MALight: A Deep Reinforcement Learning Traffic Light Control Algorithm with Pressure and Attentive Experience Replay

Yan Kong [1], *Ying Li* [1], *Chih-Hsien Hsia* [2,3*]

[1] *School of Computer and Software, Nanjing University of Information Science and Technology, China*
[2] *Department of Computer Science and Information Engineering, National Ilan University, Taiwan*
[3] *Department of Business Administration, Chaoyang University of Technology, Taiwan*
kongyan4282@163.com, ilyyli8@163.com, hsiach@niu.edu.tw

Abstract

This study proposes a new algorithm MALight based on multi-step deep Q network (DQN) and attentive experience replay (AER). Multi-step DQN samples multiple consecutive experiences within a time step, combines them into a long-term sample, and uses them to update the Q network to reduce the bias caused by inaccurate Q value estimation, which could accelerate the convergence of Q network. During training, we adopted the concept of AER to prioritize learning experiences close to the current state to enable the agent to learn better strategies. Finally, we conducted simulation experiments in the city traffic simulator CityFlow using both synthetic and real-world traffic flow datasets. The evaluation results show that MALight can accelerate the convergence speed of the network, effectively improve the traffic capacity of intersections, and optimize the average travel time of intersections.

Keywords: Deep reinforcement learning, Traffic signal control lights, Multi-step DQN, Experience replay

1 Introduction

With the rapid increase in the number of motor vehicles in this world, traffic congestion has become a challenging and terrible problem worldwide. Nowadays, traditional traffic signal lights can only switch according to preset time cycles and cannot dynamically adjust according to real-time traffic conditions, this results in the waste of road resources and thus alleviates traffic congestion to some extent. In contrast, self-adaptive traffic signal control lights can adjust the signal lights in real-time based on road traffic conditions to adapt to actual traffic needs, thereby achieving the goal of optimizing traffic flow and reducing traffic congestion. There are many deep reinforcement learning-based methods for self-adaptive traffic signal light optimization. However, the neural networks used in this research always result in the training process does not converge, are unstable, or even trapped in local optima.

To alleviate traffic congestion, methods such as limiting the growth of demand for vehicles and increasing the construction of road infrastructure are usually adopted. However, both of these methods have their limitations. As we all know, the traffic signal control system plays an important role in traffic conditions, and a desirable traffic signal control system can increase traffic volume and alleviate traffic congestion. Unfortunately, most of the currently used traffic signal lights are still traditional fixed-time traffic signal lights. Traditional traffic signal control (TSC) can be divided into three main categories: fixed-time control [1], actuated control [2], and adaptive control [3-7]. Fixed-time control is the most common traffic signal control way, traffic lights switch according to a predetermined schedule. Each traffic signal state (red, yellow, green) persists for a specific period and then switches in a predetermined sequence. This type of control is suitable for situations where traffic flow changes relatively little and is predictable, such as during nighttime or periods of low traffic flow.Actuated control triggers the switching of signal light states by detecting traffic flow or driver demands. This can include detecting the presence of vehicles, pedestrians, or bicycles and making corresponding adjustments to signal lights.Actuated control is suitable for situations that require flexible adjustments based on actual traffic demands, such as busy intersections or locations where pedestrians cross roads. Adaptive control is one of the most effective traffic signal control methods, which uses advanced algorithms and ideas to train network models to converge and achieve intelligent control of traffic signals. Adaptive control is suitable for high-traffic and frequently changing situations. It allows for real-time adjustments of signal lights to adapt to traffic conditions, providing higher traffic efficiency and reducing traffic congestion.MaxPressure [5] is one of the state-of-the-art in adaptive control methods, which chooses the appropriate phase based on traffic conditions to minimize traffic pressure. When traffic volume is too high, traditional fixed-time traffic signal lights cannot effectively handle traffic flow, thereby aggravating traffic congestion problems. Compared to traditional traffic signal lights, adaptive traffic signal light control has three advantages. Firstly, it can dynamically adjust the signal lights to adapt to the actual traffic demand, thus enabling more effective control of traffic flow. Secondly, it can optimize traffic lights according to real-time traffic conditions, thereby reducing traffic congestion and improving the efficiency of roads and pedestrian travel. Finally, it can achieve precise traffic flow prediction and

*Corresponding Author: Chih-Hsien Hsia; E-mail: hsiach@niu.edu.tw

DOI: https://doi.org/10.70003/160792642024122507001

control, thereby improving traffic safety and reliability.

In recent years, with the development of artificial intelligence (AI), many AI-related technologies have attracted the attention of researchers and have been applied to solve the problem of smart traffic signal control [8-10], e.g., Reinforcement learning (RL) which allows an agent to interact with the environment to learn how to achieve long-term goals. Some recent RL-based research has focused on the DQN (deep Q network) algorithm. However, the DQN algorithm suffers from the problem of biased Q-value estimation. To address this issue, we propose MALight, a multi-step DQN-based algorithm that can speed up the training process and improve the convergence rate. Moreover, since the random experience replay used in DQN fails to capture the importance of different experiences, we propose to replace it with a higher-priority sampling method that targets experiences similar to the current state.

The network framework of DQN is improved by accumulating *n* single-step experiences into one and using the accumulated experience for learning, thus accelerating the convergence speed of the network and enhances the accuracy of decision-making. Attentive experience replay (AER) [11] is combined with the multi-step DQN network to prioritize learning experiences that are similar to the current state, enabling the intelligent agent to learn better strategies. Comparative analysis with traditional control methods and the Presslight algorithm reveals that MALight outperforms in terms of reducing average vehicle travel times and increasing the average throughput of intersections.

2 Related Work

Deep reinforcement learning (DRL) [12] is a learning framework that combines deep learning and reinforcement learning, which is considered to be one of the most advanced frameworks in control systems [6, 8-10, 13-24]. In DRL, agents adopt deep neural networks to learn better policies to adapt to various complex traffic environments. In recent years, DRL-based methods and frameworks have been proposed and developed for smart traffic light control, in which the traffic lights (red, green, or yellow) are controlled automatically based on real-time and dynamic traffic conditions. For example, Genders et al. [13] used the DQN framework in traffic control systems, defining the state of an intersection as a two-dimensional value that includes the position and velocity information of vehicles at this intersection, through collecting traffic information in the vehicle network and then using convolutional neural networks (CNNs) for feature extraction to more accurately represent real-time environmental state information of the intersection. However, the action selection module and evaluation module in DQN are in the same network, which can cause overestimation issues. Some research has tried to alleviate the overestimation problem by improving the used DRL network frameworks. For example, Pol et al. [14] used an integrated learning method with double deep Q network (DDQN), prioritized experience replay (PER), and shadow target networks to enable adjacent intersections to interact and share their learned joint Q-function for optimal overall performance. DDQN can prevent the overestimation of Q-values and accelerate network convergence, resulting in better performance than DQN. In addition, In [15-16] used an integrated learning method with dueling double deep Q network (D3QN) and PER, which outperformed other models in terms of performance and learning speed and demonstrated better control effects than previous models. PER [25] selects samples from the experience pool based on their priorities, which are modified based on the values according to temporal difference (TD) errors, to enable the policy network to learn final policies better and more quickly.

Some earlier related studies based on the DQN [26] framework use complex states embedding and rewards calculation. For example, IntelliLight [9] used the DQN method to solve the TSC problem, and its state features consisted of five parts (queue length on the lanes, average waiting time, current traffic light state, number of waiting vehicles, and the state feature of the intersection extracted by Convolutional Neural Network). The reward was comprehensively decided by six elements (sum of queue lengths on the lanes, average waiting time, current traffic light, delay time of vehicles, number of vehicles passing through the intersection, and travel time of all vehicles). Currently, most reinforcement learning methods use more simple state representations and reward designs. Light-IntellighT (LIT) [10] only embeds the current phase and number of vehicles to a state, and the reward only depends on the vehicle queue length. Under the same network structure, its results were better than IntelliLight. PressLight [17] is a maxpressure-based method, and its state consists of the current phase, the number of vehicles entering and leaving the lane. The reward is designed to be pressure, which is defined as the difference between the numbers of vehicles entering and leaving a lane respectively. Experimental evaluation results have shown that PressLight performs better than LIT in terms of reducing average travel time. A RL model design called FRAP (which is invariant to symmetric operations like Flip and Rotation and considers All Phase configurations) [18] improved the network structure to capture the phase competition relation between different traffic movements. It can find the optimal policy faster and improve the network convergence speed.

CoLight [19] uses graph attention networks (GANs) to enhance the coordination and cooperation between agents, which can better handle cooperative control of multiple intersections. HiLight [20] introduces a hierarchical structure where each agent learns a high-level policy by selecting among several sub-policies to minimize the average travel time. In this structure, high-level policies focus on long-term goals while sub-policies optimize different short-term goals, allowing agents to learn to cooperate and optimize the average travel time.

In addition, there are several studies that have used various novel methods to control intelligent traffic signal lights. For example, AttentionLight [21] defines the state and reward of traffic phase selection as Max-QueueLength and proposes a new reinforcement learning-based model that uses self-attention mechanisms to capture phase correlations. AttendLight [22] adopts a policy-based model that can be applied to any traffic environment with different crossroads and phase combinations. It incorporates two attention mechanisms, one for handling different numbers of lanes and the other for handling different phases of intersections. In addition, DemoLight [23] integrates the idea of imitation learning, while MetaLight [24] proposes more generalized models using meta-learning strategies.

3 Preliminaries

This article explores the dynamic control of traffic signal phases in multi-intersection scenarios. For general reason, we assume that in the traffic network, each intersection is divided into four directions ("W", "E", "N", "S"), each of which includes six lanes, with three entry lanes and three exit lanes, as shown in Figure 1. It is notable that such an assumption is just an example and the proposed algorithm in this study could be generalized to more scenarios.

Traffic Movement: The lane entering an intersection is the entry lane, and the lane leaving the intersection is the exit lane. Traffic movement is defined as the traffic trajectory from entering the incoming lane to leaving the exit lane. With three types of traffic flow depending on the movement mode: straight, left turn, and right turn. The rightmost lane only allows straight and right turns, the middle lane only allows straight going, and the leftmost lane only allows left turns.

Traffic Environment: Traffic environment simulates the movement of vehicles on the road network based on the given traffic flow and pushes vehicles into the network at the corresponding time. The traffic control algorithm controls each traffic signal based on the current traffic volume.

Intersection and road network: The traffic network is described as a directed graph in which each node represents an intersection, each intersection is controlled by traffic signals and is designed as a two-way six-lane intersection.

Travel time: The time a vehicle takes from entering to leaving a specific area.

Phase: Traffic signals control corresponded traffic movements, with green indicating traffic movement is allowed and red indicating it is prohibited.

Traffic pressure: Defined as the difference between the numbers of vehicles on the entrance and exit lanes respectively, reflecting the degree of imbalance in terms of vehicle density between the entrance and exit lanes.

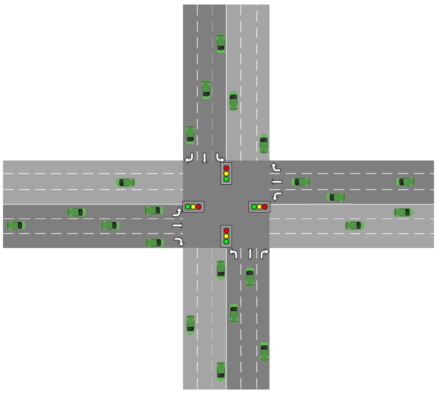


Figure 1. Intersections

4 MALight Algorithm

4.1 Problem Modelling

In DRL-based traffic control models, the traffic lights with control algorithms integrated at intersections are modeled as agents, while the controlling objects are the real-time changing traffic flows in the traffic network. The closed-loop interaction process between the agent and the controlling object is abstracted into a Markov decision process (MDP). Given real-time traffic flows and current traffic signals, the agent's goal is to select optimal actions to maximize the total reward. In the MALight model proposed in this study, an agent is set for each intersection in the road network, and models the control process as a MDP, employing multi-step DQN as the framework to control the traffic lights at intersections. It mainly consists of four modules as follows.

State representation module: Representing the current traffic signal control state as a vector that includes the respective numbers of vehicles on the entering and exiting lanes.

Action selection module: Using a deep neural network to learn the next action to be taken, controlling the phase of the signal lights.

Reward calculation module: Calculating immediate rewards according to the pressure Eq. (1).

Experience replay module: Saving the interaction history between the agent and the environment to train the neural network.

4.2 Multi-step DQN

In multi-step DQN, the agent captures the current phase and the numbers of vehicles in each lane based on the current state of the traffic intersection, and uses this observational data as the input for the neural network, as shown in Figure 2. Then, based on the pressure definition, the corresponding pressure is determined, and an action is accordingly selected from the action space to control the signal light phase, thus achieving intelligent control to the signal light and effective traffic management. The pressure is defined as:

$$\mathrm{P}_i = \mathrm{N}_{\mathrm{in}} - \mathrm{N}_{\mathrm{out}} \tag{1}$$

where P_i represents the pressure at intersection i, N_{in} and $\mathrm{N}_{\mathrm{out}}$ indicates the number of vehicles entering and exiting the lane, respectively. The reward is defined as the opposite number of pressure:

$$\mathrm{r}_i = -\mathrm{P}_i \tag{2}$$

The total reward of all traffic movements at an intersection is:

$$\mathrm{R}\left(\mathrm{s}_t, \mathrm{a}_t\right) = \sum \mathrm{r}_i \tag{3}$$

The traditional one-step DQN is a deep reinforcement learning algorithm that uses two neural networks with identical structures but different parameters: an evaluation value network used to approximate the Q values and a target

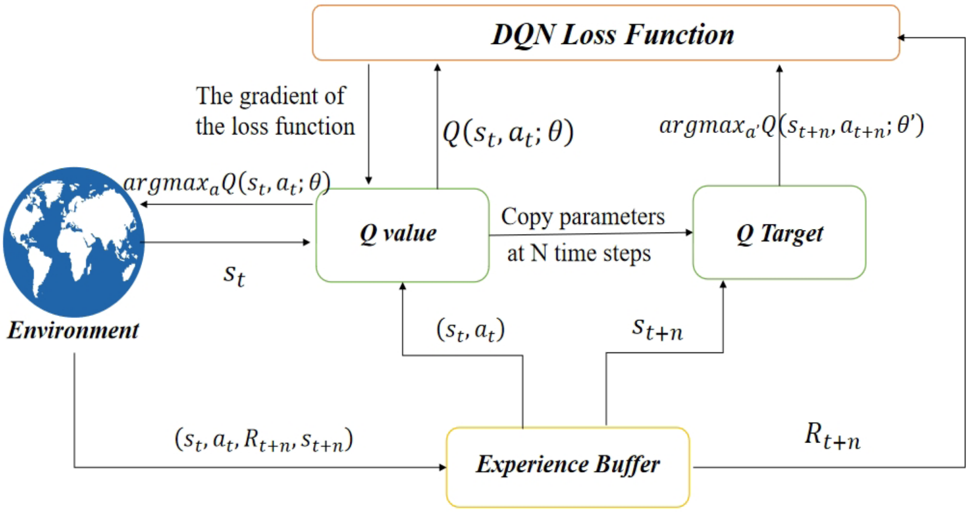

DQN Loss Function
The gradient of the loss function
$Q(s_t, a_t; \theta)$
$argmax_{a'}Q(s_{t+n}, a_{t+n}; \theta')$
$argmax_a Q(s_t, a_t; \theta)$
Q value
Copy parameters at N time steps
Q Target
s_t
Environment
(s_t, a_t)
s_{t+n}
$(s_t, a_t, R_{t+n}, s_{t+n})$
Experience Buffer
R_{t+n}

Figure 2. Diagram of multi-step DQN structure

value network used to calculate the target Q values. Since the DQN's target Q value is determined by the immediate reward and the maximum Q value at the next time step, the traditional one-step DQN has a slower learning speed in the early period. Multi-step DQN accumulates *n* single-step experiences and learns using the accumulated experience to improve the learning efficiency of the algorithm. The multi-step reward and multi-step target respectively are:

$$R_t^{(n)} = \sum_{k=0}^{n-1} \gamma_t^{(k)} R_{t+k+1} \tag{4}$$

and

$$\begin{aligned} y_{\text{target}} &= r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \max_a Q(s_{t+n+1}, a; \theta) \\ &= \sum_{i=0}^{n-1} \gamma^i r_{t+i+1} + \gamma^n \max_a Q\left(s_{t+n+1}, a; \theta\right) \end{aligned} \tag{5}$$

To update the network parameters of DQN, the mean squared error (MSE) is commonly adopted as the loss of the network, which is:

$$\begin{aligned} L(\theta) = \Sigma \frac{1}{B} (&\sum_{i=0}^{n-1} \gamma^i r_{t+i+1} \\ &+ \gamma^n \max_a Q\left(s_{t+n+1}, a; \theta\right) - Q\left(s_t, a_t; \theta\right)\Big) \end{aligned} \tag{6}$$

where $Q(s_t, a_t; \theta)$ is the predicted Q value and $\sum_{i=0}^{n-1} \gamma^i r_{t+i+1}$ $+\gamma^n \max_a Q\left(s_{t+n+1}, a; \theta\right)$ is the target value.

4.3 Attentive Experience Replay

During the interaction between the agent and the environment, the accumulation of experience is crucial. These experiences are represented by many quad-tuple (s, a, r, s') and stored in an experience pool, as shown in Figure 3. To update the network, the target network randomly selects a batch of samples from the experience pool. The advantages of experience replay are as follows. Firstly, that it can reduce the correlation between different samples through random sampling. Secondly, samples can repeatedly use for multiple times to improve the utilization efficiency of experiences. However, experience replay suffers from the problem that it cannot reflect the importance of experiences and thus past policies may not match the current policy when updating, affecting overall performance. To solve this problem, AER mechanism was proposed [11], which compares the distribution of states in the experience pool with the current state distribution, and accordingly selects the experiences with higher similarity for replay, thereby improving the convergence performance of the neural network. This further promotes more accurate value estimation and better action selection. Therefore, AER can be seen as an improved mechanism of experience replay that can effectively improve the learning efficiency and performance of the intelligent agent. Compared to random experience replay, AER can more effectively utilize historical experience data to improve the model's learning ability.

To further improve performance, we made some improvements to AER. We extract the experiences sampled from the most recent episodes in the experience pool and calculate their similarity with the remaining samples in the pool one by one. Then, the top K most similar experiences are selected as a mini batch for gradient updates of the neural network. Euclidean distance is adopted to calculate the similarity between two states.

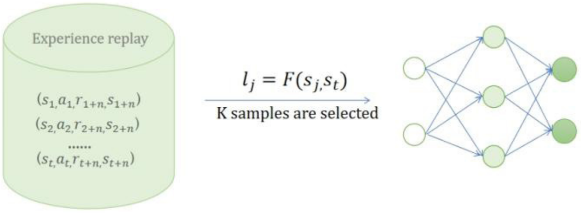

Experience replay
$(s_1, a_1, r_{1+n}, s_{1+n})$
$(s_2, a_2, r_{2+n}, s_{2+n})$
......
$(s_t, a_t, r_{t+n}, s_{t+n})$
$l_j = F(s_j, s_t)$
K samples are selected

Figure 3. Sample picking

Multi-step DQN is an improvement over the original DQN that introduces experience samples from multiple time steps to improve the accuracy, efficiency, and stability of value function estimation during training. Traditional DQN can only train on experience samples from a single time step, while Multi-step DQN can construct a more accurate value function estimate by using rewards from multiple time steps. The pseudocode of the detailed implementation of our proposed MALight is as follows.

```
MALight
Input: replay memory $D$, multi-step replay memory $D'$, multi-step replay memory size $N$, multi-step $n$, sample size $B$, mini-batch $k$, episode length $T$, discount factor $\gamma$, greedy $\varepsilon$, learning rate $\alpha$, replacement frequency $C$, similarity measure $F$.
 Initialize $Q$ with parameters $\theta$, $\hat{Q}$ with parameters $\hat{\theta}$
 for each episode do
   Initialize step number $t$ as 0, total time $t_{sum}$ as 0
   while $t_{sum} < T$ do
     Select a random phase $pha$ with probability $\varepsilon$
     Otherwise $pha \leftarrow \mathrm{argmax}_{pha}\, Q(s_t, pha; \theta)$
     Receive the green phase duration time $t_{green}$ from the environment
     Execute $a_t \leftarrow \{pha, t_{green}\}$
     Observe the new state $s_{t+1}$
     Calculate the reward based on max pressure
     $r_t = -(N_{in} - N_{out})$
     Store transition $(s_t, a_t, r_t, s_{t+1})$ in $D$
     if $t > n$ then
              $R_t^{(n)} = \sum_{k=0}^{n-1} \gamma_t^{(k)} R_{t+k+1}$
       Store transition $(s_t, a_t, R_t^{(n)}, s_{t+n})$ in $D$'
     end if
    $t_{sum} \leftarrow t_{sum} + t_{green}$, $t \leftarrow t+1$
     if $N > B$ then
       for $j = 1$ to $N$ do
         Uniformly sample transition $j$:$(s_j, a_j, R_j^{(n)}, s_{j+n})$
         Compute similar $l_j = F(s_j, s_t)$
       end for
     Select $k$ most similar transition $B$
     Calculate weight-change $\Delta$ using transitions $B$
     Update weights
     end if
   Calculate the loss $J$ by Eq. 6 and update $\theta$ by gradient descent
   with learning rate as $\alpha$
   Every $C$ step update $\hat{Q} \leftarrow Q$
   end while
 end for
```

5 Experimental Results

In this experiment, we employed the CityFlow [27] traffic simulator, which supports large-scale traffic signal control simulation, to evaluate the performance of MALight. By inputting traffic data into the simulator, vehicles are directed to their destinations according to the environmental settings, thereby simulating traffic flow.

5.1 Datasets

To evaluate the performance of MALight, both synthetic and real-world datasets are utilized in this evaluation. The synthetic dataset was synthesized from statistical analysis of real-world traffic patterns and used to learn the properties of different algorithms by manually designing traffic volume. The real-world dataset was collected from traffic data streams in the real world and has characteristics of randomness and irregularity.

For the synthetic dataset, a 1x6 intersection traffic network is used, where there is one road in the W-E direction and six roads in the N-S direction. The length of the lanes in both directions is 300 meters. Each intersection has four approaches, with each approach having three lanes for left turn, though, and right turn. The speed of the vehicles is 40 km/h. For the real-world dataset, some history traffic data flows of HangZhou and Jinan were used. The HangZhou dataset contains a 4x4 grid of intersections, with four roads in the W-E direction and four roads in the N-S direction, with the length of the lanes in the N-S direction being 800 meters and 600 meters respectively. The Jinan dataset contains 12 intersections in a 3x4 grid, with three roads in the W-E direction and four roads in the N-S direction. The real-world dataset was obtained through roadside surveillance cameras, recording the movement trajectory, speed, and entry time of vehicles, which was used to reproduce traffic in the simulator. Compared to the synthetic dataset, the real-world dataset is more random and unpredictable.

5.2 Experiment Settings and Baseline

In this experiment, a universal traffic signal control strategy was adopted for all intersections. Specifically, a green signal is followed by a 3-second yellow signal and a 2-second red signal to clear the intersection and prepare for the next phase of signal control. The simulation round length was set to 3600 seconds, and the signal control period for each phase was 10 seconds. In reinforcement learning training, the value of multi-step was set to 5, which means cumulative rewards were calculated every 5 steps. The size of the experience pool was set to 10000, and a sample batch size of 20 was used for each network update. A learning rate of 0.001 was used to update the parameters of the Q network, and a discount factor of 0.99 was set to calculate cumulative rewards. At the beginning of training, the initial exploration probability was set to 0.8 and gradually reduced to a minimum exploration probability of 0.2 to promote the model's learning of new traffic scenarios, as shown in Table 1.

Table 1. Settings for method

Model parameter	Value
Round length	3600 seconds
Action time interval Δt	10 seconds
n for multi-step	5
Memory size	10000
Sample size	1000
Bacth size	20
α for learning rate	0.001
γ for discount factor	0.99
$\mathcal{E}$ for exploration	0.8

To demonstrate the effectiveness of MALight, we used a traditional traffic light control model FixedTime [1] and two state-of-the-art reinforcement learning-based models, Maxpressure [5] and PressLight [17] as baselines.

FixedTime: a widely used approach that adopts a predetermined cycle length and is applied when traffic flow is stable.

Maxpressure: a reinforcement learning-based model selects the phase that maximizes the pressure.

PressLight: a reinforcement learning-based model that designs the reward function based on traffic pressure, minimizing traffic pressure to achieve uniform distribution of vehicles at intersections, while maximizing the reward.

5.3 Evaluation Metrics

Two representative metrics, the average travel time in seconds and the average traffic flow at intersections, are employed to evaluate different models.

Average Travel Time: The travel time of a vehicle is defined as the difference between the time of entering and leaving a road network, and the average travel time of all vehicles is employed as one of the metrics in this study.

Average Traffic Flow: Traffic flow is defined as the number of vehicles passing through a road network, and the higher the traffic flow is, the better the control strategy.

5.4 Results

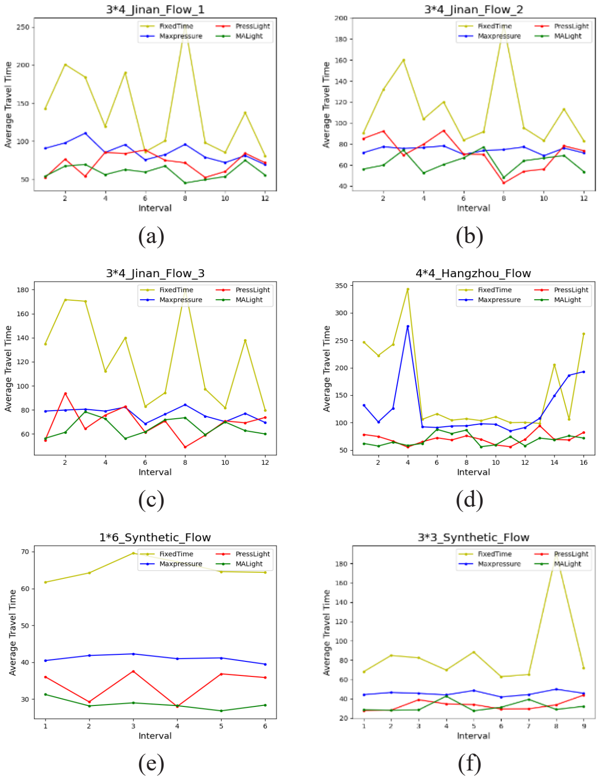
The evaluation results demonstrate that MALight significantly outperforms the original PressLight models in terms of reducing average travel time and increasing average throughput. In specific, the average travel time and average throughput for different models in each dataset, Table 2 and Table 3, respectively. Figure 4 shows the average travel time for different methods. In order to compare the network training time, we recorded the data while the process of network training, as shown in Figure 5. From Figure 4 and Figure 5, it can be seen that MALight not only achieves better performance, but also reduces the average training time of the network as a whole.

Table 2. Average travel time (seconds)

	Jinan-1	Jinan-2	Jinan-3	HangZhou	1-6	3-3
FixedTime	139.81	123.61	112.45	160.99	65.28	87.10
Maxpressure	86.28	76.77	74.38	125.87	40.99	45.67
PressLight	71.34	68.74	72.09	70.40	33.89	33.40
MALight	59.68	65.32	62.43	68.48	28.59	31.93

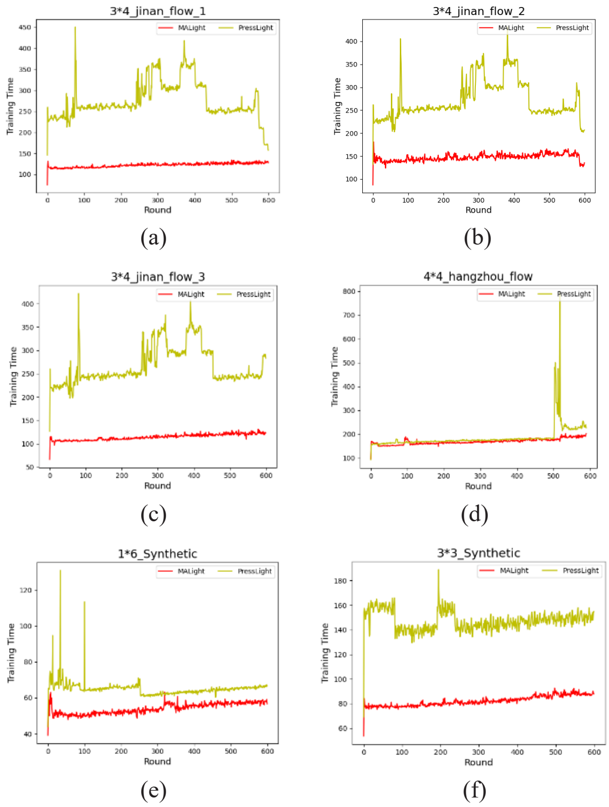
Table 3. Average throughput

	Jinan-1	Jinan-2	Jinan-3	HangZhou	1-6	3-3
FixedTime	1410	1308	1153	925	1223	1167
Maxpressure	1530	1420	1197	1226	1250	1207
PressLight	1513	1367	1169	1234	1256	1220
MALight	1540	1428	1206	1239	1257	1221



(a) (b) (c) (d) (e) (f)

Figure 4. The average travel time of intersections



(a) (b)

(c) (d)

(e) (f)

Figure 5. Training time

6 Conclusion

In this work, we propose MALight, a traffic light control algorithm with multi-step DQN and AER. We extensively tested our method and compared it with other algorithms, and the results showed that our method outperforms other reinforcement learning algorithms. Our method not only improves the average traffic flow at intersections, but also shortens the average travel time of vehicles. This indicates that our method has broad application prospects in practical applications and can provide strong support for urban traffic management. However, for complex traffic situations, considering only the current traffic conditions at individual intersections is insufficient. It's essential to take into account the traffic conditions at multiple intersections. In our future research, we aim to enable reinforcement learning agents to collaborate with other intelligent entities to address traffic congestion more effectively. Furthermore, to enhance the performance of TSC, we are also exploring more sophisticated state representations and innovative network structures.

References

[1] P. Koonce, R. Lee, *Traffic signal timing manual*, No. FHWA-HOP-08-024, United States, Federal Highway Administration, June, 2008.

[2] S. B. Cools, C. Gershenson, B. D'Hooghe, Self-organizing traffic lights: a realistic simulation, in: M. Prokopenko (Eds.), *Advances in Applied Self-Organizing Systems*, Springer, London, 2013, pp. 45-55.

[3] P. B. Hunt, D. I. Robertson, R. D. Bretherton, M. Cr Royle, The SCOOT on-line traffic signal optimisation technique, *Traffic Engineering & Control*, Vol. 23, No. 4, pp. 190-192, 1982.

[4] P. R. Lowrie, *Scats, Sydney co-ordinated adaptive traffic system, a traffic responsive method of controlling urban traffic*, 1990.

[5] P. Varaiya, Max pressure control of a network of signalized intersections, *Transportation Research Part C: Emerging Technologies*, Vol. 36, pp. 177-195, November, 2013.

[6] T. Le, P. Kovács, N. Walton, H. L. Vu, L. L. H. Andrew, S. S. P. Hoogendoorn, Decentralized signal control for urban road networks, *Transportation Research Part C: Emerging Technologies*, Vol. 58, pp. 431-450, September, 2015.

[7] X. Sun, Y. Yin, A simulation study on max pressure control of signalized intersections, *Transportation Research Record*, Vol. 2672, No. 18, pp. 117-127, December, 2018.

[8] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, Z. Li, Toward a thousand lights: decentralized deep reinforcement learning for large-scale traffic signal control, *AAAI Conference on Artificial Intelligence*, Vol. 34, No. 4, pp. 3414-3421, April, 2020.

[9] H. Wei, G. Zheng, H. Yao, Z. Li, IntelliLight: a reinforcement learning approach for intelligent traffic light control, *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom, 2018, pp. 2496-2505.

[10] G. Zheng, X. Zang, N. Xu, H. Wei, Z. Yu, V. Gayah, K. Xu, Z. Li, Diagnosing reinforcement learning for traffic signal control, *arXiv preprint arXiv:1905.04716*, May, 2019. https://arxiv.org/abs/1905.04716

[11] P. Sun, W. Zhou, H. Li, Attentive experience replay, *AAAI Conference on Artificial Intelligence*, Vol. 34, No. 4, pp. 5900-5907, April, 2020.

[12] Z. Yang, Y. Kong, C.-H. Hsia, DERLight: a deep reinforcement learning traffic light control algorithm with dual experience replay, *Journal of Internet Technology*, Vol. 25, No. 1, pp. 79-86, January, 2024.

[13] W. Genders, S. Razavi, Using a deep reinforcement learning agent for traffic signal control, *arXiv preprint arXiv:1611.01142*, November, 2016. https://arxiv.org/abs/1611.01142

[14] E. V. D. Pol, *Deep reinforcement learning for coordination in traffic light control*, Master's Thesis, University of Amsterdam, Amsterdam, Netherlands, 2016.

[15] X. Liang, X. Du, G. Wang, Z. Han, A deep reinforcement learning network for traffic light cycle control, *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 2, pp. 1243-1253, February, 2019.

[16] Z. Fu, J. Zhang, F. Tao, B. Ji, Traffic signal phase control at urban isolated intersections: an adaptive strategy utilizing the improved D3QN algorithm, *Measurement Science and Technology*, Vol. 36, No. 1, Article No. 016203, January, 2025.

[17] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, Z. Li, Presslight: learning max pressure control

to coordinate traffic signals in arterial network, *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 1290-1298.

[18] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, Z. Li, Learning phase competition for traffic signal control, *ACM International Conference on Information and Knowledge Management*, Beijing, China, 2019, pp. 1963-1972.

[19] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, Z. Li, Colight: learning network-level cooperation for traffic signal control, *ACM International Conference on Information and Knowledge Management*, Beijing, China, 2019, pp. 1913-1922.

[20] B. Xu, Y. Wang, Z. Wang, H. Jia, Z. Lu, Hierarchically and cooperatively learning traffic signal control, *AAAI Conference on Artificial Intelligence*, Vol. 35, No. 1, pp. 669–677, May, 2021.

[21] L. Zhang, Q. Wu, J. Deng, AttentionLight: rethinking queue length and attention mechanism for traffic signal control, *arXiv preprint arXiv:2201.00006*, April, 2022. https://arxiv.org/abs/2201.00006v2

[22] A. Oroojlooy, M. Nazari, D. Hajinezhad, J. Silva, AttendLight: universal attention-based reinforcement learning model for traffic signal control, *Advances in Neural Information Processing Systems*, vol. 33, Vancouver, BC Canada, 2020, pp. 4079-4090.

[23] Y. Xiong, G. Zheng, K. Xu, Z. Li, Learning traffic signal control from demonstrations, *ACM International Conference*, Beijing, China, 2019, pp. 2289-2292.

[24] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, Z. Li, MetaLight: value-based meta-reinforcement learning for traffic signal control, *AAAI Conference on Artificial Intelligence*, Vo. 34, No. 1, pp. 1153-1160, April, 2020.

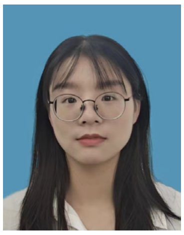
[25] T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay, *International Conference on Learning Representations*, San Juan, Puerto Rico, 2016, pp. 1-21.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, *NIPS Deep Learning Workshop*, 2013, pp. 1-9. https://arxiv.org/abs/1312.5602

[27] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, Z. Li, Cityflow: a multi-agent reinforcement learning environment for large scale city traffic scenario, *World Wide Web Conference*, San Francisco CA USA, 2019, pp. 3620-3624.

Biographies

Yan Kong received her Ph.D. degree in Computer Science from the University of Wollongong, Australia. Currently, she works as a faculty in Nanjing University of Information, Science and Technology, China. Her research interests include Deep learning, Multi-agent system, and Machine Learning. Her research focuses on the smart control on the traffic signal lights to alleviate the traffic congestion.



Ying Li received his M.S. degree in School of Software from Nanjing University of Information Science and Technology, China. His research interests include Deep Reinforcement Learning in the application of intelligent traffic signal light control.

Chih-Hsien Hsia received the Ph.D. degree in Electrical and Computer Engineering from Tamkang University, and the second Ph.D. degree from National Cheng Kung University, Taiwan, respectively. He currently is a Full Professor and a Chairperson with the Department of Computer Science and Information Engineering, NIU. His research interests include DSP IC Design, AI in Multimedia, and Cognitive Learning.