processes

MDPI

*Article*

# Deep Reinforcement Learning for Traffic Light Timing Optimization

**Bin Wang, Zhengkun He, Jinfang Sheng * and Yu Chen**

School of Computer Science and Engineering, Central South University, Changsha 410083, China
* Correspondence: jfsheng@csu.edu.cn

**Abstract:** Existing inflexible and ineffective traffic light control at a key intersection can often lead to traffic congestion due to the complexity of traffic dynamics, how to find the optimal traffic light timing strategy is a significant challenge. This paper proposes a traffic light timing optimization method based on double dueling deep Q-network, MaxPressure, and Self-organizing traffic lights (SOTL), namely EP-D3QN, which controls traffic flows by dynamically adjusting the duration of traffic lights in a cycle, whether the phase is switched based on the rules we set in advance and the pressure of the lane. In EP-D3QN, each intersection corresponds to an agent, and the road entering the intersection is divided into grids, each grid stores the speed and position of a car, thus forming the vehicle information matrix, and as the state of the agent. The action of the agent is a set of traffic light phase in a signal cycle, which has four values. The effective duration of the traffic lights is 0–60 s, and the traffic light phases switching depends on its press and the rules we set. The reward of the agent is the difference between the sum of the accumulated waiting time of all vehicles in two consecutive signal cycles. The SUMO is used to simulate two traffic scenarios. We selected two types of evaluation indicators and compared four methods to verify the effectiveness of EP-D3QN. The experimental results show that EP-D3QN has superior performance in light and heavy traffic flow scenarios, which can reduce the waiting time and travel time of vehicles, and improve the traffic efficiency of an intersection.

**Keywords:** traffic light control; deep reinforcement learning



**Citation:** Wang, B.; He, Z.; Sheng, J.; Chen, Y. Deep Reinforcement Learning for Traffic Light Timing Optimization. *Processes* **2022**, *10*, 2458. https://doi.org/10.3390/pr10112458

Academic Editors: Jie Zhang and Meihong Wang

Received: 27 October 2022
Accepted: 17 November 2022
Published: 20 November 2022



## 1. Introduction

Traffic congestion has increasingly become one of the major problems in cities. Traffic light control can effectively alleviate traffic congestion and improve traffic efficiency in urban intersections. The existing traffic light control methods are divided into timing control and adaptive traffic signal control (ATSC) [1]. The timing control is FixedTime [2], and the most representative ATSC is SCOOT [3] and SCATS [4]. Self-organizing traffic lights (SOTL) [5] and max pressure (MP) control [6] aim to maximize the global throughput from observation of traffic states.

These conventional methods are not effective as the complexity of the traffic network increases. Recently, reinforcement learning (RL) has been widely used for traffic light control. Reinforcement learning defines traffic light control as a Markov decision process (MDP) and learns an optimal control strategy through continuous iteration with the environment. Reinforcement learning based on table Q-learning can only deal with discrete intersection states [7]. Deep reinforcement learning (DRL) can deal with discrete or continuous intersection states. Some DRL-based methods have shown better performance than many traditional methods in specific scenarios, which can be used to control traffic lights and improve the traffic efficiency of the intersection [8,9].

Most DRL-based methods focus on learning a strategy to switch the current traffic lights phase in a signal cycle [10]. The duration of traffic lights is a fixed-length interval, which is not flexible enough to cope with changing traffic conditions. Liang et al., tried

to control the traffic light duration in a cycle based on deep reinforcement learning and the extracted information from vehicular networks [11]. It shows good performance in specific scenarios, but it will become unstable with the increase of intersection complexity. Hua et al., also attempted to introduce the concept of max pressure from the traffic field as the reward for control model optimization [12–14].

This paper proposes the EP-D3QN algorithm based on 3DQN [11], MP [6] algorithm, and SOTL [5] algorithm. In EP-D3QN, the road entering the intersection is divided into grids, and each grid stores the information of each vehicle. The matrix formed by these grids serves as the state of the agent, whose action is the combination of different traffic light phases in a cycle, and the reward is the difference between the sum of accumulated waiting times of all vehicles in two consecutive cycles. The EP-D3QN can dynamically adjust the duration of traffic lights in a signal cycle and activate the effective phase during phase switching. The experiment uses the Simulation of Urban Mobility (SUMO) [15] to simulate two traffic scenarios to verify the effectiveness of EP-D3QN.

The remainder of this paper is organized as follows. The related work is introduced in Section 2. Section 3 shows the problem statement and the details of the EP-D3QN. Section 4 shows the result of the experiment and analysis. Finally, the paper is concluded in Section 5.

## 2. Related Work

Traffic light control approaches can be divided into two types: traditional methods and RL-based methods.

### 2.1. Traditional Methods

Early research works on traffic light control mainly are Fixed-time Traffic Light Control (FT) [2], which fix the duration of traffic lights according to historical traffic information. Subsequently, SCOOT [3], SCATS [4], and other adaptive traffic signal control methods emerged. These methods are still widely deployed in many cities. SOTL [5] and MP [6] followed. Both MP and SOTL switch the traffic lights phase based on current traffic conditions. In the SOTL method, whether the traffic light phases is switched depends on the current observed traffic conditions and the rules defined in advance. Compared with fixed time, it is more flexible. The MP method introduces the concept of max pressure. The pressure is defined as the difference between the number of vehicles on incoming lanes and the number of vehicles on outgoing lanes. When the traffic light phases are switched, the phase with the max pressure is preferentially activated. Hua et al., introduced the concept of maximum pressure as a reward function, so as to learn more optimal strategies [12–14]. Wu et al., also optimized the MP algorithm and proposed the efficient MP [16]. Liang et al., also integrated the advantages of SOTL and MaxPressure and applied them in traffic light control [17]. Despite the high performance of the MP-based control, it lacks flexibility as the duration of traffic lights is a fixed-length interval.

### 2.2. RL-Based Methods

RL-based traffic light control has attracted wide attention from both academia and industry in the last two decades. Traditional RL-based methods mainly use table Q-learning, which can only handle discrete intersection state representation [7]. Later, deep reinforcement learning (DRL) appeared, which can deal with discrete or continuous intersection states. Compared with traditional traffic light methods, some DRL-based methods show better performance in certain situations [18–20]. However, DRL-based methods will overestimate Q-value, and due to the complexity of traffic conditions, DRL-based methods will become unstable. Some scholars also proposed multiple optimization elements to improve the DRL's performance, such as double Q-learning network [21], dueling Q-learning network [22], and prioritized experience replay [23]. Liang et al., incorporates these optimization technologies and tried to dynamically control the duration of traffic lights in a cycle [11], which showed good performance in light traffic flow scenarios, but became unstable in heavy traffic flow scenarios.

Inspired by the above work, this paper proposes the EP-D3QN algorithm based on MP, SOTL, and 3DQN algorithms. The advantage of double dueling deep Q-network can dynamically calculate the duration of the traffic light in each signal cycle. It is more flexible, and the advantages of MP and SOTL just overcome the instability of deep Q-network, which make EP-D3QN more adaptive.

## 3. Method

### *3.1. Problem Statement*

Traffic congestion often occurs at key intersections [24], so this paper focuses on an intersection. The agent corresponding to the intersection can receive its observation and obtain the duration of the traffic lights, and control the traffic light phases switching in a cycle. Each traffic light has three colors: red, yellow, and green. Each traffic light phases is a set of permissible traffic movements, and each intersection has four phases, and the combination of all phases is called a signal cycle. The vehicles always travel across an intersection from one incoming road to one outgoing road. The signal phases of a cycle are played in a fixed sequence to control the traffic flow.

Our problem is to learn a strategy based on DRL to dynamically adjust the duration of traffic lights in a cycle, and to switch the phase based on the rules we set, so as to control the traffic flow. The state of the agent is the information matrix formed by all vehicles entering the intersection, and its action is the set of all phases in a cycle. Its reward is the difference between the accumulated waiting time and the sum of all vehicles in two consecutive cycles. The goal of the agent is to maximize the reward. In each time step, the agent will obtain the observation of the intersection, and then select an action based on its own strategy. The action is a set of phases in a cycle, and the effective phase will be activated first in each time step. After the execution of the action, the agent will get the reward, and then the state of the intersection will change. The agent finally learns to get a high reward by reacting to different traffic scenarios.

### *3.2. Agent Design*

In EP-D3QN, each intersection corresponds to an agent. In order to better introduce EP-D3QN into traffic light control. First, we need to design the state, action, and reward of the agent.

#### 3.2.1. Intersection State

For each intersection, we divide the road entering the intersection into grids, and each grid can only accommodate one vehicle, as shown in Figure 1.

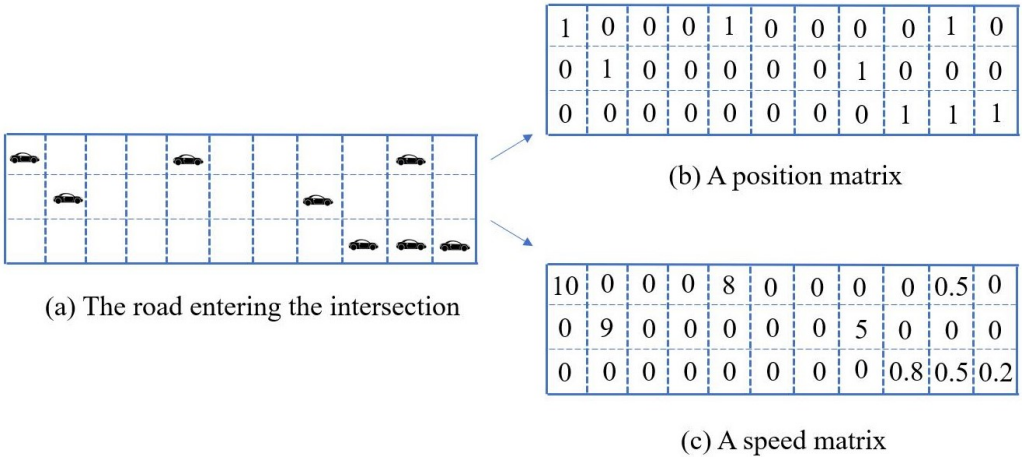


**Figure 1.** The process to build the state matrix. (**a**) is the snapshot of the road entering the intersection. (**b**) is the position matrix of vehicles at the current moment. (**c**) is the speed matrix of vehicles at the intersection at the current moment.

Among them, (b) is the position information of the vehicle. A grid with 1 means a vehicle, and 0 means no vehicle. In (b), the more grids with 1, the more vehicles stay at the intersection at the current moment. (c) represent the speed information of the vehicle. Floating data is used to represent the speed of the vehicle. Each grid is the actual speed of the vehicle in meters/second. Therefore, all the lanes entering the intersection can be represented as a matrix. This matrix is the state of the intersection.

#### 3.2.2. Agent Actions

The action of the agent is defined as $a_i \in [1, 2, \ldots, 9]$, where $a_i =< NS, NSL, WE, WEL >$, as shown in Figure 2b, NS, NSL, WE, and WEL represent the four traffic lights phases in a cycle, which indicates going straight from north to south, turning left from north to south, going straight from east to west, and turning left from east to west, respectively. We set the longest duration of the traffic light to 60 s and the shortest duration to 0 s.

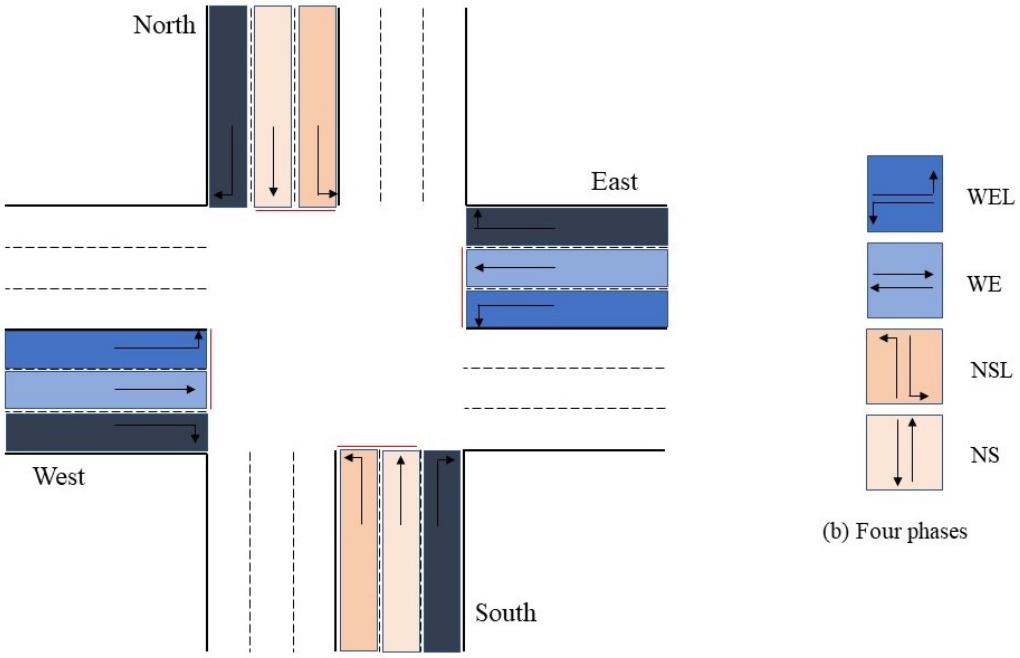


**Figure 2.** The framework of EP-D3QN.

At each time step, the agent will choose an action from the nine actions to act on the traffic light controller at the intersection. For example, the current action is $a1 =< NS, NSL, WE, WEL >$, and the next legal action is $< NS5, NSL, WE, WEL >$, $< NS, NSL5, WE, WEL >$, $< NS, NSL, WE5, WEL >$ and $< NS, NSL, WE, WEL5 >$.

#### 3.2.3. Reward

The reward of the agent is crucial for the deep reinforcement learning model. An appropriate reward can guide the agent to get better training results. Therefore, based on previous research work [7,25,26], the reward of the agent is defined as follows:

$$r_t = \sum_{i=1}^{n} w_{t-1}^{i} - \sum_{j=1}^{m} w_t^j \tag{1}$$

Among them, $w_{t-1}^{i}$ represents the waiting time of $i$-th vehicle in the $(t-1)$-th cycle, $w_t^j$ represents the waiting time of $j$-th vehicle in the $t$-th cycle, and $n$ and $m$ represent the number of vehicles entering the intersection in the $(t-1)$-th and $t$-th cycle, respectively.

### *3.3. Effective-Pressure with Double Dueing Deep Q-Network for Traffic Light Control*

In this paper, we propose the EP-D3QN algorithm based on MP, SOTL, and 3DQN algorithms. In EP-D3QN, the agent corresponding to the intersection can receive its observation and choose an action to execute. After receiving its observation, the agent encodes its observation by the convolution layer and the fully connected layer, and then obtains Q-value of each action. The greedy strategy is used to select the action with the largest Q-value, and the effective traffic lights phase is preferentially activated in a cycle during each time step. The complete process is shown in Figure 3.

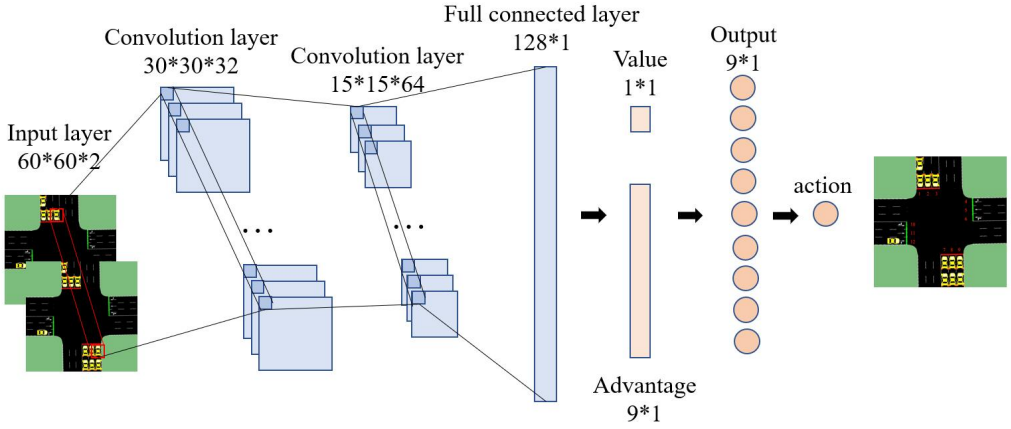


**Figure 3.** The synthetic intersection during all training episodes.

In the EP-D3QN, the state of the intersection is a vehicle information matrix with a size of 60∗60, and each grid in the matrix stores the location and speed of the vehicle. There are two convolutional layers in total. The first convolutional layer contains 32 filters, the size of each filter is 4∗4, and its moving stride is 2∗2. The second convolutional layer contains 64 filters with size 2∗2, and its moving stride is also 2∗2. The fully connected layer is responsible for integrating the information extracted by the convolutional layer. After the fully-connected layer, the data are split into two parts. The first part is then used to calculate the value and the second part is for the advantage. The advantage of action means how well it can achieve by taking an action over all the other actions. The formula can be expressed in Equation (2).

$$Q(o,a;\theta) = V(o;\theta) + (A(o,a;\theta)\frac{1}{|A|}\sum_{a'} A(o,a';\theta)) \tag{2}$$

Among them, the value of a state $V(o;\theta)$ denotes the overall expected rewards by taking probabilistic actions in future steps. The $A(o,a;\theta)$ is the advantage that corresponds to every action. Each action corresponds to a Q-value. The target Q-value is calculated as follows:

$$Q_{target}(o,a) = r + \gamma Q_{target}(o', argmax_{a'}(Q(o',a';\theta)),\theta') \tag{3}$$

where $r$ is the reward of the agent, $\gamma$ is the discount factor, $\theta$ and $\theta'$ are the parameters of the main network and target network, $o'$ and $a'$ are the state of intersection and the action of the agent at the next time step respectively.

After the agent obtains Q-value through its main network, the main network is updated by TD-error as follows:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[Q_{target}(o_i,a_i;\theta') - Q(o_i,a_i;\theta)]^2 \tag{4}$$

Among them, $m$ represents the sample size extracted by the replay buffer. The parameters of the target network $\theta'$ are updated as Equation (5).

$$\theta' = \varepsilon\theta' + (1 - \alpha)\theta \tag{5}$$

Among them, $\varepsilon$ is the update rate from the main network to the target network and $\alpha$ is the learning rate of the main network.

The detailed steps of the EP-D3QN algorithm are shown in Algorithm 1. First, to initialize the parameters of the main network and target network. Meanwhile, to initialize discount factor $\gamma$ , target network update rate $\varepsilon$, replay buffer $D$ and threshold $v_{min}$ and $v_{max}$; Then, in each time step, the agent receives state $o$, select action $a$. During the action $a$ be executed, preferentially activates the effective phase in a cycle, then get reward $r$, and new state $o'$, and stored the experience $(o, a, r, o')$ in the replay buffer (lines 4 to 13); Next, for the agent, sample a minibatch of step episodes experience trajectories $(o, a, r, o')$ from the replay buffer (line 14). Finally, update the parameters of the main network by TD-error (line 19), and then update the parameters of the target network (line 20). This process is repeated until it converges.

**Algorithm 1** EP-D3QN for traffic light control.

**Input:**
Intersections' state $o$
**Output:**
Action $a$
**Initialize:**
The parameters of main network $\theta$ and target network $\theta'$, discount factor $\gamma$,target network update rate $\varepsilon$, replay buffer $D$, threshold $v_{min}$ and $v_{max}$

1: **for** $episode = 1$ to $M$ **do**
2: Initialize observation $o$ and $t = 1$
3: **for** $t < T$ **do**
4: The agent select an action $a$
5: Calculate pressure $p_{NS}$ and $p_{WE}$ for the phases NS and WE
6: Calculate vehicles approaching red phase $v_r$
7: Calculate vehicles approaching green phase $v_g$
8: **if** $v_g < v_{min}$ and $v_r > v_{max}$ **then**
9: **if** phase = WE and $p_{NS} > p_{WE}$ or phase = NS and $p_{WE} > p_{NS}$ **then**
10: switch light
11: **end if**
12: **end if**
13: Then get reward $r$ and new observation $o'$
14: Store $(o, a, o', r)$ in $D$
15: $o \leftarrow o'$
16: **if** $T_{update} > minSteps$ **then**
17: Sample random minibatch of step $(o, a, o', r)$ from $D$
18: Calculate $Q_{target}(o, a)$ for the agent with Equation (3)
19: Update the main network $\theta$ with Equation (4)
20: Update target network $\theta'$ with Equation (5)
21: **end if**
22: **end for**
23: **end for**

## 4. Experiment and Analysis

### *4.1. Experimental Setup*

In the experiments, we use Simulation of Urban Mobility (SUMO), an open-source, micro, multi-model traffic simulation software [15], to simulate light and heavy traffic flow scenarios respectively. The intersection created by sumo is with a two-way six-lane, as shown in Figure 2a. Each direction has 6 lanes. Each lane is 180 m, and the length of the vehicles traveling is 5 m.

Vehicles entering the intersection can go straight, and turn left or right. The distance between the adjacent vehicles is set to 1 m. The maximum speed of the car is 13.89 m/s. The intersection has four traffic light phases, as shown in Figure 2b. In the light traffic flow scenario, the traffic flow is generated using the Bernoulli distribution with a probability of 0.2 (two vehicles arriving at the intersection every 10 s). In the heavy traffic flow scenario, the traffic flow is also generated using the Bernoulli distribution with a probability of 0.4.

### *4.2. Evaluation Metrics*

Following previous research work [25–27], we select two types of evaluation indicators, the first is the average reward, the second is average waiting time (AWT), average queue length (AQL), and average travel time (ATT). The average reward is used to reflect the performance of EP-D3QN, and the bigger its value, the better its performance. AVWT, AQL and ATT are used to reflect the traffic conditions of the intersection. The smaller the average, the higher the traffic efficiency of the intersection, and vice versa.

### *4.3. Compared Methods*

In order to verify the performance of EP-D3QN, we compare it with the following methods.

- **Fixed-time Traffic Light Control (FT)**. FT [2] is the most widely used traffic light control method. Each intersection sets a fixed sequence of signal phases, and the duration of traffic lights is also fixed.
- **Self-organizing traffic lights (SOTL)**. In the SOTL [5], whether the traffic light phases is switched depends on the observed traffic conditions and the rules defined in advance. Compared to FT, SOTL is very flexible.
- **MaxPressure (MP)**. In the MP [6], the traffic light controller activates the traffic light phases with max pressure in a cycle. The MP introduces the concept of pressure, which is the difference between the number of vehicles on incoming lanes and the number of vehicles on outgoing lanes. At each time step, the pressure of each phase is calculated and compared. Finally, the phase with the max pressure is activated.
- **Double dueling Deep Q-network (3DQN)**. The 3DQN [11], incorporates multiple optimization elements to improve the performance of traffic light control, such as dueling network, target network, double Q-learning network, and prioritized experience replay.

### *4.4. Result and Analysis*

#### 4.4.1. Light Traffic Flow Scenario

The simulation results are shown in Figure 4. We plotted the average waiting time during all the episodes. The red line shows the EP-D3QN, the blue line is the 3DQN, and the green line is the FixTime method, SOTL and MaxPressure correspond to the brown and orange lines, respectively. As can be seen from the figure, there is little difference in the average waiting time between SOTL and MP, but both are better than the FT method. That's because in the MP algorithm, the traffic light phases with the max pressure will be activated at each time step, while in the SOTL algorithm, whether to activate the phase according to the set threshold. Compared with 3DQN, it can be seen that EP-D3QN has faster convergence speed and stronger stability. That's because EP-D3QN activates the effective traffic phase preferentially when the action is performed at each time step.

Table 1 shows the results of the comprehensive evaluation. As can be seen from the table, compared with other methods, EP-D3QN shows better performance, its AWT, AQL, and ATT are relatively small, and the average reward is the largest. The AQL and ATT of FT are the largest, followed by MP and SOTL, and EP-D3QN is the smallest. The AWT of 3DQN and SOTL is at a medium level, the AWT of FT and MP is the highest, while the AWT of EP-D3QN is the lowest. It shows that EP-D3QN can dynamically control traffic lights more effectively, so as to improve the traffic efficiency of the intersection.

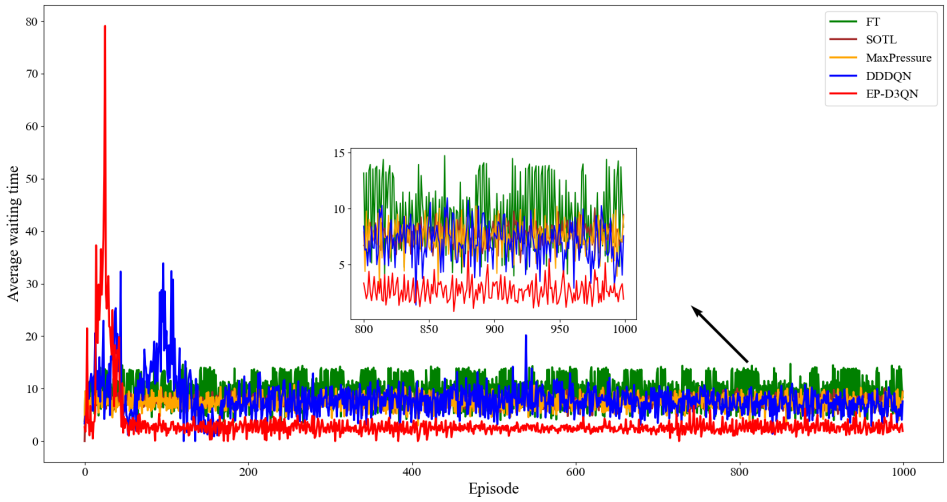


**Figure 4.** The average waiting time in the light traffic flow scenario during all training episodes.

**Table 1.** Comprehensive evaluation in the light traffic flow scenario.

| Algorithm | Reward | AQL | AWT | ATT |
|---|---|---|---|---|
| FT | 0.005 | 0.555 | 7.729 | 27.538 |
| MaxPressure | 0.172 | 0.504 | 7.240 | 15.631 |
| SOTL | 4.410 | 0.386 | 5.341 | 9.331 |
| 3DQN | 4.229 | 0.351 | 6.535 | 6.775 |
| EP-D3QN | **6.332** | **0.322** | **3.911** | **4.982** |

#### 4.4.2. Heavy Traffic Flow Scenario

Figure 5 shows the average waiting time during all training episodes in the heavy traffic flow scenario. From the figure, we can see that at the beginning, 3DQN and EP-D3QN were unstable. That's because, during the initial training episodes, the agent randomly selects actions. When the training episodes reach about 200, EP-D3QN starts to converge, and in the later training episodes, the AWT of both EP-D3QN and 3DQN could be maintained within a fixed range, but EP-D3QN showed better performance. The AWT of FT, MP, and SOTL keeps fluctuating in a fixed range, and they are all worse than EP-D3QN, but SOTL performs better than D3QN, which indicates that the traditional method is not flexible for heavy traffic flow scenarios due to the complexity of the traffic conditions.

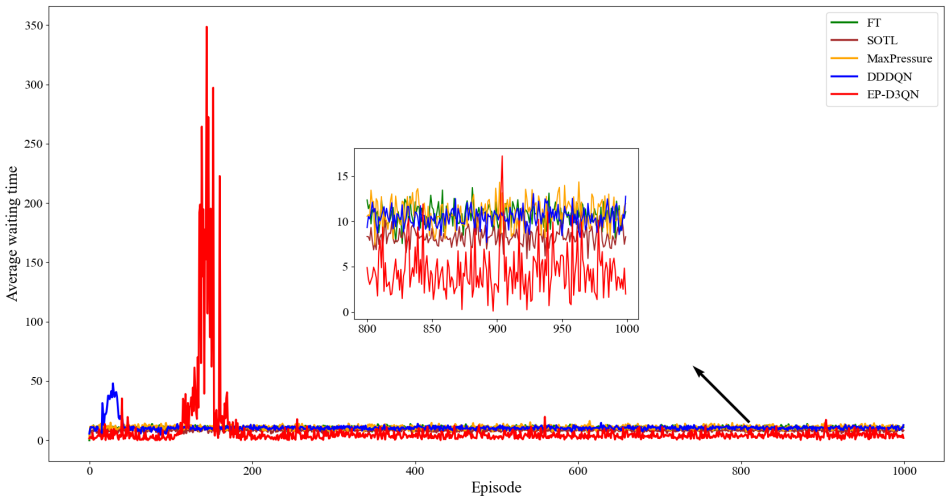


**Figure 5.** The average waiting time in the heavy traffic flow scenario during all training episodes.

Table 2 is the result of the comprehensive evaluation in the heavy traffic flow scenario. As can be seen from the table, AQL and ATT of FT are the largest, AWT of MP is the largest, while the three indexes of EP-D3QN are the smallest. Moreover, the average reward of EP-D3QN is the largest. That's because EP-D3QN incorporates the advantages of SOTL and MP. Compared with SOTL and MP, AWT is significantly reduced, and ATT is also significantly reduced compared with 3DQN.

**Table 2.** Comprehensive evaluation in the heavy traffic flow scenario.

| Algorithm | Reward | AQL | AWT | ATT |
|---|---|---|---|---|
| FT | 0.009 | 3.130 | 6.648 | 16.030 |
| MaxPressure | 0.007 | 2.740 | 9.626 | 6.377 |
| SOTL | 0.058 | 2.243 | 6.452 | 10.420 |
| 3DQN | 9.588 | 2.703 | 6.232 | 7.532 |
| EP-D3QN | **11.658** | **1.385** | **4.110** | **3.519** |

In conclusion, the EP-D3QN can perform better performance in both light and heavy traffic flow scenarios, ensuring that vehicles entering the intersection spend less waiting time and pass the intersection quickly, thus effectively improving the traffic efficiency of the intersection.

## 5. Conclusions

In this paper, we study the problem of how to control the traffic light duration in a cycle based on deep reinforcement learning in an intersection and propose an EP-D3QN algorithm based on 3DQN, MP, and SOTL algorithms. In EP-D3QN, the intersection corresponds an agent. The agent can receive its own observation and choose an action. Its state is the information matrix of the vehicles entering the intersection. The action of the agent is the traffic light phases in a cycle. During the action being executed, the traffic lights phase with the effective pressure will be activated preferentially. The reward of the agent is the sum of the accumulated waiting time of all the vehicles in two consecutive cycles. We use SUMO to simulate the traffic scenarios and verify the effectiveness of EP-D3QN. The experimental results show that EP-D3QN significantly outperforms other methods in both light and heavy traffic flow scenarios, which can improve the traffic efficiency of the intersection and relieve traffic pressure.

**Author Contributions:** Conceptualization, B.W. and J.S.; methodology, Z.H.; validation, Z.H. and Y.C.; formal analysis, Z.H. and J.S. writing—original draft preparation, Z.H.; writing—review and editing, J.S., Z.H. and Y.C.; visualization, Z.H.; supervision, B.W.; funding acquisition, J.S. and B.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Key Research and Development Program of China under grant No. 2018YFB1003602.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Noaeen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Abad, Z.S.H.; Bazzan, A.L.; Far, B. Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Syst. Appl.* **2022**, *199*, 116830. [CrossRef]
2. Li, L.; Wen, D. Parallel systems for traffic control: A rethinking. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1179–1182. [CrossRef]
3. Robertson, D.I.; Bretherton, R.D. Optimizing networks of traffic signals in real-time SCOOT method. *IEEE Trans. Veh. Technol.* **1991**, *40*, 11–15. [CrossRef]
4. Sims, A. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Trans. Veh. Technol.* **1980**, *29*, 130–137. [CrossRef]

5. Cools, S.B.; Gershenson, C.; D'Hooghe, B. Self-organizing traffic lights: A realistic simulation. In *Advances in Applied Self-Organizing Systems*; Springer: London, UK, 2013; pp. 45–55.
6. Varaiya, P. Max pressure control of a network of signalized intersections. *Transp. Res. Part C Emerg. Technol.* **2013**, *36*, 177–195. [CrossRef]
7. Haydari, A.; Yilmaz, Y. Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 11–32. [CrossRef]
8. Li, L.; Lv, Y.; Wang, F.Y. Traffic Signal Timing via Deep Reinforcement Learning. *IEEE-CAA J. Autom. Sin.* **2016**, *3*, 247–254.
9. Mousavi, S.; Schukat, M.; Howley, E. Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning. *IET Intell. Transp. Syst.* **2017**, *11*, 417–423. [CrossRef]
10. Genders, W.; Razavi, S. Policy Analysis of Adaptive Traffic Signal Control Using Reinforcement Learning. *J. Comput. Civ. Eng.* **2020**, *34*, 19–46. [CrossRef]
11. 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]
12. Wei, H.; Zheng, G.; Yao, H.; Li, Z. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 2496–2505.
13. Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Li, Z. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1290–1298.
14. Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; Li, Z. Toward a Thousand Lights: Decen-tralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19), Honolulu, HI, USA, 27 January–1 February 2019; pp. 3414–3421.
15. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent development and applications of sumo simulation of urban mobility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
16. Wu, Q.; Zhang, L.; Shen, J.; Lu, L.; Du, B.; Wu, J. Efficient pressure: Improving efficiency for signalized intersections. *arXiv* **2021**, arXiv:2112.02336.
17. Zhang, L.; Wu, Q.; Jun, S.; Lu, L.; Du, B.; Wu, J. Expression might be enough: Representing pressure and demand for re-inforcement learning based traffic signal control. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 26645–26654.
18. Shabestary, S.M.A.; Abdulhai, B. Deep learning vs. discrete reinforcement learning for adaptive traffic signal control. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 286–293.
19. Zeng, J.; Hu, J.; Zhang, Y. 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.
20. Chen, P.; Zhu, Z.; Lu, G. An Adaptive Control Method for Arterial Signal Coordination Based on Deep Rein-forcement Learning. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3553–3558.
21. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), Austin, TX, USA, 25–30 January 2015; pp. 2094–2100.
22. Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N. Dueling network architectures for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (ICML'16), New York, NY, USA, 19–24 June 2016; pp. 1995–2003.
23. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. In Proceedings of the 4th International Conference on Learning Representations (ICLR'16), San Juan, PR, USA, 2–4 May 2016.
24. Xu M.; Wu J.; Huang L.; Zhou, R.; Wang, T.; Hu, D. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. *J. Intell. Transport. Syst.* **2020**, *24*, 1–10. [CrossRef]
25. Shashi, F.I.; Md Sultan, S.; Khatun, A.; Sultana, T.; Alam, T. A Study on Deep Reinforcement Learning Based Traffic Signal Control for Mitigating Traffic Congestion. In Proceedings of the 2021 IEEE 3rd Eurasia Conference on Biomedical Engi-neering, Healthcare and Sustainability (ECBIOS), Tainan, Taiwan, 28–30 May 2021; pp. 288–291.
26. 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.* **2022**, *22*, 12–18. [CrossRef]
27. Liu, J.; Qin, S.; Luo, Y.; Wang, Y.; Yang, S. Intelligent Traffic Light Control by Exploring Strategies in an Optimised Space of Deep Q-Learning. *IEEE Trans. Veh. Technol.* **2022**, *71*, 5960–5970. [CrossRef]