Journal of Network Intelligence ©2023 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **8**, Number **3**, August **2023**

Intelligent Signal Timing Optimization Model with Bus Priority Based on Multi-Process Experience Pool

Lyuchao Liao

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology
Fujian University of Technology
Fuzhou 350118, China
fjachao@gmail.com

Penghao Tang*

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology
Fujian University of Technology
Fuzhou 350118, China
tangpenghao412@gmail.com

Zhaolin Zhao

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology
Fujian University of Technology
Fuzhou 350118, China
zhaozl@fjut.edu.cn

Qi Zheng

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology
Fujian University of Technology
Fuzhou 350118, China
451443781@qq.com

Zhengrong Li

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology
Fujian University of Technology
Fuzhou 350118, China
lee_zr@outlook.com

Yintian Zhu

Fujian Provincial Universities Key Laboratory of Industrial Control and Data Analysis
Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology
Fujian University of Technology
Fuzhou 350118, China
2469354563@qq.com

*Corresponding author: Penghao Tang

Received October 2, 2022, revised December 20, 2022, accepted February 19, 2023.

Abstract. *Nowadays, the imbalance between urban traffic demand and road infrastructures is getting more serious, resulting in serious urban traffic congestion. Unlike private cars, buses can carry more passengers under the same road segment with lower energy consumption. However, existing studies usually ignore the impacts of vehicle turn signal states on action selection and the impacts of experience pool sampling efficiency on model training. Therefore, with the increasing number of priority buses on the road, it will encounter the problem of bus response conflict. In this work, we propose an intelligent signal timing optimization model with bus priority based on a multi-process experience pool, named MP-3DQN. The model considers the turn signal state of the vehicle and the bus priority switching scheme, which can help predict the vehicles' future direction and provide a basis for the agent in action selection. We introduce a multi-process parallel method to break the correlation between samples and improve the sampling efficiency of sample data. In addition, we propose a multi-objective reward function to make the agent more sensitive to the changing detection, and we constrain the green light duration and optimize the average delay of vehicles and intersection throughput. Extensive experiments were conducted with SUMO on intersection roads in Fuzhou City, and the results show that MP-3DQN could significantly reduce the average delay time of buses without affecting traffic efficiency.*

Keywords: Bus priority signal, Deep reinforcement learning, Multi-process, Traffic light control

1. Introduction.

Today, the rapid increase in car ownership has seriously affected urban traffic conditions, which results in serious urban congestion [1]. Compared with private cars, buses have more advantages in passenger capacity and energy consumption; therefore, it is an effective way to alleviate traffic congestion by developing public transport priority. However, the current bus system still has several problems, such as low punctuality, long waiting times, and insufficient transport capacity in rush hours. These problems reduce the traffic efficiency of buses. Therefore, implementing bus signal priority is an essential strategy to improve operational efficiency and schedule adherence to the bus system.

With the rapid development of artificial intelligence technology, Deep Reinforcement Learning (DRL) technology has made a breakthrough in traffic signal control [2]. Compared with Reinforcement Learning (RL) [3], DRL technology is better at dealing with high-dimensional real-time traffic conditions [4, 5]. However, when using the DRL method to define the state in traffic signal control problems, most studies ignore the influence of vehicle turn signal state on agent selection. In addition, during the training phase of the model, it is easy to ignore the influence of sample correlation and sampling efficiency in the experience pool. Moreover, with the increasing number of priority buses on the road, it will encounter the problem of bus response conflict.

This work proposes an intelligent signal timing optimization model with bus priority based on a multi-process experience pool, named MP-3DQN. This model aims to reduce the delay of bus and bus service levels. We employed the multi-process parallel method to improve the sampling efficiency and break the correlation between samples in the experience pool. This method can generate multiple simulation environments simultaneously and store all generated data in the experience pool. To ensure agents can predict future directions of vehicles and provide a reasonable basis to select actions, we consider turn signal information of vehicles in state space. We consider a bus priority signal switching scheme in the state space to solve the problem of bus response conflict at intersections. Moreover, the method of multi-objective reward function is adopted to make the agent more sensitive to the change of reward value and optimize the indicators of average transit delay and intersection throughput simultaneously.

In this work, we combine multi-process experience pool optimization as a framework to improve the efficiency of the bus system.The experimental benefit of this work is to optimize the traffic efficiency of buses at intersections, reduce the delay of buses, and improve the throughput of intersections. This can not only alleviate the problem of urban congestion, but also improve the efficiency of residents' travel. The model is tested on SUMO, and the results show the effectiveness and efficiency of MP-3DQN. The main contributions of this work are as follows:

(1) Aiming at the low sampling efficiency of the experience pool, we propose a multi-process parallel method to improve the sampling efficiency of the experience pool and accelerate model convergence;
(2) Aiming at the problem that the existing state information description is not comprehensive enough, we consider vehicle turn signal information and bus priority phase switching scheme so that the agent can choose actions more reasonably;
(3) We propose the multi-objective reward function method to constrain the green light duration and optimize average transit delay duration and intersection throughput to make agents better feedback on the reward.

The remainder sections are organized as follows: Section 2 describes related work on applying deep reinforcement learning in bus priority signal control; Section 3 presents the agent based on deep reinforcement learning for bus priority signal control; Section 4 describes the framework of the algorithm in detail; Section 5 discusses experimental results. Section 6 summarizes our work and provides ideas for future work.

2. **Related work.** Existing bus priority measures are classified into space and time. For space measures, some researchers set bus lanes or intermittent bus lanes to improve the bus operation efficiency directly [6, 7]. But it takes up a lot of road resources and affects the efficiency of other vehicles, which may lead to congestion at intersections [8]. Unlike space measures, time measures can improve the efficiency of the bus system without using road resources. For time measures, researchers proposed bus signal priority strategies to adjust traffic lights' signal timing at intersections to ensure buses can pass through intersections quickly [9]. Bus signal priority strategies are divided into three categories: 1) passive priority strategy; 2) active priority strategy; 3) adaptive strategy [10]. The passive priority strategy is to analyze historical traffic data to generate a signal control scheme. The strategy highly depends on the stability and accuracy of traffic demand and is not efficient for real-time changing traffic scenarios [11]. The active priority strategy is to adjust real-time signal timing to ensure buses get through intersections directly when buses are detected approaching intersections [12]. The adaptive strategy is developed based on real-time detected traffic flow data for signal timing optimization control [13].

DRL technology has achieved great success in intersection signal control in recent years. Zhang et al. [14] proposed the LVQ neural network quantum genetic optimization algorithm, which has a good effect on short-term traffic flow prediction. Zhang et al. [15] proposed a traffic prediction method based on short-term memory network, which has a good effect on the management of shared bicycles. Liao et al. [16] proposed a traffic flow prediction method with space-time attention, which plays an important role in urban traffic congestion and public travel route planning. Liao et al. [17] proposed a time difference penalized traffic signal timing method by request learning technique to balance safety and throughput capacity in traffic control system. Ma et al. [18] proposed a new depth transfer learning framework, which improves accuracy by combining convolutional neural network and sparse matrix. Kang et al. [19] proposed a Gaussian process regression (GPR), which has good generalization in dealing with high-dimensional, small sample and nonlinear problems. Kumar et al. [20] proposed a traffic detection method based on LSTM network and applied it in the field of intelligent transportation. Wu et al. [21] proposed an integrated learning algorithm and applied it to model prediction, and achieved good results. Liang et al. [22] proposed an algorithm for 3DQN (Double Dueling Deep Q-Network) to control intersection signal lights. The model takes the position and speed of vehicles as the state and takes the cumulative waiting time difference between two adjacent cycles as the reward. Experiments show that the algorithm can effectively reduce the average waiting time of vehicles. Genders and Razavi [23] proposed a state method of discrete traffic state encoding (DTSE). They employed the DTSE method to define the state, which includes the position and speed of vehicles. They use the change in cumulative vehicle delay as the reward. Experiments show that the method performs better in the average cumulative delay and queue length. Gao et al. [24] proposed an adaptive traffic signal algorithm. The algorithm automatically extracts vehicles' position and speed information and takes them as states. They use the staying time of vehicles at the intersection as a reward. Experiments show that the algorithm can reduce vehicle delay effectively. Gu et al. [25] proposed a DDQN with the dual-agent algorithm. The algorithm takes traffic flow as the reward and vehicle position as state. Experiments show that the algorithm can improve the traffic capacity of vehicles effectively.

Although the above methods have made significant breakthroughs in reducing the delay of vehicles, they still have some limitations. In the state space, they ignore the influence of vehicle turn signals on agent selection. In the reward function, they seldom consider the constraints of traffic light duration. Therefore, we consider the state of the vehicle turn signal in the state space. We propose a multi-objective reward function, which considers the maximum and minimum green light constraints. Finally, the experimental results show that we propose model achieved considerable results.

3. **Agent design of traffic signal control based on deep reinforcement learning.**

DRL is one of the most successful artificial intelligence models at present, which can deal with high-dimensional and large state space problems [26]. As shown in Figure 1, it is a traffic signal control model of deep reinforcement learning. The state is received from the environment. The agent can select the corresponding action from the current traffic state. Then, the environment conducts this action and gets the next state and related rewards. After that, the model is trained iteratively, and the signal control scheme is optimized continuously to maximize the reward.

3.1. **State.**

Agent learning performance highly depends on the accuracy and comprehensiveness of state definition. Therefore, extracting critical state information from complex traffic scenes is particularly important. To describe the state information more comprehensively and accurately, we set two state variables in this work: 1) the state of vehicle perception behavior; 2) the state of bus priority phase switching.

We define the state of vehicle perception behavior as $S_t^{(1)}$. We construct a matrix that contains the types, location distribution, and turn signal information of vehicles in four directions of intersections. As shown in Figure 2, we employed DTSE [23] method to mark the type and location information of vehicles on the lane. We divide the 150m entrance road into 30 cells on average. We set the length of private vehicles and buses to 5m and 15m. If there is no vehicle in the cell, it is marked as 0; if the cell is a private vehicle, it is marked as 1; if the cell is a bus, it is marked as 3. Moreover, as shown in Table 1, the vehicle's turn signal information is represented by a three-dimensional vector (x, y, z), where x represents left turn, y represents straight ahead, and z represents right turn.

As shown in Figure 2, the state of vehicle perception behavior in the west is shown as follows: yellow indicates that there are private cars and need to turn left, orange indicates

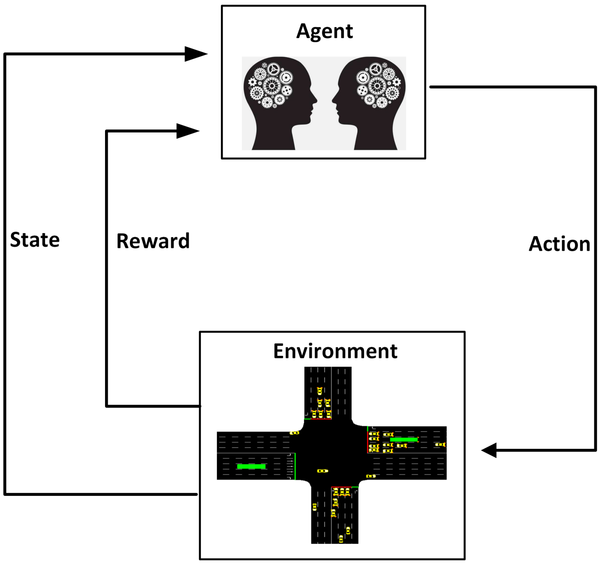


FIGURE 1. Traffic signal control based on deep reinforcement learning

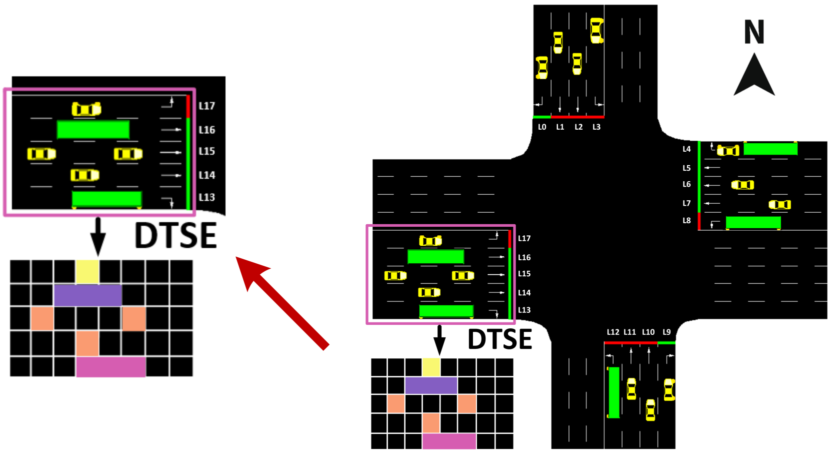


FIGURE 2. Partial schematic diagram of vehicle perception behavior state

that there are private cars and need to go straight, purple indicates that there are bus vehicles and need to go straight, and pink indicates that there are buses and need to turn right.

TABLE 1. Vehicle turn signal lamp state

Vehicle type	Turn left	Go straight	Turn right
Buses	(3,0,0)	(0,3,0)	(0,0,3)
Private vehicles	(1,0,0)	(0,1,0)	(0,0,1)

Finally, the first state $S_t^{(1)}$ integrates the state of vehicle perception behavior in the intersection, as shown in Equation 1.

$$S_t^{(1)} = [S_{N_{30*4}}, S_{E_{30*5}}, S_{S_{30*4}}, S_{W_{30*5}}]^T \tag{1}$$

where, $S_{N_{30*4}}$ represents the matrix of the north entrance, $30 * 4$ represents a matrix of 30 rows and 4 columns. 30 represents the average division of the entrance into 30 cells and 4 represents the number of the entrance. By analogy, $S_{E_{30*5}}$ represents a matrix of 30 rows and 5 columns for the eastern entrance road, $S_{S_{30*4}}$ represents a matrix of 30 rows and 4 columns for the southern entrance road, and $S_{W_{30*5}}$ represents a matrix of 30 rows and 5 columns for the western entrance road.

We define the state of bus priority phase switching as $S_t^{(2)}$, which obtains through the following two steps: first, we calculate the priority of each bus; then, we employed one-hot coding to adjust the signal phase.

First, we calculate the priority of each bus, as shown in Equation 2. When there are multiple priority requests, Lin et al. [12] proposed a method to determine the priority by the delay of each bus. Therefore, we calculate the priority of each bus by the current position and delay of each bus. We use SUMO simulation software to obtain the real-time position of the bus on the road, and use DTSE method to convert the position of the bus into a matrix form and express it in P_s. The method can ensure the bus with the largest P_r has a priority on the road.

$$P_r = P_s + D_e \tag{2}$$

where, P_r represents the priority level of the bus, P_s represents the position of the bus, D_e represents the delay of the bus.

Table 2. Bus priority switching phase method

Current phase	$S_t^{(2)}$							
	ph_1	ph_2	ph_3	ph_4	ph_5	ph_6	ph_7	ph_8
$WT - ET$	**1**	0	0	0	0	0	0	0
$WL - EL$	0	**1**	0	0	0	0	0	0
$NT - ST$	0	0	**1**	0	0	0	0	0
$NL - SL$	0	0	0	**1**	0	0	0	0
$WT - WL$	0	0	0	0	**1**	0	0	0
$NT - NL$	0	0	0	0	0	**1**	0	0
$ET - EL$	0	0	0	0	0	0	**1**	0
$ST - SL$	0	0	0	0	0	0	0	**1**

Then, we design a bus priority phase switching scheme containing 8 signal phases, as shown in Table 2. By comparing the current phase with the bus priority switching phase. If they are the same, we maintain the current phase state; otherwise, we need to use the one-hot method to set the phase position to be switched as 1 in Equation 3:

$$S_t^{(2)} = (ph_1, ph_2, ph_3, ph_4, ph_5, ph_6, ph_7, ph_8) \tag{3}$$

where, ph_1, ph_2, ph_3, ph_4, ph_5, ph_6, ph_7, ph_8 represent bus priority phase schemes. The schematic diagram corresponding to the bus flow direction is shown in Figure 3.

For example, when the current phase is $ST - SL$, but the bus with the largest P_r need to switch phase is $WL - EL$. We employed the one-hot method to set ph_2 as 1, and the remaining phase is set as 0. Finally, we obtain the state $S_t^{(2)}$, and $S_t^{(2)} = (0, 1, 0, 0, 0, 0, 0)$, as shown in Table 2.

3.2. **Action space.** The flexibility of action space has an important influence on the learning of agents. We set the time interval of the agent to select actions as 1 second, hence, the agent can select actions more flexibly. If the current action is the same as the previous action, it indicates that the duration of the current phase has been extended. If the current action is different from the previous one, it indicates that the current phase

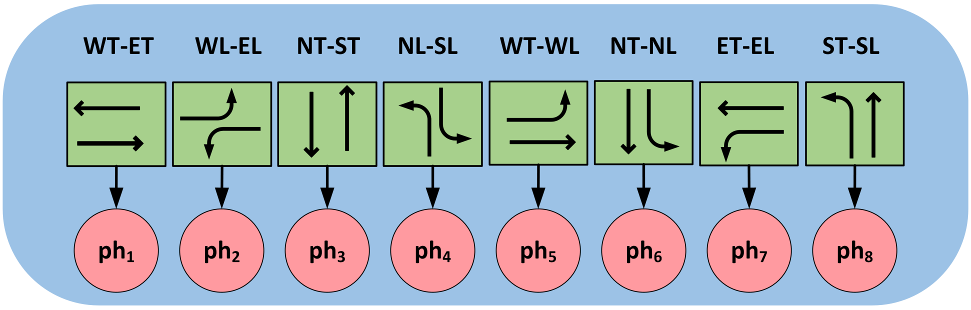
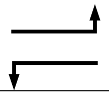
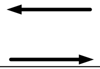
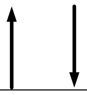
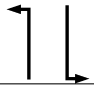
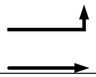
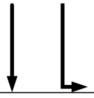
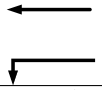
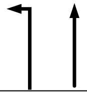


Figure 3. Bus priority phase switching direction diagram

scheme has changed. We need to automatically insert a 3-second yellow light phase scheme to ensure drivers have enough time to react. We design a switching scheme with 8 actions, as shown in Table 3. For example, if the action is a_0. The direction of the traffic flow corresponds to a_0 is $WT - ET$, which means that the vehicles in the east (E) and west (W) directions of the entrance road go straight (T). Moreover, the corresponding signal light color series of this action is ($GRRRGGGGRGRRRGGGGR$). In addition, we set the traffic lights in the right turn lane to permanent green.

Table 3. Action space of signal lights

Action	Traffic direction	Traffic light color	Graph of phase
a_0	$WT - ET$	$GRRRGGGGRGRRRGGGGR$	
a_1	$WL - EL$	$GRRRGRRRGGRRRGRRRG$	
a_2	$NT - ST$	$GGGRGRRRRGGGRGRRRR$	
a_3	$NL - SL$	$GRRGGRRRRGRRGGRRRR$	
a_4	$WT - WL$	$GRRRGRRRRGRRRGGGGG$	
a_5	$NT - NL$	$GGGGGRRRRGRRRGRRRR$	
a_6	$ET - EL$	$GRRRGGGGGGRRRGRRRR$	
a_7	$ST - SL$	$GRRRGRRRRGGGGGRRRR$	

3.3. **Reward function.** The reward function is important in the design of the model framework, which plays a vital role in the decision-making of agents' next action. Reward functions are divided into two methods: 1) single objective; 2) multi-objective. The single objective reward function is to optimize a traffic congestion indicator. For example, the average delay of vehicles, intersection throughput, and the queue length of vehicles [14, 22, 26]. In the real world, the optimization of traffic signal control requires comprehensive consideration of multiple traffic indicators. The multi-objective rewards can optimize multiple objective parameters at one time. Compared with the single objective method, the multi-objective reward function is more suitable for traffic signal control [27-29]. Therefore, we propose a multi-objective reward function, as shown in Equation 4, which contains the average delay of vehicles, intersection throughput, and the green light duration. We aim to minimize the delay of buses without affecting the throughput of intersections.

$$R_t = -R_{delay} + R_{throughput} - R_{green-min} - R_{green-max} \tag{4}$$

where, R_{delay} represents the average delay of vehicles, including buses and all other vehicles; $R_{throughput}$ represents intersection throughput, including buses and all other vehicles; $R_{green-min}$ represents the constraints of minimum green light duration; $R_{green-max}$ represents the constraints of the maximum green light duration.

(1) The first part is R_{delay}, as shown in Equation 5. We aim to reduce the delay of buses and improve the traffic efficiency of buses.

$$R_{delay} = \sqrt{\sum_{i=t-\triangle t}^{t} (\frac{1}{N_{bust}} \sum_{k=1}^{N_{bust}} y_k)} + \sqrt{\sum_{i=t-\triangle t}^{t} (\frac{1}{N_{allt}} \sum_{k=1}^{N_{allt}} y_k)} \tag{5}$$

where, N_{bust} represents the total number of buses on the entrance at time t, N_{allt} represents the number of all other vehicles at the entrance at time t. y_k represents the delay of the vehicle k. Because the duration of action selected by the agent is 1 second, the observation duration is too short to calculate vehicles accurately. Therefore, we use a duration period $\triangle t$ and set it as 20 seconds.

(2) The second part is $R_{throughput}$, as shown in Equation 6. We aim to improve the traffic efficiency of buses at intersections and ensure the traffic capacity of private vehicles.

$$R_{throughput} = \sum_{i=t-\triangle t}^{t} T_{bust} + \sum_{i=t-\triangle t}^{t} T_{allt} \tag{6}$$

where, T_{bust} represents the number of buses passing through the intersection at time t. T_{allt} represents the number of all other vehicles passing through the intersection at time t.

(3) The third part is $R_{green-min}$, as shown in Equation 7. If the green light duration is too short, which could increase the frequency of phase switching. It makes drivers stop before intersections more frequently.

$$R_{green-min} = \sum_{1}^{m} (G_{min} - L_{mt}, 0) \tag{7}$$

where, L_{mt} represents the duration of green light in lane m at time t. G_{min} represents the minimum green light duration. Xu et al. [30] set the minimum green light time for motor vehicles as 10 seconds, therefore, we set the minimum green light duration as 10 seconds.

(4) The fourth part is $R_{green-max}$, as shown in Equation 8. If the green light duration exceeds the tolerance time of drivers, it will increase the probability of the driver running the red light.

$$R_{green-max} = \sum_{1}^{m} (L_{mt} - G_{max}, 0) \tag{8}$$

where, G_{max} represents the maximum green light duration. Boltze and Friedrich [31] suggest that the maximum waiting time for motor vehicles is 120 seconds, and we set the maximum green light time to 120s.

4. **MP-3DQN Algorithm Framework.** In this work, we propose a MP-3DQN (Multi-process Parallel Double Dueling Deep Q Network) algorithm to deal with the bus signal control problem. The algorithm combines multi-process parallel, Dueling DQN, and DDQN to improve the overall performance of algorithm. We introduce DDQN [32] to solve the overestimation problem. We use two networks with different parameters to evaluate and select an action. Then, we employ the multi-process parallel method to improve the sampling efficiency of the experience pool. Finally, we adopt Dueling DQN to enhance the learning effect. The MP-3DQN algorithm framework is shown in Figure 4.

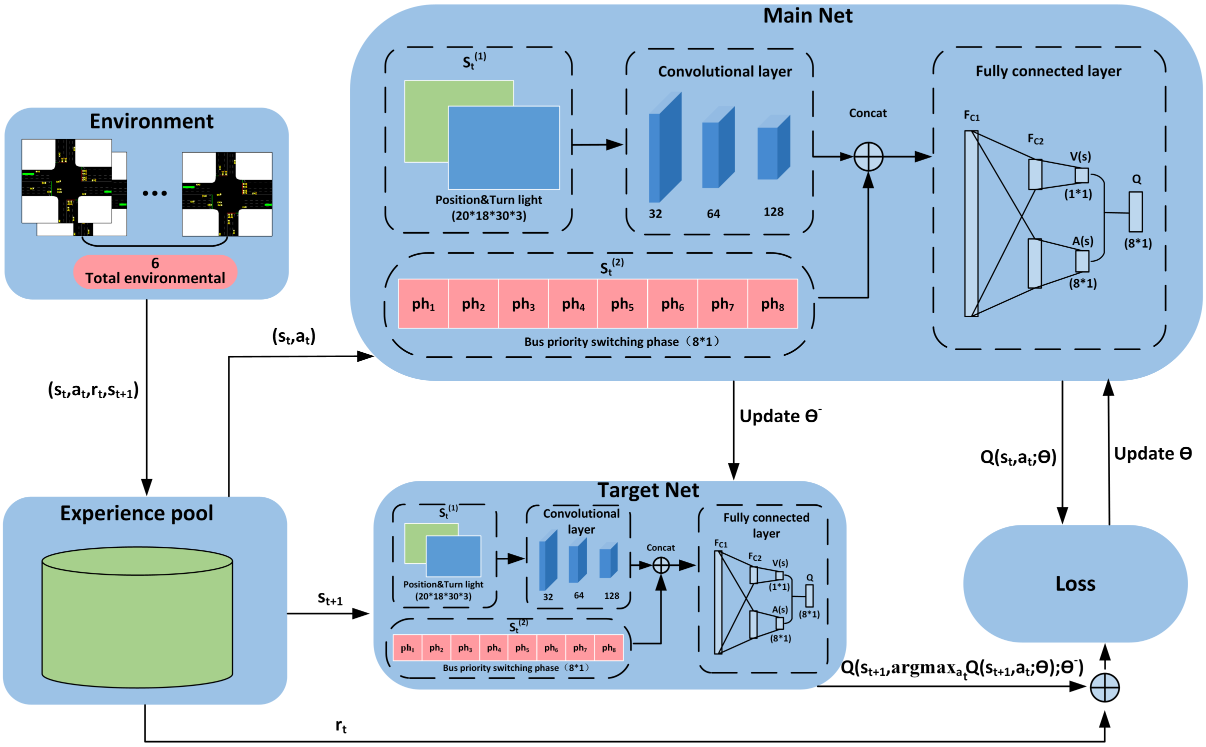


Figure 4. MP-3DQN Model Framework Diagram

MP-3DQN algorithm framework includes four parts: experience pool, main network, target network, and loss function. We use the experience pool to store the sample data. Since the DQN algorithm only generates one sample of data (s_t, a_t, r_t, s_{t+1}) in each iteration, it makes the growth rate of sample data slow, which affects the speed of model training. Therefore, we propose the multi-process parallel method to improve the sample sampling efficiency. In addition, it is an important way to improve the efficiency of model training by enhancing the independence between sample data [33]. When the sample data in the experience pool reaches a maximum value, we employ the random sampling method to extract some sample data from the experience pool for training.

The main network selects the actions of agents by three convolution layers and two full connection layers. First, we put the first state $S_t^{(1)}$ into three convolutional layers. After flattening the output result of convolutional layers, we connect it with the second state $S_t^{(2)}$. Then, we input the two connected states to F_{c1} and F_{c2} full connection layers. In addition, we employ Dueling DQN [26] to split the output results of F_{c2} into two parts: 1) the value of current state $V(s)$; 2) the advantages for each action $A(s, a)$. Finally, we add $V(s)$ and $A(s, a)$ to get a set of Q values, each Q value corresponds to an action, as shown in Equation 9.

$$Q(s_t, a_t; \alpha, \beta) = V(s_t; \alpha) + (A(s_t, a; \beta) - \frac{1}{|A|}\sum_{a_{t+1}} A(s_t, s_{t+1}; \beta)) \tag{9}$$

where α and β are the parameters of the two full connection layers. $V(s_t; \alpha)$ represents the estimation of the value function. $A(s_t, a; \beta)$ represents the estimation of the advantages of each action. In addition, we employed DDQN to solve the problem of overestimation. In DDQN, we use the main network to select actions and the target network to calculate the Q value of actions, as shown in Equation 10.

$$Y^{DDQN} = r_t + \gamma Q(s_{t+1}, argmax_{a_t} Q(s_{t+1}, a_t; \theta); \theta^-) \tag{10}$$

where θ represents the parameter of the main network. θ^- represents the parameter of the target network.

The target network has the same network structure as the main network. The difference between them is that the target network parameter is θ^-, and the main network parameter is θ. Furthermore, we use the target network to evaluate the actions.

We use Mean Square Error(MSE) as the loss function, as shown in Equation 11.

$$L(\theta) = E[(r_t + \gamma Q(s_{t+1}, argmax_{a_t} Q(s_{t+1}, a_t; \theta); \theta^-)) - Q(s_{t+1}, a_t; \theta)] \tag{11}$$

The pseudocode of MP-3DQN is shown in Algorithm 1. We propose MP-3DQN to deal with the problem of bus priority signal control at intersections. The method aims to reduce the delay of buses without affecting intersection throughput.

First, we adopt the multi-process parallel method to create m simulation environments. And, we get the state s_t through the m simulation environments. The agent uses a greedy strategy to select action a_t, and executes the action a_t to get a reward r_t. In addition, the environment produces the next state s_{t+1}. We store all sample data (s_t, a_t, r_t, s_{t+1}) into the experience pool. When the number of samples in the experience pool reaches a maximum value, we start to train the model. Each time we randomly select N samples from the experience pool for model training. We employ the queue method to remove the early sample data. This method ensures the newly generated data can be added to the experience pool.

Then, during the training of neural network, we input the first state $S_t^{(1)}$ into three convolutional layers. After flattening the output result of convolutional layers, we connect it with the second state $S_t^{(2)}$. And, we input the two connected states to F_{c1} and F_{c2} full connection layers. We use Dueling DQN to split the output results of F_{c2} into $V(s)$ and $A(s, a)$. Finally, we add $V(s)$ and $A(s, a)$ to get a set of Q values, and each Q value corresponds to an action.

Finally, we get the maximum Q value through the main network and select the corresponding action through the maximum Q value. We adopt the parameter θ to update the main network. We employ the parameter θ^- to update the target network. We use the target network to evaluate each action. In addition, we use MSE as a loss function to update the parameter θ.

5. System experiment and result analysis.

5.1. **Setting of the experimental environment.** In this work, we adopt Tensorflow-GPU-2.7.0 to build a neural network. And we employ the simulation software SUMO-1.8.0 to build road scenes at intersections. We choose the intersection of Qunzhong East Road and Wuyi Middle Road in Fuzhou as the traffic simulation scene, as shown in Figure 5. There are 18 entrance roads at the intersection, which are marked with $L_0, L_1, \ldots, L_{18}$ clockwise from the north. We set L_0, L_4, L_9, and L_{13} as right-turn lanes, L_3, L_8, L_{12}, and

Algorithm 1: Bus Priority Signal Control Algorithm

```
Input: max capacity of experience pool $M$, step number $i$, discount factor $\gamma$,
       greedy $\epsilon$, current samples of experience pool $T$, batch size $N$, total steps
       $S$
Output: the parameter $\theta$ of the main network
Create $m$ simulation environments
 Initialize step number $i = 0$
 Initialize the main network parameter $\theta$
 Initialize the target network parameter $\theta^-$
 while $i < S$ do
    if $T < M$ then
        According to $\epsilon$ greedy strategy to select an action $a_t$
         According to $a_t$ to get a reward $r_t$
         Obtain the next state $s_{t+1}$
         Store the data $(s_t, a_t, r_t, s_{t+1})$ into the experience pool
    end
    else
        Remove the oldest samples from the experience pool
         Store the latest data $(s_t, a_t, r_t, s_{t+1})$ in the experience pool
         Randomly select $N$ samples from the experience pool
         Start neural network training
         Calculate the $Q$ value of each action in the main network: $Q(s_t, a_t; \theta)$
         Select the maximum action in the main network: $argmax_{a_t} Q(s_t, a_t; \theta)$
         Calculate the $Q$ value of the action in the target network:
         $y_t = r_t + \gamma Q(s_{t+1}, argmax_{a_t} Q(s_{t+1}, a_t; \theta); \theta^-)$
         Calculate the loss function:
         $L(\theta) = E[(r_t + \gamma Q(s_{t+1}, argmax_{a_t} Q(s_{t+1}, a_t; \theta); \theta^-)) - Q(s_{t+1}, a_t; \theta)]$
         According to the loss function to update $\theta$
    end
    $i \leftarrow i + 1$
     end while
end
```

L_{17} as left-turn lanes. In addition, we set the remaining entrance roads as straight lanes. We set the length of the entrance road as 150 meters and divide it into 30 small grids. We set the length of buses and private cars to 15 and 5 meters, and the length of each bus corresponds to the size of three small grids. Moreover, we use detectors at each entrance lane to collect the traffic flow and the state of vehicles.

We design an eight-phase traffic signal to improve the utilization of traffic lights, as shown in Table 4. In addition, when the traffic light phase changes, we need to automatically insert a 3-second yellow light phase scheme to ensure drivers have enough time to react. The experimental data is collected from the intersection of Fuzhou East Qunzhong Road and Wuyi Middle Road during 17:30-18:30, March 16, 2022. The sampling interval of experimental data is 15min.

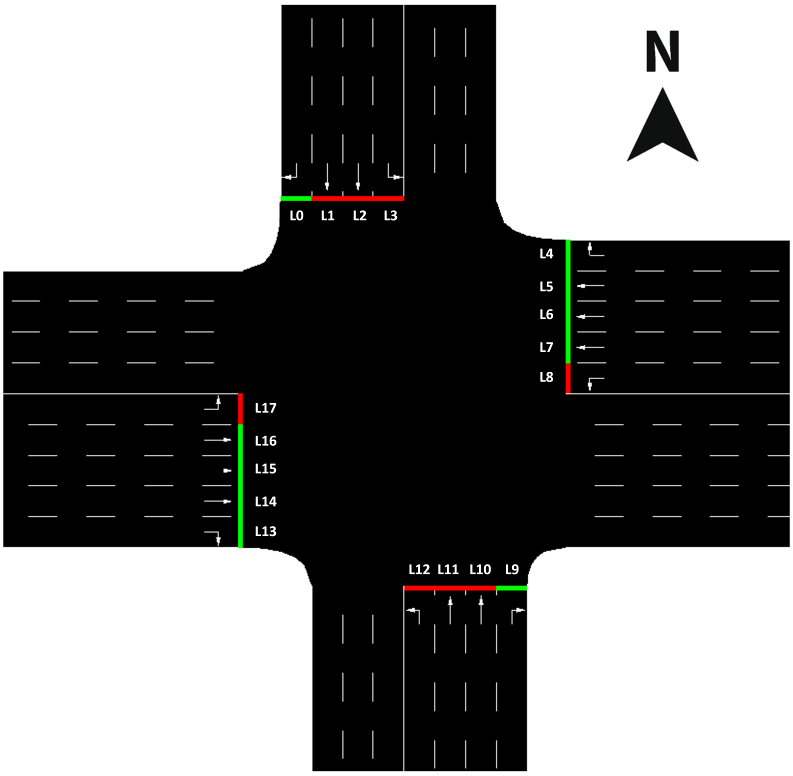
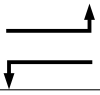
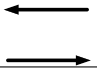
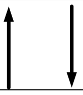
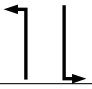
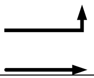
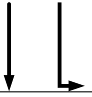
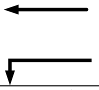
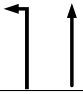


Figure 5. Road Network Structure

Table 4. Eight-phase of traffic signal

Phase code	Direction of phase	Graph of phase
1	$WT - ET$	
2	$WL - EL$	
3	$NT - ST$	
4	$NL - SL$	
5	$WT - WL$	
6	$NT - NL$	
7	$ET - EL$	
8	$ST - SL$	

In this work, we set the relevant parameters of the experiment, as shown in Table 5. We set the simulation time for each round to 6000 seconds and the number of iterations to 150 rounds. We set the action space to 8 to make the agent more flexible when selecting actions, so as to improve the throughput of the intersection. We set the capacity of the experience pool to 20000, the number of samples randomly selected each time to 64, and the number of simulation rounds to 6. These are to break the correlation between samples and accelerate the convergence of the model. We set the learning rate to 0.01 and the discount factor to 0.99. When selecting an action, we will randomly select an action with a linear decreasing probability. We set the greedy factor to a small value to prevent the action from changing too fast and failing to achieve the training effect. The number of simulation rounds is 150. The simulation duration of each round is 6000s. The maximum capacity of the experience pool is 20000. The total number of simulation environments is 6. The number of action spaces is 8. The number of random samples is 64. The greedy factor is 0.01. The learning rate is 0.1^6. The discount factor is 0.99.

Table 5. Initial setting of experimental parameters

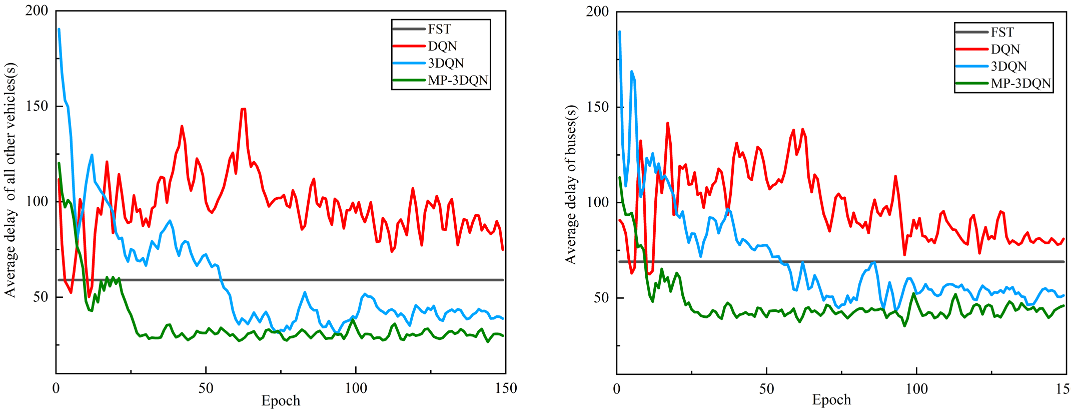
Parameter	Initial setting
Learning rate α	0.1^6
Random samples	64
Discount factor γ	0.99
Action space	8
Experience pool capacity M	20000
Total number of iteration rounds episode	150
Greedy factor ϵ	0.01
Total number of simulation environments m	6
Duration of the simulation cycle	6000

First, when the model starts training, we obtain two state variables $S_t^{(1)}$ and $S_t^{(2)}$ from six simulation environments. In addition, $S_t^{(1)}$ dimension size is $(20 * 18 * 30 * 3)$, which means the traffic status in the last 20 seconds, 30 segments in 18 entrance lanes, and 3 kinds of states in turn signal lamp (go straight, turn-left, turn-right). $S_t^{(2)}$ dimension size is $(8 * 1)$, which means that there are 8 signal schemes for bus priority switching. We use greedy strategies to choose actions. We randomly select 64 samples from the experience pool each time.

Then, during neural network training, we process the first state $S_t^{(1)}$ through three convolutional layers. And these three convolutional layers have 32, 64, 128 filters. After flattening the output result of convolutional layers, we connect it with the second state $S_t^{(2)}$. We input the two connected states into the full connection layer. The model outputs a set of Q values with a dimension size of $(8 * 1)$.

Finally, we choose actions through the main network and evaluate each action through the target network. We update the parameter through the loss function.

5.2. **Analysis of experimental results.** In this experiment, we set 150 simulation rounds. In addition, we choose the average delay of vehicles and the throughput of intersections as the evaluation indicators. The performance of MP-3DQN is compared with three models: 1)traditional DQN [35]; 2) fixed signal timing (FST) [36](we set the duration of each phase to 40 seconds); 3) the latest 3DQN. We use 3DQN as the baseline for comparative experiments.



(a) Average delay of all other vehicles (b) Average delay of buses

Figure 6. Average delay of buses

5.2.1. *Average delay of vehicles.* At the end of each simulation round, we calculate the average delay of all other vehicles and buses, as shown in Figure 6(a) and Figure 6(b). From these two figures, we can see that MP-3DQN is better than the other three models in reducing the average delay of all other vehicles and buses.

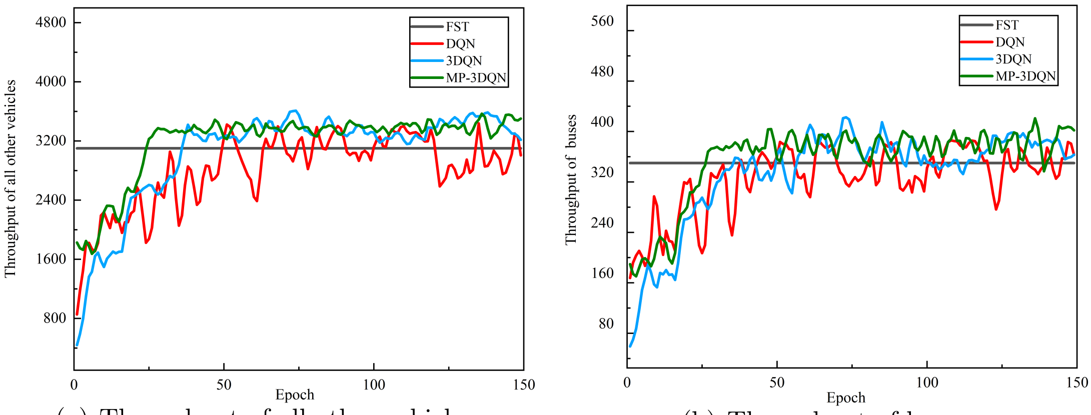
From Figure 6(a), the average delay of all other vehicles in FST is 59 seconds. In DQN, the fluctuation amplitude of the curve after 100 iterations starts to decrease. In DQN, the average delay of all other vehicles fluctuates between 75 and 95 seconds, and the stability of the model is poor. In 3DQN, the average delay of all other vehicles is reduced from 172 seconds to 60 seconds after 65 iterations. Then, 3DQN starts to converge and the convergence value is about 43 seconds. We propose that MP-3DQN can reduce the average delay of all other vehicles to about 35 seconds after 28 iterations. Compared with 3DQN model, MP-3DQN reduces the average delay of all other vehicles by 18.6%, and MP-3DQN converges faster than 3DQN.

From Figure 6(b), the average delay of buses in FST is 69 seconds. In DQN, the fluctuation amplitude of the curve after 110 iterations starts to decrease, and the average delay of buses is about 75 seconds. In 3DQN, the average delay of buses is reduced from 183 seconds to 60 seconds after 85 rounds of iteration. Then, 3DQN starts to converge and the convergence value is about 60 seconds. MP-3DQN can reduce the average delay of buses to about 46 seconds after 32 iterations. Compared with 3DQN, MP-3DQN reduces the average delay of buses by 23.3%.

5.2.2. *Throughput of intersections.* At the end of each simulation round, we calculate throughput of all other vehicles and buses, as shown in Figure 7(a) and Figure 7(b). From these two figures, we can see that MP-3DQN slightly improves over the other three models in throughput of all other vehicles and buses.

From Figure 7(a), the throughput of all other vehicles in FST is 3150. In DQN, the throughput of all other vehicles fluctuates greatly and the stability is poor. In 3DQN, the throughput of all other vehicles increases from 510 to 3300 after 65 iterations. MP-3DQN can improve the throughput of all other vehicles to about 3350 after 30 iterations. Compared with 3DQN, MP-3DQN slightly improves the throughput of all other vehicles.

From Figure 7(b), the throughput of buses in FST is 350. In DQN, the throughput of buses fluctuates greatly and the stability is poor. In 3DQN, the throughput of buses increases from 50 to 360 after 75 iterations. MP-3DQN improves the throughput of buses to about 370 after 36 iterations. Compared with 3DQN, MP-3DQN slightly improves the throughput of buses.



(a) Throughput of all other vehicles (b) Throughput of buses

Figure 7. Throughput of intersections

6. **Conclusion.** In this work, we propose an intelligent signal timing optimization model with bus priority based on multi-process experience pool, named MP-3DQN. The model can give priority rights to buses at intersections, and reduce the delay of buses without affecting the throughput of intersections. we apply MP-3DQN to an intersection in Fuzhou and compare it with the FST, DQN, and 3DQN. Experiments show that MP-3DQN is superior to other algorithms in reducing the average delay of buses, and has achieved good results.

1 We consider the turn signal state of the vehicle and the bus priority switching scheme, which not only can provide a basis for the agent in action selection but also solve the problem of bus response conflict.
2 We propose a multi-process parallel method to improve the efficiency of the experience pool. The method not only improves the convergence speed but also reduces the correlation between samples.
3 We propose a multi-objective reward function. The reward function constrains green light duration and optimizes the average delay of vehicles and intersection throughput. The method reduces the number of vehicle stops and the delay of vehicles.

Although the experimental results have achieved the expected results, there are still some limitations. Our current research work is the signal control of bus priority at single intersection. The main solution is to reduce the average delay of buses at intersections without affecting the throughput of intersections. In the future, first of all, we will further subdivide the types of other vehicles, such as fire engines, ambulances, police cars and trucks, to make the simulation effect closer to reality. Then, our current model is applied to the case of single intersection.

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

REFERENCES

[1] Y. Wang, B. Wu, and L. Li, "Traffic congestion management and control strategy for Land redevelopment," *Urban Redevelopment and Traffic Congestion Management Strategies*, pp. 115-145, 2022.

[2] F. Rasheed, K.A. Yau, R. Noor, C. Wu, and Y. Low, “Deep reinforcement learning for traffic signal control: a review,” *IEEE Access*, vol. 8, pp. 208016-208044, 2020.

[3] K.A. Yau, J. Qadir, H.L. khoo, M.H. Mee, and P. Komisarczuk, “A survey on reinforcement learning models and algorithms for traffic signal control,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1-38, 2017.

[4] B.R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A.A. Al Sallab, S. Yogamani, and P. Pérez, “Deep reinforcement learning for autonomous driving: a survey,” *arXiv e-prints*, pp. arXiv: 2002.00444, 2020.

[5] N.P. Farazi, T. Ahamed, L. Barua, and B. Zou, “Deep reinforcement learning and transportation research: a comprehensive review,” *arXiv preprint arXiv:2010.06187*, 2020.

[6] H. Haitao, M. Menendez and S.I. Guler, “Analytical evaluation of flexible-sharing strategies on multimodal arterials,” *Transportation Research Part A: Policy and Practice*, vol. 114, pp. 364-379, 2018.

[7] S.I. Guler, and M.J. Cassidy, “Strategies for sharing bottleneck capacity among buses and cars,” *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1334-1345, 2012.

[8] W. Ma, K.K. Head, and Y. Feng, “Integrated optimization of transit priority operation at isolated intersections: a person-capacity-based approach,” *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 49-62, 2014.

[9] M. Long, X. Zou, Y. Zhou, and E. Chung, “Deep reinforcement learning for transit signal priority in a connected environment,” *Transportation Research Part C: Emerging Technologies*, vol. 142, pp. 103814, 2022.

[10] M. Bagherian, M. Mesbah, and L. Ferreira, “Using delay functions to evaluate transit priority at signals,” *Public Transport*, vol. 7, no. 1, pp. 61-75, 2015.

[11] K. Yang, M. Menendez, and S.I. Guler, “Implementing transit signal priority in a connected vehicle environment with and without bus stops,” *Transportmetrica B: Transport Dynamic*, vol. 7, no. 1, pp. 423-425, 2019.

[12] Y. Lin, X. Yang, N. Zou, and M. Franz, “Transit signal priority control at signalized intersections: a comprehensive review,” *Transportation letters*, vol. 7, no. 3, pp. 168-180, 2015.

[13] D. Wang, W. Qiao, and C. Shao, “Relieving the impact of transit signal priority on passenger cars through a bilevel model,” *Journal of Advanced Transportation*, vol. 2017, 7696094, 2017.

[14] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, “Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction,” *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[15] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, and T.-Y. Wu, “A traffic prediction method of bicycle-sharing based on long and short term memory network,” *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17-29, 2019.

[16] L. Liao, Z. Hu, Y. Zheng, S. Bi, F. Zou, H. Qiu, and M. Zhang, “An improved dynamic chebyshev graph convolution network for traffic flow prediction with spatial-temporal attention,” *Applied Intelligence*, pp. 1-13, 2022.

[17] L. Liao, J. Liu, X. Wu, F. Zou, J. Pan, Q. Sun, S. Li, M. Zhang, “Time difference penalized traffic signal timing by lstm q-network to balance safety and capacity at intersections,” *IEEE Access*, vol. 8, pp. 80086–80096, 2020.

[18] Y. Ma, Y. Peng, and T.-Y. Wu, “Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning,” *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.

[19] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, “Selecting hyper-parameters of Gaussian process regression based on non-inertial particle swarm optimization in internet of things,” *IEEE Access*, vol. 7, pp. 59504-59513, 2019.

[20] S. Kumar, A. Damaraju, A. Kumar, S. Kumari, and C.-M. Chen, “Lstm network for transportation mode detection,” *Journal of Internet Technology*, vol. 22, no. 4, pp. 891-902, 2021.

[21] M.-E. Wu, J.-H. Syu, and C.-M. Chen, “Kelly-based options trading strategies on settlement date via supervised learning algorithms,” *Computational Economics*, vol. 59, no. 4, pp. 1627-1644, 2022.

[22] X. Liang, X. Du, G. Wang, and Z. Han, “A deep reinforcement learning network for traffic light cycle control,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243-1253, 2019.

[23] W. Genders, and S. Razavi, “Using a deep reinforcement learning agent for traffic signal control,” *arXiv e-prints*, pp. arXiv: 1611.01142, 2016.

[24] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive traffic signal control: deep reinforcement learning algorithm with experience replay and target network," *arXiv e-prints*, pp. arXiv: 1705.02755, 2017.

[25] J. Gu, Y. Fang, Z. Sheng, and P. Wen, "Double deep q-network with a dual-agent for traffic signal control," *Applied Sciences*, vol. 10, no. 5, pp. 1622, 2020.

[26] A.Haydari, and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: a survey," *arXiv e-prints*, pp. arXiv: 2005.00935, 2020.

[27] A.R.M. Jamil, and N. Nower, "Dynamic weight-based multi-objective reward architecture for adaptive traffic signal control system," *International Journal of Intelligent Transportation Systems Research*, pp. 1-13, 2022.

[28] A.R.M. Jamil, K.K. Ganguly, and N. Nower, "Adaptive traffic signal control system using composite reward architecture based deep reinforcement learning," *IET Intelligent Transport Systems*, vol. 14, no. 14, pp. 2030-2041, 2020.

[29] T. Brys, A. Harutyunyan, P. Vrancx, M.E. Taylor, D. Kudenko, and A. Nowé, "Multi-objectivization of reinforcement learning problems by reward shaping," *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 2315-2322, 2014.

[30] H. Xu, J. Sun, and M. Zheng, "Comparative analysis of unconditional and conditional priority for use at isolated signalized intersections," *Journal of Transportation Engineeringn*, vol. 136, no. 12, pp. 1092-1103, 2010.

[31] M. Boltze, and B. Friedrich, "Innovation in der lichtsignalsteuerung–die neufassung der richtlinien fur lichtsignalanlagen (rilsa)," *Straßenverkehrstechnik*, vol. 54, no. 4, pp. 192-197, 2007.

[32] H.V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2094-2100, 2015.

[33] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," *Advances in Neural Information Processing Systems*, vol. 31, pp. 8224-8234, 2019.

[34] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," *International Conference on Machine Learning*, pp. 1995-2003, 2016.

[35] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247-254, 2016.

[36] A.S. Abdelfatah, and H.S. Mahmassani, "System optimal time-dependent path assignment and signal timing in traffic network," *Transportation Research Record*, vol. 1645, no. 4, pp. 185-193, 1998.