method, the waiting time of pedestrians was reduced by about 49.78% and the number of waiting pedestrians was reduced by 71.72%. Compared with Dueling DQN, the waiting time was reduced by 40.1% and the number of waiting people was reduced by 61.19%. The number of pedestrians waiting and the waiting time were reduced significantly, which is related to the fact that these models did not add the pedestrian factor.

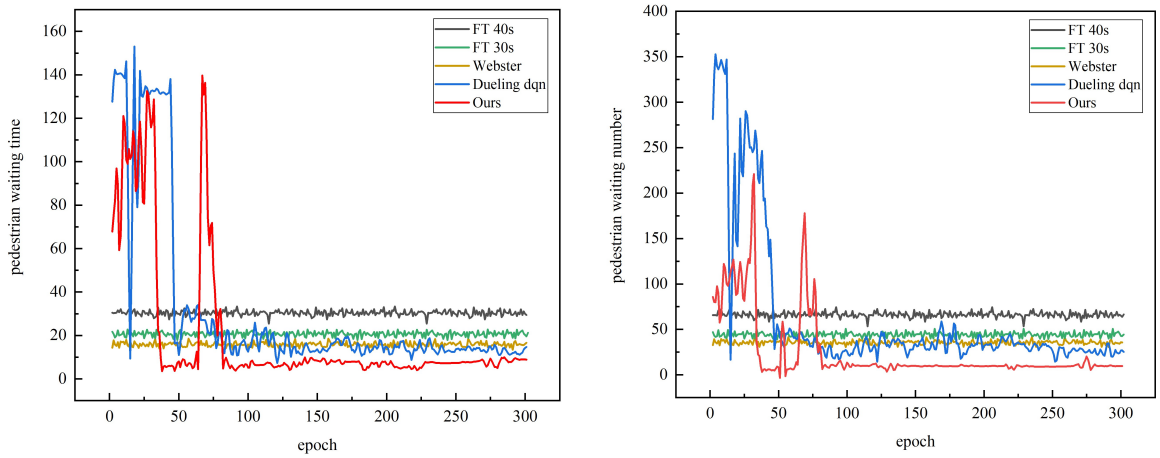


Figure 5. Standard intersection results for pedestrians. (**a**) Pedestrian waiting time; (**b**) Number of pedestrians waiting.

Table 3. Comparison of evaluation indicators of standard intersections.

Method	Pedestrian Waiting Time (s)	Pedestrian Waiting Number
FT 40s	30.35	66.13
FT 30s	20.66	44.38
Webster	15.91	35.61
Dueling DQN	13.34	25.95
Ours	7.99	10.07

4.3. Actual Intersection Experiment Results

Intersection setting: As shown in Figure 6, the Wuyi Middle Road and Qunzhong Road intersection is used as a traffic simulation scene of a single intersection. The intersection is a crossroad that consists of four roads, each with a length of 100 m. There are five vehicle entrance lanes at all four entrances of the intersection and a right-turn lane on the far right side of each direction. The setup includes pedestrian waiting areas, eight sidewalks, and four pedestrian crossings.

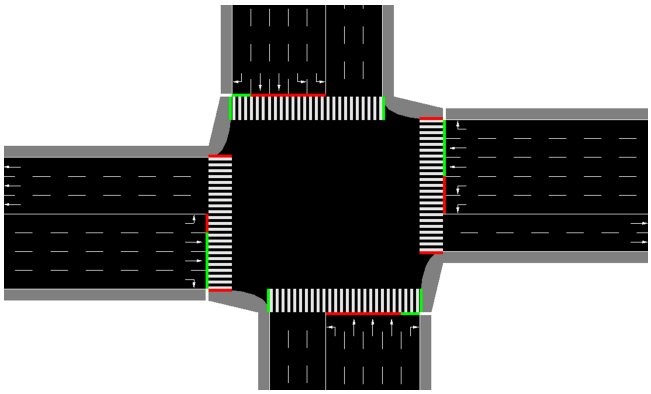


Figure 6. Schematic diagram of Wuyi Middle Road–Qunzhong Road intersection.

To align with the actual situation and help the model make a more accurate plan, we collected the traffic flow and pedestrian flow information at the Wuyi Middle Road–Qunzhong Road intersection in Fuzhou City. The experimental data collection time was from 5:30 to 5:45 p.m. on 16 March 2022. We expanded the 15 min traffic volume at the intersection to an hourly traffic volume and input this into the road network file as the traffic flow data for the experiment. The traffic flow at the entrance is shown in Table 4.

Table 4. Actual Traffic Flow at Intersection.

Lane	Vehicle Type	Turn Left (pcu/h)	Go Straight (pcu/h)	Turn Right (pcu/h)
1	Vehicle	338	427	137
	Pedestrian		1768	
2	Vehicle	344	423	109
	Pedestrian		1392	
3	Vehicle	176	235	182
	Pedestrian		1028	
4	Vehicle	53	160	109
	Pedestrian		544	

The experiment results are shown in Figure 7. It can be seen in the figures that our method outperformed the other three methods. Among them, the pedestrian waiting time of the Webster method was about 48 s and the number of waiting pedestrians was about 83. The waiting time for the fixed time of 40 s was about 32 s and the number of waiting people was about 67. After 25 iterations of our model, the waiting time fluctuated between 10 s and 50 s and the number of waiting people fluctuated between 30 and 200. The model stabilized after 200 epochs; the waiting time was about 12 s and the number of people was about 27. The results show that the pedestrian waiting time was significantly reduced compared with the previous two methods. Compared with the Webster method and the fixed-time method with 40 s, Dueling DQN obtained a shorter waiting time and number of people and the performance was better. However, its convergence speed was slow and its performance was not stable. Compared with Dueling DQN, the waiting time of our method decreased by 58.76% and the number of people decreased by 51.54%. It can be seen that the overall impact of Dueling DQN was lower than that of our method. Therefore, our

method outperformed the other three methods in reducing the waiting time and number of pedestrians.

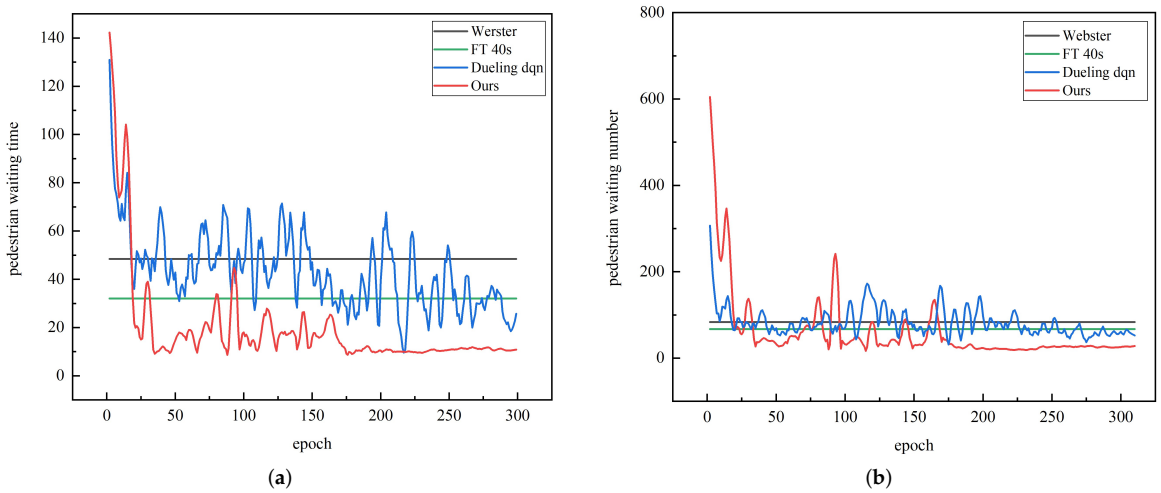


Figure 7. Actual intersection results for pedestrians. (**a**) Pedestrian waiting time; (**b**) Number of pedestrians waiting.

The comparison results of the throughputs at the intersections are shown in Figure 8. The general trend in the three strategies was relatively consistent. Our method fluctuated wildly before 200 epochs, showing the agent's continuous learning and exploration process. When the training exceeded 200 epochs, the curve tended to be stable and the throughput was about 1190 pcu/h. The throughputs of the Webster and fixed-time 40 s methods were 1154 pcu/h and 1119 pcu/h. In Table 5, it can be seen that, compared with the Webster and fixed-time methods, the throughput was increased by 3.11% and 6.34% in our method. It can be seen that the throughput was slightly higher in our method than in the other two methods, indicating that our method did not significantly affect the throughput of vehicles when considering pedestrian traffic at the intersection.

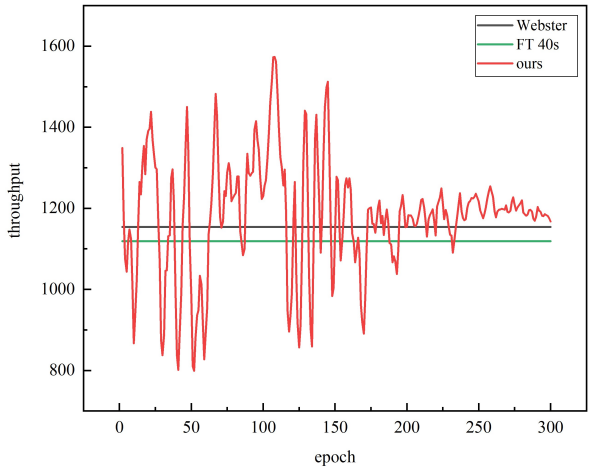


Figure 8. Throughput at actual intersection.

Table 5. Comparison of evaluation indicators of actual intersection.

Method	Pedestrian Waiting Time (s)	Pedestrian Waiting Number	Throughput (pcu/h)
Webster	48.42	83.25	1154
FT 40s	32.06	67.01	1119
Dueling DQN	28.30	55.08	/
Ours	11.67	26.69	1190

The average delay and waiting number of vehicles are shown in Figure 9. We fit the delay and waiting number of vehicles and drew their fitting prediction band and confidence band so we could more intuitively see from the figure that with the increase in the number of epochs, the delay and the waiting number of vehicles gradually decreased and converged. After 300 epochs of training, the average delay per vehicle was reduced to about 98 s and the average number of waiting vehicles for each road also decreased slightly. We understand that the data were relatively scattered at the beginning of training and the data distribution became dense as the agent continued to explore and learn, which shows that the training of the agent in our method is effective.

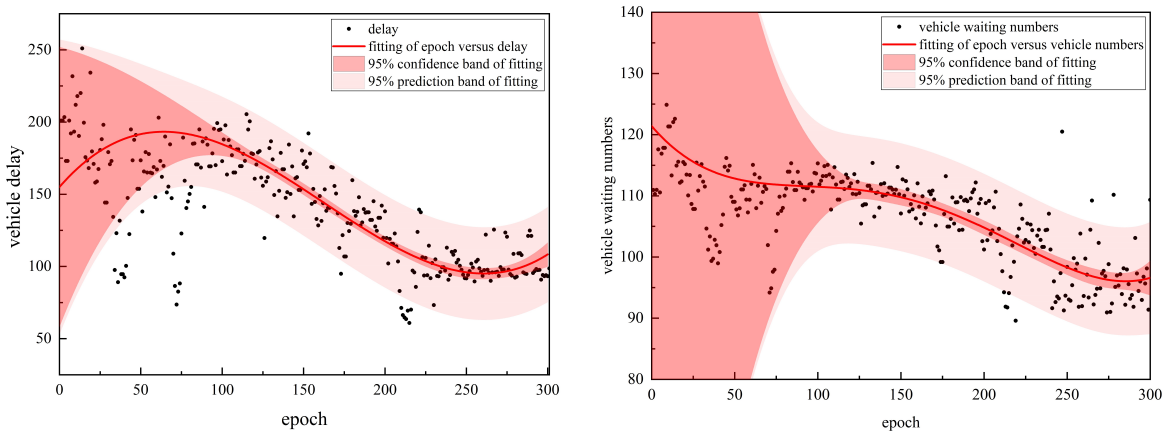


Figure 9. Actual intersection results for vehicles. (**a**) The average delay; (**b**) The waiting numbers.

Traffic scenarios are more complex at a realistic intersection during rush hour. However, through training, our model reduced the waiting time for pedestrians and the number of pedestrians waiting. At the same time, compared with the traditional traffic signal timing method, the delay and the waiting number of vehicles decreased and traffic efficiency was not significantly affected. This demonstrates the effectiveness of our strategy.

5. Conclusions

This work proposed a pedestrian-considered deep reinforcement learning single-intersection signal control method. This method utilized the influencing factors between pedestrians and intersection signal control optimization, effectively alleviated the inconvenience caused by the signal timing method controlled by vehicle status, balanced the demands of pedestrians and vehicles, and was more realistic. At the same time, we ran multiple environments to improve the training efficiency of the system. In addition, we built a standard intersection and a simulation environment based on an actual intersection in Fuzhou to verify the effectiveness of our method. It was observed from the simulation results of SUMO that our method effectively reduced the waiting time of pedestrians and also achieved better results in the actual intersection environment.

Nevertheless, our study has some limitations. First, there may be inaccuracies in the vehicle and pedestrian traffic information collected by manual counting. Second, the impact of latency in inter-intersection communications on our model was not taken into account.

In future work, it is worth considering the latency in inter-intersection communications and a more accurate statistical scheme. In addition, we should consider more intersection types, such as T-shaped intersections and roundabouts, and apply the single intersection method to multiple intersections.

Author Contributions: Conceptualization, G.H. and Q.Z.; methodology, L.L.; validation, P.T., Y.Z. and Z.L.; formal analysis, Q.Z.; investigation, G.H.; data curation, L.L.; writing, all authors; visualization, Z.L.; supervision, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a project of the Fujian University of Technology (GY-Z19066), projects of the National Natural Science Foundation of China (41971340), projects of the Fujian Provincial Department of Science and Technology (2021Y4019), and a project of Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology (Fujian University of Technology) (KF-J21012).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DQN	Deep Q-network
ITSC	Intelligent Traffic Signal Control
RL	reinforcement learning
SUMO	Simulation of Urban Mobility
DTSE	Discrete Traffic State Encoding

References

1. Wei, H.; Zheng, G.; Gayah, V.; Li, Z. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explor. Newsl.* **2021**, *22*, 12–18. [CrossRef]
2. Ding, T.; Wang, S.; Xi, J.; Zheng, L.; Wang, Q. Psychology-Based Research on Unsafe Behavior by Pedestrians When Crossing the Street. *Adv. Mech. Eng.* **2015**, *7*, 203867. [CrossRef]
3. World Health Organization. *Global Status Report on Road Safety 2018: Summary*; World Health Organization: Geneva, Switzerland, 2018.
4. Wei, H.; Zheng, G.; Gayah, V.; Li, Z. A survey on traffic signal control methods. *Assoc. Comput. Mach.* **2019**, *1*, 1–32.
5. Ye, Q.; Huang, W.; Sun, W.; He, J.; Li, S.; Hu, Y. Research on Signal Control Optimization Based on Pedestrian Oriented Crossing. *Mod. Transp. Technol.* **2014**, *11*, 65–68.
6. Genders, W.; Razavi, S. Using a deep reinforcement learning agent for traffic signal control. *arXiv* **2016**, arXiv:1611.01142.
7. Robertson, D.I. TRANSYT: A traffic network study tool. *Road Res. Lab. UK* **1969**, *253*, 14–25.
8. Hunt, P.B.; Robertson, D.I.; Bretherton, R.D.; Royle, M.C. The SCOOT on-line traffic signal optimisation technique. *Traffic Eng. Control* **1982**, *23*, 190–192.
9. Luk, J.Y.K. Two traffic-responsive area traffic control methods: SCAT and SCOOT. *Traffic Eng. Control* **1984**, *25*, 14.
10. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
11. Li, L.; Lv, Y.; Wang, F.-Y. Traffic signal timing via deep reinforcement learning. *IEEE/CAA J. Autom. Sin.* **2016**, *3*, 247–254.
12. Lin, Y.; Dai, X.; Li, L.; Wang, F.-Y. An efficient deep reinforcement learning model for urban traffic control. *arXiv* **2018**, arXiv:1808.01876.
13. Mikami, S.; Kakazu, Y. Genetic reinforcement learning for cooperative traffic signal control. In Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 27–29 June 1994; pp. 223–228.
14. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
15. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]

16. Haydari, A.; Yilmaz, Y. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *32*, 11–32. [CrossRef]
17. Zeng, J.; Hu, J.; Zhang, Y.N. Adaptive Traffic Signal Control with Deep Recurrent Q-learning. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1215–1220.
18. Liang, X.; Du, X.; Wang, G.; Han, Z. A Deep Reinforcement Learning Network for Traffic Light Cycle Control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1243–1253. [CrossRef]
19. Gu, J.; Fang, Y.; Sheng, Z.; Wen, P. Double Deep Q-Network with a Dual-Agent for Traffic Signal Control. *Appl. Sci.* **2020**, *10*, 1622. [CrossRef]
20. Zhuang, X.; Wu, C.; Ma, S. Cross or wait? Pedestrian decision making during clearance phase at signalized intersections. *Accid. Anal. Prev.* **2018**, *111*, 115–124. [CrossRef]
21. Ma, W.; Liu, Y.; Head, K.L. Optimization of Pedestrian Phase Patterns at Signalized Intersections: A Multiobjective Approach. *J. Adv. Transp.* **2014**, *48*, 1138–1152. [CrossRef]
22. Liu, Y.; Liu, L.; Chen, W.-P. Intelligent traffic light control using distributed multi-agent Q learning. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–8.
23. Zhang, Y.; Zhang, Y.; Su, R. Pedestrian-safety-aware traffic light control strategy for urban traffic congestion alleviation. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 178–193. [CrossRef]
24. Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; Wu, D.O. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8243–8256. [CrossRef]
25. Zhu, R.; Wu, S.; Li, L.; Lv, P.; Xu, M. Context-Aware Multiagent Broad Reinforcement Learning for Mixed Pedestrian-Vehicle Adaptive Traffic Light Control. *IEEE Internet Things J.* **2022**, *9*, 19694–19705. [CrossRef]
26. Xu, T.; Bika, Y.; Levin, M. Ped-Mp: A Pedestrian-Friendly Max-Pressure Signal Control Policy for City Networks. Available online: https://ssrn.com/abstract=4186588 (accessed on 2 October 2022).
27. Lu, C.; Huang, J.; Deng, L.; Gong, J. Coordinated ramp metering with equity consideration using reinforcement learning. *J. Transp. Eng. Part A Syst.* **2017**, *143*, 04017028. [CrossRef]
28. Liu, G.X.; Li, K.; Sun, J. Research on pedestrian's waiting time at signal control intersection. *China Saf. Sci. J.* **2009**, *19*, 159–166.
29. Lu, S.; Wang, H.; Liu, X. Maximum waiting time for pedestrian crossing based on survival analysis. *J Transp. Inf. Saf.* **2009**, *27*, 69–76.
30. Xu, H.; Sun, J.; Zheng, M. Comparative analysis of unconditional and conditional priority for use at isolated signalized intersections. *J. Transp. Eng.* **2010**, *136*, 1092–1103. [CrossRef]
31. Liao, L.; Liu, J.; Wu, X.; Zou, F.; Pan, J.; Sun, Q.; Li, S.E.; Zhang, M. Time difference penalized traffic signal timing by LSTM Q-network to balance safety and capacity at intersections. *IEEE Access* **2020**, *8*, 80086–80096. [CrossRef]
32. Webster, F.V. *Traffic Signal Settings*; Road Research Lab Tech Papers; Road Research Lab: London, UK, 1958; Volume 39.