A Comparison of the adaptation to the sudden change of the traffic conditions of Deep Q network algorithms

Jiayu Ju

China University of Mining & Technology

Abstract: This study investigates the adaptability of three deep reinforcement learning algorithms—Deep Q-Network (DQN), Double Dueling Deep Q-Network (D3QN), and Prioritized Experience Replay D3QN (PER-D3QN)—under sudden traffic condition changes, such as traffic flow surges, lane blockages, and their combination. A single-intersection model was developed in the SUMO simulation environment, and the algorithms were compared using average queue length, average delay, and cumulative rewards as evaluation metrics. The results demonstrate that DQN struggles to adapt to sudden changes, showing instability and oscillations in training. In contrast, D3QN significantly improves stability and overall performance by mitigating Q-value overestimation, particularly in high-flow scenarios. PER-D3QN exhibits the fastest convergence and strong short-term adaptability, though its performance fluctuates more than D3QN in extended training. Overall, both D3QN and PER-D3QN outperform the baseline DQN, effectively reducing delay and congestion in dynamic traffic environments. D3QN proves to be more stable in the long term, while PER-D3QN provides rapid adaptation in the early stages after sudden disturbances.

1. Introduction

Nowadays the need of the transportation has been much denser as the development of the urban scale. The government needs to deal with this denser and denser issue effectively to reduce the emission of the green house gases like carbon dioxide, and to optimize the traffic throughout and efficiency (Ma et al., 2025). Additionally, with the huge amount of traffic flow and more complicated traffic conditions, the risk of traffic accidents has been increased dramatically (Gokasar et al., 2022). Therefore, to mitigate these issues due to the complex traffic conditions, modifying the traffic signal control system is the most straightforward and feasible approach.

Recently, to better determine the optimized signal control strategies, researchers try to design the algorithms combined the reinforcement learning to optimize the traffic signal control system (Liang et al., 2019). In these methods, artificial intelligence agents are developed (Wang., 2021). Compared to the conventional methods, reinforcement learning (RL) could make the agent interact with the environment, which allows the agent to learn the optimal strategy and make the performance better and achieve the long-term goals. This approach has already got some promotion in some research, especially for the single-intersection scenario. However, this approach has some issues, like traffic explosion can not be resolved by using the conventional Q-learning. With the development of the research of artificial intelligence and machine learning, the deep reinforcement learning (DRL) has been introduced (Wang et al., 2022). Researchers in different areas started to combined DRL with researches in different real-world application, including the traffic signal control, which literally increase the traffic throughout and decrease the traffic delay effectively.

In the researches of the signal control technology by using DRL, researchers use different states to represent different traffic conditions, and get the optimal action of the traffic signal phases from the interaction of the agent and the environment (Quadri et al., 2020). However, there are still some gaps in these researches. In most of researches, they concentrate on the single-intersection scenario and normal road conditions, whereas few researches focus on the adaptation of the signal control technology using DRL to the sudden change of the traffic environment. This research focuses on the comparison of the robustness of deep Q network algorithm (DQN) and its optimized algorithms in alternative traffic environment, which means the traffic environment in this research will be more realistic since it simulates the dynamic change of the traffic conditions in the real world. The different conditions considered in this article include the lane blockage due to the construction and the dramatic increase of the traffic flow.

To compare the difference of the adaptation of the DQN and its optimized algorithms, the basic framework of this research is constructed according to the existed three popular algorithms. The contributions are as follow:

The main objects chosen in the comparison are conventional DQN algorithm, D3QN algorithm combined dueling network and deep double Q network, and D3QN algorithm with prioritized experience replay mechanism. These three algorithms differentiate the adaptation to the different traffic environment.

In the comparison, based on the average queue length, delay, and cumulative awards, the traffic performance and efficiency are compared. The optimal action of phase and its duration are updated mainly based on the time interval of the duration, also based on the throughout and congestion partially.

The literature review related to this research is described in section 2, and the methodology used in the research is introduced in Section 3. The specific experiment setting and results are demonstrated in section 4. Finally, section 5 concludes this research.

2. Literature Review

More and more academics nowadays concentrates on the traffic signal control by using DRL as a result of the traffic demands growth. These researches make certain progress in the signal control performance and efficiency. There are still some issues that these researches neglect, including the influence of sudden change traffic conditions.

Wan and Hwang (2018) proposed the DQN algorithm with optimized discount factor, improving the devote of the action in the model, optimize the model by comparing the reduction of the average delay, however, the stability of the model depends on the environment, which means the model performance will be influenced drastically by the environment conditions. Gao et al. (2017) modify the network architecture from Fully Connected Neural Network to the Convolutional Neural Network (CNN), optimize the average waiting time compared to the conventional DQN algorithm. Faqir et al. (2022) combine eXtreme Gradient Boosting (XGBoost) algorithm and DQN with CNN to build up the ConvXGB grids to optimize the parameters, improving the performance and accuracy, comparing to the conventional DQN, it reduces the cumulative waiting time of vehicles. Li et al (2016) introduce a special architecture called stacked autoencoder (SAE), which is used to estimate the Q value in the algorithm, comparing to the conventional one, improve the traffic throughout especially the throughout at the peak.

Due to the obvious drawbacks of DQN algorithm, Wang (2021) concludes the Double Deep Q network (DDQN) algorithm which uses two set of Q networks, avoiding the influence of the overestimation. Additionally, combining the advantages of DQN and Dueling DQN, Zheng et al.

propose the Pri-DDQN algorithm, the power function is used to dynamically change the exploration rate of the agent. And the experience samples are sorted, and the samples with high learning value are prioritized. Therefore, it has the best convergence speed and experimental effect, effectively alleviating traffic congestion during peak hours, and has stronger adaptability to real-time changing traffic flows. Liang et al. (2019) introduce the Dueling network to make the training process stable and use the double Q network to reduce the overestimation, improving the overall performance. Wang et al. (2022) in their research propose a specific algorithm called EP-D3QN based on the MP, SOTL, and D3QN algorithms. They test their model in different traffic conditions, including low traffic flow and high traffic flow, improving the traffic throughout and mitigating the traffic pressure. Zai and Yang (2023) propose a traffic signal control method based on the efficient channel attention mechanism (ECA-NET), long short-term memory network (LSTM) and Dueling Q network (D3QN) EL_D3QN.

Considering the partial use of the DQN and D3QN algorithm, although which create the agent to interact the environment and make the traffic performance better, the multi-intersection scenario cannot be considered reasonably. Chu et al. (2020) and Zhang et al. (2024) try creating multiple agents to improve the traffic performance in multi-intersection scenario, which is kind of systematic method to control and adjust the phase and phase duration in perfect union.

The existed literatures pay few attentions on variety of traffic scenarios. This research tries focusing on the adaptation of different algorithms to the different traffic conditions, including different traffic flows and different road conditions.

3. Methodology

3.1 Model Statement

Due to the increasingly growing traffic requirements and demands for more modifications of roads, modern signal control technology needs to deal with more complicated and variable traffic scenarios. To show the adaptation of the current popular traffic signal control technologies, which utilize deep reinforcement learning as a tool, for the sudden change in the traffic situation, a comprehensive comparison among the Deep Q Network algorithm (DQN), Double Deling Deep Q network (D3QN) and D3QN with Prioritized Experience Replay (PER-D3QN) in the sudden changing scenario, including flow increase and lane blockage, is done in this research. The basic traffic signal control model used in this research is illustrated in Fig.1 below. All models are composed of two parts: a traffic flow environment simulated in SUMO, and an agent written in Python. When choosing an action, the agent receives states from the SUMO environment as the input values. The traffic signal in the environment will be altered according to the action chosen by the agent. Then, the reward value is calculated after action choosing is completed, and learning is executed in this loop.

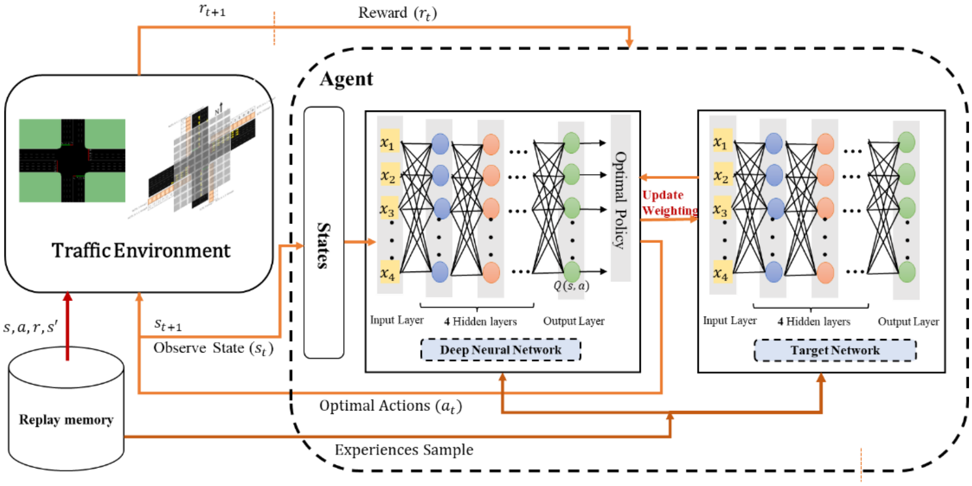


Fig.1 DQN Signal Control Model

As for the extra special structure in the model, the Target Q network is a copy of the online neural Network, but updated less frequently, which provides a relatively stable Q value when computing the Bellman equation. Instead of just training on the recent transition, the model provides a replay buffer used to restore past experiences (s, a, r, s'), which can also stabilize the training process and improve the sample efficiency (Rasheed et al., 2020).

The intersection model simulated in this research is a single intersection model with four entry lanes running North-South and four entry lanes running East-West for left turns, straight, and right turns. In the process of action making, the state will be input into the network, and the action will be chosen. When the action is chosen, it will make the next phase and its duration change, which means the action will influence the next phase and how long the phase will last.

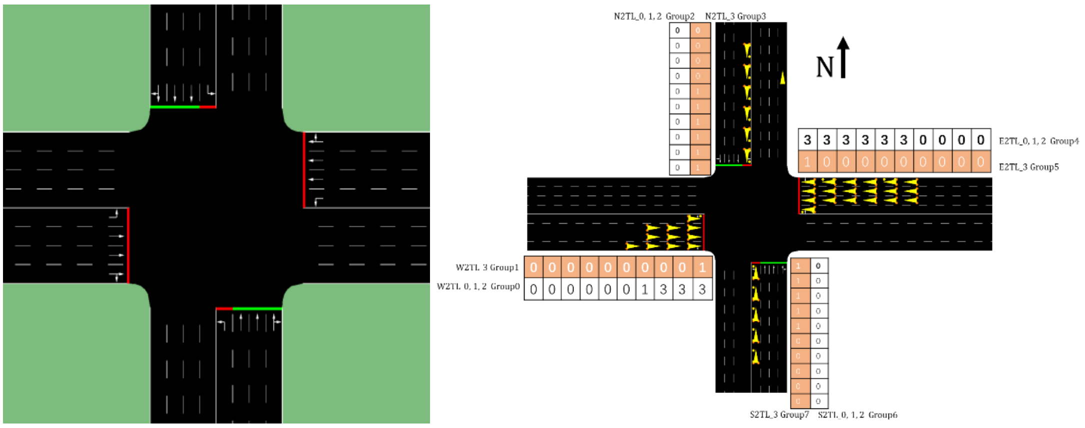


Fig.2 The intersection model and state layout

3.1.1 Intersection state

Following the DTSE representations based on the grid, the entry lanes of the intersection are discretized to capture the spatial distribution of the cars around the stop line. In the setting, four directions (W, N, E, S) are split into straight/ right-turn (lane_0, _1, _2) and left-turn (lane_3), 8 groups of lanes in total. For every group of lanes, the lanes upstream of the stop line are separated into 10 non-uniform units according to the position of the cars. At every simulation step, the car transferred into the edges (E2TL, N2TL, W2TL, S2TL) will be allocated accurately to a unit, thereby producing an 80-dimensional feature vector (8 groups * 10 units). Every unit records the number of cars, then is standardized by z-score normalization. The cars that are not transferred to the edges will be ignored. This kind of DTSE highlights that the queue accumulation and spatial concentration near the stop line, providing the RL strategy with more stable training. Fig.2 shows the intersection state layout.

3.1.2 Action Space

The agent can choose one of 4 discrete actions at each decision step; thus, the action space for the model is $A=\{0,1,2,3\}$. Each action corresponds to one green service phase of the traffic signal. The correspondence relationship is shown in Table 1. Yellow phases in this model are not directly chosen by the agent. They occur automatically as clearance phases when switching actions, and the yellow phases have a kind of relationship with the old action a, as formula (1).

$$YELLOW = 2a + 1$$

Table 1. The green phases corresponded to the actions

Action	Phase index (SUMO)	Movement served	Description
0	PHASE_NS_GREEN = 0	North–South through/right	Both N and S approaches allow straight + right-turn traffic
1	PHASE_NSL_GREEN = 2	North–South left-turn	Left-turn lanes (N2TL_3, S2TL_3) get green
2	PHASE_EW_GREEN = 4	East–West through/right	Both E and W approaches allow straight + right-turn traffic
3	PHASE_EWL_GREEN = 6	East–West left-turn	Left-turn lanes (E2TL_3, W2TL_3) get green

When the traffic flow increases, the continuous time of the green phase will grow. To avoid this situation, the maximum green time is set, the maximum green time is 50s. When the green phase duration is longer than 50s, the maximum green time will be chosen directly.

3.1.3 Rewards

In this study, a comprehensive rewards function is developed, guiding the agent to optimize the signal duration distribution. All of the agents of different algorithms are using a basic reward function that is related to the cumulative waiting time. As formula (2)-(3), the difference in value of cumulative waiting time between the adjacent moments is the basic indicator. When the total waiting time decreases, the function will grant a positive reward; otherwise, it will impart negative punishment, and this value can be modified by the zoom factor to reduce the congestion.

$$Waittime_t = \sum_{i \in \mathrm{IN}} waittime_i$$

$$R_{Waittime_t} = Waittime_{t-1} - Waittime_t$$

Where IN is the number of import lanes, *waittime$_i$* is the waiting time of the vehicle for import lane i, and *Waittime$_t$* is the total waiting time of all vehicles for import lanes.

Besides the basic reward function, extra reward mechanisms are also introduced in the D3QN and PER-D3QN agents, which are congestion punishment and throughout reward. For the congestion punishment, if the count of cars in a certain entry lane exceeds the threshold value of congestion, the function will impose extra punishment, and the value is correlated to the extent of congestion, which can avoid the cars accumulating in the single approach. As for the throughout reward, the reward can be improved when more cars pass through the stop line, which can provide an encouraging signal to take better action to improve the traffic efficiency.

3.1.4 Greedy Strategy

The method used in the stage of action choosing is the ε-greedy strategy. Specifically, the agent chooses an alternative action randomly with a probability of ε in every decision step to explore, outputting the most optimized action for the current Q network with probability 1- ε. The ε here is adjusted in an exponential decay mechanism dynamically; the formula of this mechanism (4) is:

$$\varepsilon_t = \varepsilon_{end} + \left(\varepsilon_{start} - \varepsilon_{end} \right) \cdot e^{-\frac{t}{\varepsilon_{decay}}}$$

With the increase in the training steps, the probability of exploration gradually decreases, and the probability of utilization gradually increases, which not only satisfies the sufficient exploration at the early stage, but also achieves the stable utilization. In addition, to make sure the physical constraint of the traffic signal, the action mask is also added into the process of action choosing, shielding the illegal actions or actions exceeding the maximum green duration.

3.2 Agent Architecture

This research concentrates on the comparison of the adaptation to the suddenness of DQN, D3QN, and the PER-D3QN algorithm. The distinction in the function and adaptation is related to the network architecture. Thus, this part demonstrates the difference in the network architecture of these three algorithms to help people better understand the further comparison of adaptation.

The traditional DQN agent developed in this study is built with 4 hidden layers of a fully connected neural network, shown in Fig.1. The action chosen strategy has been mentioned in Section 2.1. The agent is composed of three main parts: the main network, the target network, and the replay buffer. The function of every part has been introduced. More importantly, the traditional DQN architecture has obvious drawbacks:

1st, the Q value is easy to overestimate due to the calculation of the target in the traditional DQN algorithm, as formula (5):

$$y = r + \gamma \max_a Q(s', a; \theta)$$

The max is used here to solve the action choice, and estimating the action value will create a relatively large deviation.

2nd, the output of traditional DQN is the Q value of different kinds of actions. When there is no obvious difference among different actions, the network cannot identify whether the state is good or bad.

To solve these issues, researchers introduce the D3QN architecture, which uses Double DQN to solve the Q value overestimation. While the traditional DQN algorithm uses the same max operator and values for both selection and evaluation of an action, the DDQN algorithm uses the target as follows (Rasheed et al., 2020):

$$y_j^{DDQN} = r_{j+1}(s_{j+1}) \\ +\gamma Q(s_{j+1}, \arg\max_a Q(s_{j+1}, a; \theta_j); \theta_j^-)$$

Besides this, the D3QN algorithm also introduces a Dueling network, which makes the stream split into 2 FC layers, the state value $V(s)$: Dense (128)→ Dense (1), and the advantage value $A(s)$: Dense (128)→Dense (4), the value Q is sum of the state value V and the function of advantage value A, as the formula (7) which allows the network to learn more efficiently and stably, especially when the difference between actions is not obvious.

$$Q(s,a;\theta) = V(s;\theta) + \left(A(s,a;\theta) - \frac{1}{|A|}\sum_{a'} A(s,a';\theta)\right)$$

$A(s,a;\theta)$ indicates how significant a certain action is to the value function across all actions. If the A-value of an action is positive, it means that it performs better than the average of all possible actions in terms of rewards. If it is negative, it implies that the potential reward of a certain action is lower than the average (Rasheed et al., 2020). θ is a parameter in the main network, and θ^-in the target network, is updated according to the formula (8).

$$\theta^- = \alpha\theta^- + (1-\alpha)\theta$$

α in the formula, is the update rate, which shows the extent of the influence on the components of the target network from the most recent parameters.

The target Q of the D3QN network is calculated as the formula (9), where r is the reward value, and γ is the discount factor.

$$Q_{target}(s,a) = r + \gamma Q\left(s', argmax(Q(s',a';\theta));\theta^-\right)$$

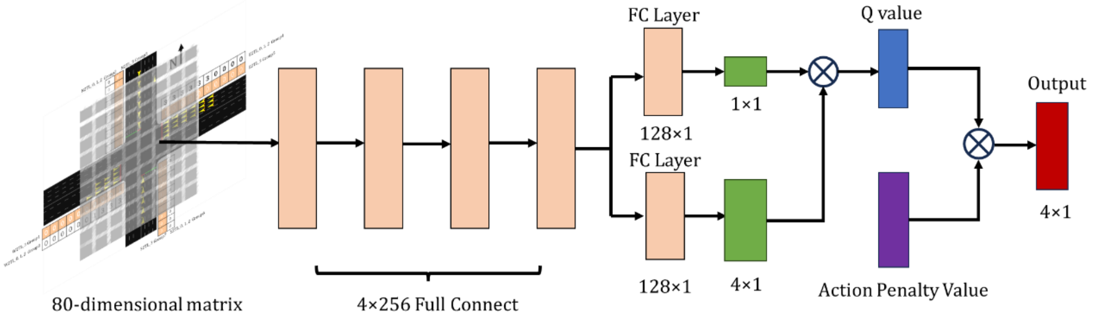


Fig.3 D3QN Network Architecture

In this research, another algorithm in this comparison is PER-D3QN, which has the Prioritized Experience Replay (PER) mechanism, a modification of the normal replay buffer. Compared to the normal experience replay, which samples randomly from memory, PER makes the sampling process more efficient by selecting more valuable experiences from memory, depending on the Temporal difference error (TD error). The calculation of the error is shown as formula (10). When the TD error is bigger, it will have a higher probability of being sampled. The probability is calculated as formula (11):

$$\delta = y - Q(s,a)$$

$$P(i) = \frac{p_i^\alpha}{\sum_j p_j^\alpha}$$

In the formula (8), p is provided according to the value of $|\delta|$.

With the mechanism of PER, the agent can sample efficiently and has a higher rate of convergence to improve the overall performance of the adaptation to more complicated traffic conditions.

4. Experiment Design and Analysis

4.1 Experiment Setup

Three main comparative objects are defined in this study to compare their adaptation to the sudden change of the traffic environment, including the flow increase and the lane blockage. Thus, to these three objects, several experiments were designed in this research.

4.1.1 Traffic Flow Generation

In this research, there will be N cars generated in each training episode, which is defined in the training setting files, and the maximum simulation steps are also defined in the training setting. The original samples of the arrival time t_i obey a Weibull distribution (shape parameter = 2), then the arrival order will be sorted in ascending order. After this, get t_{min} and t_{max}, map [t_{min}, t_{max}] to [0, max steps]. Finally, the discrete leave step is defined as formula (12):

$$\widehat{t}_i = round\left(\frac{max\ steps}{t_{max} - t_{min}}\left(t_{(i)} - t_{max}\right) + max\ steps\right),\ \ \widehat{t}_i \in \{0, \ldots, max\ steps\}$$

This step will make sure the shape feature of the Weibull, the arrival intensity that first increases and then descends, also distributes the time scale into the simulation range.

The flow generator defines 12 routes, which are 4 straight routes and 8 turning routes. For every vehicle, it will be sampled by a fixed turning ratio, 75% straight or 25% turning. When the vehicle is set to the straight routes, it will be distributed onto the {W_E, E_W, N_S, S_N} edges randomly. The leave step is $\widehat{t}_i$, the lane departed is random, and the initial speed is set to 10 m/s. The probability of these options on which entry to choose is equal. In addition, the vehicles in the simulation are set up with uniform parameters: the maximum acceleration is 1 m/s^2, the minimum deceleration is 4.5 m/s^2, the vehicles have a length of 5m, and the minimum following distance between two vehicles is 2.5 m.

4.1.2 Parameter Setting

Tables 2 and 3 provide lists of the parameters needed for DQN, D3QN, and PER-D3QN algorithms. Every simulation lasts for 100 episodes, with each episode running the traffic flow defined in the setting.

Table 4 provides the specific parameters needed for the PER mechanism.

Table 2. Parameter setting of the DQN algorithm

Parameter	Value	Parameter	Value
Discount factor	0.75	minimum ε	0.1
Learning rate lr	0.001	maximum ε	1
Target network update freq	1000	episode decay	83000
Batch size	100	Experience pool capacity	50000

Number of episodes	100	Activation	ReLu

Table 3. Parameter setting of the D3QN algorithm

Parameter	Value	Parameter	Value
Discount factor	0.95	minimum ε	0.05
Learning rate lr	0.0003	maximum ε	1
Target network update freq	1000	decay rate	0.97
Batch size	96	Experience pool capacity	50000
Number of episodes	100	Activation	ReLu
Congestion threshold	25	congestion penalty	-10

Table 4. Parameter setting for the PER mechanism

Parameter	Value	Parameter	Value
Priority Clip Max	10	uniform mix ratio	0.2
Importance Sampling Exponent	0.3	Priority Exponent	0.4
Beta Annealing Rate	0.004	Minimum Priority	1.00E-06

4.1.3 Comparative Methods

To compare the adaptation to the sudden changes of the different algorithms, three independent experiments are designed in this research. Through using the interface of Python and the SUMO simulator, four main scenarios are simulated:

Scenario 1: Normal road conditions; Normal traffic flow (1000 cars/5400steps)
Scenario 2: 1 Lane Blockage; Normal traffic flow (1000cars/5400steps)
Scenario 3: Normal road conditions; Peak traffic flow (2000cars/5400steps)
Scenario 4: 1 Lane Blockage; Peak traffic flow (2000cars/5400steps)

The distinct parameters in different scenarios are designed to map the more complicated situations, which might be changed from the normal traffic conditions, where 1000 cars/5400 steps and 2000 cars/5400 steps are used to simulate the normal density of flow and high density of flow; as for the lane blockage, the new road net file are created in SUMO netedit, where one lane in one edge is deleted to simulate one lane blockage. Due to the same probability of the car leaving from different edges, the indicator of the lane chosen could be neglected.

In every scenario, models will be trained for 100 episodes. Scenarios 2, 3 are designed to examine the adaptation of the algorithms to the change of increased flow and lane blockage, and scenario 4 is for the adaptation to the combined interference. In the phase of training, the models trained in scenarios 2, 3, and 4 are trained by transferring from the model trained in scenario 1. By comparing the distinction of the training process and testing these models in a simulated environment that simulates the suddenly changed traffic conditions, the results can indicate the adaptation of the different algorithms to the different sudden changes.

The overall experiment plan in this research is shown in Fig.4 below.

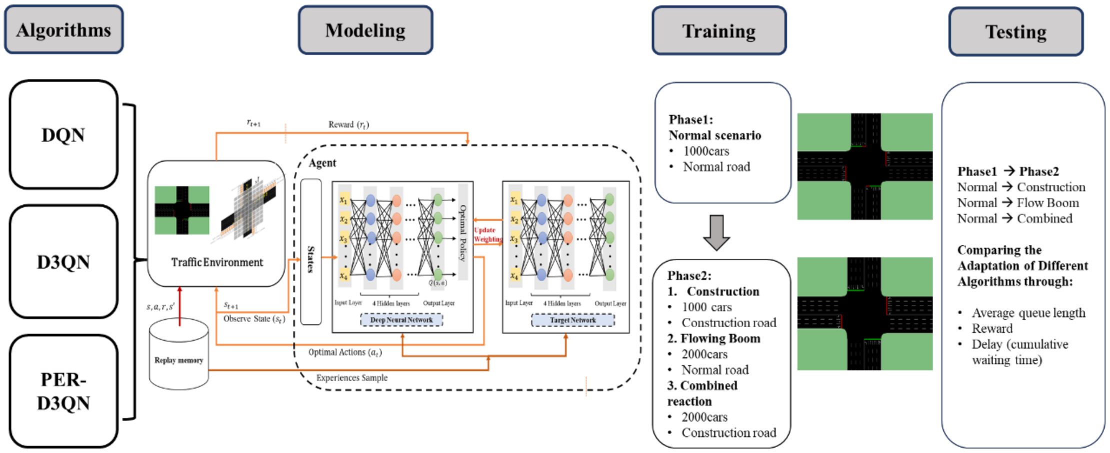


Fig.4 Experiment Framework

4.2 Results and Analysis

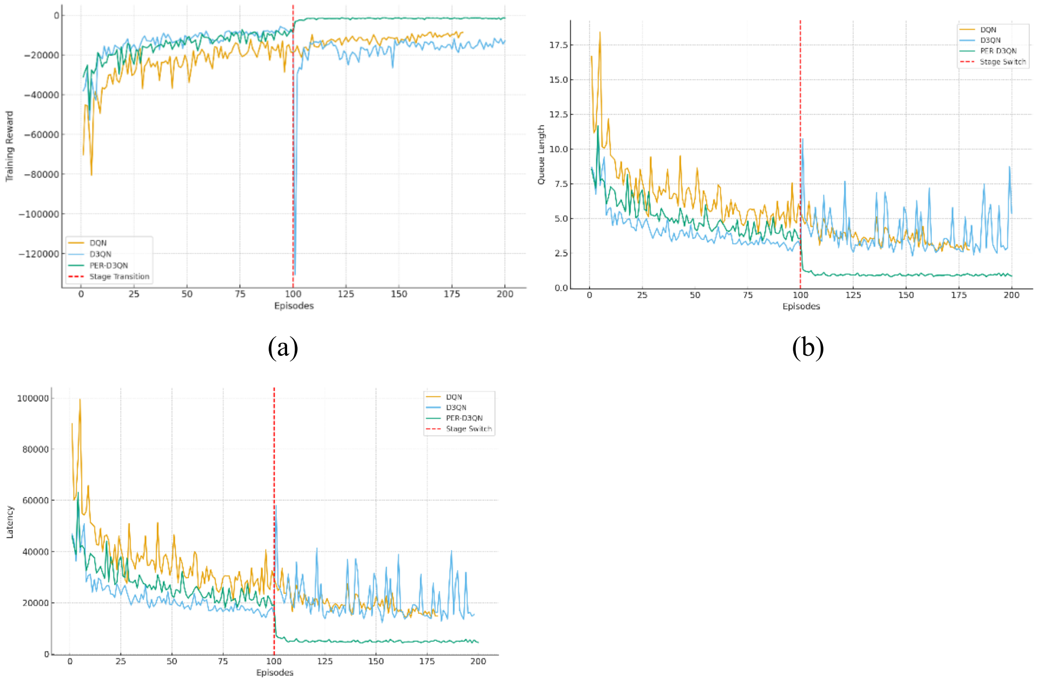
This research includes three main experiments: the comparison of three algorithms in the lane blockage condition, the comparison in the flow increasing condition, and the combination of the lane blockage and increasing flow. In this section, the results will be shown specifically, including the training process and the test results, and the result of the testing part is averaged over 5 episodes. The seeds used in these episodes are random.

4.2.1 Lane Blockage

This part compares the adaptation to the sudden change of the road condition of three algorithms: DQN, D3QN, and PER-D3QN. The red dashed line in the middle of the diagram is the sign of the sudden change. The first 100 episodes belong to phase 1, which are trained in the normal road condition. The second 100 episodes belong to phase 2, which are trained in the environment where 1 western lane is closed. In these diagrams, the performance of different models' adaptation can be identified clearly. The lines in the graphs represent different algorithms; the yellow line is DQN, the green one is PER-D3QN, and the blue line is D3QN.

By analyzing Fig.5, the performance of phase 1 is relatively uniform, and the performance of these algorithms is relatively stable, with the bad starts then converging gradually. Specifically, by contrast, the D3QN and the PER-D3QN both have better performance than the DQN, and the D3QN seems to get the best grade in phase 1. Nonetheless, with the sudden change happening, 1 lane is closed, PER-D3QN gets the best performance in the whole training process, and the training line shown in diagrams is very stable. The interesting thing is that D3QN seems to show more vibration in the training process of phase 2, and compared to the performance of phase 1, it falls behind the DQN.

Table 5 shows the test results of different algorithms. Aspects of the performance of D3QN and PER-D3QN seem to have improved compared to the DQN algorithm: the average queue length of the D3QN gets about a 65% decrease, the PER-D3QN is better, reducing 74%; the rewards are also improved, both rewards of the D3QN and PER-D3QN are improved from -2.04 of the DQN to 0. For the delay, the D3QN and the PER-D3QN also get better performance than the DQN does, 82% decline for the D3QN and 89% decline for the PER-D3QN. These results show that the D3QN and the PER-D3QN, contrary to the DQN algorithm, have better generalization ability.



(a)

(b)

(c)

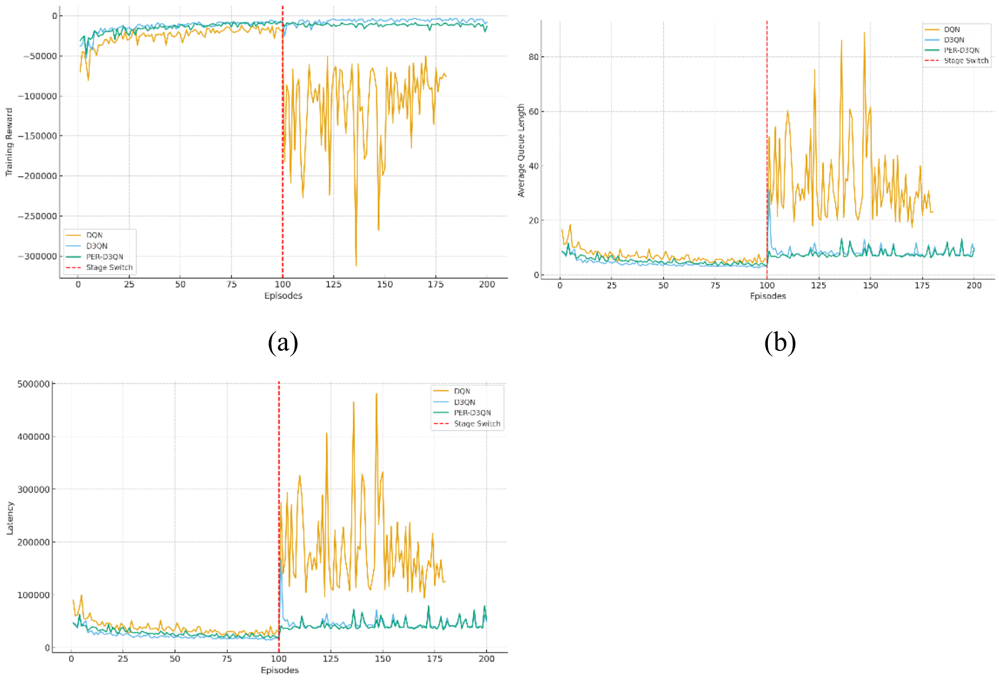
Fig.5 Comparison of Performance of Different Algorithms in Lane Blockage Conditions (train)
(a) cumulative reward, (b) average queue length, (c) delay

Table 5. Different Algorithms Performance Results (test)

Algorithms	**Normal→1 Lane Blockage**		
	Average queue length	rewards	Delay
DQN (BaseLine)	3.93	-2.04	130.29
D3QN	1.36 (↓65%)	0	23.18 (↓82%)
PER-D3QN	1.04 (↓74%)	0	14.43 (↓89%)

4.2.2 Increasing Flow

In Fig.6, the performance of phase 1 has been analyzed in Section 3.2.1. As for phase 2, with the sudden combined change of the lane blockage and the flow increasing, the DQN performance is influenced by the change drastically, producing huge vibrations, which means that the sudden change makes the DQN difficult to adapt and converge. Compared to the DQN, the D3QN and the PER-D3QN seem to get better performance. Specifically, the PER-D3QN seems to have better performance at the initial stage, but in the training process, the D3QN overtakes the PER-D3QN algorithm. Generally, the comparison of these training processes shows the good adaptation ability of the PER-D3QN algorithm in the early stage after the sudden change happens, but after training enough episodes, the performance of the D3QN gets better. This shows that, with the sudden increase of traffic flow, both D3QN and PER-D3QN can remain stable and adapt to the complicated environment after training, but the PER-D3QN converges more quickly, and the D3QN shows a better performance eventually.



(a)

(b)

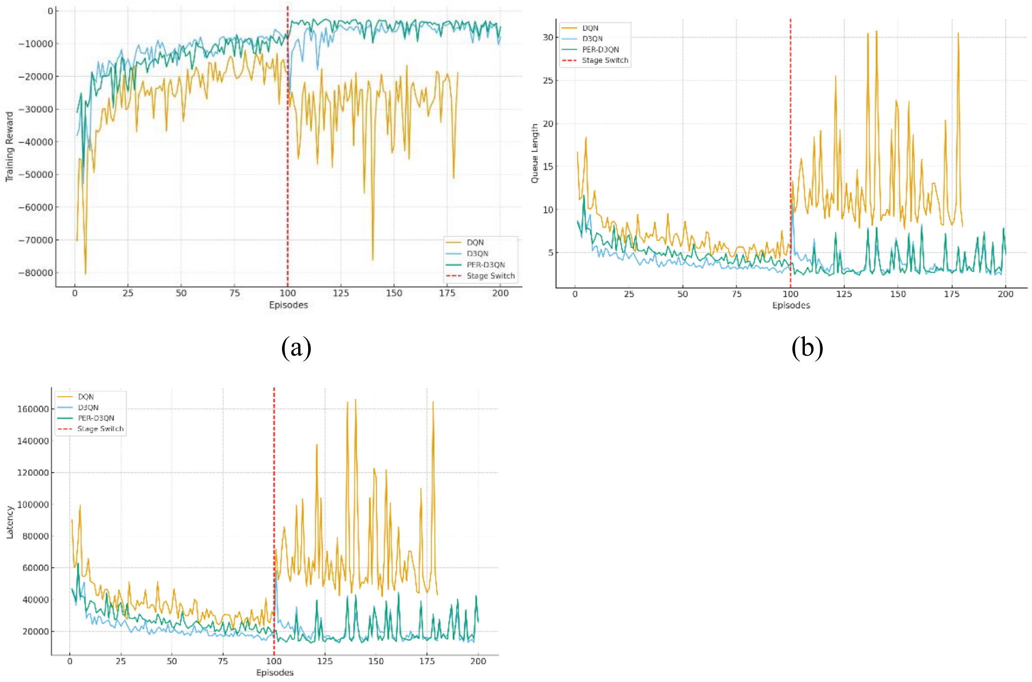
(c)

Fig.6 Comparison of Performance of Different Algorithms in Increasing Flow Conditions (train) (a) cumulative reward, (b) average queue length, (c) delay

4.2.3 Combined Interference

In Fig.7, the performance of phase 1 has been analyzed in Section 3.2.1. As for phase 2, with the sudden combined change of the lane blockage and the flow increasing, the DQN performance is influenced by the change drastically, producing huge vibrations, which means that the sudden change makes the DQN difficult to adapt and converge. Compared to the DQN, the D3QN and the PER-D3QN seem to get better performance. Specifically, the PER-D3QN seems to have better performance at the initial stage, but in the training process, the D3QN catches up with the PER-D3QN algorithm, and the overall performance line of PER-D3QN is more vibrated than D3QN's. Generally, the comparison of these training processes shows the good adaptation ability of the PER-D3QN algorithm. With the change of the environment, it can remain stable and adapt to the complicated environment at a relatively fast speed, but it is not as stable compared to the D3QN algorithm.

Table 7 shows the algorithm's performance results in the test. Similar to the training process, the D3QN and PER-D3QN improve the overall performance. For the average queue length, D3QN has reduced 78% compared to the DQN, and PER-D3QN has reduced 75%. As for the delay, both the D3QN and the PER-D3QN have good optimization, getting 92-93% reduction. These results show that both the D3QN and the PER-D3QN are fit for the high traffic flow situation with lane blockage conditions.



(a)

(b)

(c)

Fig.7 Comparison of Performance of Different Algorithms with Combined Interference (train) (a) cumulative reward, (b) average queue length, (c) delay

Table 7. Different Algorithms Performance Results (test)

Algorithms	**Normal→Combined Interference**		
	Average queue length	rewards	Delay
DQN (BaseLine)	8.84	0	312.5
D3QN	1.91 (↓ 78%)	0	22.37 (↓ 93%)
PER-D3QN	2.17 (↓ 75%)	0	25.45 (↓ 92%)

5. Conclusion

This study compares the adaptability of three deep reinforcement learning algorithms—DQN, D3QN, and PER-D3QN—under sudden traffic condition changes such as lane blockages, increased traffic flow, and combined disturbances. Using a SUMO-based single-intersection model, the algorithms were evaluated in terms of average queue length, delay, and cumulative rewards. Results indicate that the baseline DQN struggles with instability and poor convergence under dynamic conditions, while D3QN significantly improves stability and overall performance by addressing Q-value overestimation. PER-D3QN achieves the fastest convergence and strong early-stage adaptability, though with higher fluctuations during extended training. Both D3QN and PER-D3QN outperform DQN, effectively reducing delays and congestion in complex traffic scenarios; however, D3QN proves more stable in the long run, whereas PER-D3QN excels at rapid adaptation immediately after sudden changes. These findings highlight the advantages of advanced DQN variants for robust traffic signal control in dynamic urban environments.

References

Ma Changxi, Liu Yiyi, Zhao Hongxing, & Ma Cunrui. (2025). A review of the optimization of signal control at intersections considering low emissions. Journal of Lanzhou Jiaotong University, 44(3), 51–61.

Gokasar, I., Timurogullari, A., Deveci, M., & Garg, H. (2022). SWSCAV: Real-time traffic management using connected autonomous vehicles. *ISA Transactions*. https://doi.org/10.1016/j.isatra.2022.06.025

Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, *68*(2), 1243–1253. https://doi.org/10.1109/TVT.2018.2890726

Wang Tianxiang. (2021). Research on Collaborative Control of Large-scale Intersection Signal Lights Based on Multi-agent Deep Reinforcement Learning [Master's Thesis, Hefei University of Technology]. https://doi.org/10.27101/d.cnki.ghfgu.2021.002129

Wang, B., He, Z., Sheng, J., & Chen, Y. (2022). Deep reinforcement learning for traffic light timing optimization. *Processes*, *10*(11), 2458. https://doi.org/10.3390/pr10112458

Qadri, S. S. S. M., Gökçe, M. A., & Öner, E. (2020). State-of-art review of traffic signal control methods: Challenges and opportunities. *European Transport Research Review*, *12*(1), Article 1. https://doi.org/10.1186/s12544-020-00439-1

Wan, C., Hwang, M. (2018). Value-based deep reinforcement learning for adaptive isolated intersection signal control. *IET intelligent transport systems—Wiley online library*. (n.d.). Retrieved July 31, 2025, from https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2018.5170

Faqir, N., Loqman, C., & Boumhidi, J. (2022). Deep Q-learning approach based on CNN and XGBoost for traffic signal control. *International Journal of Advanced Computer Science and Applications*, *13*(9).

Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N. (2017). *Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network* (No. arXiv:1705.02755). arXiv. https://doi.org/10.48550/arXiv.1705.02755

Li, L., Lv, Y., & Wang, F. (2016) Traffic signal timing via deep reinforcement learning. IEEE/CAA Journal of Automatica Sinica, 3(3), 247–254. https://doi.org/10.1109/jas.2016.7508798

Wang, B., He, Z., Sheng, J., & Chen, Y. (2022). Deep reinforcement learning for traffic light timing optimization. Processes, 10(11), 2458. https://doi.org/10.3390/pr10112458

Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, *68*(2), 1243–1253. https://doi.org/10.1109/TVT.2018.2890726

Wang Tianxiang. (2021). Research on Collaborative Control of Large-scale Intersection Signal Lights Based on Multi-agent Deep Reinforcement Learning [Master's Thesis, Hefei University of Technology]. https://doi.org/10.27101/d.cnki.ghfgu.2021.002129

Zheng, Y., Luo, J., Gao, H., Zhou, Y., & Li, K. (2024). Pri-DDQN: learning adaptive traffic signal control strategy through a hybrid agent. *Complex & Intelligent Systems*, *11*(1). https://doi.org/10.1007/s40747-024-01651-5

Zai, W., & Yang, D. (2023). Improved deep reinforcement learning for intelligent traffic signal control using ECA_LSTM network. *Sustainability*, *15*(18), 13668. https://doi.org/10.3390/su151813668

Chu, T., Wang, J., Codecà, L., & Li, Z. (2020). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 1086–1095. https://doi.org/10.1109/TITS.2019.2901791

Zhang, G., Chang, F., Jin, J., Yang, F., & Huang, H. (2024). Multi-objective deep reinforcement learning approach for adaptive traffic signal control system with concurrent optimization of safety, efficiency, and decarbonization at intersections. *Accident Analysis and Prevention*, *199*, 107451. https://doi.org/10.1016/j.aap.2023.107451

AndreaVidali. (2019). *GitHub - AndreaVidali/Deep-QLearning-Agent-for-Traffic-Signal-Control: A framework where a deep Q-Learning Reinforcement Learning agent tries to choose the correct traffic light phase at an intersection to maximize traffic efficiency.* GitHub. https://github.com/AndreaVidali/Deep-QLearning-Agent-for-Traffic-Signal-Control

Rasheed, F., Yau, K.-L. A., Noor, R. Md., Wu, C., & Low, Y.-C. (2020). Deep Reinforcement Learning for Traffic Signal Control: A Review. IEEE Access, 8, 208016–208044. https://doi.org/10.1109/access.2020.3034141

Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; Wu, D.O. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. IEEE Trans. Veh. Technol. 2020, 69, 8243–8256.