Traffic Light Control at Intersections in Case of Heterogeneous Traffic Using Reinforcement Learning

Do Thai Giang[1,2], Truong Cong Doan[1], and Phan Duy Hung[2(✉)]

[1] International School, Vietnam National University, Hanoi, Vietnam
{24078012,tcdoan}@vnu.edu.vn, giangdt26@fe.edu.vn
[2] FPT University, Hanoi, Vietnam
hungpd2@fe.edu.vn

Abstract. One of the urgent problems in traffic is to reduce congestion and waiting time of vehicles participating in traffic. This paper focuses on the employment of reinforcement learning algorithms such as Double Deep Q-Network and Proximal Policy Optimization to reduce vehicle waiting times in three-way and four-way intersections with heterogeneous traffic conditions. At the same time, we also investigate the performance of the algorithm in both heavy and clear traffic scenarios. Our method dramatically lowers waiting times and improves traffic flow efficiency, according to extensive testing. Significantly, the outcomes reveal remarkable performance gains in both three-way and four-way intersection, indicating the possibility of widespread use in urban traffic control. When compared to the fixed time strategy, the experimental results likewise perform better in both clear and congested traffic scenarios. This study demonstrates the effectiveness of intelligent traffic control systems in varied and complex traffic environments.

Keywords: reinforcement learning · traffic light control · heterogeneous traffic · DDQN · PPO · SUMO simulation

1 Introduction

Traffic management at isolated crossroads is quite difficult, especially in situations with heterogeneous traffic. Heterogeneous traffic means that complicated traffic situation due to the participation of many types of vehicles (cars, buses, motorbikes, trucks, etc.) interspersed, where the mix of vehicles might differ greatly in terms of size, speed, and driving style. Such complexity is frequently too hard for traditional traffic signal management systems to handle effectively, which results in more traffic congestion, longer waiting times, and higher emissions.

The traditional traffic light control method that uses fixed light phase times is often ineffective. For instance, when traffic is more congested during rush hour than during other times, if the light phases are set to deal with congestion, it is not appropriate for clear traffic conditions, which creates an unreasonable situation and forces vehicles to stop and wait for traffic lights when the road is very clear. Recent developments in

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15896, pp. 419–431, 2026.
https://doi.org/10.1007/978-3-031-97654-4_26

reinforcement learning (RL) have demonstrated encouraging promise for dynamic, data-driven traffic light control optimization. In order to maximize flow and minimize delays, RL algorithms are able to continuously improve signal timings by learning and adapting to real-time traffic conditions.

The main contribution of this research is to experimentally and evaluate the ability to reduce waiting time of vehicles by applying PPO and DDQN algorithms in controlling traffic lights at three-way and four-way intersections. At the same time, these algorithms are also applied in congestion situations and smooth traffic flow situations. Although there have been previous studies using the PPO algorithm in traffic light control, this study mainly focuses on comparing PPO and DDQN in increasing the reward value of the algorithms without clarifying the ability to reduce waiting time and limit congestion of the PPO algorithm in different traffic situations.

2 Literature Review

2.1 Reinforcement Learning in Traffic Management

In recent years, with the development of artificial intelligence and optimization algorithms, there has been increasing interest in applying these algorithms to various areas of life, in which traffic congestion is always an urgent problem in large cities. Traditional traffic light control systems are increasingly showing their ineffectiveness in modern traffic when traffic density is very high as well as traffic situations change rapidly and are complex. There is growing interest in and studying powerful optimization algorithms for traffic signal control to mitigate congestion. Traditional traffic light control techniques were the mainstays of early traffic signal control systems. Whereas actuated systems modify signals in response to traffic demand at intersections, pre-timed systems follow set schedules. Even while they work well in some situations, these conventional methods find it difficult to adjust to changes in traffic in real time, especially in locations with diverse traffic patterns.

A dynamic and adaptable method for controlling traffic signals is provided by reinforcement learning, a subfield of machine learning. Through interaction with the environment and incentives based on behaviors, RL algorithms develop the best rules. The potential of RL resides in its capacity to dynamically optimize traffic flow and adjust to shifting traffic conditions. The optimization of signal timings at isolated crossings was the main focus of early RL research for traffic management. Li et al. (2016) [1], for example, used Q-learning at a four-way intersection and showed that, in comparison to conventional techniques, average vehicle waiting times were decreased.

Deep reinforcement learning approaches including deep Q-networks (DQN) and policy gradient methods were added to later studies, which demonstrated even greater gains in traffic efficiency. Romain Ducrocq and Nadir Farhipropose (2021) [2] offer a deep reinforcement Q-learning model to enhance traffic signal regulation at an isolated intersection. DQN model sub-stantially improves the performance of traffic light control compared to deployed actuated methods Max Pressure [3] and Self-Organising Traffic Lights (SOTL) [4], achieving both fairness between vehicles and global efficiency at the intersection.

Li et al. (2022) [5] presents an enhanced Distributional DQN to develop a signal optimization decision-making model employing reinforcement learning grounded in value distribution. The simulation results demonstrate that the Distributional DQN proposed in the paper has a faster convergence rate than the original DQN, and the cumulative delay at the intersection is reduced by about 13.1%, and the average driving speed is increased by 7.1%, which further improves the control efficiency of signalized intersections.

Another notable contribution is the publication of Yue Zhu et al. (2022) [6] on the use of Proximal Policy Optimization (PPO) algorithm in traffic light control at an intersection. The paper showed that PPO has better results than DQN and DDQN. In this paper, the authors compared the rewards of the algorithms but did not give results on its effectiveness in improving the traffic situation. In this study, the authors also only focused on applying the PPO algorithm to the situation of an intersection with unbalanced traffic but did not pay attention to considering other traffic situations such as at three-way intersections, situations with smooth traffic flow and balance between directions.

Liu et al. (2023) [7] examines the utilization of deep reinforcement learning (DRL) for the management of traffic signals in intelligent transportation systems (ITSs). The authors propose using two DRL algorithms - DQN and Deep Deterministic Policy Gradient (DDPG) - to manage traffic lights in both single road intersections and grid road networks. Theoretical analysis and numerical experiments demonstrate that the DQN algorithm delivers optimal control, while the DDPG algorithm produces intelligent "greenwave" behavior, where vehicles experience a progressive cascade of green lights without needing to brake. The results highlight the scalability and effectiveness of DRL algorithms in improving traffic efficiency and reducing congestion.

2.2 Heterogeneous Traffic

Heterogeneous traffic, defined by a variety of vehicle kinds and behaviors, poses distinct issues for traffic management. Zhang et al. (2017) [8] proposes a novel model for heterogeneous traffic networks that includes both signalized and non-signalized crossings. The authors propose and validate a mixed integer programming approach for signal control using VISSIM simulations.

Eom, M., and Kim, BI (2020) [9] published a review paper that gives a complete examination of challenges, methodologies, and practices in traffic signal control at intersections, including diverse traffic situations.

Hung, P.D., and Giang, D.T. (2021) [10] released a paper on the use of fuzzy logic to optimize traffic signals at isolated crossroads in heterogeneous traffic conditions. The system is simulated with SUMO simulation, which shows substantial advantages over fixed traffic lights.

There are very few researches on traffic light control under current heterogeneous traffic conditions, and most of them use old algorithms with low efficiency.

2.3 Three-Way and Four-Way Configurations

The majority of traffic light control research focuses on traffic problems at four-way junctions, with very few studies examining traffic light management at three-way intersections. One study that should be mentioned is the publication of Maheen Firdous et al. (2019) [11]. The paper describes a traffic signal control system that uses fuzzy logic to reduce the number of queues and wait times for cars at crossings.

Another recent study by Ni et al. (2024) [12] provides traffic signal control solutions for T-shaped crossings in urban road networks based on deep Q network algorithms. The simulation findings demonstrate that combining the Dueling-DQN approach with dynamic time aggregation considerably increased vehicle throughput. Compared to DQN and fixed-time techniques, this strategy can reduce average journey time by up to 43% in low-traffic hours and 15% in high-traffic periods.

3 Proposed Method

3.1 Algorithms

Previous articles have examined a variety of methods and techniques for traffic light regulation at crossings. Following a comprehensive evaluation of the works, we chose two state-of-the-art algorithms to experiment in this paper including Double Deep Q-Network (DDQN) and Proximal Policy Optimization (PPO). In this section, we'll talk about these two algorithms and their optimization goals.

A. *Double Deep Q-Network*

DDQN is an extension of the Deep Q-Network algorithm that is intended to overcome the overestimation bias of Q-values in regular DQN. Hado van Hasselt et al. [13] introduced DDQN in 2015, which distinguishes between selecting the best action and evaluating it. It employs two networks: a selection network to select the action and a target network to evaluate it. This separation reduces overestimation of Q-values and increases the stability and performance of the learning process. The DDQN architecture is shown as Fig. 1.

The DDQN technique is based on Q-learning, which is a policy temporal difference (TD) learning algorithm that approximates the value of a state-action value function, or Q-function, using previously obtained estimates of this function. The Q-function is defined as the expected discounted reward that an agent would earn when starting the game in state s with action a and then acting according to policy π. The mathematical description is as follows:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[R_t | s_t = s, a_t = a] \tag{1}$$

where R_t represents the expected reward at the end of the session. The Q-function evaluates an agent's performance as it moves from one state to another based on the set policy. The policy is a greedy policy defined as *argmax(Q(s, a))* for all actions. During training, the agent learns and converges on the optimal strategy to maximize total reward per episode. The Q-function itself can be expressed recursively:

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q\left(s', a'\right) \tag{2}$$

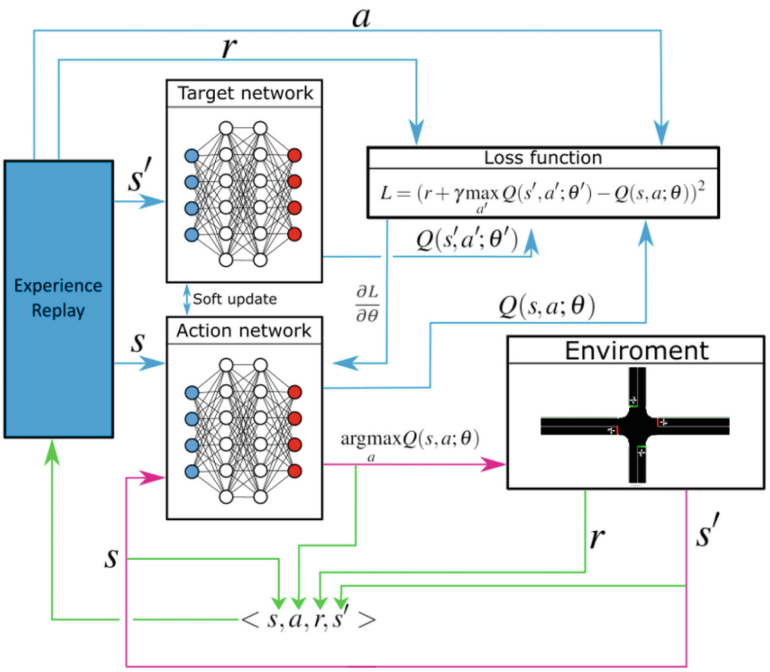

Fig. 1. The architecture of double deep Q-learning network [14]

where *r(s, a)* is the reward for moving from state *s* to state s' by doing action *a*, and γ ∈ [0, 1] is a discount factor that sets the relative value of future and present rewards. ϵ-greedy approach is used to select an action throughout the learning process, allowing the agent to explore the environment in many ways. Throughout the episode, the agent chooses actions with the highest value of the Q-function in state *s* with probability (1 - ϵ) or randomly with probability ϵ, where ϵ ∈ [0, 1]. After the agent has passed from state s to state s' and received a reward *r*, the value of the Q-function is changed using

$$Q^{new}(s, a) = Q(s, a) + \alpha\left(r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right) \tag{3}$$

We get a deep reinforcement learning method when we employ deep neural networks to approximate the policy, Q-functions, or another RL function. This method is utilized when working with continuous situations where the number of states or actions is limitless. The Deep Q-Network (DQN) approach uses neural networks (NN) to approximate the values of the function $Q(s, a, \theta)$, where θ represents the weights in the deep neural network. In this scenario, we utilize a multilayer neural network that accepts state *s* as input and outputs a vector of values $Q(s, a_i, \theta)$ for each potential action a_i.

The neural network is trained using the backpropagation method, where the loss function is described as the square of the difference between $Q(s, a)$ and $Q^{new}(s, a)$, which is obtained from Eq. (3):

$$L = \left(r + \gamma \max_{a'} Q(s', a', \theta') - Q(s, a, \theta)\right)^2 \tag{4}$$

where θ and θ' are the weights of two neural networks with the same topology, called action network and target network, respectively (see Fig. 1). During error propagation training, the action network is updated, and the target network is updated by copying the action network's weights (θ) every few episodes. This technique, known as the Double Deep Q-learning Network, allows to prevent overestimating the action-state function in the learning process.

B. *Proximal Policy Optimization*

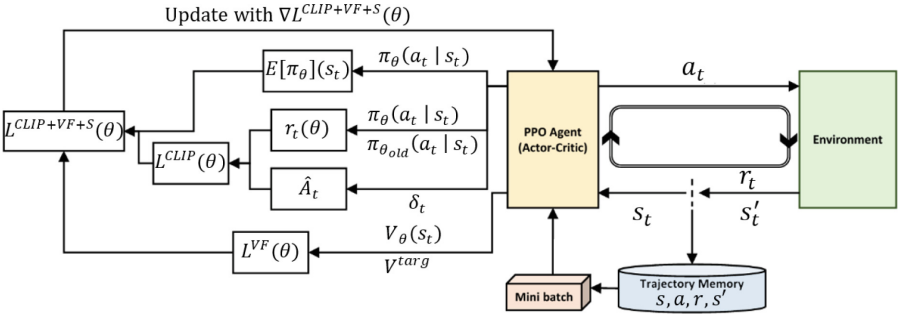

Fig. 2. The architecture of Proximal Policy Optimization algorithm

PPO is a widely utilized reinforcement learning technique developed by John Schulman et al. in 2017. PPO aims to improve the stability and efficiency of policy gradient methods by striking a balance between data efficiency and ease of implementation.

The key innovation in PPO is the clipping of the objective function, which prevents overly large policy updates. This clipping mechanism ensures that the new policy does not deviate too much from the old policy, maintaining stability during training. PPO uses an actor-critic approach, where the actor network predicts actions and the critic network estimates the state values, providing helpful training signals to guide the actor's learning. PPO is a method to simplify the complex calculation of trust region policy optimization (TRPO) [15]. TRPO maximizes a surrogate objective as follows:

$$L_t^{TRPO}(\theta) = \mathbb{E}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \mathbb{E}\left[r_t(\theta)\,\hat{A}_t\right] \tag{5}$$

where a_t and s_t represent an action and state, respectively, and $r_t(\theta)$ represents the probability ratio of policy distribution in time step t. Maximizing the surrogate objective function L^{TRPO} of TRPO without a constraint would result in an unreasonably large policy update. Additionally, the surrogate objective function L^{TRPO} has a complicated formula expansion and must determine the second derivative [15]. Therefore, by using the clipping method to approximate the first derivative, PPO was able to compensate for the inadequacies of TRPO. The clipping is applied to the following goal function:

$$L_t^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\,\hat{A}_t, \text{clip}\Big(r_t(\theta), 1-\epsilon, 1+\epsilon\Big)\hat{A}_t\right)\right] \tag{6}$$

where ϵ is a tiny hyperparameter that determines how far the new policy can deviate from the previous one. The main objective of PPO is optimizing the fusion loss function [16]:

$$L_t^{CLIP+VF+S}(\theta) = \mathbb{E}\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 E[\pi_\theta](s_t) \right] \tag{7}$$

where c_1, c_2 are coefficients, and E denotes an entropy bonus, and $L_t^{VF}(\theta)$ is a squared-error loss in Eq. 8.

$$L_t^{VF}(\theta) = \left(V_\theta(s_t) - V_t^{targ} \right)^2 \tag{8}$$

3.2 Environment

According to Markov Decision Process (MDP), the reinforcement learning agent learned from interacting with environment. Let's discuss about environment in Figs. 1 and 2. The purpose of this project is to control traffic lights at an intersection in order to optimize light phases and reduce vehicle wait time, therefore we used The SUMO Reinforcement Learning (SUMO-RL) [17] framework as environment. SUMO-RL is an effective tool for developing and testing reinforcement learning algorithms for traffic management and control utilizing the Simulation of Urban MObility (SUMO) simulator (Fig. 3). This framework creates a seamless interface between RL algorithms and SUMO, allowing simulate and optimize traffic signal control, vehicle routing, and other traffic-related activities.

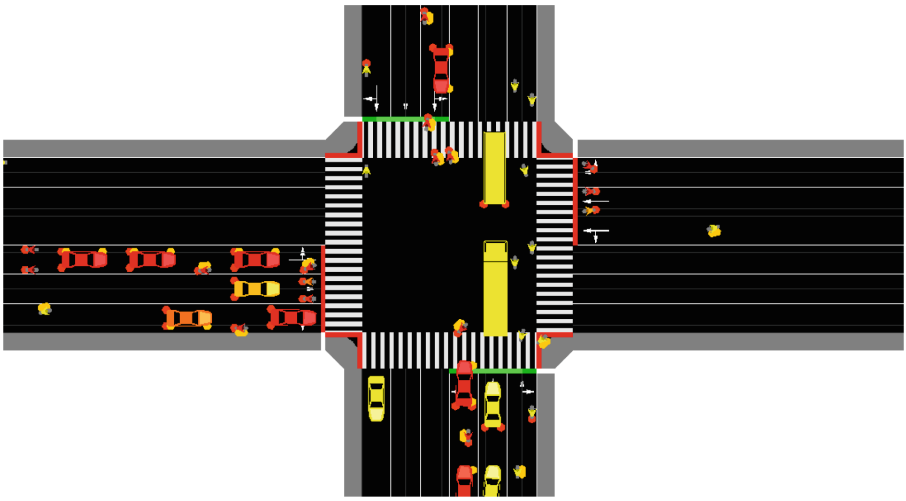

Fig. 3: SUMO Simulator

Key features of the SUMO RL framework include:

- *Compatibility*: It supports popular RL libraries such as Gymnasium, PettingZoo, stable-baselines3, and RLlib [17].
- *Multiagent RL*: The framework supports multiagent RL environments, allowing for the simulation of complex interactions between multiple agents [17].
- *Customization*: Easy define custom states, actions, and rewards to tailor the RL environment to their specific needs [17].
- *Efficiency*: The framework leverages SUMO's capabilities to create realistic and scalable traffic simulations, making it suitable for large-scale experiments.

The SUMO RL framework is widely used for developing and testing RL-based traffic management solutions, such as optimizing traffic signal timings, reducing congestion, and improving overall traffic flow [18].

3.3 State Space

The state space for traffic signal agent in SUMO-RL environment is a vector:

$$state = \left[phase, \min_green, L1_density, \ldots, Ln_density, L1_queue, \ldots, Ln_queue\right]$$

where:

- *phase* is a one-hot encoded vector indicating the current active green phase
- *min_green* is a binary variable indicating whether *min_green* seconds have already passed in the current phase
- *Li_density* is the number of vehicles in incoming lane *i* divided by the total capacity of the lane
- *Li_queue* is the number of queued (speed below 0.1 m/s) vehicles in incoming lane *i* divided by the total capacity of the lane

3.4 Action Space

The action space in the SUMO-RL environment is discrete1. Every *delta_time* seconds, each traffic signal agent can choose the next green phase configuration. For example, in a 2-way single intersection, there are 4 discrete actions $A = \{1, 2, 3, 4\}$, corresponding to the following green phase configurations:

1. Green for the north-south direction
2. Green for the east-west direction
3. All-red phase
4. Green for left turns (if applicable)

Sudden phase changes can create a collision probability, to limit that problem we have used yellow light phase as a buffer time for phase change so that vehicles have time to react to the change. Every time a phase change occurs, it is preceded by a yellow phase lasting *yellow_time* seconds.

3.5 Reward Function

The reward function is chosen for the purpose of reducing the waiting time of vehicles, so that it is defined as:

$$r = D_t - D_{t+1} \tag{9}$$

where D_t is sum of the waiting times of all approaching vehicles in time step t. That is, the reward is how much the total delay changed in relation to the previous time-step.

4 Experiment

4.1 Experiment Scenario

We will conduct experiments in two scenarios at crossroads: three-way and four-way intersections (see Fig. 4). In the four-way intersection situation, each direction is a two-way road with a total of 6 lanes. In the three-way intersection situation, the main direction will be a two-way road with a total of 6 lanes, and the secondary direction will be a two-way road with 4 lanes. For the 4-way intersection, the traffic light configuration for fixed phase time, each phase will be set to 37 seconds of green light and 3 second for the yellow light. With 3-way intersection, the green phase is set to 45 seconds for the main direction and 25 seconds for the sub direction.

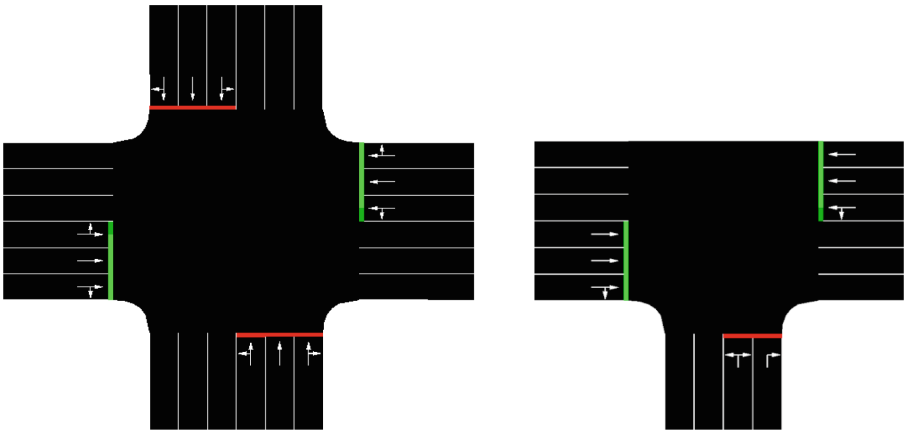

Fig. 4. Four-way and three-way configuration

4.2 Traffic Flow Scenarios

To evaluate the algorithm's efficacy in diverse scenarios, we also set up simulated traffic flow configurations for rush hour scenarios with large traffic volumes and the potential for congestion if fixed light timing is used. We also wanted to investigate the algorithm in

relatively clear traffic situations where fixed time methods can handle it effectively. The traffic flow in the congested scenario was set to be around 3400–3600 vehicles passing through the intersection per hour. In the clear traffic scenario, the flow was set to be around half of the congested scenario, meaning a flow of around 1800–2000 vehicles per hour.

4.3 Heterogeneous Traffic

As mentioned earlier, an important issue that this paper focuses on is the heterogeneous traffic conditions where there are many vehicles of different sizes and different speeds participating in traffic at the same time. In this paper, the situation mentioned here will include the types of vehicles mentioned in Table 1.

Table 1. Distribution probability of the types of vehicles

Vehicle type	Probability
Motor bike	63%
Car	31%
Bus	3%
Truck	3%

4.4 Result Analysis

The experimental process shows very impressive results when the algorithms have great improvement. Figure 5 shows that the waiting time of vehicles is greatly reduced. The congestion situations have also been completely resolved. The traffic flow through the intersection is more stable when the algorithms are applied.

Even in clear traffic situations, when the vehicles are not congested, the waiting time of vehicles is also reduced. When the traffic light phase times are fixed, although the traffic situation is very clear, the vehicles still have to wait in vain. While the algorithms have continuously adjusted the headlight phase time to suit the situation, when the vehicles approach the intersection, they can quickly pass the intersection, which obviously leads to a reduction in waiting time.

The experimental results also show that the PPO algorithm outperforms DDQN in all scenarios. Especially in congested scenarios, the performance of PPO over DDQN is even more evident. PPO shows better stability while DDQN results show some points where the waiting time increases quite sharply.

Table 2 shows the quantitative comparison results between the control methods. The data shows that the average waiting time per second when applying algorithms is completely superior to using the fixed time method. In all scenarios, the PPO algorithm always gives better results than DDQN.

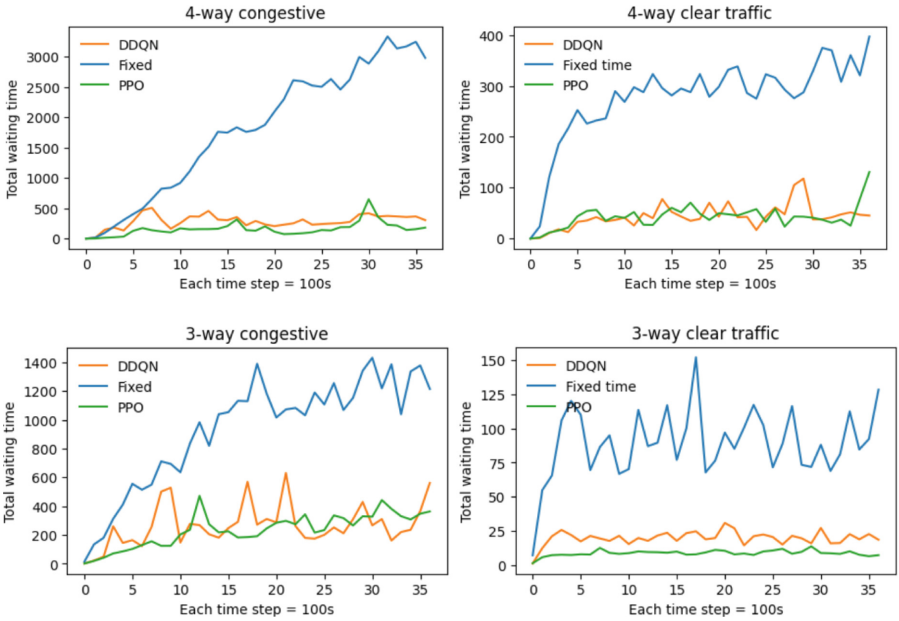

Fig. 5. Accumulated waiting time for every 100 seconds.

Table 2. Compare average waiting times per second

	3-way congestive	3-way clear traffic	4-way congestive	4-way clear traffic
Fixed time	927.74	197.44	1386.93	272.66
DDQN	259.37	19.7	285.11	43.23
PPO	**231.62**	**8.61**	**157.90**	**40.5**

The ability to process and respond to real-time traffic conditions is an essential component of traffic light control. In our trials, the inference time per decision cycle for both PPO and DDQN was 1.8 ms on an Intel Core i7 8750H CPU. This finding illustrates that the algorithms can perform effectively within practical restrictions, allowing for rapid modifications to traffic lights with little delays.

Compared to traditional traffic light control methods that rely on fixed time phases, reinforcement learning offers dynamic decision-making capabilities, allowing the system to continuously optimize light timings based on real-time traffic flow. The measured inference time shows that, despite the complexity of heterogeneous traffic situations, the algorithms retain low latency while significantly lowering vehicle waiting times.

5 Conclusion and Perspectives

The experimental results of the study have shown the feasibility and superiority of reinforcement learning algorithms in traffic light control. The algorithms can continuously adjust the timing of the light phases in accordance with the current traffic situation, significantly reducing the waiting time of vehicles. The paper also shows that the algorithms are effective in traffic situations including heterogeneous traffic conditions with many different types of vehicles.

By combining real-time traffic data and learning from observed traffic patterns, these algorithms dynamically adjust to changing traffic conditions, assuring optimal traffic flow. Notably, the adaptability of these algorithms enables them to handle complicated circumstances like intersections with a wide range of vehicle kinds and volumes. This versatility is critical in metropolitan areas where traffic patterns might alter quickly owing to unforeseen occurrences or shifting traffic needs.

Looking foward, there are several intriguing areas for future research. One area of interest is the possibility that multi-agent reinforcement learning (MARL) approaches can lead to progressively more complex traffic management solutions. MARL allows many traffic light controllers to collaborate and share information, resulting in more coordinated and harmonic traffic flow across bigger networked intersections.

In conclusion, the use of reinforcement learning in traffic signal regulation represents a significant improvement in intelligent transportation systems. The excellent findings of this study indicate the potential for widespread acceptance and ongoing innovation in this field, paving the way for smarter and more effective urban traffic management.

Acknowledgments. This research was supported by the **Doctoral Scholarship Program** of the **International School, Vietnam National University, Hanoi**. The financial assistance provided through this scholarship has enabled the authors to conduct extensive experiments and simulations, contributing to the advancement of intelligent traffic management using reinforcement learning techniques.

References

1. Li, Y., Xu, K., Li, D.: Traffic light control with reinforcement learning. J. Traf. Manag. **12**(3), 45–53 (2016)
2. Ducrocq, R., Farhi, N.: Deep reinforcement Q-learning for intelligent traffic signal control with partial detection. Int. J. Intell. Transp. Syst. Res. **21**, 192–206 (2021)
3. Minnesota Transportation Research. Investigating Max-Pressure Traffic Signal Timing. Available at: https://mntransportationresearch.org/2023/02/14/investigating-max-pressure-traffic-signal-timing/ (Accessed 10 Oct 2024)
4. Reztsov, A.: Self-Organising Traffic Lights (SOTL) as an upper bound estimate. (2014). https://doi.org/10.2139/ssrn.2467948
5. Li, J., Chen, T., Zhou, F., Lv, X., Peng, F.: Research on signal control method of deep reinforcement learning based on value distribution. In: Journal of Physics: Conference Series (2330/012019), vol. 2330, p. 012019. IOP Publishing (2022)
6. Zhu, Y., Cai, M., Schwarz, C.W., et al.: Intelligent traffic light via policy-based deep reinforcement learning. Int. J. ITS Res. **20**, 734–744 (2022). https://doi.org/10.1007/s13177-022-00321-5

7. Liu, X. Y., Zhu, M., Borst, S., & Walid, A.: Deep reinforcement learning for traffic light control in intelligent transportation systems. arXiv:2302.03669 (2023)
8. Zhang, Y., Su, R., Sun, C., Zhang, Y.: Modelling and traffic signal control of a heterogeneous traffic network with signalized and non-signalized intersections. In: Proceedings off the IEEE Conference on Control Technology and Applications (CCTA), Maui, HI, USA, pp. 1581–1586 (2017)
9. Eom, M., Kim, B.I.: The traffic signal control problem for intersections: a review. Eur. Transp. Res. Rev. **12**, 50 (2020)
10. Hung, P.D., Giang, D.T.: Traffic light control at isolated intersections in case of heterogeneous traffic. In: Kreinovich, V., Hoang Phuong, N. (eds.) Soft Computing for Biomedical Applications and Related Topics Studies in Computational Intelligence, vol. 899. Springer, Cham (2021)
11. Firdous, M., Din Iqbal, F.U., Ghafoor, N., Qureshi, N.K., Naseer, N.: Traffic light control system for four-way intersection and t-crossing using fuzzy logic. In: Proceedings of the IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 178–182 (2019)
12. Ni, W., Li, C., Wang, P., Li, Z.: Traffic signal optimization at T-shaped intersections based on deep Q networks. In: Luo, B., Cheng, L., Wu, Z.G., Li, H., Li, C. (eds.) Neural Information Processing. ICONIP 2023 Lecture Notes in Computer Science, vol. 14449. Springer, Singapore (2024)
13. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-Learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16), pp. 2094–2100. AAAI Press (2016)
14. Kuprikov, E., Kokhanovskiy, A., Serebrennikov, K., Turitsyn, S.: Deep reinforcement learning for self-tuning laser source of dissipative solitons. Sci. Rep. **12**(1), 7185 (2022)
15. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: Proceedings of the International conference on machine learning (ICML), Lille, France (2015)
16. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 (2017)
17. Alegre, L: SUMO-RL: Traffic signal control with reinforcement learning (n.d.). Retrieved from https://lucasalegre.github.io/sumo-rl/
18. Song, Y., Zhao, H., Luo, R., Huang, L., Zhang, Y., Su, R.: A SUMO framework for deep reinforcement learning experiments solving electric vehicle charging dispatching problem. arXiv:2209.02921 (2022)