

Article

Intelligent Traffic Control Decision-Making Based on Type-2 Fuzzy and Reinforcement Learning

Yunrui Bi, Qinglin Ding, Yijun Du *, Di Liu and Shuaihang Ren

School of Automation, Nanjing Institute of Technology, Nanjing 211167, China; biyunrui@njit.edu.cn (Y.B.); y00450220307@njit.edu.cn (Q.D.); liudinust@163.com (D.L.); z1013153165@163.com (S.R.)

* Correspondence: duyijun@njit.edu.cn

Abstract: Intelligent traffic control decision-making has long been a crucial issue for improving the efficiency and safety of the intelligent transportation system. The deficiencies of the Type-1 fuzzy traffic control system in dealing with uncertainty have led to a reduced ability to address traffic congestion. Therefore, this paper proposes a Type-2 fuzzy controller for a single intersection. Based on real-time traffic flow information, the green timing of each phase is dynamically determined to achieve the minimum average vehicle delay. Additionally, in traffic light control, various factors (such as vehicle delay and queue length) need to be balanced to define the appropriate reward. Improper reward design may fail to guide the Deep Q-Network algorithm to learn the optimal strategy. To address these issues, this paper proposes a deep reinforcement learning traffic control strategy combined with Type-2 fuzzy control. The output action of the Type-2 fuzzy control system replaces the action of selecting the maximum output Q-value of the target network in the DQN algorithm, reducing the error caused by the use of the max operation of the target network. This approach improves the online learning rate of the agent and increases the reward value of the signal control action. The simulation results using the Simulation of Urban MObility platform show that the traffic signal optimization control proposed in this paper has achieved significant improvement in traffic flow optimization and congestion alleviation, which can effectively improve the traffic efficiency in front of the signal light and improve the overall operation level of traffic flow.

Keywords: intelligent transportation; deep Q-network; Type-2 fuzzy control; traffic signal optimization control



Citation: Bi, Y.; Ding, Q.; Du, Y.; Liu, D.; Ren, S. Intelligent Traffic Control Decision-Making Based on Type-2 Fuzzy and Reinforcement Learning. *Electronics* **2024**, *13*, 3894. <https://doi.org/10.3390/electronics13193894>

Academic Editor: Mohamed Karray

Received: 11 August 2024

Revised: 27 September 2024

Accepted: 27 September 2024

Published: 1 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous growth of the global economy and population, cars have become indispensable modes of transportation in people's daily lives. Although many people can afford to buy cars, this also presents significant challenges to urban transportation systems. Studies have found that traffic crashes caused by congestion are on the rise annually, and traffic jams also lead to increased fuel consumption and vehicle exhaust emissions, resulting in serious environmental pollution [1,2]. This not only restricts urban development but also causes substantial economic losses for countries due to traffic congestion every year. Therefore, urban traffic congestion has become a severe problem faced by countries worldwide [3]. To alleviate urban traffic pressure, various countries have introduced intelligent transportation systems (ITSs) for traffic management in major cities. These systems utilize advanced technologies to regulate roads, vehicles, and pedestrians, effectively optimizing the utilization of traffic resources to mitigate congestion, reduce traffic crashes, and lower environmental pollution [4,5].

Intelligent transportation systems have long been a research focus in society, and academic researchers both domestically and internationally have been studying ITS optimization in recent decades [6–8]. With the advancement of artificial intelligence theory, reinforcement learning has emerged as a crucial approach for optimizing urban traffic signal control and driving theoretical research in this field.

For single-intersection scenarios, some studies have adopted a dynamic programming approach and a distributed signal control system that integrates automated vehicle path planning information to optimize signal timing [9]. Without oversimplification, because the traffic system is highly complex, it is difficult to model the mechanism with desirable mathematical characteristics. Therefore, non-deterministic optimization methods such as intelligent optimization algorithms and neural networks can often yield better results. For example, a genetic algorithm combined with a decomposing fuzzy system has been used to develop a genetic algorithm-based fuzzy variable division signal timing optimization method for a single-intersection scenario, which has achieved promising results [10]. Some studies have simulated the optimal timing schemes for different traffic flow scenarios to generate an initial sample dataset, in which the average queue length of all methods is used to evaluate the optimal scheme. Finally, the Webster model was used to verify the rationality of the initial sample set, and a machine learning sample database for signal timing optimization was constructed [11].

In addition, deep reinforcement learning (DRL) combined with deep neural networks has also been widely used in adaptive traffic signal control research and has achieved varying degrees of optimization effects. Table 1 summarizes several studies on traffic light control systems based on (deep) reinforcement learning. Among them, the “RL” column indicates the specific reinforcement learning algorithm used by the researchers, while the “Function Approximation” column describes the function estimation method that was investigated to express the mapping relationship between the real-time road condition state and the signal light control decision.

Table 1. Research on adaptive traffic signal control.

Research	Network	RL	Function Approximation
Research [12]	Grid	Q-learning	Bayesian
Research [13]	Barcelona, Spain	DDPG	DNN
Research [14]	Tehran, Iran	Actor-Critic	RBF, Tile Coding
Research [15]	Changsha, China	Q-learning	Linear
Research [16]	Luxembourg City	DDPG	DNN

Deep Neural Network (DNN). Radial Basis Function (RBF).

Noaen et al. [17] have summarized the application of reinforcement learning in various fields of traffic signal optimization control in recent years, explored all application methods, and provided development suggestions. According to existing studies [18–20], reinforcement learning has been found to continuously acquire the environmental state through interaction with the urban road environment, enabling the learning of optimal traffic control strategies and the formation of an adaptive control system for intelligent traffic [21]. Although the DDPG algorithm can achieve good performance in certain scenarios, it still has some limitations in terms of hyperparameter tuning or other parameter adjustment actions. Additionally, due to errors in the estimation of the Q-function, the value function of the Critic is often overestimated, and the accumulation of such errors can eventually lead to the failure of the strategy in obtaining the optimal solution. Therefore, in 2018, Fujimoto et al. [22] proposed the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, which is based on the DDPG algorithm. In 2021, Yang et al. [23] designed a unique reward function that continuously selects the most appropriate strategy as the control mechanism, thereby tracking the actions of traffic signals. This system effectively reduced vehicle delay time.

Fuzzy control is an intelligent approach distinct from traditional control methods. It does not require establishing a specific mathematical model but instead relies on summarizing the experience and control strategies related to the research object or extracting control rules from a large dataset, to achieve intelligent control of the research object. The fuzzy controller is a key means of implementing fuzzy control, and the fuzzy control of a specific intersection can be realized by constructing the fuzzy controller [24]. As technology

has advanced, researchers have found that traditional Type-1 fuzzy sets have limitations in addressing system uncertainty [25]. The use of Type-2 fuzzy control can offer some novel partitions of the input domain and demonstrate excellent performance in various applications, such as signal control, particularly in modeling and control [26]. For example, Bi et al. [27] proposed a Type-2 fuzzy coordinated control method to address coordination and dynamic uncertainty issues in trunk traffic. The gravity search algorithm was employed to iteratively optimize the membership function parameters and rules of two controllers, with the aim of better configuring the high-dimensional complex parameters of the coordinated two-layer Type-2 fuzzy logic controller.

The traffic system is a nonlinear, time-varying, and hysteretic large-scale system, making it difficult to obtain satisfactory results using traditional control methods. Furthermore, the increasing complexity of the urban road environment leads to a rapid expansion of the state-action space during the acquisition of reinforcement learning knowledge. To address these challenges, some researchers [28–31] have attempted to combine fuzzy control with reinforcement learning to make adjustments. Zhao et al. [32] designed a traffic signal controller based on reinforcement learning and a fuzzy neural network (FNN). They made full use of reinforcement learning to enable the online learning of traffic signal control algorithms and incorporated a standard two-input and one-output fuzzy neural network structure. To improve the stability and robustness of the control system, Tunc I et al. [33] used a deep Q-Learning algorithm to control the phase sequence, and a Fuzzy Logic Controller to regulate the duration of the green light. This organic combination of the deep Q-learning algorithm and fuzzy logic control (FLC) enabled the optimization of signal timing. However, the nonlinear and stochastic nature of traffic systems makes modeling a challenging task. In order to overcome the unreasonable shortcomings of manually determining the variable membership functions and fuzzy control rules of fuzzy controllers, Lin et al. [34] proposed that a multi-objective differential evolution algorithm (DEA) could be used to optimize the membership functions and fuzzy control rules. DEA employs the principle of natural evolution to achieve global fast search in the solution space, making it widely used to solve large-scale combinatorial optimization problems. The simulation results show that the intelligent control technology based on the combination of multi-objective DEA and fuzzy control can effectively reduce the average delay time of passing vehicles at intersections and adapt to the complex and dynamic traffic environment.

To summarize, the application of fuzzy control and reinforcement learning technology shows great potential in a wide range of traffic control applications. Considering the shortcomings of network error, lack of interpretability, and limited self-learning and generalization ability in Type-2 fuzzy control combined with reinforcement learning, this paper presents relevant improvements to leverage the advantages of both approaches and create a more robust control system.

The key contributions of this study are as follows:

1. We developed a model for the traffic signal control process, and established a Type-2 fuzzy control system based on the inherent fuzziness of real-time traffic state information, such as queue length and vehicle waiting time.
2. Fuzzy inference is performed on the input traffic state data. The output action of the fuzzy control system is replaced by selecting the maximum Q value from the output of the target network in the DQN algorithm, which reduces the error caused by the maximum operation of the target network. This improves the online learning rate of the agent and increases the reward value of the traffic light control action.
3. The SUMO-1.18.0 simulation software was used to model and simulate the experiment, and the effectiveness of the Type-2-FDQN algorithm was verified by comparing it with four other methods.

The rest of this paper is summarized as follows. Section 2 introduces the relevant modeling process. In Section 3, the algorithm principle and implementation flow of Type2-FDQN are presented. Section 4 describes the parameters of the simulation experiment and

analyzes the experimental results. Finally, Section 5 summarizes the key findings of this paper and proposes future research directions.

2. Related Work

2.1. Single Intersection Signal Light Control Model

Single intersection signal control is the foundation of road coordination control. Exploring the optimal method of signal timing optimization for a single intersection is key to determining the best timing period for traffic lights. This paper utilizes SUMO to model the road network at intersections. The intelligent transportation system decision center employs a deep double Q network combined with fuzzy logic to train the Agent, and the fuzzy logic is integrated to determine the operation mode. The environment is an intersection road, the state space S represents the position and speed of all vehicles, and the action state space A includes four phases and timing quantities of traffic lights at two intersections. Within a fixed period T , for each intersection signal light, the Agent can adaptively select an optimal action from the action space as a decision based on the environmental state, thereby improving the overall driving speed of all vehicles and reducing their travel time. The traffic light control model at intersections established by SUMO software [35] is shown in Figure 1.

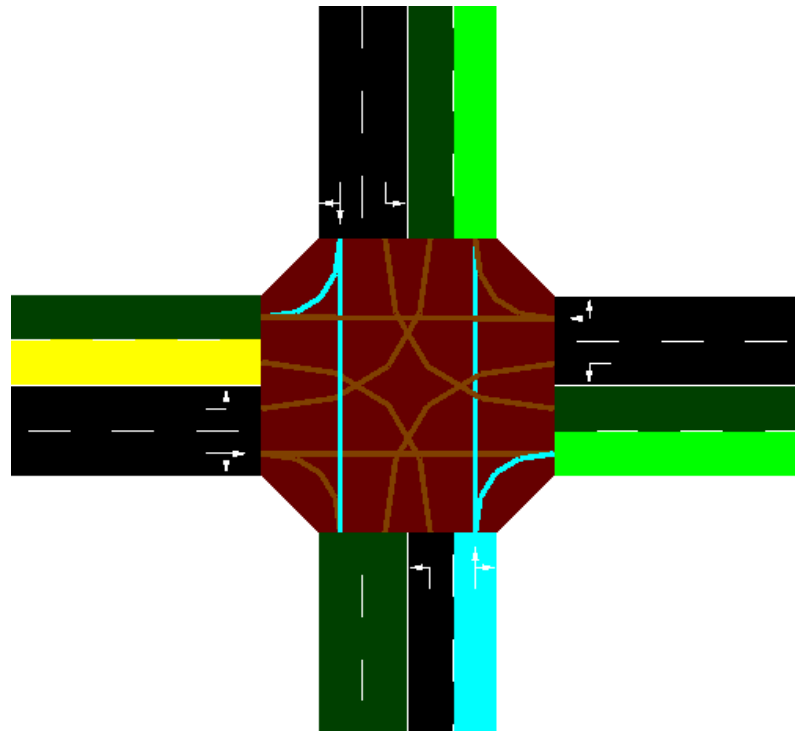


Figure 1. Single intersection signal light control model.

2.2. Definition of State Space

In this paper, the traffic state of a single intersection is defined by two parameters: the current vehicle position and speed. Using the lanes leading to the west entrance of the intersection, shown in Figure 1, the intersection is evenly divided into square grids of equal size. The side length of each grid is set to accommodate only one vehicle, ensuring that no two vehicles occupy the same grid simultaneously. Within each grid, the vehicle status is represented by a binary group: the position coordinate is a binary value, where 1 indicates the presence of a vehicle and 0 indicates no vehicle. The speed coordinate is a floating-point value representing the current vehicle speed in meters per second. By obtaining the vehicle information in all grids on each lane, the position matrix and speed matrix corresponding

to each entrance direction of the intersection can be established. The process of converting the traffic state into the input matrix is shown in Figure 2.

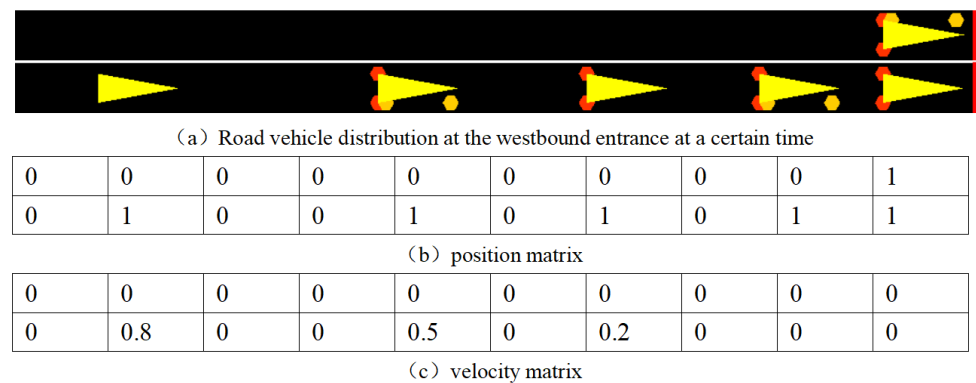


Figure 2. There are three multiple panels in the process of converting the traffic state into the input matrix.

2.3. Definition of Action Space

There are four phases in this paper, which are as follows: east–west straight and right turn (EW), east–west left turn (EWL), north–south straight and right turn (NS), and north–south left turn (NSL). The agent is responsible for selecting the appropriate actions based on the current traffic situation to ensure the smooth flow of vehicles at the intersection. In this system, the agent scans the traffic state and chooses one of two actions: 0—do not change the traffic signal phase, or 1—turn on the green light for the next traffic signal phase in the sequence. At the end of each control step, the agent performs the action to either maintain the current phase or execute the next phase in the sequence. Through the execution of this series of actions, the agent indirectly realizes the dynamic update of the intersection traffic signal timing scheme. The four phases selected in this paper are shown in Figure 3.

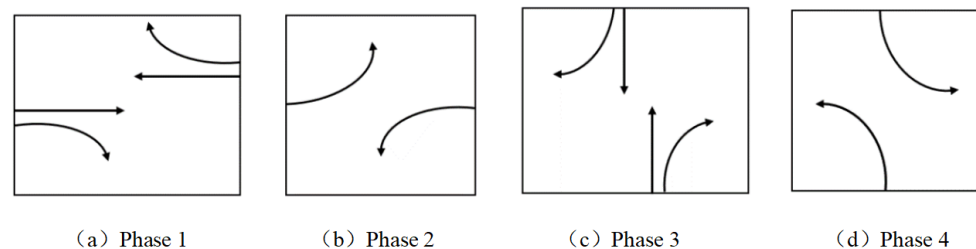


Figure 3. Four-phase signal diagram.

2.4. Definition of Reward Value Space

After the agent takes an action, the environment makes corresponding changes and generates certain reactions, which are digitized into feedback rewards for the agent. The agent receives the reward value from the environment, indicating whether the action had a positive or negative impact on the current environmental state. This allows the agent to learn and take actions that can maximize the reward. The primary goal of the agent is to improve the efficiency of intersections and reduce vehicle delay times. The length of the queue or the waiting time of vehicles can effectively convey this result, so this paper defines the reward as the change in the cumulative waiting time between two adjacent cycles. Let i_t represent the observed vehicles in a cycle, the waiting time of vehicle i_t in a cycle is w_{i_t} , and N_t represents the total number of vehicles in a cycle, where $1 < i_t < N_t$, then the reward value in the t cycle is as follows:

$$R_w = W_t - W_{t+1} \tag{1}$$

$$W_t = \sum_{i_t=1}^{N_t} w_{i_t,t} \tag{2}$$

In summary, the reward is equal to the increment of the accumulated waiting time of the vehicle before and after the action is taken. If the reward is larger, it means that the wait time has increased by less than before.

3. Traffic Control Decision Based on Type2-FDQN Algorithm

3.1. Design Principle of Type 2 Fuzzy Controller

The function of the fuzzer is to map the exact input value to a fuzzy number, while a Type-2 fuzzer maps the system’s input and output variables to a Type-2 fuzzy set. In other words, building on Type-1 fuzzification, the membership degree is further blurred to create a three-dimensional membership function. It is important to note that if any input or output variable is a Type-2 fuzzy set, the entire fuzzy system is considered a Type-2 system. To simplify calculations, single-point fuzzification is usually performed.

In the Type-2 case, the structure of the rules is the same as in the Type-1 case, still consisting of a series of “IF-THEN” statements. However, some or all of the Type-2 rules are of Type-2. Consider a Type-2 fuzzy system with P inputs $x_1 \in X_1, \dots, x_p \in X_p$ and an input $y \in Y$, assuming it has M rules, in the case of Type-2 for both Mamdani and TSK rule forms, the first rule can be expressed as follows:

R^1 : If x_1 is $\tilde{F}_1^l, \dots, x_p$ is \tilde{F}_p^l , Then y^l is $\tilde{G}^l, 1 = 1, \dots, M$ (Mamdani)

R^l : If x_1 is $\tilde{F}_1^l, \dots, x_p$ is \tilde{F}_p^l , Then $y^l = f^l(x_1, \dots, x_p), 1 = 1, \dots, M$ (TSK)

When determining the membership function, the influence of the membership function curve shape on the control performance of the system should be considered. The Gaussian membership function curve has a relatively smooth and stable shape and control characteristics, making it a reasonable form to describe fuzzy subsets. Therefore, this paper selects the interval type-2 Gaussian membership function for the uncertainty deviation, as shown in Equation (3).

$$\mu_{\tilde{A}}(x) = \exp\left(-\frac{(x - m)^2}{2\sigma^2}\right), \sigma \in [\sigma_1, \sigma_2] \tag{3}$$

where m is the center of the membership function, respectively, σ_1 and σ_2 are the two deviations of the membership function.

In a Type-1 fuzzy system, the inference engine is used to combine rules and map input fuzzy sets to output fuzzy sets. The multiple antecedents of a rule are connected by a T-norm operation, the membership of the input set and the membership of the output set are combined, and the combination of multiple rules can be obtained by a T-conorm operation or weighted summation during the defuzzification process. The reasoning process of Type-2 fuzzy systems is highly similar. The inference engine is used to combine rules and generate a mapping from an input Type-2 fuzzy set to an output Type-2 fuzzy set. In this paper, Mamdani fuzzy inference is applied to the established fuzzy inference rule to obtain a fuzzy quantity, and the inference result of the first triggering rule is presented in Equation (4).

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^l}(y) \cap \left\{ \sqcup_{x \in X} \left\{ \left[\mu_{\tilde{X}_1}(x_1) \cap \mu_{\tilde{F}_1^l}(x_1) \right] \cap \dots \cap \left[\mu_{\tilde{X}_p}(x_p) \cap \mu_{\tilde{F}_p^l}(x_p) \right] \right\} \right\} \tag{4}$$

where $\mu_{\tilde{B}^l}(y)$ is the fuzzy inference value of the first triggering rule, and $\mu_{\tilde{F}_i^l}(x_i)$ is the Type-2 fuzzy membership value of the i th input.

If the above input is blurred by a single point, the above formula can be simplified as follows:

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^l}(y) \cap \left\{ \sqcup_{x \in X} \left\{ \mu_{\tilde{F}_1^l}(x_1) \cap \dots \cap \mu_{\tilde{F}_p^l}(x_p) \right\} \right\} \tag{5}$$

Assuming that N of the M rules are triggered, the final inference result is as follows:

$$\mu_{\tilde{B}}(y) = \sqcup_{i=1}^N \mu_{\tilde{B}^i}(y) \tag{6}$$

The main structure of the Type-2 fuzzy controller is very similar to that of the Type-1 fuzzy control, but the output part differs. The Type-1 fuzzy system has only one defuzzification module, while the Type-2 fuzzy inference machine produces a Type-2 fuzzy output. This Type-2 fuzzy output must first be transformed into a Type-1 fuzzy output through a process called type reduction, before the final defuzzification step. Therefore, the output part of the Type-2 fuzzy system requires an additional type reduction module before the defuzzification module. Type reduction is a unique feature of Type-2 fuzzy systems, and is also considered a challenging aspect of this approach. The output of a Type-2 fuzzy inference system is a Type-2 fuzzy set, which needs to be transformed into a Type-1 fuzzy set before further processing. Fuzzy reduction is an extension of the defuzzification process used in Type-1 fuzzy systems, but its computation and complexity are significantly greater. This paper primarily employs a defuzzification method based on set center reduction to obtain the final crisp output. The set center reduction approach is used to transform the Type-2 fuzzy set into a Type-1 fuzzy set, which can then be defuzzified using standard techniques.

In the Type I set center defuzzification method, each rule consequent fuzzy set is replaced with a single crisp value located at its center of gravity. Then, the center of gravity of the Type I fuzzy set composed of these single crisp values is calculated to determine the final output. The mathematical expression for this process is as follows:

$$y_{cos}(x) = \frac{\sum_{i=1}^M c^l T_{i=1}^p \mu_{F_i^l}(x_i)}{\sum_{i=1}^M T_{i=1}^p \mu_{F_i^l}(x_i)} \tag{7}$$

where T represents the chosen T-norm, and c^l is the center of gravity of the first after-set.

Defuzzification is the process of mapping a fuzzy number to a precise or crisp number. In a Type-1 fuzzy system, the output of the inference engine is a type-1 fuzzy set, and the precise output of the system can be directly obtained by solving the fuzzy module. However, in a Type-2 fuzzy system, the output of the inference engine is a Type-2 fuzzy set, which must first undergo a type-reduction operation before the defuzzification process can be carried out to determine the final crisp output.

In this paper, a two-type fuzzy signal controller is established at a single intersection. The input variables of the fuzzy control system are the vehicle queue lengths L_1 and L_2 at the current phase and the next phase, respectively, at time T . The output variable is the green light extension time T . The discourse domain of the number of queued vehicles is set as $[0, 50]$, the discourse domain of the vehicle speeds is set as $[0, 35]$, and the fuzzy discourse domain of the output variable T is set as $[0, 30]$. The fuzzy sets are divided into three subsets based on the input values: short (S), medium (M), and long (L). Based on daily experience and the expertise of the traffic police, the fuzzy rules are shown in Table 2.

Table 2. Fuzzy inference rules.

T				L_1	
		S	M	L	
L_2	S	S	M	L	
	M	S	M	L	
	L	S	S	M	

T —Green extension. L_1 —Current phase vehicle queue length. L_2 —Next phase vehicle queue length.

The implementation of traffic control on a four-phase single intersection using the designed two-type fuzzy controller can be summarized in the following steps:

Step 1: Determine the values of the input variables L_1 and L_2 based on the traffic model.

Step 2: Map the input variable $L_1(L_2)$ to the fuzzy domain as $X = k_1 * L_1(L_2)$, $k_1 = 3/10$.

Step 3: Establish the input and output membership functions in the Type-2 fuzzy controller according to Equation (3).

Step 4: According to Equations (4) and (7) in the fuzzy rule table, type-2 fuzzy reasoning and type reduction are carried out, respectively, and the final precise output value Y is obtained through defuzzification.

Step 5: The fuzzy-domain output variable is converted to the actual output value using the equation $T = k_2 * Y + c$, where the coefficient $c = 15$, and $k_2 = 25/6$ during the straight phase or $k_2 = 5/2$ during the left turn phase.

Step 6: Apply the calculated value of T to the traffic model in order to obtain the corresponding average vehicle delay and queue length. Then, return to step 1 and repeat the process until the set simulation time is reached.

3.2. Principle of Type2-FDQN Algorithm

The DQN (Deep Q-Network) algorithm combines neural network and reinforcement learning techniques. However, the use of the max operation to select and evaluate the state-action value function can lead to overestimation issues caused by the neural network. The max operation always tends to select the action corresponding to the amplified state-action value function, resulting in a biased optimal strategy learned by the model. Consequently, the agent's action decisions may not be optimal, leading to a reduction in the reward value.

In order to reduce the output Q value error of the target network selected by the max operation in the DQN algorithm, the DDQN algorithm is commonly used at present. However, the essence of the DDQN algorithm is to decouple action selection and strategy evaluation by using a predictive network, which can lead to low estimation accuracy. To address this, a Type-2 fuzzy control system is introduced to select the action based on the output Q value of the target network. This gives rise to a Type-2-FDQN-based reinforcement learning algorithm, which aims to obtain more accurate agent actions and reduce the error of the output Q value of the target network. Figure 4 illustrates the traffic decision-making principle diagram based on the Type-2-FDQN algorithm. Figure 5 illustrates the flow chart of the traffic decision-making process based on the Type-2-FDQN algorithm.

When calculating the loss function, the DQN algorithm adopts max operation when selecting the output Q value of the target network. At this time, the output Q value of the target network is the maximum value, and $Q_{target}(s_{t+1}, a_{t+1}, \theta') = \max_{a+1} Q_{target}(s_{t+1}, a_{t+1}, \theta')$ can be obtained; while the output Q value of the target network selected by the FDQN algorithm is the output action $a_f (a_f \in \{a_{on}, a_{off}\})$ of the fuzzy control system. If the output action a_f of the fuzzy control system is not equal to the action a_{t+1} that takes the max operation to obtain the output Q value of the target network, and the corresponding output Q value of the target network is not equal, then $Q_{target}(s_{t+1}, a_f, \theta') \leq \max_{a+1} Q_{target}(s_{t+1}, a_{t+1}, \theta')$. It can be seen that selecting the output Q value of the target network in the FDQN algorithm can reduce the error caused by using the max operation in the DQN algorithm, and alleviate the overestimation phenomenon.

During the training process, the agent interacts with the environment to obtain a quadruple of sampled data (s_t, a_t, R_t, s_{t+1}) , and starts to update the network parameters. The FDQN algorithm uses two independent state-action value functions. Instead of using max operation, the output action of the fuzzy control system is used to determine the target network action:

$$\arg \max_{a_{t+1}} Q_{target}(s_{t+1}, a_{t+1}, \theta') = a_f \quad (8)$$

The output action a_f of the Type-2 fuzzy control system is used to calculate the estimated Q value of the policy update return:

$$Q_T = R_t + \gamma \max_{a_{t+1}} Q_{target}(s_{t+1}, a_{t+1}, \theta') = R_t + \gamma Q_{target}(s_{t+1}, a_f, \theta') \quad (9)$$

where γ is the discount factor.

Finally, according to the estimated Q value of the updated return according to the strategy, the value function iteration process of the FDQN algorithm is as follows:

$$Q_{estimation}(s_t, a_t, \theta) + \alpha [Q_T - Q_{estimation}(s_t, a_t, \theta)] \rightarrow Q_{estimation}(s_t, a_t, \theta) \quad (10)$$

where α is the learning rate.

Then the loss function of the FDQN algorithm is as follows:

$$L(\theta) = E \left[\left(R_t + \gamma Q_{target}(s_{t+1}, a_f, \theta') - Q_{estimation}(s_t, a_t, \theta) \right)^2 \right] \quad (11)$$

where E is expectation.

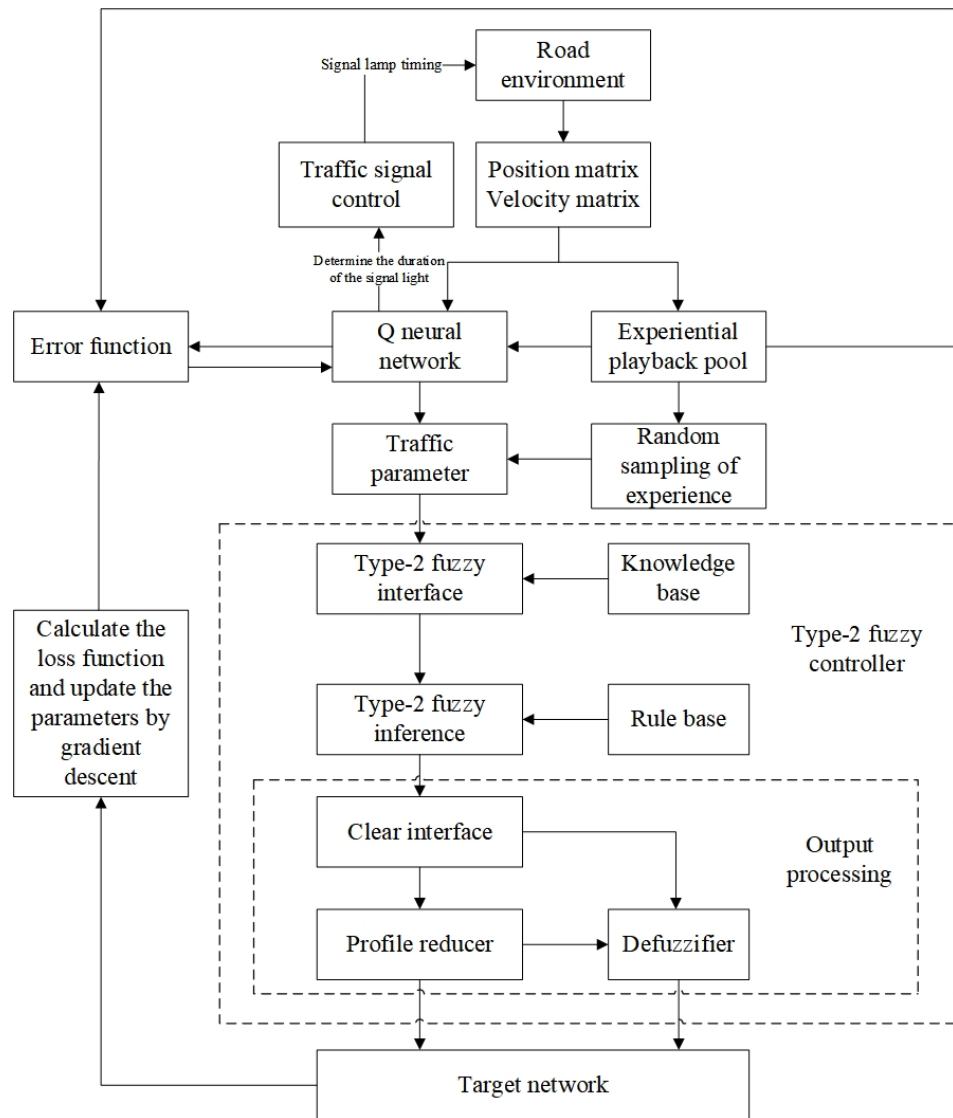


Figure 4. Traffic decision principle diagram based on Type2-FDQN algorithm.

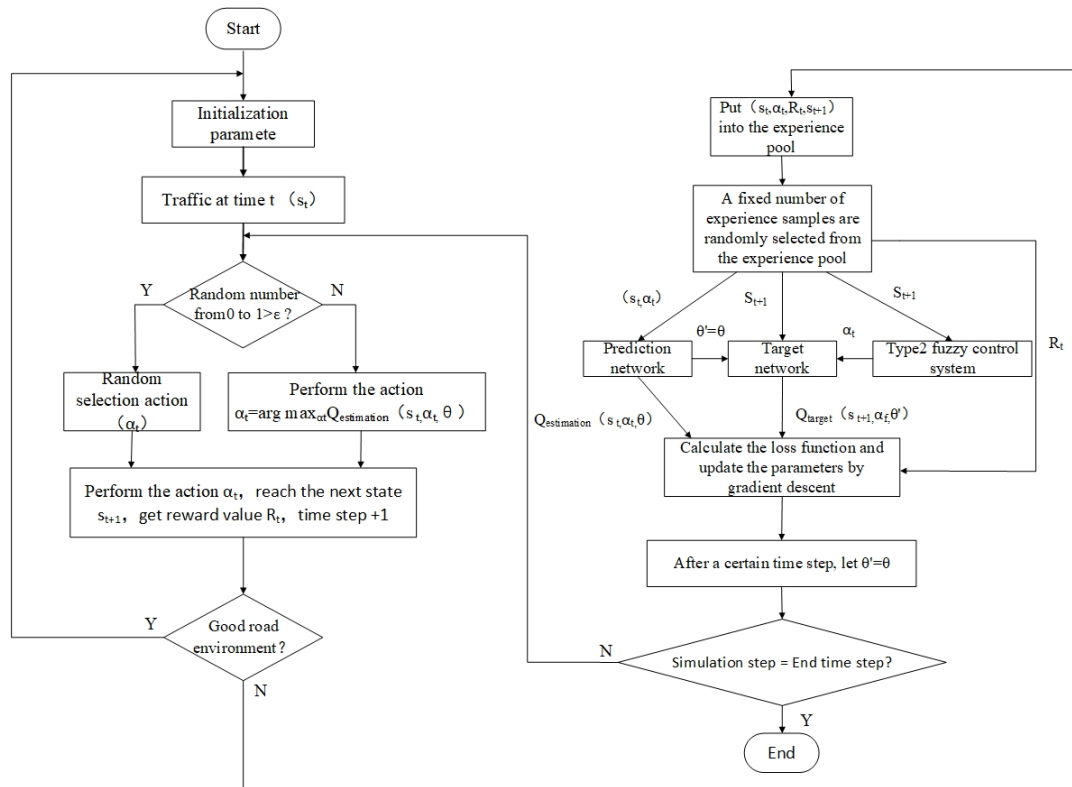


Figure 5. The workflow of the traffic decision-making process based on the Type-2-FDQN algorithm.

4. Simulation Experiments and Analysis of Results

4.1. Experiment Settings

This study utilizes the SUMO software [35] to build the simulation platform, and the Traci interface of traffic control software is used to obtain real-time traffic flow data and modify traffic control state through Python 3.10. As shown in Figure 6, the single-intersection simulation area features lane lengths of 300 m and a maximum allowable speed of 70 km/h. All simulated vehicles enter the traffic junction from the starting position of the road, with each vehicle being 3 m in length and maintaining a minimum distance of 2 m between vehicles.

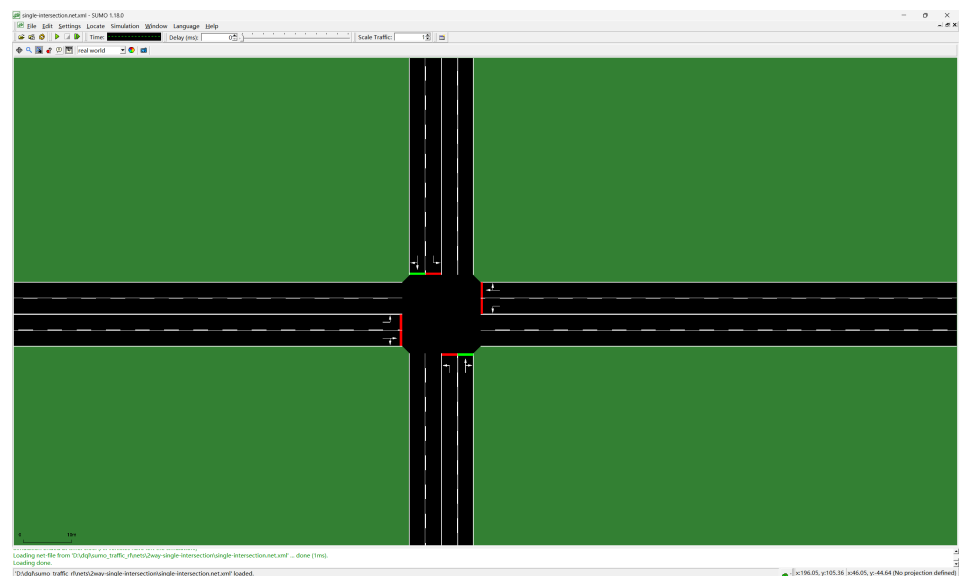


Figure 6. SUMO simulation single-intersection simulation environment.

In order to verify the effectiveness of the proposed Type2-FDQN algorithm in traffic signal control, this paper conducted several comparative experiments with different algorithms. During training, all algorithms used the same network structure and hyperparameter settings. The simulation environment parameters used in the Python platform [36] are shown in Table 3. Figure 7 illustrates the software simulation process.

Table 3. Parameter Settings in the simulation process.

Hyperparameter	Value
Experience pool size M	20,000
Number of training rounds episodes N	35
Number of training steps per round steps T	3000
Discount Factor γ	0.99
Learning Rate α	0.001
Sample set size B	512
Training frequency	50

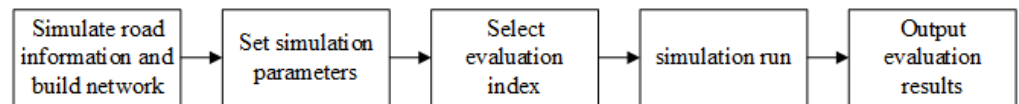


Figure 7. Software simulation process.

The average cumulative reward value of the two algorithms established by Python platform [36] is shown in Figure 8. The horizontal axis represents the training time, while the vertical axis shows the average cumulative reward value. Initially, due to the lack of experience samples, the agent employs exploration strategies, resulting in a relatively low reward value. However, as training progresses, the agent continuously interacts with the environment, accumulating a large number of experience samples. Consequently, the algorithm’s reward value steadily increases, eventually converging in the 2000s.

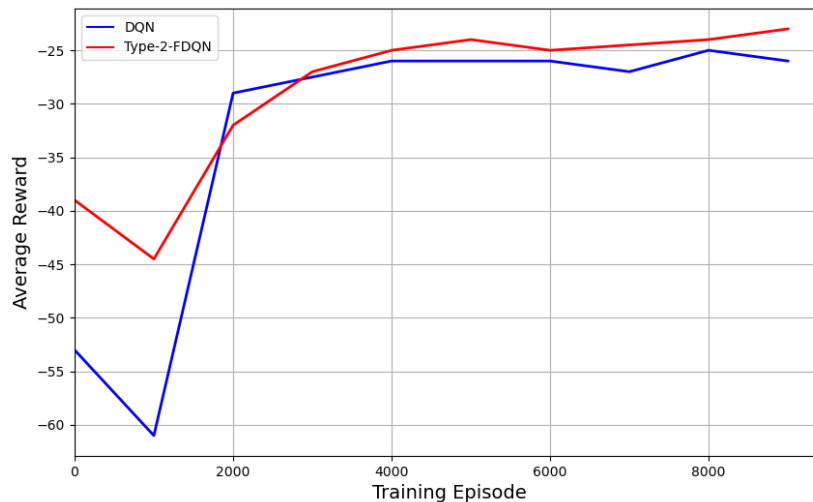


Figure 8. Trend chart of average cumulative reward value.

4.2. Comparison Experiment with the Same Traffic Flow

In order to evaluate the performance of the traffic control strategy (Type-2-FDQN) proposed in this paper, the control performance evaluation indexes used are the average queue length, average waiting time, average driving speed, and average delay time. In the simulation experiment stage, the four traffic control methods were simulated several times. The fixed signal cycle length was set to 120 s, with each phase including 27 s of green time and 3 s of yellow light time. By averaging the experimental results of the four methods,

the comparison results are shown in Table 4. The data indicates that the Type-2-FDQN algorithm has the best performance. Compared to the DQN algorithm, the Type-2-FDQN algorithm reduces the average queue length by 19.4%, the average waiting time by 18.9%, and increases the average driving speed by 20.8%. The average delay time is also reduced by 10.1%. These results demonstrate that the Type-2-FDQN algorithm proposed in this paper can effectively alleviate traffic congestion and realize efficient traffic signal control. All figures of experimental results shown below were generated using Python software [36].

Table 4. Performance comparison of different control algorithms under the same traffic.

Evaluation Index	Fixed-Time	DQN	Type-1-FDQN	Type-2-FDQN
Average queue length (car)	23.7913	19.7945	18.5506	15.9352
Average waiting time (s)	66.2488	59.8790	52.4288	48.5093
Average speed (m/s)	4.4865	5.7062	6.3319	6.8944
Average delay time(s)	87.6432	76.3715	72.4598	68.6564

During the 2000th simulation step, the system entered a congested state, resulting in an increase in the average queue length and a decrease in the average vehicle speed. Figure 9 compares the average queue length under four different traffic control strategies. It is evident that the timed traffic signal control method is ineffective in managing congested traffic flows. In contrast, the Deep Q-Network (DQN) control strategy can effectively regulate traffic and be appropriately adjusted when traffic is congested, preventing the queue length from continuing to rise. The algorithm based on Type-2 Fuzzy and reinforcement learning demonstrates better control performance during vehicle congestion. The Type-1-FDQN algorithm shows a flattening trend earlier than the DQN algorithm, with a slight downward fluctuation. After the Type-2-FDQN algorithm achieves a stable state, it further adjusts to reduce the queue length, thus better alleviating traffic congestion. Figure 10 shows the comparison of average vehicle speeds under the four control strategies. In the beginning period, all indicators under the control of fixed signal timing and various algorithms are poor, and the speed decreases at this time. This is because the agent mainly accumulates an experience pool and learns in the beginning stage. At this time, the number of samples in the experience pool is small and the network parameters are updated less, so the action prediction is not accurate. As the agent accumulates more experience pool samples and learns and updates network parameters many times, the vehicle speed parameter indicators are improved to a certain extent, and it can quickly recover to a stable average vehicle speed. Among them, the control strategy based on Type-2 fuzzy and reinforcement learning can recover the stable speed faster and reduce the queue length in the congested environment.

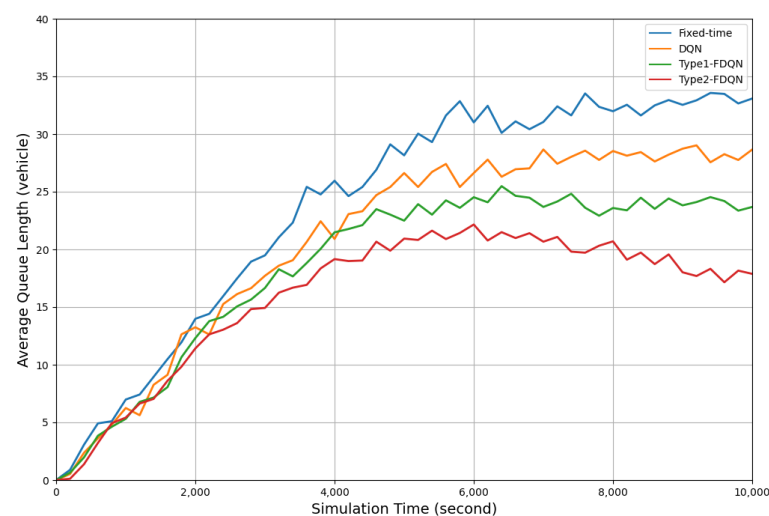


Figure 9. Average queue length of the vehicle.

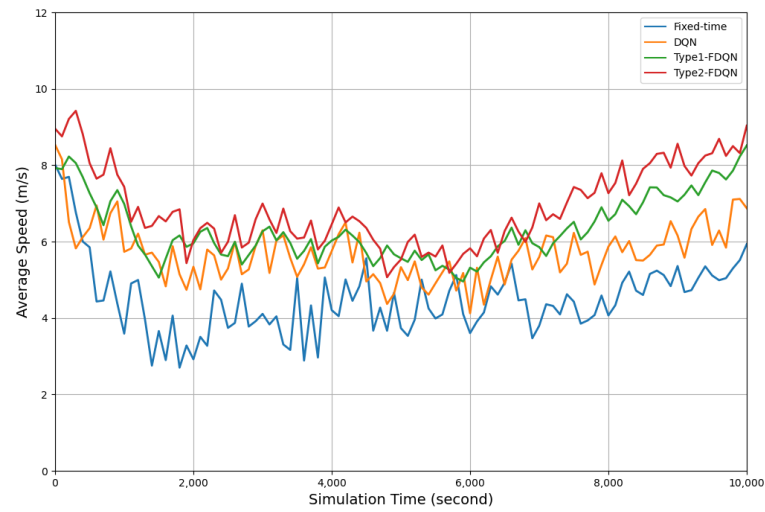


Figure 10. Average speed of vehicle.

Figures 11 and 12 present a comparison of the average waiting time and total waiting time of vehicles under the four control strategies, respectively. The fixed timing strategy is a simple and static control method that fails to adapt flexibly to changes in traffic flow, resulting in long waiting times. In contrast, the DQN algorithm can learn and adjust according to real-time environmental information, enabling it to optimize traffic control to a certain extent and reduce the average waiting time of vehicles. The Type-1-FDQN algorithm considers the fuzziness and uncertainty of the environment to a certain degree, leading to a slightly reduced average waiting time compared to the DQN algorithm. Finally, the Type-2-FDQN algorithm exhibits higher complexity and adaptability and demonstrates better average and total waiting times to some extent.

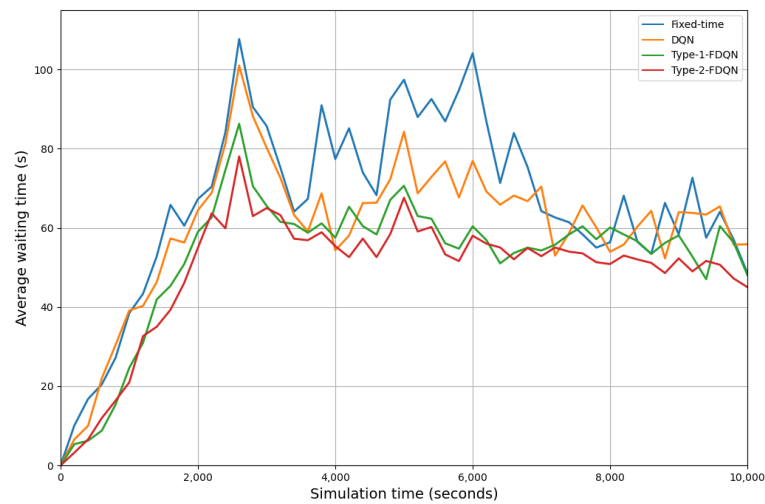


Figure 11. Average waiting time of the vehicle.

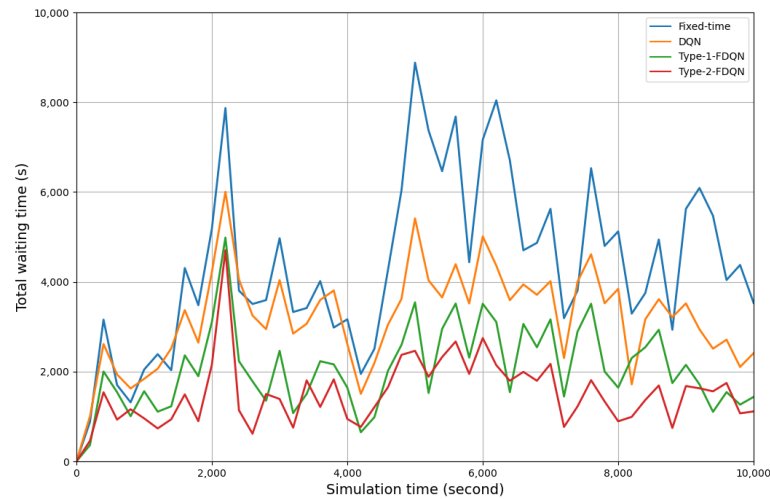


Figure 12. Total waiting time of vehicle.

4.3. Comparison Experiment with Different Traffic Flow

To evaluate the performance of the traffic control strategy (Type-2-FDQN) proposed in this paper, the four control strategies are compared under different traffic flows.

Figures 13 and 14 respectively present the average speed and average queue length of vehicles when the traffic flow ranges from 500 to 3000 vehicles. When the traffic flow is relatively low and not too congested, the average speed and average queue length across the four control methods do not differ significantly. However, as the traffic flow gradually increases and the traffic becomes more congested, the performance of the traditional fixed-timing strategy begins to decline compared to the other three algorithms. Notably, the DQN algorithm exhibits similar effects to the Type-1-FDQN and Type-2-FDQN algorithms. When the traffic flow exceeds 2000 vehicles (indicating severe congestion), the traditional fixed-timing strategy is evidently unable to flexibly adapt to the changing traffic conditions. In contrast, the control strategy of the Type-2-FDQN algorithm cannot only maintain a high average vehicle speed but also effectively reduce the average queue length as much as possible. Overall, the findings suggest that the Type-2-FDQN algorithm outperforms the other control methods in terms of both preserving high average vehicle speeds and minimizing queue lengths, especially under congested traffic conditions.

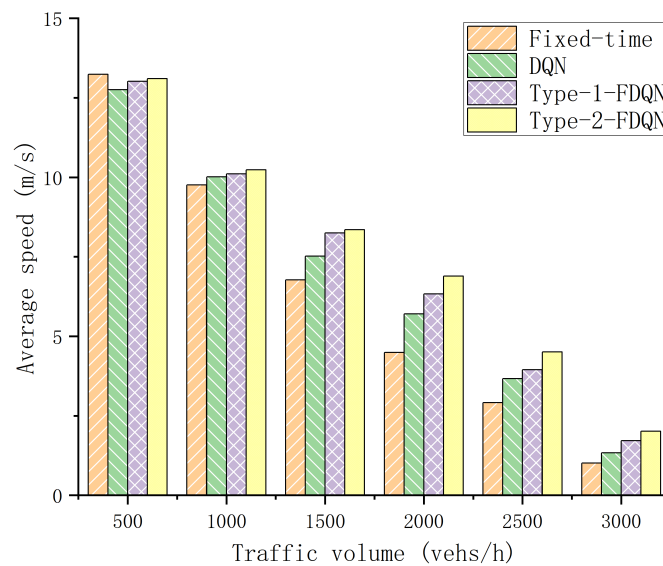


Figure 13. Average vehicle speed under different traffic volumes.

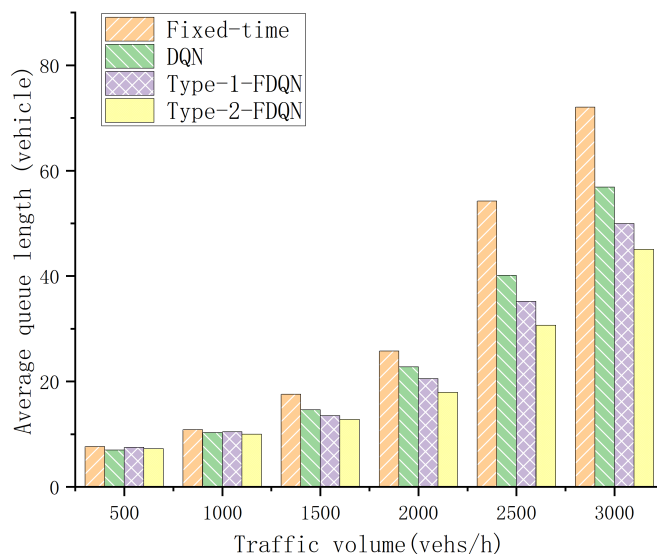


Figure 14. Average vehicle queue length under different traffic volumes.

5. Conclusions and Future Work

5.1. Conclusions

When addressing current traffic challenges, new intersection control strategies are needed to manage a large number of intersections, minimize delays, and improve traffic capacity and safety, as intersections are of vital practical significance. Focusing on a single intersection as the research object, this paper proposes a novel traffic control strategy that combines deep reinforcement learning with Type-2 fuzzy control. The proposed algorithm not only leverages many advantages of reinforcement learning but also utilizes the output action of the fuzzy control system to replace the action of selecting the maximum Q-value from the output of the target network in the DQN algorithm. This reduces the error caused by the max operation in the target network, improves the online learning rate of the agent, and increases the reward value of the control action for the traffic signal. As a result, the autonomous learning and adaptive capabilities of the intelligent traffic control algorithm are further enhanced. This paper simulates the road network environment using the Traci interface and SUMO to evaluate different traffic demand scenarios. The experimental results show that compared to fixed timing decision-making, the Type-2-FDQN algorithm can converge faster and maintain more stable performance, further improving the key evaluation metrics for the traffic system. This study not only brings new ideas to the field of intelligent traffic control but also provides strong support for improving the efficiency and adaptability of traffic signal control. This research is expected to promote intelligent transportation systems that better meet the challenges of the urban road environment and achieve more intelligent, flexible, and efficient traffic management.

5.2. Future Work

Through the combination of Type-2 fuzzy control and reinforcement learning, this paper has achieved some research results. However, there are still some limitations to this study. In view of these limitations, the following directions for future work are proposed:

1. The model studied in this paper is optimized for single-crossing intersections, which may not guarantee the operation efficiency of arterial roads or regions in general. Due to the coupling of various factors between intersections and road sections, it is more challenging to optimize the timing for arterial roads and regions. The next step could be to study the optimization of multiple performance indicators for arterial roads and regions.
2. This paper employs the SUMO traffic simulation software and Python programming language to realize the secondary development of a deep reinforcement learning

framework. Live simulation is conducted with the SUMO-simulated road network environment through the Traci interface to verify the rationality of the control method. However, real-world traffic scenarios involve complex factors such as pedestrians, non-motor vehicles, and weather conditions, which are issues that need to be considered when simulating a realistic traffic network.

3. With the advancement of artificial intelligence, improved optimization algorithms have continued to emerge in the research field, such as the Ivy algorithm (LVYA). We plan to utilize the Ivy algorithm for optimization in the next step, set up multiple experimental groups for comparison, and constantly refine the control system to address urban traffic problems and enhance traffic efficiency.

Author Contributions: Conceptualization, Y.B. and Q.D.; methodology, Y.B. and Q.D.; software, Q.D.; validation, Y.B. and S.R.; formal analysis, Y.B. and Q.D.; investigation, Y.D.; resources, D.L.; data curation, Q.D.; writing—original draft preparation, Q.D.; writing—review and editing, Q.D.; visualization, S.R.; supervision, Y.D.; project administration, Y.B. and D.L.; funding acquisition, Y.B. and Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by grants from the National Natural Science Foundation (No. 62303214), the Jiangsu Province Natural Science Foundation (No. BK20201043), the Nanjing Institute of Technology Innovation Fund Project (No. CKJB202203), and the Key Project of Basic Science Research in Universities of Jiangsu Province (No. 23KJA460008).

Data Availability Statement: Data are contained within the article.

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

References

1. Kelley, S.B.; Lane, B.W.; Stanley, B.W.; Kane, K.; Nielsen, E.; Strachan, S. Smart transportation for all? A typology of recent US smart transportation projects in midsized cities. *Ann. Am. Assoc. Geogr.* **2020**, *110*, 547–558.
2. Oladimeji, D.; Gupta, K.; Kose, N.A.; Gundogan, K.; Ge, L.; Liang, F. Smart transportation: An overview of technologies and applications. *Sensors* **2023**, *23*, 3880. [[CrossRef](#)]
3. Cao, K.; Wang, L.; Zhang, S.; Duan, L.; Jiang, G.; Sfarra, S.; Zhang, H.; Jung, H. Optimization Control of Adaptive Traffic Signal with Deep Reinforcement Learning. *Electronics* **2024**, *13*, 198. [[CrossRef](#)]
4. Wang, F.Y.; Lin, Y.; Ioannou, P.A.; Vlacic, L.; Liu, X.; Eskandarian, A.; Lv, Y.; Na, X.; Cebon, D.; Ma, J.; et al. Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10262–10278. [[CrossRef](#)]
5. Li, H.; Chen, Y.; Li, K.; Wang, C.; Chen, B. Transportation internet: A sustainable solution for intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 15818–15829. [[CrossRef](#)]
6. Song, W.; Rajak, S.; Dang, S.; Liu, R.; Li, J.; Chinnadurai, S. Deep learning enabled IRS for 6G intelligent transportation systems: A comprehensive study. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 12973–12990. [[CrossRef](#)]
7. Kaffash, S.; Nguyen, A.T.; Zhu, J. Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis. *Int. J. Prod. Econ.* **2021**, *231*, 107868. [[CrossRef](#)]
8. Li, Q.; Wang, W.; Zhu, Y.; Ying, Z. BOppCL: Blockchain-Enabled Opportunistic Federated Learning Applied in Intelligent Transportation Systems. *Electronics* **2023**, *13*, 136. [[CrossRef](#)]
9. Rasheed, F.; Yau, K.L.A.; Noor, R.M.; Wu, C.; Low, Y.C. Deep reinforcement learning for traffic signal control: A review. *IEEE Access* **2020**, *8*, 208016–208044. [[CrossRef](#)]
10. Li, J.; Peng, L.; Xu, S.; Li, Z. Distributed edge signal control for cooperating pre-planned connected automated vehicle path and signal timing at edge computing-enabled intersections. *Expert Syst. Appl.* **2024**, *241*, 122570. [[CrossRef](#)]
11. Li, R.; Xu, S. Traffic signal control using genetic decomposed fuzzy systems. *Int. J. Fuzzy Syst.* **2020**, *22*, 1939–1947. [[CrossRef](#)]
12. Khamis, M.A.; Gomaa, W. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Eng. Appl. Artif. Intell.* **2014**, *29*, 134–151. [[CrossRef](#)]
13. Casas, N. Deep deterministic policy gradient for urban traffic light control. *arXiv* **2017**, arXiv:1703.09035.
14. Aslani, M.; Mesgari, M.S.; Wiering, M. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 732–752. [[CrossRef](#)]
15. Liu, W.; Qin, G.; He, Y.; Jiang, F. Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8667–8681. [[CrossRef](#)]
16. Genders, W. Deep Reinforcement Learning Adaptive Traffic Signal Control. Ph.D. Thesis, McMaster University, Hamilton, ON, Canada, 2018.

17. Noaen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Abad, Z.S.H.; Bazzan, A.L.; Far, B. Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Syst. Appl.* **2022**, *199*, 116830. [[CrossRef](#)]
18. Zhang, R.; Ishikawa, A.; Wang, W.; Striner, B.; Tonguz, O.K. Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 404–415. [[CrossRef](#)]
19. Liang, X.; Du, X.; Wang, G.; Han, Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1243–1253. [[CrossRef](#)]
20. Li, L.; Lv, Y.; Wang, F.Y. Traffic signal timing via deep reinforcement learning. *IEEE/CAA J. Autom. Sin.* **2016**, *3*, 247–254. [[CrossRef](#)]
21. Ning, Z.; Zhang, K.; Wang, X.; Obaidat, M.S.; Guo, L.; Hu, X.; Hu, B.; Guo, Y.; Sadoun, B.; Kwok, R.Y. Joint computing and caching in 5G-envisioned Internet of vehicles: A deep reinforcement learning-based traffic control system. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5201–5212. [[CrossRef](#)]
22. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
23. Yang, X.; Xu, Y.; Kuang, L.; Wang, Z.; Gao, H.; Wang, X. An information fusion approach to intelligent traffic signal control using the joint methods of multiagent reinforcement learning and artificial intelligence of things. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 9335–9345. [[CrossRef](#)]
24. Nae, A.C.; Dumitrache, I. Neuro-fuzzy traffic signal control in urban traffic junction. In Proceedings of the 2019 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 28–30 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 629–635.
25. Jovanović, A.; Teodorović, D. Type-2 fuzzy logic based transit priority strategy. *Expert Syst. Appl.* **2022**, *187*, 115875. [[CrossRef](#)]
26. Wu, D.; Mendel, J.M. Recommendations on designing practical interval type-2 fuzzy systems. *Eng. Appl. Artif. Intell.* **2019**, *85*, 182–193. [[CrossRef](#)]
27. Bi, Y.; Lu, X.; Sun, Z.; Srinivasan, D.; Sun, Z. Optimal type-2 fuzzy system for arterial traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 3009–3027. [[CrossRef](#)]
28. Kumar, N.; Rahman, S.S.; Dhakad, N. Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4919–4928. [[CrossRef](#)]
29. Zhang, Y.; Chadli, M.; Xiang, Z. Prescribed-time formation control for a class of multiagent systems via fuzzy reinforcement learning. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 4195–4204. [[CrossRef](#)]
30. Xiao, B.; Lam, H.K.; Xuan, C.; Wang, Z.; Yeatman, E.M. Optimization for interval type-2 polynomial fuzzy systems: A deep reinforcement learning approach. *IEEE Trans. Artif. Intell.* **2022**, *4*, 1269–1280. [[CrossRef](#)]
31. Khooban, M.H.; Gheisarnejad, M. A novel deep reinforcement learning controller based type-II fuzzy system: Frequency regulation in microgrids. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *5*, 689–699. [[CrossRef](#)]
32. Zhao, H.; Chen, S.; Zhu, F.; Tang, H. Traffic signal control based on reinforcement learning and fuzzy neural network. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 4030–4035.
33. Tunc, I.; Soylemez, M.T. Fuzzy logic and deep Q learning based control for traffic lights. *Alex. Eng. J.* **2023**, *67*, 343–359. [[CrossRef](#)]
34. Lin, H.; Han, Y.; Cai, W.; Jin, B. Traffic signal optimization based on fuzzy control and differential evolution algorithm. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 8555–8566. [[CrossRef](#)]
35. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wiefner, E. Microscopic traffic simulation using sumo. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2575–2582.
36. Reitz, K. *Python Guide Documentation*; Release 0.01; Python Software Foundation: Wilmington, DE, USA, 2017.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.