Proceedings of the 4th International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/2026.33417

Intelligent Traffic Light Control Based on Reinforcement Learning

Yubo Wang

School of Statistics and Data Science, Capital University of Economics and Business, Beijing, China
32024230103@cued.edu.cn

Abstract. With the advancement of technology, the number of vehicles has increased, and the traffic problems have become more severe. During the morning rush hour, traffic congestion has become a common occurrence. The traditional traffic lights cannot determine when to change the lights as the cycle changes. Therefore, the traditional traffic lights that change according to a fixed cycle are no longer suitable for the current traffic conditions. This experiment aims to investigate the feasibility of an intelligent traffic light control system based on deep reinforcement learning (DQN). By constructing a simplified simulation environment of a crossroads, the DQN intelligent agent was designed to dynamically adjust the phase switching, and a comparative experiment was conducted with the fixed-duration strategy. The aim of this study is to cut down queuing time, lower time costs and fuel consumption. The experimental results show that the DQN agent can autonomously learn the optimal control strategy, reducing the average queue length by approximately 28.6%, verifying the application potential of reinforcement learning in intelligent traffic control. This research conducted a systematic analysis of aspects such as technical feasibility, data feasibility, and one's own capabilities, providing a framework for the development of intelligent traffic lights.

Keywords: Reinforcement Learning, DQN, Intelligent Transportation, Traffic Light Control, Feasibility Study

1. Introduction

The traditional traffic lights, due to their periodic changes, cannot adapt to the situation of a large number of vehicles in big cities, resulting in inevitable traffic congestion and unnecessary congestion. The economic losses and time costs caused by this are incalculable. Even some traffic jams could have been avoided and controlled. According to statistics, traffic congestion in major cities accounts for over 30% of the travel time [1], and the economic losses caused each year are as high as hundreds of billions [2]. And traffic lights, as the basic infrastructure for traffic control, can directly affect the efficiency of traffic flow at intersections [3]. At present, most urban intersections adopt a fixed-time control strategy - the traffic lights switch between red and green in a predetermined cycle [4]. The limitations of this strategy are obvious. During the morning rush hour, when the east-to-west traffic flow is heavy, the green light time in the north-south direction is wasted; while at night when the traffic flow is sparse, vehicles still have to wait for the red light.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.33417

This "one-size-fits-all" control method is unable to adapt to the dynamic changes of traffic flow [5]. Although reinforcement learning for controlling traffic lights can reduce queuing time, its practical operation still requires time [6]. The following are the research and social values of this article. The social value lies in reducing vehicle waiting times, alleviating traffic congestion, lowering carbon emissions, and minimizing environmental pollution. The economic benefits are that it can increase the traffic flow at intersections, save fuel consumption, and reduce time costs. According to the relevant research of this experiment, the average queue length has decreased by 23.85% [7]. Therefore, this experiment established a single intersection traffic flow simulation environment, trained an intelligent traffic light control agent based on the DQN algorithm [8], compared the control effects of the reinforcement learning strategy and the fixed duration strategy, and analyzed the influence of key parameters on the control performance [9]. This thesis focuses on crossroads, but there is still a lot of research space for multi-intersection studies [10].

This research has great value. Its social value lies in reducing vehicle waiting time, alleviating traffic congestion, and reducing carbon emissions. The economic value of this research is increasing intersection throughput, saving fuel consumption, and reducing time costs.

Build a simulation environment for single intersection traffic flow. Train an intelligent traffic light control agent based on the DQN algorithm. Compare the control effects of reinforcement learning strategies and fixed duration strategies. Analyze the influence of key parameters on control performance.

2. Research methods

Model the traffic light control as a Markov Decision Process (MDP), where the agent (the controller of the traffic lights at the intersection) can be regarded as an intelligent entity. The environment is the traffic flow at the intersection. The state is the queue length of vehicles in each direction lane. The action is to select the next green light phase. The reward is the negative total waiting time of vehicles (to minimize it).

Traditional Q-learning stores Q-values in a table, but when the state space is continuous or has a high dimension, the table-based method is no longer applicable. Deep Q-Network (DQN) uses a neural network to approximate the Q-function, solving this problem. The two core technologies of DQN are experience replay and the target network. Experience replay refers to storing interaction data in an experience replay pool and randomly sampling for training, breaking the data correlation. The target network refers to using a separate network to calculate the target Q-value and synchronizing it regularly to improve training stability.

A typical intersection has four phases, namely east-west straight, east-west left turn, north-south straight, and north-south left turn [11]. The control objective is to dynamically select the green light phase to maximize the traffic efficiency at the intersection. The evaluation indicators mainly include the average queue length, average waiting time, and throughput. The average queue length refers to the average number of vehicles queuing in each direction. The average waiting time refers to the average waiting time of vehicles at the intersection. Throughput refers to the number of vehicles passing through the intersection per unit time. System Modeling and Experimental Design.

DOI: 10.54254/2753-8818/2026.33417

3. Experiment

3.1. Environment modeling

The main assumption in building a single intersection model is that the intersection is a cross shape, with 3 lanes in each of the 4 directions. The arrival of vehicles follows a Poisson distribution, with an average arrival rate of $\lambda = 2$ vehicles per step. The green light release rate is fixed, with a maximum of 3 vehicles released per step. The minimum green light duration is 10 steps to prevent frequent phase switching. The state space is defined as a 5-dimensional vector, namely [east queue number, south queue number, west queue number, north queue number, current phase], and each dimension is normalized to the range [0, 1]. The action space contains 2 discrete actions, namely action 0 and action 1. Action 0 indicates maintaining the current phase. Action 1 indicates switching to the next phase. The reward function is defined as the negative total queue length: reward = -sum(queues), that is, the less the queue, the greater the reward.

3.2. Design of DQN model

The neural network structure consists of a 3-layer fully connected network, namely the output layer, the hidden layer 1, the hidden layer 2, and the output layer. The input layer refers to 5 neurons (representing the state dimension), the hidden layer 1 consists of 24 neurons with ReLU activation, the hidden layer 2 also has 24 neurons with ReLU activation, and the output layer comprises 2 neurons (representing the action dimension) with linear activation. The parameter configuration of this study is shown in Table 1.

Table 1. Rules to format sections

Level	parameter	Vaule
1	Learning rate	0.001
2	Discount rate	0.95
3	Exploration rate	1 to 0.01
4	Experience pool capacity	2000
5	Batch size	32

3.3. Contrast test design

Set a fixed duration strategy as the benchmark for comparison, The rule is to switch the phase every 20 steps. (10 steps of maintenance, 10 steps of switching, alternatingly carried out). The DQN strategy and the fixed-duration strategy each ran for 200 episodes, Each episode lasts for 500 steps. The evaluation metric is the average queue length.Keywords (use style: Keywords)

The keywords part starts with "Keywords". Each paper must have three to five keywords. Use commas to separate the keywords. All keywords should be in lower case except specific terms.

DOI: 10.54254/2753-8818/2026.33417

4. Experimental results and analysis

4.1. Analysis of the training process

Table 2. Training process

Before 50 episodes	50-150 episodes	150 episodes
The rewards are rising rapidly.	Entering the stage of stable optimization	The exploration rate gradually declined.

Table 2 shows the analysis of the training process. The DQN agent was trained for 200 episodes. The rewards increased rapidly in the first 50 episodes, indicating that the agent began to learn how to reduce queuing. From 50 to 150 episodes, it entered a stable optimization stage, during which the agent gradually became stable and continuously optimized. After 150 episodes, it tended to converge. The exploration rate gradually decreased from 1.0 to 0.01, achieving a smooth transition from exploration to exploitation.

4.2. The results of the comparative experiment

The average queue length of the fixed-duration strategy was 7.24, while that of the DQN strategy was 5.17, representing a reduction of 28.6%. This result indicates that the DQN agent is capable of adjusting the phase in accordance with the real-time traffic dynamics, which is significantly superior to the "one-size-fits-all" strategy with a fixed duration. From the perspective of the distribution of queue lengths, the queueing under the fixed-duration strategy fluctuates significantly, with the queue length reaching over 15 during peak hours. The queuing of the DQN strategy is relatively stable with a small fluctuation range, demonstrating the robustness of intelligent control. The parameter analysis of this experiment is shown in Table 3.

Table 3. Rules to format sections

Learning rate	0.0001	0.001	0.01
Final queue length	6.8	5.2	6.1

Take the learning rate as an example. Three values [0.0001, 0.001, 0.01] were tested, which respectively represent a learning rate of 0.0001, a learning rate of 0.001, and a learning rate of 0.01. Among them, a learning rate of 0.0001 indicates a slow convergence, and the final queue length is approximately 6.8. The learning rate of 0.001 indicates the fastest convergence speed and the best final performance, with a queue length of 5.2. A learning rate of 0.01 indicates that the training is unstable and there is significant oscillation. The final queue length of 6.1 suggests that the learning rate is one of the most sensitive parameters, and a balance needs to be struck between convergence speed and stability.

4.3. Feasibility analysis

The following is an introduction to the algorithm maturity. Since the DQN algorithm was proposed in 2013, it has been widely applied to various decision-making problems. Its principle is clear and its implementation is stable. The development framework, TensorFlow/Keras, provides a complete set of deep learning APIs, enabling the rapid construction of neural networks [12]. The simulation environment is a self-built simplified environment without complex dependencies. The code length

DOI: 10.54254/2753-8818/2026.33417

is approximately 200 lines, making it easy to debug and modify. The data is generated interactively by the simulation environment itself and does not require any external datasets. The dataset consists of 200 episodes × 500 steps = 100,000+ samples, which is sufficient for training a small neural network. The diversity of the data is ensured by the random arrival of vehicles, thereby guaranteeing the diversity and coverage of the data. The potential risks and countermeasures of this experiment are presented in Table 4.

Table 4. The potential risks and countermeasures

potential risks	countermeasures
Unstable training	Run multiple times and take the average, fix the random seed
The environment is overly simplistic.	Subsequently, it can be connected to the SUMO professional software for verification
Difficulty in parameter tuning	First, conduct a single-factor analysis to identify the key parameters.

5. Conclusion

This report systematically demonstrates the feasibility of the study on intelligent traffic light control based on reinforcement learning. From a technical perspective, the DQN algorithm is mature and stable, and the simulation environment can be built and implemented independently. From a data perspective, the interaction data can be generated automatically and does not require external dependencies. From the perspective of its own capabilities, the existing knowledge reserves and time arrangements can support the completion of the project. The preliminary experimental results show that the DQN agent can reduce the average queue length by 28.6%, significantly outperforming the fixed-duration strategy. This research includes executable code, complete DQN intelligent traffic light control code, including environment, training, and evaluation modules. Experimental data, training curves, comparison results, and parameter sensitivity analysis charts. Optimization guide, summary of the impact of key parameters on performance and setting suggestions. Currently, most traffic lights are of periodic variation. Intelligent traffic lights have not been widely adopted. Whether it is for reducing traffic congestion time, reducing fuel emissions for environmental protection, or for reducing time costs, intelligent traffic lights can demonstrate that they are very promising.

References

[1] Tsuboi, T. (2018). Traffic service quantitative analysis method under developing country. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 443–448). IEEE.
[2] Texas A& M Transportation Institute. (2025). 2025 urban mobility report. College Station, TX. https: //mobility.tamu.edu/umr/
[3] Liang, S., et al. (2023). Signalized intersection dynamic straight-right lane design and evaluation. Physica A: Statistical Mechanics and Its Applications, 620, 128771.
[4] Yu, C., Ma, W., & Yang, X. (2020). A time-slot based signal scheme model for fixed-time control at isolated intersections. Transportation Research Part B: Methodological, 140, 176–192.
[5] Saadi, A., Abghour, N., Chiba, Z., et al. (2025). A survey of reinforcement and deep reinforcement learning for coordination in intelligent traffic light control. Journal of Big Data, 12(1), 84.
[6] Xiao, F., Lu, J., Li, L., et al. (2025). Advances in reinforcement learning for traffic signal control: A review of recent progress. Intelligent Transportation Infrastructure, 4, liaf009.
[7] Wei, M., & Huang, J. (2025). A traffic signal reinforcement learning optimization method based on dual-policy network. Computer Applications, 45(3), 712–720.

DOI: 10.54254/2753-8818/2026.33417

[8] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533.
[9] Varela, G. S., Santos, P. P., Sardinha, A., et al. (2021). A methodology for the development of RL-based adaptive traffic signal controllers. arXiv preprint arXiv: 2101.09614.
[10] Agrahari, A., Dhabu, M. M., Deshpande, P. S., et al. (2024). Artificial intelligence-based adaptive traffic signal control system: A comprehensive review. Electronics, 13(19), 3875.
[11] Chen, X., & Xu, J. (2021). Design and implementation of urban traffic light control model based on deep reinforcement learning (Master's thesis, Soochow University).
[12] Dumić, E. (2024). Learning neural network design with TensorFlow and Keras. In ICERI2024 Proceedings (pp. 10689–10696). IATED.