Article

A Bus Signal Priority Control Method Based on Deep Reinforcement Learning

Wenchao Shen 1, Liang Zou 1, Ruisheng Deng 2, Hongyu Wu 2 and Jiabin Wu 3,*

1 College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518060, China; supershen@xsjsoft.com (W.S.); zouliang@szu.edu.cn (L.Z.)
2 Guangzhou Schoolplus Software Technology Co., Ltd., Guangzhou 510650, China; rsdeng@xsjsoft.com (R.D.); hyywu@xsjsoft.com (H.W.)
3 School of Transportation, Civil Engineering & Architecture, Foshan University, Foshan 528225, China
* Correspondence: 13670000003@sina.cn

Abstract: To investigate the issue of multi-entry bus priority at intersections, an intelligent priority control method based on deep reinforcement learning was constructed in the bus network environment. Firstly, a dimension reduction method for the state vector based on the key lane was proposed, which contains characteristic parameters such as the bus states, the flow states, and the signal timing. Secondly, a control action method that can adjust phase sequence and phase green time at the same time was proposed under the constraints of maximum green and minimum green. Furthermore, a reward function, which can be uniformly converted into the number of standard cars, was established focusing on the indexes such as the busload and maximum waiting time. Finally, through building an experimental environment based on SUMO simulation, a real-time bus signal priority control method based on deep reinforcement learning was constructed. The results show that the algorithm can effectively reduce the waiting time of buses without affecting overall traffic efficiency. The findings can provide a theoretical basis for the signal control method considering bus priority and improve the operation efficiency of public transport.

Keywords: traffic engineering; bus priority; signal control; deep reinforcement learning; sumo simulation

1. Introduction

With the continued growth of motorized mobility in recent years, the passenger flow of public transport has declined year by year, and the lack of competitiveness of buses is one of the important reasons. This is reflected in two aspects: On the one hand, bus travel is unreliable, reflected in the low speed and punctuality; on the other hand, the imbalance between supply and demand leads to poor service and a waste of supply surplus. As for the unreliability of bus travel, studies have shown that bus delays at intersections account for 10–20% of the total bus travel time, and this bus delay is more than half of the entire delay time [1]. Therefore, improving public transport operation efficiency at intersections is one of the critical choices to enhance public transport competitiveness.

The traditional bus priority control is mainly divided into passive priority control, active priority control, and adaptive priority control [2–4]. These methods are usually designed for specific traffic flow scenarios and periods. They use an optimization mathematical model with traffic flow theory and solve the model based on optimization theory. However, traffic flow has substantial spatiotemporal variability, so this method usually cannot be used at different periods at the same intersection, nor can it be used at other intersections at the same period. In addition, these methods mainly focus on a single entrance direction of bus signal priority with green extension, red light shortening, and special bus phase insertion, so it is too limited for all entrance direction bus priority [5]. However, according to the advantages of closed-loop feedback control and global optimization for reinforcement learning, which does not rely on mathematical models [6], a series of
achievements has been made in traffic signal control optimization [7–9], which is mainly reflected in optimizing the state space, behavior action, reward function, and the solution method. Due to the complex traffic flow characteristics, there are several apparent hurdles in the bus signal priority control method based on deep reinforcement learning.

Firstly, how is state space represented in fewer dimensions? The current research mainly focuses on three points: pixel matrix representation based on images [10–13], DTSE (discrete traffic state encoding) based on each lane [14], and vector representation based on each lane [15,16]. The indexes involved mainly include queue length [17], vehicle location parameters [18], density, speed [19], and current signal phase time [20], etc. However, the construction of the traffic state space with the lane as the unit is prone to redundant information, leading to the curse of dimensionality and difficulty in solving problems.

Secondly, how can the control action design be more flexible? The current research mainly focuses on two types of methods: one is the adjustable phase sequence and the fixed green time [15,16,18,21,22]; one is a fixed phase sequence and has adjustable green time [23,24]. Meanwhile, the frequency of adjustment of the green time can be divided into two types: (1) adjust once a cycle; (2) adjust once a phase [25,26]. Generally, the above action design method lacks flexibility, and cannot adjust phase sequence and phase green time at the same time.

Thirdly, how can an action reward function considering multi-objective optimization be designed? The previous research mainly focused on the design of the reward function with a single optimization objective, the optimization objective involved mainly includes delay, vehicle waiting time, queue length, number of stops, traffic volume, vehicle emissions, etc. [13,20,27–30]. Given that urban traffic signal control needs to consider multi-objective optimization control problems such as safety, efficiency, and order [31], it is difficult to meet the application scenarios only with the improvement of a single objective as an action reward. Many researchers have also designed a multi-objective reward function based on weight [16,31–34]. However, the objective weight setting is mainly based on artificial experience, which is difficult to effectively calibrate, resulting in a certain degree of influence on the training convergence speed and stability of the algorithm. At the same time, the scale of different objectives is inconsistent, leading to poor interpretability.

Finally, how can this model be solved? Traditional reinforcement learning is mainly based on the Q-value table learning algorithm [35] and SARSA algorithm [36], which is not good at dealing with high-dimensional and continuous traffic-state information, restricting further application. Thus, many studies have begun to take advantage of deep learning in data processing to carry out research on deep reinforcement learning in the field of signal control [37,38].

The current research on bus signal priority control based on reinforcement learning exhibits the above problems to different degrees [39–42]. To address these issues, a new method for bus signal priority control based on deep reinforcement learning is proposed. The main contributions of this study can be summarized as follows:

1. We propose a unified framework for bus signal priority real-time control, which is able to migrate to different intersections and different time periods quickly;
2. By using key lanes to characterize traffic states, the dimension of the state space is effectively reduced, and the rapid training ability of the model can be improved;
3. The design of the action space is more flexible, which can adjust phase sequence and phase green time at the same time;
4. Under the multi-objective optimization, we establish a reward function that can be uniformly converted into the number of standard cars, which effectively solves the problem of weight calibration. Accordingly, the rest of this article is structured as follows. Section 2 provides a model description for bus signal priority control based on reinforcement learning. Section 3 introduces the reinforcement learning agent training process to achieve the proposed method. Section 4 presents the simulated experiment results to verify the effectiveness of the proposed method. Lastly, conclusions and recommendations for future research are made in Section 5.
2. Model of Bus Signal Priority Control Reinforcement Learning Agent

A traffic light controller is defined as an “Agent”, which can assign the access right to traffic flows.

2.1. Traffic State Space Construction

Considering that signal timing is the core principle of green time allocation based on the characteristics of traffic flow in key lanes [43,44], this paper built the state space based on the key lanes, in which the state parameters are composed of the information of buses, the flow characteristics of intersections, and the information of signal timing. Bus information is represented by two parameters: bus location and passenger capacity, which are aggregated by the phase key lanes and 6 m per cell. The flow characteristics of intersections are selected based on the number of queuing vehicles and aggregated by the phase key lanes. The signal timing contains four parameters, that is, the current vehicle phase number, the current vehicle release time, the vehicle phase red light duration, and the pedestrian phase red light waiting time. The intersection state space is expressed as:

\[
S_v = (s_1, \ldots, s_i, \ldots, s_n) \tag{1}
\]

\[
s_i = (d_i, ld_i, q_i, rv_i) \tag{2}
\]

\[
S_p = (p_{1i}, \ldots, p_{pi}, \ldots, p_n) \tag{3}
\]

\[
S = \{S_v, S_p, la, gl\} \tag{4}
\]

where the state vector \( S \) is a \( 1 \times m \) vector, merged through vector concatenation; \( n \) denotes the number of vehicle phases; \( k \) denotes the number of pedestrian phases; \( s_i \) denotes the state space of the motor vehicle phase in phase \( i \); \( d_i, ld_i, q_i, rv_i \) denote the bus running position vector, the busload vector, the queuing vehicles vector in the \( k \) lane, and the red light duration vector in phase \( i \) of the motor vehicle, respectively; \( p_i \) indicates the red light waiting time of pedestrians in phase \( i \); \( la \) indicates the current vehicle phase number; \( gl \) indicates the current vehicle phase green light duration time.

Based on key lanes, the division of cells does not need to consider the number of lanes on the approach, which can reduce feature dimension. Taking the east approach in an intersection as an example, as shown in Figure 1, the east approach consists of five lanes. The length between the parking line of the east approach and the upstream intersection is 180 m, and the length of each cell is 6 m. By using the DTSE method (as shown in Figure 1), there needs to be \( 30 \times 5 = 150 \) elements to represent the bus position in the east approach. However, there only needs to be 30 elements that represent the bus position by using the proposed method (key lanes-based method, as shown in Figure 2), and the dimension can be reduced by 80%.

2.2. Behavior Action Space Construction

The phase number is used as the action set. When the next phase obtained by the agent decision is the current phase, it means that the current phase extends the unit green light time; when the next phase is not the current phase, the phase transitions are performed, and the switching transitions are made with a fixed yellow light time and full red time. For example, at a four-phase intersection, the action set \( A = \{ \text{switch to phase 1}, \text{switch to phase 2}, \text{switch to phase 3}, \text{switch to phase 4} \} \).

At the same time, considering the safety problems caused by insufficient pedestrian crossing time, insufficient motor vehicle passing time, and the long waiting time of pedestrians or vehicles, the minimum green and maximum green of pedestrian crossing or motor vehicle release duration time are rigidly constrained. After the phase duration time meets the minimum green, the action decision is started, and the phase must be switched when
the maximum green is exceeded. The values and calculation methods of minimum green and maximum green refer to the results of Liu [45] and Wang [46]. At the same time, the unit extension time is usually taken as 3–5 s [47]. Considering that the saturated time headway of the bus (a time gap between the beginning of one vehicle and the beginning of the next) is larger than the car, the unit extension time is taken as 6 s.

2.2. Behavior Action Space Construction

The phase number is used as the action set. When the next phase obtained by the agent decision is the current phase, it means that the current phase extends the unit green light time; when the next phase is not the current phase, the phase transitions are performed, and the switching transitions are made with a fixed yellow light time and full red time. For example, at a four-phase intersection, the action set $A = \{\text{switch to phase 1}, \text{switch to phase 2}, \text{switch to phase 3}, \text{switch to phase 4}\}$.

At the same time, considering the safety problems caused by insufficient pedestrian crossing time, insufficient motor vehicle passing time, and the long waiting time of pedestrians or vehicles, the minimum green and maximum green of pedestrian crossing or motor vehicle release duration time are rigidly constrained. After the phase duration time meets the minimum green, the action decision is started, and the phase must be switched when the maximum green is exceeded. The values and calculation methods of minimum green and maximum green refer to the results of Liu [45] and Wang [46]. At the same time, the unit extension time is usually taken as 3–5 s [47]. Considering that the saturated time headway of the bus (a time gap between the beginning of one vehicle and the beginning of the next) is larger than the car, the unit extension time is taken as 6 s.

**Figure 1.** DTSE method represents bus position in the east approach.

**Figure 2.** Key lane-based method represents bus position in the east approach.

2.3. Action Reward Function Construction

For the bus priority scenario, considering that the main purpose of bus priority is to reduce the bus delay (evaluation index) as much as possible without significantly increasing the vehicle delay (constraint index) as much as possible. Therefore, by balancing the contradiction between them, this paper designs a set of multi-objective reward functions with a unified scale.

In terms of the selection of evaluation indexes, considering that the calculation of delay indexes is lagging, it is difficult to meet the requirements of real-time feedback rewards. Taking into account the real-time availability of indexes, this paper uses the number of vehicles passing in the action execution as the evaluation index. At the same time, by combining the vehicle networking environment of the bus, the busload can be obtained,
and the average passenger capacity of the car can be converted into the number of cars with a unified scale:

\[ rv = \sum_{i=1}^{vn} \frac{b_i}{cl} + N_v \]

where \( rv \), \( vn \), and \( N_v \) denote the number of standard cars, buses and cars passing through the intersection during the period of action execution respectively; \( b_i \) indicates the number of passengers on bus \( i \); \( cl \) represents the average passenger capacity of a car, which is taken as two in this paper.

In the selection of constraint indexes, the main factors such as traffic efficiency, safety and fairness are considered. The growth of the number of queuing vehicles \( r_q \), the loss caused by the stop of the bus in front of the queue at the end of the original phase due to phase transition \( r_s \), the maximum red time of motor vehicles \( rc \), and the maximum red time of pedestrians \( rp \) are selected as constraint indexes:

\[ r_q = \sum_{i=1}^{n} q_i^t - q_i^{t-1} \]

\[ r_s = \sum_{i=1}^{ln} \frac{bl_{qj}}{cl} \]

\[ rc = \sum_{i=1}^{m} \frac{\max(Rc_{it} - Rc_m, 0)}{hd} \]

\[ rp = \sum_{i=1}^{m} \frac{\max(Rp_{it} - Rp_m, 0)}{hd} \]

where \( q_i^t \) denotes the maximum number of queuing vehicles in phase \( i \) at time \( t \); \( ln \) denotes the number of lanes corresponding to the release direction in the current release phase at time \( t \); \( bl_{qj} \) denotes the number of passengers carried by the first bus in lane queue \( j \) corresponding to the current release phase at time \( t \); \( Rc_{it} \) denotes the duration of the red light of the motor vehicle in phase \( i \) at time \( t \); \( Rc_m \) denotes the maximum red time of the motor vehicle; \( hd \) denotes the saturation time headway, which is taken as 2 s [17]; \( Rp_{it} \) denotes the duration of the red light of pedestrians in phase \( i \) at time \( t \); \( Rp_m \) denotes the maximum red time of pedestrians.

Therefore, the unified scale multi-index reward function proposed in this paper is designed as:

\[ rt = rv - rq - rs - rc - rp \]

3. Bus Signal Priority Control Reinforcement Learning Agent Training

Considering that the traffic state space is difficult to be exhaustively characterized by the traditional Q-value table, this paper uses a deep Q network (DQN) to solve the problem of Q-value table storage. At the same time, by employing the mechanism of experience playback [48], the correlation of data in agent training is eliminated, ensuring the fast iterative convergence of the training. The deep Q network adopts a three-layer network structure. The agent training framework is shown in Figure 3.

The architecture of this model includes the following modules: state extraction module, Deep Q network module, action selection module, action execution module, reward calculation module, experience pool, and model training module.

Among them, the state extraction module observes the bus state, intersection operation state, and signal timing state at the current moment, and concatenates them as the intersection state vector \( S \), which is taken as the inputs of the Deep Q network module, and the output of the Deep Q network module is the probability Q value for each action (phase). Then, through the action selection module, we select the next phase based on the greedy strategy. At the same time, the signal controller changes the green light duration time through the action execution module. After the execution is completed, the state
extraction module observes and extracts the intersection state vector again and runs the cycle according to the above process. After each decision action is executed, the reward value $R$ is calculated through the reward calculation module, and the previous state, action, next state, and reward are saved to the experience pool. In every step, the model training module randomly selects a certain number of samples in the experience pool to train and update the deep Q network.

Figure 3. Bus signal priority control under the deep reinforcement learning framework.

The detailed training process is shown in the Algorithm 1:

**Algorithm 1: Detailed training process**

1: **Build** SUMO simulation environment
2: **Initialize** network $net$, the experience pool $M$, observation interval $N_o$, update interval $N_u$, iterations $N$, single simulation duration $N_t$, decision time point $s_n$, action $a_n$
3: **For** $i = 0: N$
4:     **For** step = 0: $N_t$
5:         **If** step = $s_n$ **Then**
6:             Observe the traffic state $S_l$
7:             Compute the next action $a_n$ based on $S_l$
8:             Execute action $a_n$
9:             Update $s_n$
10:            Observe the traffic state $S_n$
11:            Compute the reward $R$
12:            Save $(S_l, a_n, S_n, R)$ to the memory pool $M$
13:            Count = Count + 1
14:            Step = Step + 1
15:        **End if**
16:     **If** Count > $N_b$ and $N_b \% N_u == 0 **Then**
17:         Randomly select training samples in $M$
18:         Using $net$ computes predictive Q value for all actions
19:         $Q_t = R + \Gamma \cdot \text{Max}[Q(S_n, \text{all actions})]$
20:        **Train and update** $net$
21:        **End if**
22:    **End for**
23: **End for** $net \ M$.
4. The Analysis of Experiments
4.1. Construction of Simulation Test Environment

The simulation experimental intersection is Luotian/Xinhu intersection in Shenzhen, Guangdong Province, China. It is surrounded by Bao’an Stadium and large residential areas, with dense bus lines passing through intersections. The intersection is shown in Figure 4.

![SUMO simulation environment of Luotian–Xinhu intersection.](image)

As shown in Figure 5, the current existing traffic signal plan of this intersection is a fixed-time signal plan. Due to its proximity to the nearby Bao’an Stadium and large residential areas, the intersection has a large number of pedestrians. So, the traffic signal plan of the intersection should not only consider motor vehicles but also consider cross-street pedestrians. Compared to the control strategy for straight-ahead directions in one phase, and left turns (from opposite approach) in another phase, the current existing traffic signal planning (as shown in Figure 5) has the following advantages: (1) the waiting time for pedestrians to cross the street can be reduced; (2) the waiting time of pedestrians on the central safety island can also be reduced. In Figure 5, the double-arrow dotted line indicates the direction of pedestrians crossing, and pedestrian twice crossing the street, and the single-arrow solid line indicates the direction of motor vehicles passing. In a fixed-time control, the maximum green time of each phase is used as the actual green light time of each phase. This paper still follows the current existing traffic signal planning. On the one hand, the current traffic signal planning is the optimal planning under the long-term practice of professional signal timing social service teams. On the other hand, compared with the current traffic signal planning for a fixed-time control, only the pedestrians in 41P and 31P will increase the waiting time of 31–38 s in the central safety island (1 m × 6 m). It would not significantly increase the risk of pedestrians staying on the central safety island.

According to the definition of the traffic state space of the intelligent agent in 2.1, and considering that the maximum vehicle queue length at Luotian/Xinhu intersection in the real world is less than 100 m, to order to reduce the representation dimension of the state space and considering the redundant design, so only describe the traffic state within a range of 180 m in Luotian/Xinhu intersection, and the state vector S can be represented by a 1 × 262 vector. And the state parameters, such as bus location and busload, queuing vehicles, the information of signal timing, can be obtained in the bus networking environment.
The bus location can be obtained based on the positioning information; the busload can be obtained by counting the on-board camera in the bus; the number of queuing vehicles can be obtained by radar vehicle detectors; and the signal timing state, such as the current phase green time, can be obtained by reading the information from the traffic signal controller.

Figure 5. Signal control scheme.

The experimental traffic data at the intersection were obtained from the manual survey data from 12:00 to 13:00 on 22 November 2021, as shown in Table 1. The busload data were based on the manual statistics of the bus video, as shown in Table 2. In Tables 1 and 2, column “Direction” is defined as “entrance direction: turn direction”. A, L, S, and R indicate turn around, turn left, go straight, and turn right, respectively.

Table 1. Intersection traffic flow statistics.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Passing Cars (veh/h)</th>
<th>Passing Buses (veh/h)</th>
<th>Direction</th>
<th>Passing Cars (veh/h)</th>
<th>Passing Buses (veh/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE: A</td>
<td>40</td>
<td>0</td>
<td>SE: A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>NE: L</td>
<td>55</td>
<td>3</td>
<td>SE: L</td>
<td>97</td>
<td>5</td>
</tr>
<tr>
<td>NE: S</td>
<td>62</td>
<td>6</td>
<td>SE: S</td>
<td>368</td>
<td>28</td>
</tr>
<tr>
<td>NE: R</td>
<td>52</td>
<td>2</td>
<td>SE: R</td>
<td>84</td>
<td>5</td>
</tr>
<tr>
<td>SW: A</td>
<td>21</td>
<td>0</td>
<td>NW: A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>SW: L</td>
<td>115</td>
<td>8</td>
<td>NW: L</td>
<td>47</td>
<td>2</td>
</tr>
<tr>
<td>SW: S</td>
<td>167</td>
<td>11</td>
<td>NW: S</td>
<td>218</td>
<td>17</td>
</tr>
<tr>
<td>SW: R</td>
<td>66</td>
<td>2</td>
<td>NW: R</td>
<td>54</td>
<td>2</td>
</tr>
</tbody>
</table>

The algorithm was built on the Windows 10 operating system with i7-9700 CPU and 16.0 GB memory, implemented on python 3.8.5. The neural network was built by TensorFlow 2.8.0, the traffic operation simulation environment was SUMO 1.8.0. The simulation time is 3600 s for an iteration and there are 600 iterations in total. The deep neural network adopts a three-layer structure, with 262 nodes in the input layer, 200 nodes in the hidden layer, and 4 nodes in the output layer. The neural network parameters are updated every 10 decision steps after 100 decision steps. The capacity of the experience pool is set to 10,000, and 150 samples are randomly sampled to train the neural network each time.

The learning rate of the agent is 0.0001, and the empirical loss rate in the target Q value estimation is 0.9, adopting the greedy strategy in the learning process. The initial exploration probability is set to 0.5, the final exploration probability is set to 0.0001, and the number of attenuation steps is 50,000. The exploration probability is linearly attenuated, and the number of attenuation steps is set to 50,000.

In the simulation, the traffic flow models such as the car-following model and the lane-change model all adopt the default settings in Sumo, and the vehicle parameters such as the speed distribution and acceleration also take the default settings in SUMO. The signal control parameters, such as maximum green time, and minimum green time for each stage were used in Figure 5.
Table 2. Intersection busload statistics.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Bus ID</th>
<th>Busload</th>
<th>Direction</th>
<th>Bus ID</th>
<th>Busload</th>
<th>Direction</th>
<th>Bus ID</th>
<th>Busload</th>
<th>Direction</th>
<th>Bus ID</th>
<th>Busload</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE: L</td>
<td>1</td>
<td>33</td>
<td>SW: S</td>
<td>5</td>
<td>7</td>
<td>SE: S</td>
<td>10</td>
<td>28</td>
<td>SE: R</td>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>NE: L</td>
<td>2</td>
<td>30</td>
<td>SW: S</td>
<td>6</td>
<td>20</td>
<td>SE: S</td>
<td>11</td>
<td>10</td>
<td>NW: L</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>NE: L</td>
<td>3</td>
<td>27</td>
<td>SW: S</td>
<td>7</td>
<td>11</td>
<td>SE: S</td>
<td>12</td>
<td>20</td>
<td>NW: L</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>NES</td>
<td>1</td>
<td>14</td>
<td>SW: S</td>
<td>8</td>
<td>7</td>
<td>SE: S</td>
<td>13</td>
<td>14</td>
<td>NW: S</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>NE: S</td>
<td>2</td>
<td>27</td>
<td>SW: S</td>
<td>9</td>
<td>28</td>
<td>SE: S</td>
<td>14</td>
<td>18</td>
<td>NW: S</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>NE: S</td>
<td>3</td>
<td>29</td>
<td>SW: S</td>
<td>10</td>
<td>26</td>
<td>SE: S</td>
<td>15</td>
<td>9</td>
<td>NW: S</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>NES</td>
<td>4</td>
<td>26</td>
<td>SW: S</td>
<td>11</td>
<td>32</td>
<td>SE: S</td>
<td>16</td>
<td>11</td>
<td>NW: S</td>
<td>4</td>
<td>31</td>
</tr>
<tr>
<td>NE: S</td>
<td>5</td>
<td>10</td>
<td>SW: R</td>
<td>1</td>
<td>19</td>
<td>SE: S</td>
<td>17</td>
<td>7</td>
<td>NW: S</td>
<td>5</td>
<td>29</td>
</tr>
<tr>
<td>NES</td>
<td>6</td>
<td>19</td>
<td>SW: R</td>
<td>2</td>
<td>20</td>
<td>SE: S</td>
<td>18</td>
<td>26</td>
<td>NW: S</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>NE: R</td>
<td>1</td>
<td>32</td>
<td>SE: L</td>
<td>1</td>
<td>17</td>
<td>SE: S</td>
<td>19</td>
<td>16</td>
<td>NW: S</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>NE: R</td>
<td>2</td>
<td>26</td>
<td>SE: L</td>
<td>2</td>
<td>11</td>
<td>SE: S</td>
<td>20</td>
<td>6</td>
<td>NW: S</td>
<td>8</td>
<td>33</td>
</tr>
<tr>
<td>SW: L</td>
<td>1</td>
<td>33</td>
<td>SE: L</td>
<td>3</td>
<td>34</td>
<td>SE: S</td>
<td>21</td>
<td>20</td>
<td>NW: S</td>
<td>9</td>
<td>31</td>
</tr>
<tr>
<td>SW: L</td>
<td>2</td>
<td>7</td>
<td>SE: L</td>
<td>4</td>
<td>24</td>
<td>SE: S</td>
<td>22</td>
<td>6</td>
<td>NW: S</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>SW: L</td>
<td>3</td>
<td>9</td>
<td>SE: L</td>
<td>5</td>
<td>17</td>
<td>SE: S</td>
<td>23</td>
<td>31</td>
<td>NW: S</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>SW: L</td>
<td>4</td>
<td>9</td>
<td>SE: S</td>
<td>1</td>
<td>34</td>
<td>SE: S</td>
<td>24</td>
<td>12</td>
<td>NW: S</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>SW: L</td>
<td>5</td>
<td>19</td>
<td>SE: S</td>
<td>2</td>
<td>11</td>
<td>SE: S</td>
<td>25</td>
<td>16</td>
<td>NW: S</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>SW: L</td>
<td>6</td>
<td>6</td>
<td>SE: S</td>
<td>3</td>
<td>22</td>
<td>SE: S</td>
<td>26</td>
<td>6</td>
<td>NW: S</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td>SW: L</td>
<td>7</td>
<td>30</td>
<td>SE: S</td>
<td>4</td>
<td>32</td>
<td>SE: S</td>
<td>27</td>
<td>10</td>
<td>NW: S</td>
<td>15</td>
<td>29</td>
</tr>
<tr>
<td>SW: L</td>
<td>8</td>
<td>5</td>
<td>SE: S</td>
<td>5</td>
<td>7</td>
<td>SE: S</td>
<td>28</td>
<td>10</td>
<td>NW: S</td>
<td>16</td>
<td>26</td>
</tr>
<tr>
<td>SW: S</td>
<td>1</td>
<td>13</td>
<td>SE: S</td>
<td>6</td>
<td>6</td>
<td>SE: R</td>
<td>1</td>
<td>30</td>
<td>NW: S</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>SW: S</td>
<td>2</td>
<td>31</td>
<td>SE: S</td>
<td>7</td>
<td>7</td>
<td>SE: R</td>
<td>2</td>
<td>27</td>
<td>NW: R</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>SW: S</td>
<td>3</td>
<td>8</td>
<td>SE: S</td>
<td>8</td>
<td>8</td>
<td>SE: R</td>
<td>3</td>
<td>7</td>
<td>NW: R</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>SW: S</td>
<td>4</td>
<td>32</td>
<td>SE: S</td>
<td>9</td>
<td>10</td>
<td>SE: R</td>
<td>4</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2. Training Results Analysis

The simulation time of 3600 s is an iteration. Six indexes are selected as traffic efficiency indexes, namely, the cumulative travel time of all vehicles, cumulative waiting time of all vehicles, cumulative travel time of all buses, cumulative waiting time of all buses, cumulative waiting time of all passengers, and cumulative travel time of all passengers. Figure 6 shows the curve of the prediction error and the number of iterations of the neural network. Figure 7 shows the curve of the six traffic efficiency indexes and the number of iterations. The results show that with the increase in the number of iterations, the prediction error and traffic efficiency index of the neural network shows a downward trend. When the number of iterations reaches 330, the prediction error and six traffic efficiency indexes of the neural network converge, indicating the effectiveness of the method. Through calculation, comparing the first 300 iterations with the last 300 iterations, the cumulative travel time of all vehicles was reduced by 20%, the cumulative waiting time of all vehicles was reduced by 53%, the cumulative travel time of all buses was reduced by 16%, the cumulative waiting time of all buses was reduced by 46%, the cumulative waiting time of all passengers was reduced by 19%, and the cumulative travel time of all passengers reduced by 51%.
4.3. Comparison of Intelligent Agent Operation Effect

To verify the effect, the proposed method was compared with a fixed-time control and an actuated control. In the fixed-time control, as shown in Figure 5, the signal planning uses the maximum green time as the duration time of each phase. In the full actuated control, the signal phase, the minimum green time, and the maximum green time are also shown in Figure 5. The green light time of phase changes without changing the sequence of phases. The unit green light extension time is taken as 6 s, and the inductive loop detector is set at the range of 30 m from the stop line. When the bus or other vehicle is detected within a preset time interval of 6 s, the phase will be extended until the maximum green time. When no vehicle is detected within the preset time interval of 6 s, the phase will be changed. In the experiment, 10 groups of random seeds were added to ensure the fairness of the experiment by simulating 10 groups of different departure rules under the specified traffic flow conditions, and the average value of 10 groups of result data was taken to
compare the operation effect of three methods (as shown in Figure 8). The results of the proposed method show that the average cumulative travel time of all vehicles is 10.2% less than that of a fixed-time control and 3.4% less than that of an actuated control. The average cumulative waiting time of all vehicles is reduced by 35.9% compared with a fixed-time control and 14.3% with an actuated control. The average cumulative travel time of buses is 9.7% less than a fixed-time control and 6.7% less than an actuated control. The average cumulative waiting time of buses is 36.9% less than a fixed-time control and 27.2% less than an actuated control. The average cumulative waiting time of all passengers is reduced by 9.9% compared with a fixed-time control and 3.9% with an actuated control. The average cumulative travel time of all passengers is reduced by 35.9% compared with a fixed-time control and 16.7% with an actuated control.

![Figure 8: Effect comparison between three methods: fixed-time control method, actuated control method and DQN method: fixed-time control method, actuated: (a) cumulative travel time of all vehicles; (b) cumulative waiting time of all vehicles; (c) cumulative travel time of all buses; (d) cumulative waiting time of all buses; (e) cumulative travel time of all passengers; (f) cumulative waiting time of all passengers.](image-url)
We designed “high”, “shoulder”, and “low” traffic demand scenarios to verify the adaptability of our algorithm. The traffic flow of “shoulder” traffic demand is shown in Table 1, the traffic flow of “high” and “low” traffic demand increases and decreases by 10% proportionally, respectively on the basis of “shoulder” traffic demand. In “high”, “shoulder”, and “low” traffic demand, the DQN control method proposed in this paper is superior to the fixed-time control method and actuate control method. As shown in Table 3, the average travel time of all vehicles can be reduced by 1.6–10.6%; the average waiting time of all vehicles can be reduced by 12.1–38.3%; the average travel time of buses can be reduced by 5.5–11.9%; the average cumulative waiting time of buses can be reduced by 35.9–37.0%; the average waiting time of passengers can be reduced by 2.4–10.8%; and the average travel time of passengers can be reduced by 15.4–50.7%.

### Table 3. Effect comparison in three traffic demand.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Traffic Demand</th>
<th>All Buses</th>
<th>All Vehicles</th>
<th>All Passengers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Demand</td>
<td>Travel Time (s)</td>
<td>Waiting Time (s)</td>
<td>Travel Time (s)</td>
</tr>
<tr>
<td>Fixed-time control</td>
<td>high</td>
<td>14,066</td>
<td>3317</td>
<td>224,430</td>
</tr>
<tr>
<td></td>
<td>shoulder</td>
<td>11,553</td>
<td>3119</td>
<td>200,843</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>10,808</td>
<td>2814</td>
<td>178,793</td>
</tr>
<tr>
<td>Actuated control</td>
<td>high</td>
<td>13,877</td>
<td>2901</td>
<td>212,796</td>
</tr>
<tr>
<td></td>
<td>shoulder</td>
<td>11,197</td>
<td>2705</td>
<td>186,728</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>10,566</td>
<td>2559</td>
<td>163,089</td>
</tr>
<tr>
<td>DQN</td>
<td>high</td>
<td>12,388</td>
<td>2155</td>
<td>200,460</td>
</tr>
<tr>
<td></td>
<td>shoulder</td>
<td>10,429</td>
<td>1966</td>
<td>180,316</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>9984</td>
<td>1774</td>
<td>160,398</td>
</tr>
</tbody>
</table>

### 5. Conclusions

This paper proposes a deep Q-learning control method for multi-flow bus signal priority at intersections in the bus networking environment. By taking into account the needs of social vehicles and pedestrians crossing the street, it provides priority access for each traffic flow of buses at intersections. Based on the core objective of bus priority, the proposed method optimizes the design of state, action and reward functions, building a bus signal priority deep Q-learning simulation platform through the COM interface of the SUMO simulation platform and Python programming software. As shown in the simulation experiment, compared with a fixed-time control and an actuated control, the waiting time of buses decreased by 36.9% and 27.2%, respectively, and the average waiting time of all vehicles decreased by 35.9% and 16.7%, respectively.

Although the proposed method has achieved some achievement in addressing the signal control problem considering bus priority, it is still less successful and needs further development. Moreover, there is still a risk that bus priority control can lead to increased emissions and thus be in conflict with sustainability. Therefore, with careful consideration of sustainability, further work will be focused on researching bus priority control as well as trunk and regional bus priority control.

### Author Contributions:

Conceptualization, W.S.; methodology, W.S.; software, L.Z.; validation, L.Z.; formal analysis, R.D.; investigation, R.D.; resources, H.W.; data curation, H.W.; writing—original draft preparation, W.S.; writing—review and editing, J.W.; visualization, H.W.; supervision, J.W.; project administration, J.W.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

### Funding:

This research was supported by [the Guangdong Basic and Applied Basic Research Foundation] under grant number [2022A15151111039]. The APC was funded by [the Guangdong Basic and Applied Basic Research Foundation].

### Institutional Review Board Statement:

Not applicable.

### Informed Consent Statement:

Not applicable.
References


41. Xue, P. Research on Bus Priority Signal Control Based on Reinforcement Learning. Ph.D. Dissertation, Xi’an University of Technology, Xi’an, China, 2021. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.