Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084267

ARTICLE

A Hybrid Bio-inspired Type-2 Fuzzy Reinforcement Learning Framework for Regional Traffic Signal Coordination Control

Yunrui Bi[1,*], **Qiliang Yang**[1], **Qinglin Ding**[1], **Bin Ran**[2], **Kun Liu**[1] and **Mingjie Zhang**[1]

[1]School of Automation, Nanjing Institute of Technology, Nanjing, China
[2]Department of Civil and Environmental Engineering, University of Wisconsin–Madison, Madison, WI, USA

*Corresponding Author: Yunrui Bi. Email: biyunrui@njit.edu.cn

Received: 19 April 2026; Accepted: 13 July 2026; Published: 13 August 2026

ABSTRACT: To improve regional traffic signal coordination under uncertain and dynamic traffic conditions, this paper proposes a hybrid Type-2 fuzzy reinforcement learning framework integrated with Beetle Antennae Search (BAS) and Deep Q-Network (DQN), named Type-2 fuzzy Beetle Antennae Search and Deep Q-Network (T2-BAS-DQN). In this framework, DQN remains active during online signal control, while the Type-2 fuzzy module provides uncertainty-aware correction for phase selection and green-time adjustment. BAS is used only in the offline training stage to optimize a low-dimensional parameter vector related to fuzzy correction, reward adjustment, and coordination pressure. A 3×3 Simulation of Urban MObility (SUMO) road network is constructed to compare the proposed method with Fixed-Time, Max-Pressure, CoLight, DQN, and related baselines. The results show that T2-BAS-DQN achieves lower queue length, shorter waiting time, and higher average speed under different traffic demand levels. In addition, a real-geometry SUMO simulation calibrated with field traffic counts from Jiangning District, Nanjing, is conducted to further examine its applicability. Compared with DQN in this calibrated scenario, T2-BAS-DQN reduces average queue length by 47.4%, decreases average waiting time by 24.4%, and increases average travel speed by 11.0%. These results indicate that the proposed framework provides competitive simulation performance and improves several traffic operation indicators in the tested scenarios.

KEYWORDS: Regional traffic signals; Type-2 fuzzy control; beetle antennae search algorithm; deep Q network; coordination control

1 Introduction

In recent years, the acceleration of urbanization has intensified the mismatch between traffic supply and demand, making low road network operational efficiency a major constraint on sustainable urban economic growth [1–3]. Optimizing regional traffic signal coordination strategies is of great significance for improving the spatiotemporal utilization of road network resources and reducing travel delays. To achieve intelligent traffic scheduling, artificial intelligence has been widely applied to this field. For instance, Graph Convolutional Networks (GCN) are used to capture the spatiotemporal correlations of traffic flow for precise prediction [4,5]; Deep Reinforcement Learning (DRL) is employed to build adaptive control models that respond to traffic fluctuations in real time [6,7]; and multi-agent coordination mechanisms are utilized to address synchronization challenges in large-scale traffic grids [8,9].

Regarding the handling of high uncertainty in traffic flows, the Type-2 Fuzzy Deep Q-Network (Type-2-FDQN) algorithm has gained significant attention due to its integration of fuzzy logic's capability to represent nonlinear systems with the self-learning characteristics of deep reinforcement learning [10].

By introducing Type-2 fuzzy logic, this method can smooth stochastic traffic fluctuations more effectively than conventional algorithms, demonstrating notable stability in single-intersection scenarios [11]. However, Type-2-FDQN reveals limitations when faced with complex regional multi-intersection networks. First, these methods often rely on independent agent mechanisms and lack cross-spatial coordination, making it difficult to achieve global optimization without a regional perspective [12]. Second, the conventional Deep Q-Network (DQN) architecture may suffer from slow initial convergence and limited exploration efficiency, while purely rule-based fuzzy controllers without online learning or adaptive correction may struggle to maintain long-term robustness in highly dynamic traffic environments.

The motivation for integrating Type-2 fuzzy logic, Beetle Antennae Search (BAS), and DQN comes from the complementary characteristics of the three techniques in regional traffic signal control. DQN provides adaptive decision-making ability by learning signal control policies from real-time traffic states. However, urban traffic states are highly stochastic due to random vehicle arrivals, turning movements, and queue spillback among adjacent intersections, which may lead to unstable Q-value estimation and inefficient exploration. Type-2 fuzzy logic is therefore introduced to explicitly model the uncertainty of traffic states and smooth the influence of traffic fluctuations on control decisions. Meanwhile, the performance of fuzzy-reinforcement learning controllers is sensitive to fuzzy correction weights, reward weights, and coordination-related parameters. BAS is employed as a lightweight offline parameter optimizer to improve parameter adaptability and reduce manual tuning. In this way, the proposed Type-2 fuzzy Beetle Antennae Search and Deep Q-Network (T2-BAS-DQN) framework integrates adaptive learning, uncertainty modeling, and parameter optimization into a unified regional signal coordination controller.

To overcome the aforementioned challenges, this paper proposes a T2-BAS-DQN hybrid framework for regional traffic signal coordination. First, a two-level Type-2 fuzzy control framework, comprising an intersection-level controller and a coordination-level controller, is designed to improve signal timing coordination between adjacent intersections. Second, state and reward coordination mechanisms are established among intersection groups to enhance regional cooperative control. Third, the Beetle Antennae Search (BAS) algorithm is introduced as a lightweight offline parameter optimization method. Instead of claiming universal superiority over other metaheuristic algorithms, BAS is used in this study to tune a low-dimensional parameter vector related to fuzzy correction, reward adjustment, and coordination pressure. BAS is not applied to the full high-dimensional DQN weight space; the DQN network parameters are still trained through temporal-difference learning and gradient-based optimization. Compared with population-based optimization methods such as differential evolution, genetic algorithms, or DNA evolutionary algorithms, BAS has a simple update rule and fewer control parameters, making it convenient for lightweight parameter tuning in the proposed fuzzy-reinforcement learning framework [13–15]. During online signal control, the trained DQN network remains active, while the BAS-optimized parameters and the Type-2 fuzzy module are used to improve the adaptability and robustness of the hybrid controller.

For experimental evaluation, a 3×3 regional road network is first constructed in Simulation of Urban MObility (SUMO) to test the proposed method under controlled traffic demand conditions. In addition, to further examine the applicability of the proposed method under more realistic road topology and turning-movement conditions, a real-geometry SUMO simulation scenario is constructed using field-count traffic demand from Jiangning District, Nanjing. This experiment is not presented as an on-road real-world validation, but as a simulation with real-world geometry and demand calibration. The limitations related to single-day field-count data, missing complete time-period information, and the absence of comparison with deployed signal timing plans are further discussed in the experimental and discussion sections.

The present study is closely related to the authors' earlier work on Type-2 fuzzy traffic signal control. Bi et al. [16] developed a Type-2 fuzzy multi-intersection traffic signal controller in which differential evolution was used to optimize fuzzy-controller parameters. Subsequently, Bi et al. [17] proposed an adaptive Type-2 fuzzy traffic signal control method with an evolutionary optimization mechanism based on a DNA evolutionary algorithm. These studies established an important optimization-based Type-2 fuzzy control lineage for traffic signal control. However, their deployed controllers were mainly fuzzy-control-based systems in which membership functions, rule parameters, or control parameters were optimized by metaheuristic algorithms.

Different from these predecessor studies, the present work does not simply replace differential evolution or DNA evolutionary optimization with BAS. Instead, DQN remains the online signal decision-making module, while BAS is used only as an offline optimizer of a low-dimensional parameter vector associated with fuzzy correction, reward adjustment, and coordination pressure. Therefore, the incremental contribution of this study lies in integrating online DQN-based phase decision-making, Type-2 fuzzy uncertainty-aware correction, BAS-based lightweight parameter tuning, and regional coordination information within a unified traffic signal control framework. The role of BAS is not claimed to be universally superior to differential evolution or DNA evolutionary algorithms; rather, it is adopted here as a simple and low-dimensional tuning mechanism suitable for the proposed hybrid fuzzy-reinforcement learning controller.

2 Regional Traffic Modeling Methods

2.1 Road Network Modeling

During the network modeling stage, the geometric characteristics of each intersection in the target area were first collected through on-site traffic surveys, encompassing key attributes such as the number of inbound approaches, link lengths, and lane configurations [18,19]. The collected data were then converted into a digital road-network model using Netedit, SUMO's visual topology-editing tool. The network model adopts a node–edge topology representation: nodes (Node) encode an intersection's spatial coordinates, channelization scheme, and traffic rules, while connecting edges (Edge) describe link geometry (length, curvature) and operational parameters (number of lanes, speed limits, etc.). As shown in Fig. 1, a 3×3 network was created in Netedit in which each link is a two-way six-lane road (three lanes per direction), yielding a total of 24 two-way, six-lane road segments [20].

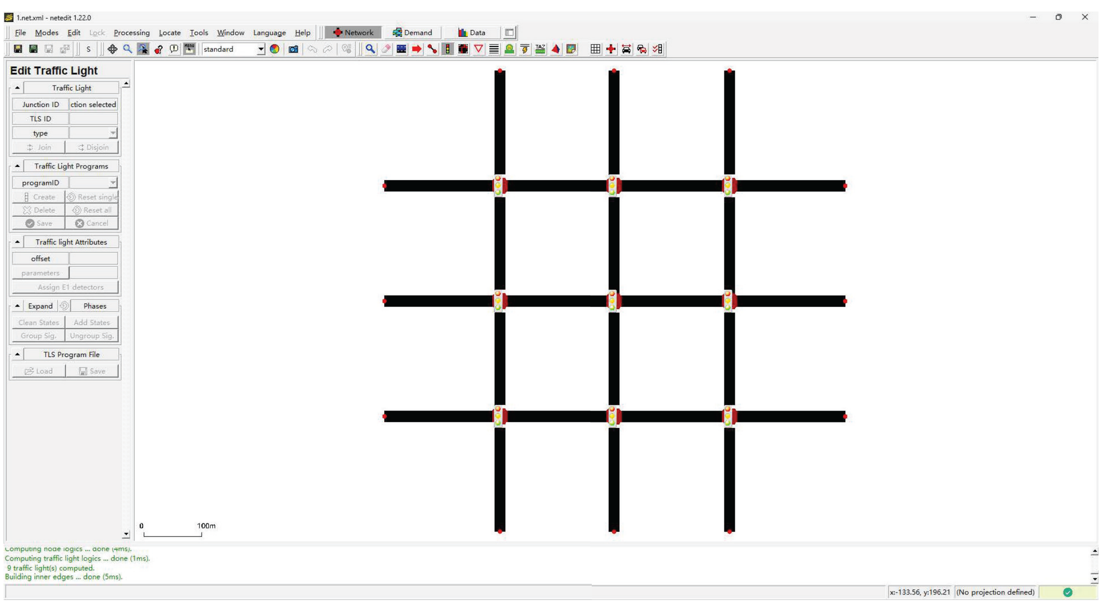


Figure 1: Road network structure diagram.

The network model contains nine signal-controlled nodes, each equipped with an independent traffic signal. The attributes of nodes and edges are listed in Tables 1 and 2, respectively; node attributes include node name ID, coordinates, radius, type, and the traffic-signal control algorithm, while edge attributes include edge name ID, start and end nodes, number of lanes, speed limit, length, etc.

Table 1: Node file attribute information.

Description	Type	Attribute
Node name	String	id
Node X-coordinate	float	x
Node Y-coordinate	float	y
Intersection radius	float	radius
Intersection type	priority, traffic_light, unregulated, allwaystop, rail_signal, . . .	type
Traffic signal control type	static, actuated	tltype
Edges controlled by the node	list of edge ids	controlledInner

Table 2: Edge file attribute information.

Description	Type	Attribute
Border name	String	id
Types of roads	SUMO edge type file	type
Starting node of the edge	Referenced node id	from
Terminal node of the edge	Referenced node id	to
Priority of the road	Right-of-way	priority
Number of lanes	int	numLanes
Vehicles prohibited from entering the road	List of vehicle	disallow
Vehicles allowed on the road	List of vehicle	allow
Road name	String	name
Road length	float	length
Road width	float	width
Speed limit on the road	float	speed
Road coordinate position	List of positions	shape
Sidewalk width	float	sidewalkWidth

The node and edge attributes listed in Tables 1 and 2 are provided only to support the reproducibility of the SUMO network construction. The subsequent analysis focuses on traffic-control-related configurations, including network topology, lane settings, link length, traffic demand, signal phases, and training parameters, which are summarized in the implementation and experimental setup sections.

2.2 Traffic Flow Model

After the generation of the road network model, it is necessary to construct a traffic flow simulation model. SUMO provides two mechanisms for generating traffic flow. To simulate the real-world traffic environment as closely as possible during the simulation process, this study conducts experiments based on the vehicle motion model. By constructing a vehicle configuration file (rou.xml), multi-dimensional parameters for the simulated vehicles are defined, including attributes such as the state and path of each vehicle [21,22].

In the traffic flow model, a route file must be developed to define the vehicles. Right-hand traffic is generally adopted in the vehicle motion model, which is characterized by five parameters: V_{max} denotes the maximum vehicle speed in km/h; b is the maximum deceleration in m/s^2; a is the maximum acceleration in m/s^2; and the vehicle length is measured in m. Vehicle driving behavior is primarily influenced by routing and speed. The routing design is determined by the origin, destination, and intermediate roads, while the speed follows the vehicle motion model, which fully reflects the complexity of traffic flow. In this paper, the calculation formula for safe vehicle travel is shown in Eq. (1):

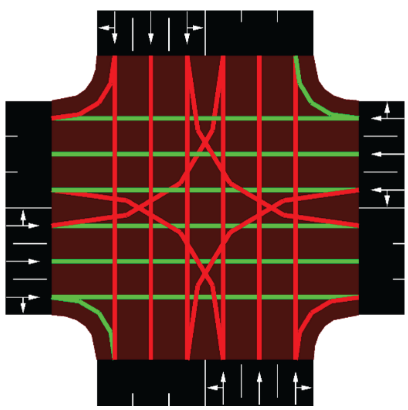
$$v_{\text{safe}} = v_l(t) + \frac{g(t) - v_l(t)\,\tau}{\dfrac{v_l(t) + v_f(t)}{2b} + \tau} \tag{1}$$

where τ is the driver's reaction time (1 s). At time t, the distance between the following vehicle and the leading vehicle is $g(t)$, the speed of the leading vehicle is $v_l(t)$, and the speed of the following vehicle is $v_f(t)$. The corresponding safe-speed calculation used in the vehicle motion model is given by Eq. (2):

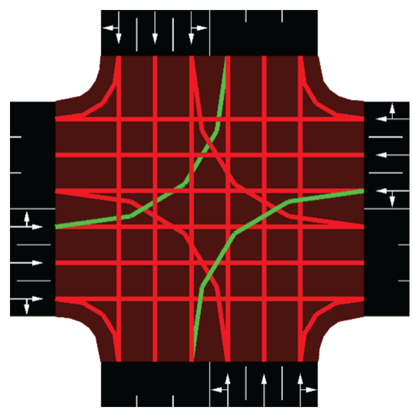
$$V_s = -\tau\, b + \sqrt{(\tau\, b)^2 + v_t^2 + 2b\, g(t)} \tag{2}$$

2.3 Traffic Light Model

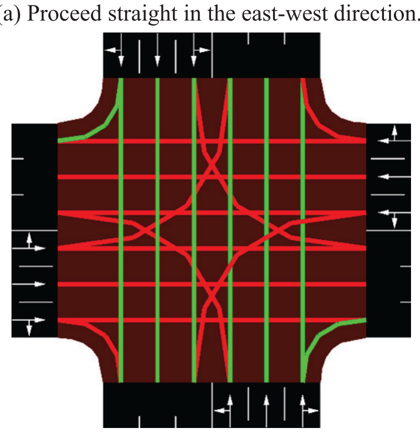
In the traffic light model, this paper defines 4 phases based on the bidirectional six-lane roads established in the aforementioned road network, as shown in Fig. 2.



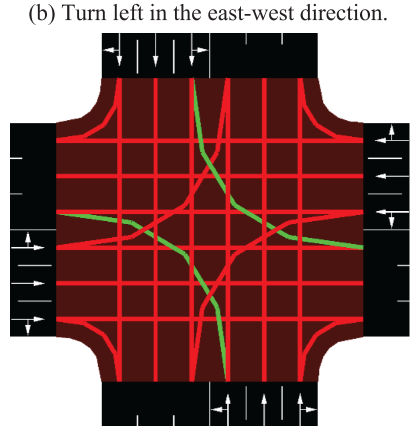
(a) Proceed straight in the east-west direction.



(b) Turn left in the east-west direction.



(c) Proceed straight in the north-south direction.



(d) Turn left in the north-south direction.

Figure 2: Traffic light model.

The model adopts a right-hand driving logic. All four-way intersections are configured with four phases: East-West through (including right turns), East-West left turns, North-South through (including right turns), and North-South left turns. In the traffic light model definition, states are represented by four parameters: *G*, *g*, *r*, and *y*. The traffic light phases can be represented as `'GGGGgrrrrrGGGGgrrrrr'`, `'yyyyyrrrrryyyyyrrrrr'`, `'rrrrrGGGGgrrrrrGGGGg'` and `'rrrrryyyyyrrrrryyyyy'`.

Taking `'GGGGgrrrrrGGGGgrrrrr'` as an example, the phase state information represents the status of the intersection approaches starting from the North (12 o'clock direction) and proceeding clockwise through East, South, and West. Every 5 characters form a group, representing the traffic light phase for each specific road [23,24].

3 Control Method Design

3.1 Design of Type-2 Fuzzy Controller (Intersection Level and Coordination Level)

To address regional traffic coordination requirements, this paper proposes a two-level Type-2 fuzzy control framework—an intersection-level controller and a coordination-level controller—as illustrated in Fig. 3.

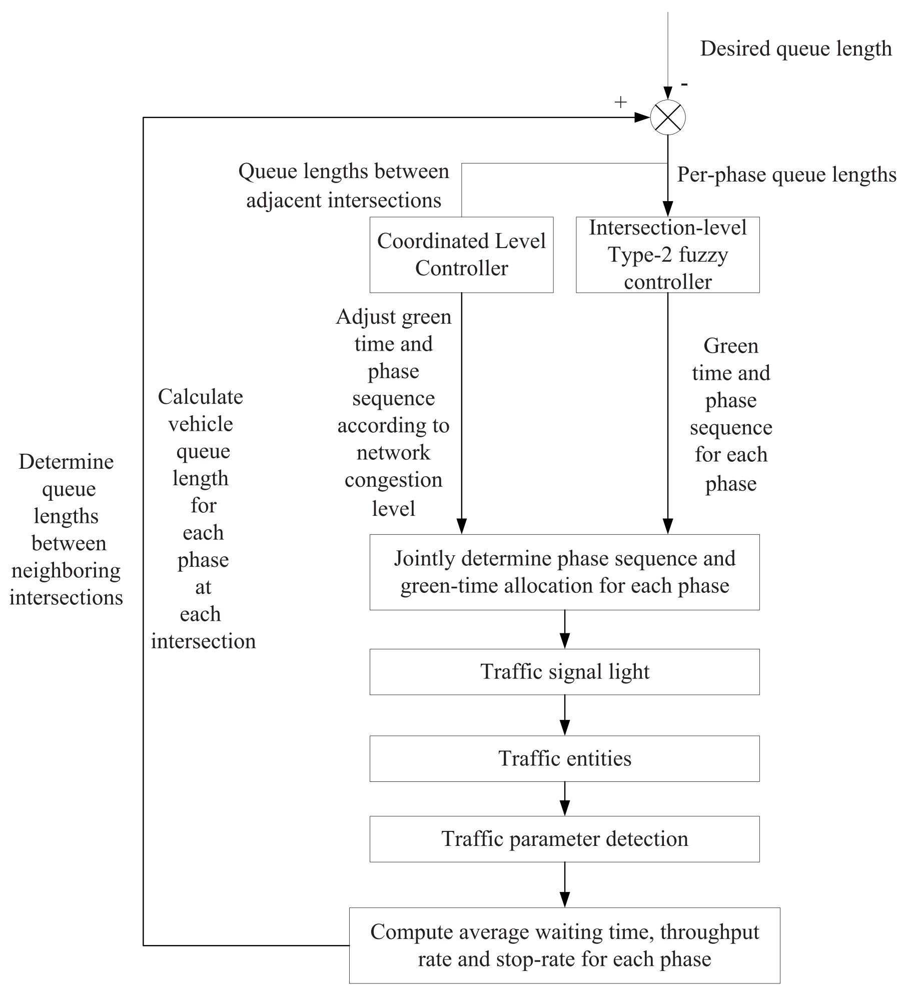


Figure 3: Regional traffic intersection Type-2 fuzzy control structure diagram.

It should be noted that Fig. 3 is used to illustrate the information flow between the intersection-level controller and the coordination-level controller, rather than a classical error-feedback controller with respect to a fixed desired queue length. In the actual rule design, the intersection-level controller determines phase priority based on absolute queue length, waiting time, current phase, and maximum red-time constraints. The coordination-level controller further adjusts the priority according to neighboring-link queue occupancy and spillback risk.

At the intersection-level controller, each intersection agent autonomously optimizes phase timing based on local microscopic traffic-flow parameters [25,26]. Specifically, the queue length and waiting time of vehicles for the current green phase are chosen as the key decision inputs; by comparing the queue lengths and red-phase waiting times across phases, the control variables L_1 and L_2 are determined. The algorithm proceeds as follows: first check whether any phase has reached the preset maximum red time. If such a phase exists, set L_1 to the queue length of that phase, set L_2 to the maximum queue length among the remaining phases, and allocate T as the green time for the L_1 phase. If no phase has reached the maximum red time, let L_1 and L_2 be the largest and second-largest queue lengths across all phases, respectively, and set T to the green time required by the largest queue. In this way, when a phase's demand is high, its green time can be extended or the controller can switch to a phase with greater demand [27].

Compared with conventional fuzzy logic, Type-2 fuzzy logic better handles traffic-flow uncertainty. This study employs Type-2 Gaussian membership functions with uncertain deviation; the three fuzzy subsets are defined as Short (S), Medium (M), and Long (L). Because the allowable green-time ranges differ between through and left-turn phases, when T corresponds to a straight-through phase the output T has the domain $[9, 60]$, whereas when T corresponds to a left-turn phase T has the domain $[9, 40]$.

The coordination-level controller accounts for interactions among neighboring intersections to achieve inter-intersection coordination. When congestion or spillback risk occurs between adjacent intersections, the controller adjusts signal priorities to reduce queue propagation and prevent local bottlenecks. The storage capacity of an inter-intersection link is estimated as

$$C_{ij} = \left\lfloor \frac{L_{ij} n_{ij}}{h} \right\rfloor, \tag{3}$$

where L_{ij} is the link length, n_{ij} is the number of lanes in the considered direction, and h is the average vehicle spacing. For the synthetic network in this study, the average link length is approximately 400 m and each direction contains three lanes. If the average vehicle spacing is set to 8 m, the per-lane storage capacity is about 50 vehicles, and the per-direction storage capacity is about 150 vehicles. When the observed queue length approaches this capacity, the coordination-level controller is activated to adjust the phase priority of adjacent intersections [28,29].

Based on the estimated link storage capacity and the maximum red-time constraint, the detailed coordination procedure is as follows:

1. Check whether intersection 1's north–south through phase or east–west left-turn phase has reached the maximum red time. If yes, go to step 2; otherwise go to step 3.
2. Set L_1 to the queue length of the phase that has reached the maximum red time, and set L_2 to the largest queue length among the remaining phases. In this case T is the green time to be allocated to the phase (either the north–south through or the east–west left-turn) that reached the maximum red waiting time.
3. Otherwise, set L_1 to the maximum queue length among intersection 1's north–south through and east–west left-turn phases, and set L_2 to the maximum queue length among the other three phases; T is then the green time allocated to the phase corresponding to L_1.

Similarly, if comparable congestion emerges at other intersections, the coordination controller will also be activated, prioritizing coordination beginning from central intersections of the region to prevent critical nodes from blocking. Especially during morning and evening peak periods, by ensuring key intersections remain unobstructed first and then coordinating surrounding intersections, the regional green-time allocation becomes more rational and large-scale gridlock can be avoided.

3.2 T2-BAS-DQN Fusion Mechanism for Regional Traffic Signal Control

This section describes the fusion mechanism of the proposed T2-BAS-DQN framework for regional traffic signal control. In this mechanism, DQN is responsible for estimating the action values of candidate signal phases, the Type-2 fuzzy module provides uncertainty-aware correction based on real-time traffic states, BAS optimizes a low-dimensional parameter vector during the training stage, and the regional coordination module introduces neighboring traffic pressure information to support coordinated phase selection.

To improve the adaptability and robustness of regional traffic signal coordination, this study proposes a T2-BAS-DQN framework by integrating DQN-based adaptive decision-making, Type-2 fuzzy uncertainty modeling, and BAS-based parameter optimization. It should be emphasized that the DQN remains active during online signal control. BAS is not used as an online controller, but is employed in the offline parameter-optimization stage after DQN training to optimize key parameters of the fuzzy-reinforcement learning controller. During BAS optimization, the trained DQN parameters are fixed, and BAS only evaluates candidate parameter vectors through validation rollouts. During online deployment, the trained DQN network outputs the Q-values of candidate signal phases, while the Type-2 fuzzy module provides uncertainty-aware correction for phase preference and green-time adjustment. The final control decision is obtained by fusing the DQN output, the fuzzy correction result, and the coordination pressure information.

The overall workflow of the proposed framework is shown in Fig. 4. The framework consists of two parts: the left part describes the online signal control process, and the right dashed box describes the offline DQN training and BAS-based parameter optimization process. In the online control process, the traffic state of the road network is first inspected, including queue length, waiting time, current phase, and neighboring-intersection queue information. These traffic state variables are used as the input of the DQN network and the Type-2 fuzzy module. The DQN network evaluates the Q-values of different candidate phases, while the Type-2 fuzzy module calculates the uncertainty correction term according to traffic fluctuation and congestion conditions.

In the offline optimization process, BAS is used to optimize a low-dimensional parameter vector rather than directly searching the full high-dimensional DQN weight space. This design avoids the problem that a four-direction BAS variant may be ineffective in a high-dimensional neural-network parameter space. In the proposed framework, the DQN parameters θ are trained through the temporal-difference loss and gradient-based optimization, whereas BAS is only responsible for tuning several key fuzzy-reinforcement learning parameters.

The optimized parameter vector is defined as

$$\Theta = [\omega_f, \omega_r, \omega_p], \tag{4}$$

where ω_f denotes the fuzzy correction weight, ω_r denotes the fuzzy reward adjustment weight, and ω_p denotes the coordination pressure weight. These parameters are used to adjust the contribution of the Type-2 fuzzy module, reward feedback, and regional coordination information in the final decision-making process. Since the dimension of Θ is small, the BAS search process is computationally tractable and does not suffer from the high-dimensional search problem associated with direct DQN weight optimization.

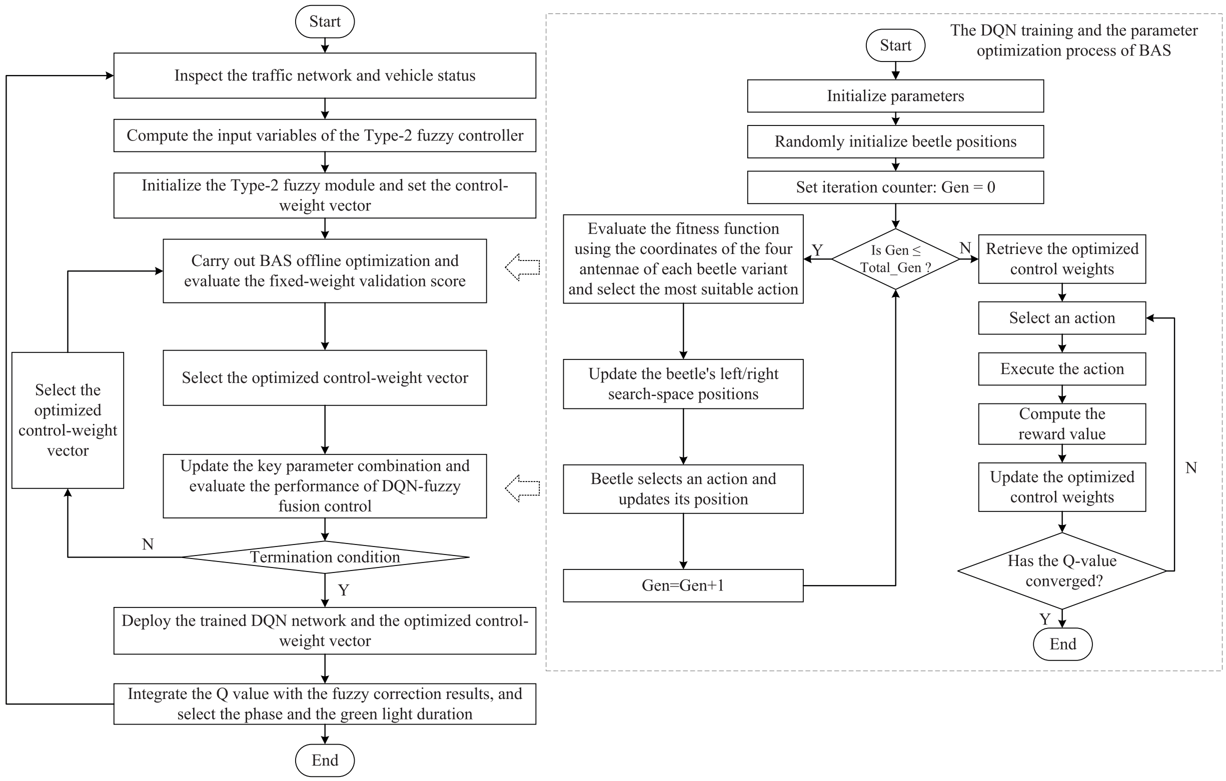


Figure 4: Offline training and online deployment process of the proposed T2-BAS-DQN framework.

In the offline parameter-optimization process, the DQN is first trained through temporal-difference learning, and the trained DQN parameters are denoted as θ^*. After the DQN training stage, BAS is used to optimize the low-dimensional parameter vector Θ through validation rollouts. For each candidate parameter vector Θ, the DQN parameters θ^* are fixed. The controller interacts with the SUMO traffic environment through the TraCI interface, observes the traffic state s_t, evaluates the candidate action values $Q(s_t, a; \theta^*)$, calculates the Type-2 fuzzy correction term and the coordination pressure priority, and then selects the signal action according to the fused action score. The fixed-weight validation score is calculated to evaluate the fitness value of the candidate parameter vector. Therefore, BAS does not retrain the DQN network for each candidate Θ; it only evaluates different parameter vectors using the fixed trained DQN policy.

$$L(\theta) = \mathbb{E}\left[\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta)\right)^2\right], \tag{5}$$

where θ and θ^- denote the parameters of the online network and target network, respectively, and γ is the discount factor.

The fitness function of BAS is defined according to the traffic performance obtained in validation rollouts rather than the raw cumulative reward containing the tunable coefficient ω_r. For a given parameter vector Θ, the validation fitness over M validation episodes is calculated as

$$F(\Theta) = -\frac{1}{M}\sum_{m=1}^{M}\left(\xi_Q \overline{Q} * m(\Theta) + \xi_W \overline{W} * m(\Theta) + \xi_S \overline{S} * m(\Theta) + \xi_H \overline{H} * m(\Theta)\right), \tag{6}$$

where $\overline{Q} * m(\Theta)$, $\overline{W} * m(\Theta)$, $\overline{S} * m(\Theta)$, and $\overline{H} * m(\Theta)$ denote the average queue length, average waiting time, number of stops, and phase-switching penalty obtained in the m-th validation episode under parameter vector Θ, respectively. The coefficients ξ_Q, ξ_W, ξ_S, and ξ_H are fixed validation weights and are not optimized by BAS. A larger value of $F(\Theta)$ indicates better validation performance because lower queue length, waiting time, stops, and switching penalty lead to a higher fitness value. During BAS optimization, the beetle position represents a candidate parameter vector. The fitness values of the left and right antennae are evaluated, and the beetle position is updated toward the direction with a higher fixed-weight validation score. This process is repeated until the termination condition is satisfied, and the optimal parameter vector Θ^* is obtained.

It should be noted that ω_r is used as a reward-shaping coefficient in the DQN reward formulation, but it is not treated as an independent additive bonus in the BAS fitness evaluation. Therefore, moving ω_r toward its boundary cannot improve the BAS fitness by itself. A candidate value of ω_r is retained only when the corresponding parameter vector leads to improved traffic performance in validation rollouts.

During online deployment, BAS is no longer executed. The trained DQN network parameters θ^* and the optimized control-weight vector Θ^* are deployed together for real-time signal control. For each decision step, the DQN network outputs the action-value function $Q(s_t, a; \theta^*)$ for each candidate phase. Meanwhile, the Type-2 fuzzy module calculates the fuzzy correction term $\Phi_t(a)$, and the coordination module calculates the pressure-related priority $P_t(a)$. The fused action score is defined as

$$\widetilde{Q}(s_t, a) = Q(s_t, a; \theta^*) + \omega_f \Phi_t(a) + \omega_p P_t(a), \tag{7}$$

where $\Phi_t(a)$ represents the uncertainty-aware fuzzy correction for action a, and $P_t(a)$ represents the regional coordination or pressure priority. The final signal action is selected by

$$a_t = \arg\max_a \widetilde{Q}(s_t, a). \tag{8}$$

After the action is selected, the corresponding phase and green time are executed in the traffic environment. Therefore, the proposed framework does not deploy a static Type-2 fuzzy controller alone. Instead, the online controller consists of the trained DQN network, the optimized control-weight vector Θ^*, and the coordination pressure module. This structure allows the controller to maintain the adaptive learning ability of DQN while improving robustness through Type-2 fuzzy uncertainty modeling and BAS-based parameter optimization.

The detailed procedure of the proposed T2-BAS-DQN framework is summarized as follows.

Step 1: Inspect the traffic network and collect traffic state variables, including queue length, waiting time, current signal phase, and neighboring-intersection queue information.

Step 2: The Type-2 fuzzy module is initialized, and the three scalar control weights ω_f, ω_r, and ω_p are encoded as the low-dimensional parameter vector to be optimized.

Step 3: Train the DQN controller through interaction with the SUMO environment. The DQN observes the current traffic state, outputs the Q-values of candidate signal phases, selects and executes an action, receives the reward, and updates the Q-network.

Step 4: Perform BAS-based offline parameter optimization. BAS evaluates the fixed-weight validation fitness of different parameter combinations and updates the parameter vector according to the fitness function.

Step 5: Deploy the trained DQN network and the optimized control-weight vector Θ^*. During online control, the DQN outputs Q-values, the Type-2 fuzzy module provides uncertainty correction, and the coordination module provides pressure priority.

Step 6: Fuse the Q-values, fuzzy correction results, and coordination pressure information to select the optimal phase and green time. The selected signal control action is then executed in the traffic environment.

3.3 Rationale for the Hybrid Integration of Type-2 Fuzzy Logic, BAS and DQN

The proposed T2-BAS-DQN framework is not a simple stacking of three independent techniques. Instead, Type-2 fuzzy logic, BAS and DQN are introduced to address three different but closely related challenges in regional traffic signal coordination, namely adaptive decision-making, uncertainty representation, and parameter optimization.

First, DQN is adopted as the basic adaptive decision-making module. In regional traffic signal control, the traffic state changes dynamically due to stochastic vehicle arrivals, turning movements, queue dissipation, and spillback propagation among adjacent intersections. Traditional fixed-time control cannot respond to these real-time variations, while purely rule-based methods have limited adaptability under complex demand fluctuations. DQN can approximate the action-value function and learn signal phase selection strategies through interactions with the traffic environment. Therefore, DQN provides the basic learning ability for adaptive signal control.

Second, Type-2 fuzzy logic is introduced to enhance the robustness of the DQN-based controller under uncertain traffic conditions. In practical traffic networks, queue length, waiting time, and downstream congestion are not deterministic variables. They are affected by stochastic vehicle arrivals, lane-changing behavior, detector noise, and spillback effects between neighboring intersections. Compared with Type-1 fuzzy logic, Type-2 fuzzy logic uses a footprint of uncertainty to describe the uncertainty of membership grades, making it more suitable for representing ambiguous and fluctuating traffic states. Therefore, Type-2 fuzzy logic is used to smooth the influence of traffic-state uncertainty on control decisions and improve the stability of the controller.

Third, BAS is introduced as a lightweight parameter optimization module. The performance of a fuzzy-reinforcement learning controller is sensitive to several key parameters, such as fuzzy control weights, reward weights, pressure-related weights, and coordination parameters. Manually tuning these parameters is inefficient and may lead to suboptimal performance under different traffic demand levels. BAS provides a simple search mechanism with few control parameters and can be used to optimize a low-dimensional parameter vector during the offline training stage. In this study, BAS is not used to search the full high-dimensional DQN weight space. Instead, it is applied to the optimization of key fuzzy-reinforcement learning parameters, which makes the search process computationally tractable.

It should also be clarified that the use of BAS in this study is different from predecessor optimization strategies such as differential evolution and DNA evolutionary algorithms used in related fuzzy-control studies. Differential evolution and DNA evolutionary algorithms are usually population-based methods, which can provide stronger global exploration ability but may require more candidate solutions, more hyperparameters, and higher fitness-evaluation cost. In contrast, BAS uses a simpler directional search mechanism and is suitable for lightweight tuning when the number of optimized parameters is small. Since the proposed framework only optimizes the low-dimensional parameter vector $\Theta = [\omega_f, \omega_r, \omega_p]$, BAS is selected as a practical optimizer to reduce manual tuning and keep the offline optimization process computationally manageable.

In this study, BAS is selected as a lightweight design choice for low-dimensional parameter tuning rather than as an empirically proven superior optimizer. The optimized vector only contains three scalar weights, namely $\Theta = [\omega_f, \omega_r, \omega_p]$. Compared with population-based optimizers such as differential evolution and DNA evolutionary algorithms, BAS has a simple directional search mechanism and fewer algorithmic control parameters, which makes it convenient for the proposed lightweight offline tuning setting. Therefore, the use of BAS should be understood as a design rationale for simple low-dimensional tuning, rather than evidence that BAS generally outperforms differential evolution, DNA evolutionary algorithms, PSO, or genetic algorithms.

However, this study does not claim that BAS is theoretically or empirically superior to differential evolution, DNA evolutionary algorithms, PSO, or genetic algorithms in all scenarios. A complete comparison among different metaheuristic optimizers under the same computational budget is beyond the scope of this paper. The incremental contribution of this study lies in integrating DQN-based online decision-making, Type-2 fuzzy uncertainty modeling, BAS-based lightweight parameter tuning, and regional coordination into a unified traffic signal control framework.

As shown in Table 3, the three components play complementary roles in the proposed framework. DQN learns adaptive signal control policies, Type-2 fuzzy logic improves the representation of uncertain traffic states, and BAS optimizes key parameters to enhance adaptability. The proposed T2-BAS-DQN can thus be regarded as a hierarchical hybrid framework that integrates learning-based decision-making, fuzzy uncertainty modeling, and bio-inspired parameter optimization for regional traffic signal coordination.

Table 3: Roles of different components in the proposed T2-BAS-DQN framework.

Component	Main Function	Problem Addressed
DQN	Learns adaptive signal control policies from real-time traffic states	Limited adaptability of fixed-time and purely rule-based signal control methods
Type-2 fuzzy logic	Models uncertain traffic states and smooths control decisions	Random traffic fluctuations, detector uncertainty, and unstable traffic-state evaluation
BAS	Optimizes key fuzzy-reinforcement learning parameters	Manual parameter tuning and limited parameter adaptability under different demand levels
Coordination mechanism	Exchanges queue and pressure information among neighboring intersections	Lack of cooperation and spillback awareness among independent intersection controllers

3.4 Implementation Details and Training Procedure

To improve the reproducibility of the proposed T2-BAS-DQN framework, this subsection provides detailed implementation settings, including agent configuration, state representation, action space, reward function, DQN network architecture, training configuration, BAS optimization parameters, convergence criterion, and computational complexity. These details are added to make the training and evaluation procedure clear and reproducible.

In the proposed regional traffic signal control framework, each signalized intersection is modeled as one DQN agent. Therefore, for the 3×3 synthetic road network, the number of agents is 9. Each agent observes its local traffic state and neighboring-intersection coordination information. The agents share the same network architecture and training hyperparameters, while each agent selects actions according to its own local state. Neighboring queue and pressure information is exchanged to support regional coordination.

At each decision step t, the state of intersection i is defined as

$$s_t^i = \left[Q_t^i, W_t^i, P_t^i, G_t^i, Q_{nei,t}^i\right], \tag{9}$$

where Q_t^i denotes the queue length of each signal phase at intersection i, W_t^i denotes the accumulated waiting time of each phase, P_t^i denotes the current signal phase, G_t^i denotes the elapsed green time of the current phase, and $Q_{nei,t}^i$ denotes the queue information from neighboring intersections. This state representation enables each agent to consider both local traffic conditions and regional congestion propagation.

The action space is defined according to the four-phase signal control structure used in the simulation network. Each action corresponds to selecting or maintaining one of the four candidate signal phases:

$$\mathcal{A} = \{a_1, a_2, a_3, a_4\}, \tag{10}$$

where a_1 denotes the east–west through phase, a_2 denotes the east–west left-turn phase, a_3 denotes the north–south through phase, and a_4 denotes the north–south left-turn phase. To avoid frequent phase switching and ensure traffic safety, the selected action is constrained by the minimum green time and yellow transition time. In the implementation, the selected action can be interpreted as either maintaining the current phase or switching to another candidate phase when the minimum green-time constraint is satisfied.

The reward function is designed to reduce queue accumulation, vehicle waiting time, stop frequency, and unnecessary phase switching. To explicitly incorporate the reward adjustment weight optimized by BAS, for intersection i, the local reward at decision step t is defined as

$$r_t^i(\Theta) = -\left(\alpha Q_t^i + \beta W_t^i + \eta S_t^i + \lambda H_t^i + \omega_r C_t^i\right), \tag{11}$$

where Q_t^i is the total queue length, W_t^i is the total waiting time, S_t^i is the number of vehicle stops, H_t^i is the phase-switching penalty, and C_t^i denotes the congestion degree estimated by the Type-2 fuzzy module. The coefficients α, β, η, and λ are weighting parameters used to balance different optimization objectives. The term $\omega_r C_t^i$ represents the fuzzy reward adjustment term, where ω_r is optimized by BAS to tune the influence of the uncertainty-aware congestion evaluation on the reinforcement learning reward. This term is used for reward shaping during DQN training. It is not directly used as an independent additive term in the BAS fitness evaluation.

The regional reward is obtained by averaging the local rewards of all signalized intersections:

$$R_t(\Theta) = \frac{1}{N}\sum_{i=1}^{N} r_t^i(\Theta), \tag{12}$$

where N is the number of controlled intersections. In this study, $N = 9$ for the 3×3 synthetic road network. Therefore, the optimized parameter vector $\Theta = [\omega_f, \omega_r, \omega_p]$ affects the hybrid controller in two ways: ω_r is used as a reward-shaping coefficient during DQN training, while ω_f and ω_p affect the fused action score during online phase selection. During BAS-based validation, candidate parameter vectors are evaluated using the fixed-weight traffic-performance score defined in Eq. (15), rather than an independent additive reward term containing ω_r.

The DQN is used to approximate the action-value function $Q(s, a; \theta)$, where θ denotes the parameters of the online Q-network. A target network with parameters θ^- is used to calculate the target Q-value and improve the stability of training. Since the scalar reward used in the DQN update is the regional reward $R_t(\Theta)$, the loss function of DQN is defined as

$$L(\theta;\Theta) = \mathbb{E}\left[\left(R_t(\Theta) + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta)\right)^2\right], \tag{13}$$

where γ is the discount factor, s_t and s_{t+1} are the current and next states, a_t is the selected action, and a' denotes the candidate action at the next state. In this formulation, ω_r participates in the DQN learning process through the reward signal $R_t(\Theta)$. During BAS-based validation, however, the candidate parameter vector is evaluated using the fixed-weight traffic-performance score rather than the raw cumulative reward containing an independent additive $\omega_r C_t^i$ term.

The detailed DQN architecture used in this study is listed in Table 4. The input dimension depends on the number of state variables used for each intersection agent, while the output dimension is equal to the number of candidate signal phases.

Table 4: DQN network architecture.

Item	Setting
Input variables	Queue length of each phase, waiting time of each phase, current phase, elapsed green time, and neighboring-intersection queue information
Input dimension	Dimension of s_t^i defined in Eq. (9)
Network type	Fully connected neural network
Number of hidden layers	2
Number of neurons per hidden layer	64
Activation function	ReLU
Output dimension	4 candidate signal phases
Output meaning	Action-value estimates $Q(s_t^i, a; \theta)$ for candidate signal phases
Optimizer	Adam
Loss function	Mean squared temporal-difference error

The training configuration is summarized in Table 5. The simulation is implemented in SUMO through the Traffic Control Interface (TraCI)-Python interface. The car-following model adopts the default Krauss model in SUMO unless otherwise specified. All learning-based methods are trained and evaluated under the same simulation horizon, decision interval, signal phase constraints, and random seeds to ensure fair comparison.

Table 5: Training and simulation configuration.

Parameter	Value
Simulation platform	SUMO with TraCI-Python interface
SUMO version	SUMO 1.21.0
Car-following model	Krauss model
Number of agents	9 DQN agents for the 3 × 3 network
Agent setting	One agent per signalized intersection with neighboring-information exchange
Training episodes	1000
Simulation horizon	2000 steps

(Continued)

Table 5 (continued)

Parameter	Value
Decision interval	5 s
Replay buffer size	10,000
Batch size	64
Learning rate	0.001
Discount factor γ	0.99
Target network update interval	100 steps
Initial exploration rate ε	1.0
Minimum exploration rate $\varepsilon_{\min}$	0.05
Exploration decay rate	0.995
Minimum green time	9 s
Maximum green time for through phases	60 s
Maximum green time for left-turn phases	40 s
Yellow time	3 s
Random seeds	1–10

In the proposed framework, BAS is used during offline training to optimize a low-dimensional parameter vector related to fuzzy correction, reward adjustment, and coordination pressure. The optimized parameter vector is defined as

$$\Theta = \left[\omega_f, \omega_r, \omega_p\right], \tag{14}$$

where ω_f denotes the fuzzy correction weight, ω_r denotes the fuzzy reward adjustment weight, and ω_p denotes the coordination pressure weight. BAS is not used to search the full high-dimensional DQN weight space. Instead, it optimizes the key parameters of the fuzzy-reinforcement learning controller, which reduces computational cost and improves parameter adaptability.

The fitness function of BAS is defined according to the traffic performance obtained in validation rollouts rather than the raw cumulative reward containing the tunable coefficient ω_r. For a given parameter vector Θ, the validation fitness over M validation episodes is calculated as

$$F(\Theta) = -\frac{1}{M}\sum_{m=1}^{M}\left(\xi_Q \overline{Q} * m(\Theta) + \xi_W \overline{W} * m(\Theta) + \xi_S \overline{S} * m(\Theta) + \xi_H \overline{H} * m(\Theta)\right), \tag{15}$$

where M is the number of validation episodes. $\overline{Q} * m(\Theta)$, $\overline{W} * m(\Theta)$, $\overline{S} * m(\Theta)$, and $\overline{H} * m(\Theta)$ denote the average queue length, average waiting time, number of stops, and phase-switching penalty obtained in the m-th validation episode under parameter vector Θ, respectively. The coefficients ξ_Q, ξ_W, ξ_S, and ξ_H are fixed validation weights and are not optimized by BAS. A larger value of $F(\Theta)$ indicates better validation performance because lower queue length, waiting time, stops, and switching penalty lead to a higher fitness value.

It should be noted that ω_r is used as a reward-shaping coefficient in the DQN reward formulation, but it is not treated as an independent additive bonus in the BAS fitness evaluation. Therefore, moving ω_r toward its boundary cannot improve the BAS fitness by itself. A candidate value of ω_r is retained only when the corresponding parameter vector leads to improved traffic performance in validation rollouts. The detailed BAS parameter settings are listed in Table 6.

Table 6: BAS optimization parameters.

Parameter	Value
Optimized parameter vector	$\Theta = [\omega_f, \omega_r, \omega_p]$
Dimension of search space	3
Number of beetles/candidate solutions	1 beetle with antenna-based directional search
Maximum number of BAS iterations	50
Initial antenna length d_0	1.0
Initial step size δ_0	0.5
Antenna length decay factor	0.95
Step size decay factor	0.95
Parameter boundary handling	Clipping to predefined bounds
Fitness evaluation	Fixed-weight validation score based on queue length, waiting time, stops, and phase-switching penalty
Number of validation episodes for each fitness evaluation	1–3
Optimization objective	Maximize $F(\Theta)$
Termination condition	Maximum iterations or fitness convergence

The complete training procedure consists of five stages. First, the SUMO network, route files, signal phases, vehicle detectors, and neighboring-intersection relationships are initialized. Second, each DQN agent observes the traffic state from the SUMO environment through the TraCI interface and selects signal actions according to the ε-greedy strategy during training. Third, the transition samples are stored in the replay buffer, and the DQN parameters are updated using mini-batch training. The target network is synchronized with the online network at a fixed interval. Fourth, BAS evaluates candidate parameter vectors according to the fixed-weight validation fitness and updates the parameter vector Θ. Fifth, after training and optimization are completed, the trained DQN network and the optimized control-weight vector Θ^* are deployed for online signal control.

The convergence of training is evaluated using the moving average cumulative reward. Let $\overline{R}_k$ denote the average reward over the latest K_r episodes. The training process is regarded as converged when

$$\left|\overline{R}_k - \overline{R}_{k-K_r}\right| < \varepsilon_c, \tag{16}$$

where ε_c is the convergence threshold. If the convergence condition is not satisfied, the training process stops when the maximum number of training episodes is reached.

The complete training procedure of the proposed T2-BAS-DQN framework is summarized in Algorithm 1. It should be noted that BAS only optimizes the low-dimensional parameter vector $\Theta = [\omega_f, \omega_r, \omega_p]$, while the DQN network parameters θ are updated by the temporal-difference loss.

Algorithm 1: Training and offline optimization procedure of T2-BAS-DQN

Input: SUMO environment, Type-2 fuzzy module, DQN hyperparameters, BAS parameters
Output: Trained DQN parameters θ^* and optimized parameter vector Θ^*

Initialize the SUMO environment, signal phases, detectors, DQN networks, replay buffer $\mathcal{D}$, and parameter vector $\Theta = [\omega_f, \omega_r, \omega_p]$.
Stage 1: DQN training
for each training episode **do**
    Observe traffic state s_t, calculate $Q(s_t, a; \theta)$, Type-2 fuzzy correction $\Phi_t(a)$, and pressure priority $P_t(a)$.
    Select and execute action a_t using the fused action score and the ε-greedy strategy.
    Obtain reward $r_t(\Theta)$ and next state s_{t+1}, and store transition $(s_t, a_t, r_t(\Theta), s_{t+1})$ in $\mathcal{D}$.
    Update θ by minimizing the temporal-difference loss and periodically update the target network θ^-.
end for
Obtain the trained DQN parameters θ^* and freeze them.
Stage 2: BAS-based offline parameter optimization
for each BAS iteration **do**
    Generate candidate parameter vectors Θ using BAS.
    Evaluate each candidate Θ through validation rollouts with fixed DQN parameters θ^*.
    Calculate the fitness value $F(\Theta)$ using the fixed-weight validation score defined in Eq. (15).
    Update the BAS position according to the fitness value.
end for
Obtain the optimized parameter vector Θ^*.
Stage 3: Online deployment
Deploy θ^* and Θ^* for online signal control.
During online deployment, BAS is no longer executed; the action is selected by
$a_t = \arg\max_a \left[Q(s_t, a; \theta^*) + \omega_f^* \Phi_t(a) + \omega_p^* P_t(a) \right]$.

It should be noted that the DQN network is trained once in Stage 1. During the BAS optimization stage, the DQN parameters θ^* are fixed, and BAS only evaluates different low-dimensional parameter vectors Θ through validation rollouts. Therefore, the DQN training cost and the BAS validation cost are additive rather than multiplicative, which is consistent with the computational complexity analysis presented later.

3.4.1 Computational Complexity

The computational cost of the proposed framework mainly consists of DQN training and BAS-based parameter optimization. For DQN training, the main computational cost comes from the forward and backward propagation of the neural network. If the number of training episodes is denoted as E, the number of decision steps in each episode is denoted as T, the batch size is denoted as B, and the number of DQN network parameters is denoted as $|\theta|$, the approximate training complexity can be expressed as

$$O(ETB|\theta|). \tag{17}$$

For BAS optimization, suppose that the number of BAS iterations is K and each fitness evaluation runs M validation episodes. The additional offline optimization complexity can be approximately expressed as

$$O(KMT). \tag{18}$$

Therefore, the total offline training complexity of the proposed framework can be written as

$$O(ETB|\theta| + KMT). \tag{19}$$

It should be noted that BAS optimization is performed in the offline parameter-optimization stage after DQN training. During this stage, the trained DQN parameters are fixed, and BAS only evaluates different low-dimensional parameter vectors through validation rollouts. During online deployment, BAS is no longer executed; the controller only needs to perform traffic state observation, Type-2 fuzzy inference, one forward pass of the trained DQN network, and fused action selection. Therefore, the online computational cost remains acceptable for real-time traffic signal control.

This implementation is consistent with Algorithm 1: the DQN is trained once, and BAS is subsequently performed using validation rollouts with fixed DQN parameters. Therefore, the total offline complexity is additive rather than multiplicative and can be expressed as Eqs. (17)–(19).

3.4.2 Reproducibility Statement

To further support reproducibility, the main methodological elements required for implementation have been summarized in Tables 4–6, including the state representation, action space, reward function, DQN architecture, training hyperparameters, BAS optimization parameters, number of agents, random seeds, and simulation settings. At the time of submission, the SUMO network files, route files, additional files, controller scripts, and configuration files have not yet been deposited in a public repository. These materials can be provided by the corresponding author upon reasonable request for academic verification. Exact numerical reproduction of the reported results requires these materials, including the SUMO network, route files, random seeds, controller implementation, hyperparameter configuration, and evaluation scripts. A public repository with anonymized simulation artifacts will be considered after publication or after institutional approval.

4 Experiment and Result Analysis

4.1 Experimental Setup and Compared Methods

To evaluate the performance of the proposed T2-BAS-DQN algorithm for regional multi-intersection traffic signal coordination, a simulation environment was constructed in the SUMO platform, and the interaction between the control algorithm and the traffic simulation environment was implemented through the TraCI-Python interface. The simulation scenario is based on a 3 × 3 regional road network, as shown in Fig. 5. The network contains 9 signalized intersections and 24 road sections. Each road section contains three lanes, and the average segment length is approximately 400 m. The vehicle maximum speed is set to 60 km/h.

The turning probabilities are configured according to the traffic flow design, where the probability of left turn is 12.5%, the probability of going straight is 75%, and the probability of right turn is 12.5%. The main traffic parameters of the experimental road network are summarized in Table 7.

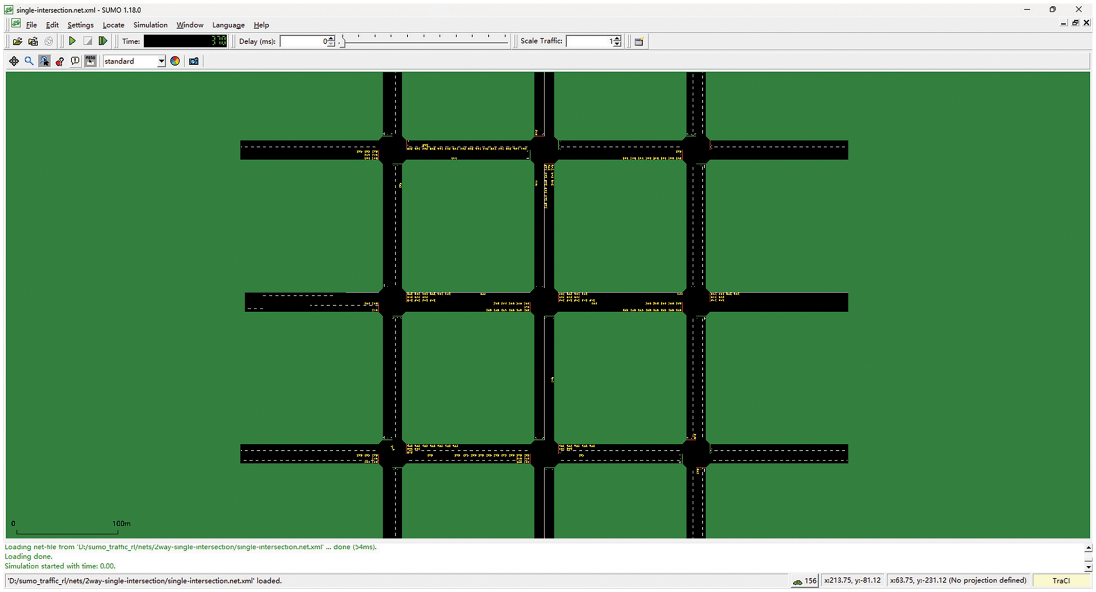


Figure 5: SUMO simulation area intersection simulation interface.

Table 7: Traffic parameters of the experimental road network.

Traffic Parameter	Value
Number of intersections	9
Number of lanes per road section	3
Number of road sections	24
Average segment length	400 m
Traffic volume	2000 veh/h
Probability of turning left	12.5%
Probability of going straight	75%
Probability of turning right	12.5%
Maximum speed limit	60 km/h

To provide a comprehensive comparison, five control methods are selected in the main experiments: Fixed-Time, Max-Pressure, CoLight, DQN, and the proposed T2-BAS-DQN. Fixed-Time is used as a conventional non-adaptive signal timing baseline. Max-Pressure is introduced as a representative analytical pressure-based signal control method, which determines signal priority according to queue pressure and movement competition. CoLight is adopted as a representative multi-agent reinforcement learning baseline for regional signal coordination. DQN is used as the basic deep reinforcement learning baseline without Type-2 fuzzy uncertainty modeling or BAS-based parameter optimization. The proposed T2-BAS-DQN integrates DQN-based online decision-making, Type-2 fuzzy uncertainty correction, BAS-based offline parameter optimization, and regional coordination information.

It should be noted that Type-2-FDQN is used in the component contribution analysis rather than in the main baseline comparison. Specifically, DQN, Type-2-FDQN, and T2-BAS-DQN are compared to evaluate the incremental contribution of the Type-2 fuzzy module and the BAS-based parameter optimization module. In the main comparison experiments, Max-Pressure and CoLight are added to cover representative analytical pressure-based control and multi-agent reinforcement learning baselines. PressLight and MPLight are also important pressure-based multi-agent reinforcement learning methods. However, they require specific network-wide multi-agent training settings and pressure-based reward designs that are different from the current implementation framework. Therefore, they are left for future comparative studies.

To ensure a fair comparison, all control methods are evaluated under the same SUMO network, traffic demand, simulation horizon, decision interval, and random seeds. For learning-based methods, the same state representation, action space, reward indicators, and evaluation protocol are adopted unless otherwise specified. The performance indicators reported in the experimental tables are obtained during the evaluation stage after model training or parameter optimization is completed.

For the learning-based baselines, the implementation settings were kept consistent as much as possible. DQN used the state representation defined in Eq. (9), including queue length, waiting time, current signal phase, elapsed green time, and neighboring-intersection queue information. The action space followed the four-phase signal control structure defined in Eq. (10). The reward function was defined by the local reward and regional reward formulations in Eqs. (11) and (12), respectively. The DQN network adopted the fully connected architecture summarized in Table 4. The training hyperparameters, including the number of training episodes, learning rate, discount factor, replay buffer size, batch size, exploration strategy, minimum green time, yellow time, and random seeds, followed the common settings in Table 5.

CoLight was implemented as a graph-attention-based multi-agent reinforcement learning baseline under the same SUMO–TraCI interface. The road-network adjacency relationship was used to construct the coordination graph, and neighboring-intersection information was aggregated through the attention-based coordination module. To avoid introducing additional differences from the reward or action design, CoLight used the same action space, reward indicators, training episodes, learning rate, and random seeds as the other learning-based methods. The proposed T2-BAS-DQN used the same DQN architecture and basic training configuration, while additionally incorporating Type-2 fuzzy correction, coordination pressure priority, and BAS-based offline optimization of the low-dimensional parameter vector $\Theta = [\omega_f, \omega_r, \omega_p]$. During BAS optimization, the trained DQN parameters were fixed, and different parameter vectors were evaluated through validation rollouts. Unless otherwise specified, the hyperparameters of the learning-based baselines were reused from the common settings in Table 5, while only the method-specific modules, such as Type-2 fuzzy correction, graph-attention coordination, and BAS-based parameter optimization, were changed.

The proposed T2-BAS-DQN introduces an additional BAS-based offline parameter optimization stage. Therefore, the additional optimization cost should be distinguished from the online control performance. In this study, traffic performance and computational cost are reported separately. The traffic performance indicators, such as average queue length, average waiting time, average speed, and number of stops, are used to evaluate the final control effect under the same testing scenarios. The training and optimization cost comparison is summarized in Table 8. Since BAS is not executed during online deployment, the online control process of T2-BAS-DQN mainly includes traffic state observation, one DQN forward pass, Type-2 fuzzy inference, and fused action selection.

Table 8: Training and optimization cost comparison of different methods.

Method	Offline Training Required	Additional Offline Optimization	Online Control Process	Cost Characteristic
Fixed-time	No	No	Fixed signal timing plan	Lowest cost, non-adaptive
Max-pressure	No	No	Queue-pressure calculation	Low online cost
CoLight	Yes	No	Multi-agent network inference	Higher training cost
DQN	Yes	No	DQN forward pass	Learning cost only
T2-BAS-DQN	Yes	BAS optimization of Θ	DQN forward pass, Type-2 fuzzy inference, and action fusion	Extra offline cost, acceptable online cost

4.2 Training Performance and Component Contribution Analysis

To evaluate the learning performance of the reinforcement-learning-based methods, the average cumulative reward values of DQN, Type-2-FDQN, and T2-BAS-DQN during the training process are compared, as shown in Fig. 6.

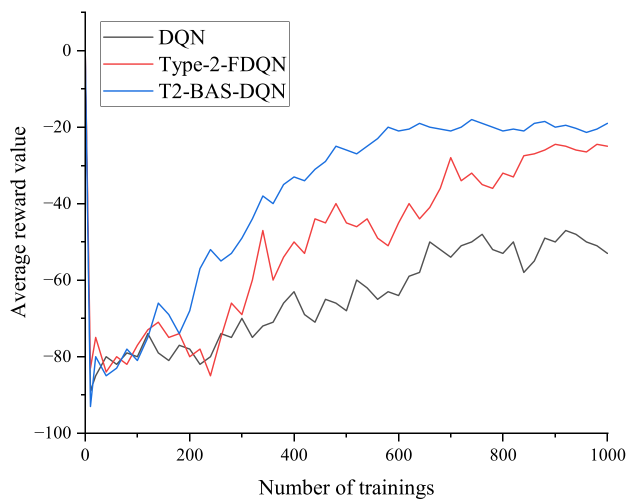


Figure 6: Average reward value during training.

As shown in Fig. 6, DQN has a relatively low reward value in the early training stage and shows obvious fluctuations during the learning process. This is mainly because the basic DQN method relies on the original Q-network to explore the action space, and its exploration efficiency is limited when the traffic state changes dynamically. Compared with DQN, Type-2-FDQN introduces Type-2 fuzzy logic into the reinforcement learning framework, which improves the representation ability of uncertain traffic states and enhances the stability of the learning process.

The proposed T2-BAS-DQN further combines Type-2 fuzzy control and BAS-based parameter optimization. The reward value increases faster and gradually converges to a higher level, indicating that the BAS optimization mechanism can improve the parameter adaptability of the controller and help the reinforcement learning model find a better control strategy. Therefore, the comparison among DQN, Type-2-FDQN, and T2-BAS-DQN can also reflect the contribution of the Type-2 fuzzy module and BAS optimization module.

To further analyze the contribution of different modules, the performance of DQN, Type-2-FDQN, and T2-BAS-DQN is compared in Fig. 7. DQN is regarded as the basic reinforcement learning controller. Type-2-FDQN adds the Type-2 fuzzy module on the basis of DQN, while T2-BAS-DQN further introduces BAS-based parameter optimization. From the comparison results, Type-2-FDQN improves the performance of DQN in most indicators, which indicates that Type-2 fuzzy logic can enhance the robustness of the controller under stochastic traffic conditions. T2-BAS-DQN further reduces queue length and waiting time or maintains competitive performance, showing that BAS optimization contributes to better parameter adjustment and more stable phase decision-making.

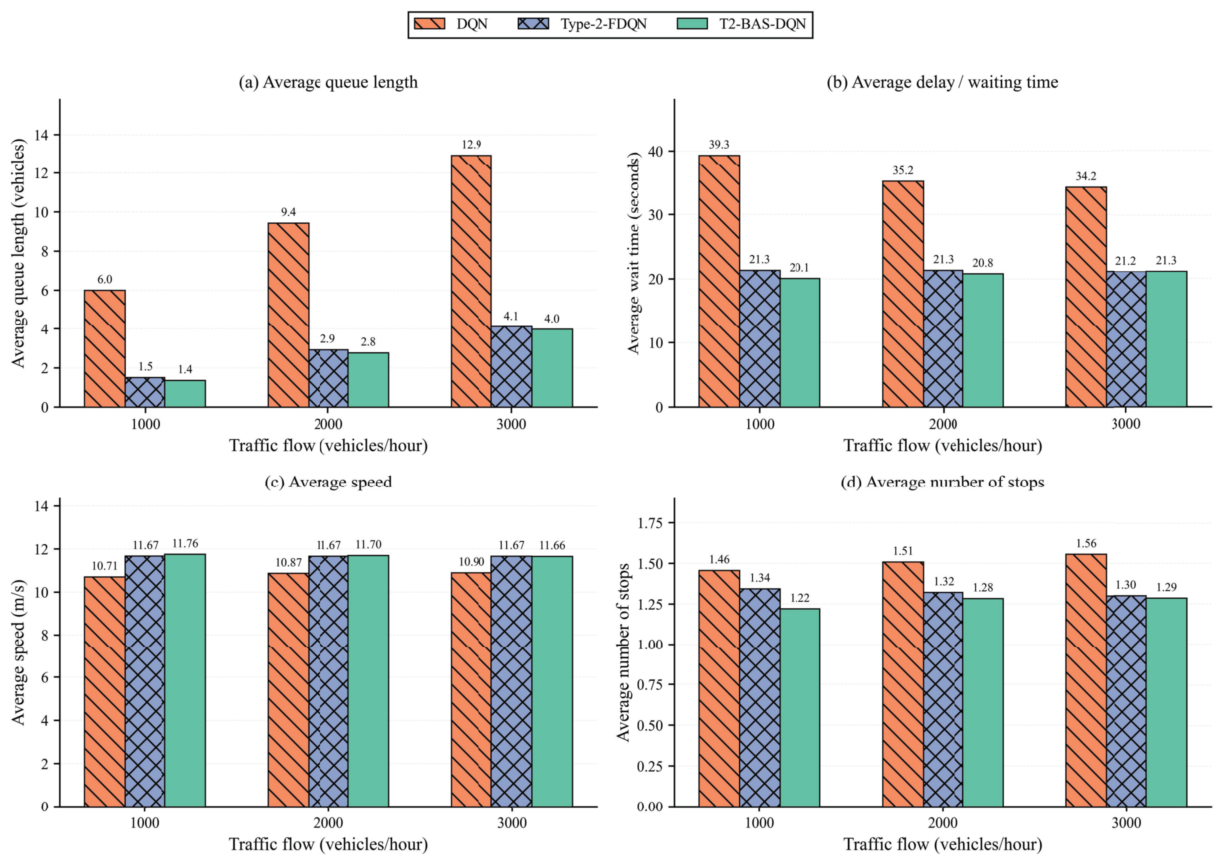


Figure 7: Component contribution comparison of DQN, Type-2-FDQN and T2-BAS-DQN.

4.3 Performance Comparison under the Same Traffic Flow

To further compare the control performance of different algorithms under the same traffic demand, Fixed-Time, Max-Pressure, CoLight, DQN, and T2-BAS-DQN are evaluated under the same traffic flow condition.

To reduce the influence of randomness caused by vehicle generation and reinforcement learning training, each control method was independently evaluated over $N = 10$ runs using random seeds 1–10. The results are reported in the form of mean ± standard deviation, which reflects both the average performance and the fluctuation range of each algorithm. Since the same traffic flow condition was used for all compared methods, the statistical comparison can more reliably reflect the robustness and stability of different control strategies. The detailed statistical results are summarized in Table 9.

Table 9: Statistical performance comparison of different control algorithms under the same traffic flow.

Evaluation Indicator	Fixed-Time	Max-Pressure	CoLight	DQN	T2-BAS-DQN
Average queue length (vehicles)	23.55 ± 1.18	17.20 ± 0.65	17.52 ± 0.72	20.88 ± 0.93	16.61 ± 0.56
Average waiting time (s)	52.16 ± 2.40	38.60 ± 1.32	39.12 ± 1.45	44.91 ± 1.88	37.15 ± 1.31
Average speed (m/s)	6.22 ± 0.22	9.18 ± 0.18	9.05 ± 0.20	7.63 ± 0.24	9.36 ± 0.15

As shown in Table 9, compared with Fixed-Time, all adaptive and coordination-based methods improve the overall traffic operation performance to different degrees. Fixed-Time produces the largest queue length and waiting time, indicating that static signal timing cannot adapt to real-time traffic fluctuations. DQN reduces the average queue length and waiting time compared with Fixed-Time, showing that reinforcement learning can improve adaptive phase decision-making. However, the relatively larger standard deviation of DQN indicates that its control performance is still affected by stochastic traffic states.

Max-Pressure achieves better performance than DQN by directly considering queue imbalance and movement pressure. CoLight further improves regional signal coordination through multi-agent information interaction. Compared with these baselines, T2-BAS-DQN achieves the lowest average queue length and waiting time, as well as the highest average speed. In addition, its standard deviation remains relatively small, indicating that the proposed method not only improves the average control performance but also maintains better stability under repeated simulations. These results suggest that the combination of DQN-based online decision-making, Type-2 fuzzy uncertainty correction, BAS-based offline parameter optimization, and regional coordination can improve the robustness and adaptability of regional traffic signal control.

The dynamic variation of average queue length under the same traffic flow is shown in Fig. 8. At the beginning of the simulation, the number of vehicles in the road network is small, and the average queue length of all methods remains close to zero. As vehicles continuously enter the network, the queue length gradually increases. During the congested period, Fixed-Time produces the highest queue accumulation because its signal timing cannot respond to real-time traffic fluctuations. DQN reduces the queue length to a certain extent, but its queue dissipation ability is still limited. Max-Pressure reduces queue accumulation by considering queue pressure, while CoLight improves the coordination among intersections through multi-agent information interaction. Compared with these methods, T2-BAS-DQN maintains a lower queue level during the congested stage and shows better queue dissipation ability in the later stage.

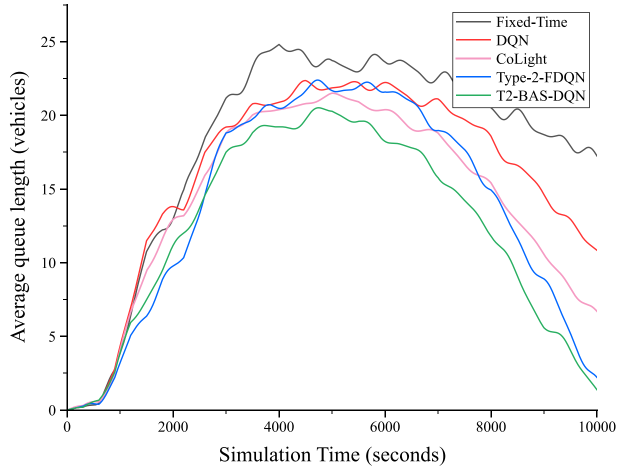


Figure 8: Average queue length under the same traffic flow.

Fig. 9 shows the variation of average speed under the same traffic flow. At the early stage of simulation, the road network is relatively uncongested, and all methods maintain a relatively high speed. With the increase of vehicles in the network, queue accumulation and stop-and-go behavior gradually reduce the average speed. Fixed-Time has the lowest speed during most of the simulation process, indicating poor adaptability to dynamic traffic demand. DQN improves the average speed compared with Fixed-Time, but the speed curve still fluctuates. Max-Pressure improves the average speed by reducing queue pressure, while CoLight shows better speed maintenance ability through regional coordination. T2-BAS-DQN maintains a relatively high speed in the middle and later stages, indicating that the proposed method can improve vehicle progression and reduce stop-and-go behavior.

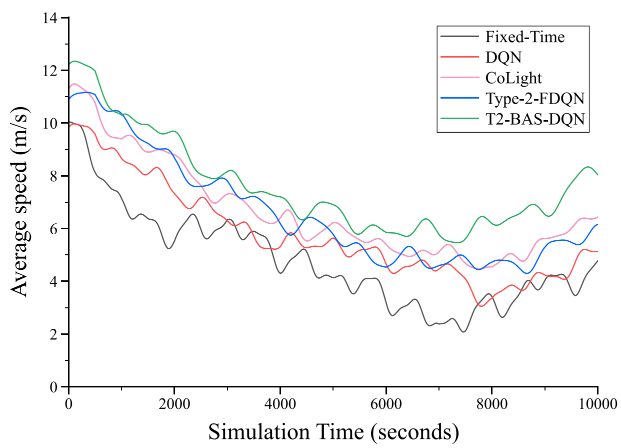


Figure 9: Average speed under the same traffic flow.

4.4 Demand Generalization under Different Traffic Flows

To evaluate the adaptability of different algorithms under varying traffic demand levels, comparative experiments are conducted under three traffic flow conditions: 1000, 2000, and 3000 veh/h. These three demand levels represent light, medium, and heavy traffic conditions, respectively. Compared with the single-demand evaluation in Section 4.3, this experiment further examines whether each control strategy can maintain stable performance when traffic demand changes. In this subsection, Fixed-Time, Max-Pressure, CoLight, DQN, and T2-BAS-DQN are treated as the main comparison methods. Type-2-FDQN is retained only as a component-analysis reference to examine the contribution of the Type-2 fuzzy module without BAS-based parameter optimization. The average queue length and average waiting time under different traffic flows are shown in Figs. 10 and 11, respectively.

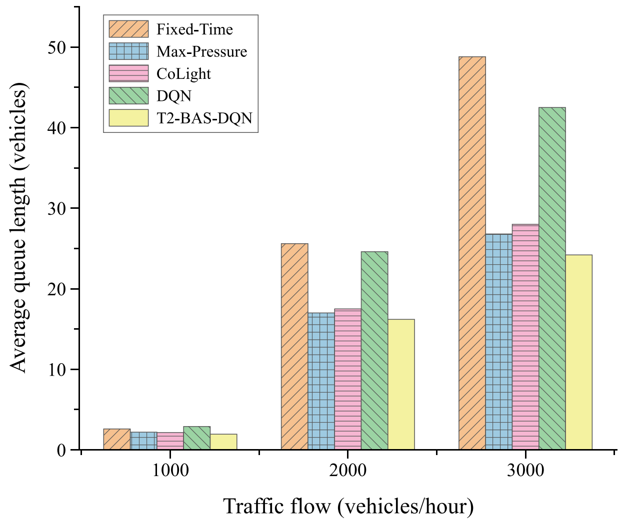


Figure 10: Average queue length under different traffic flows

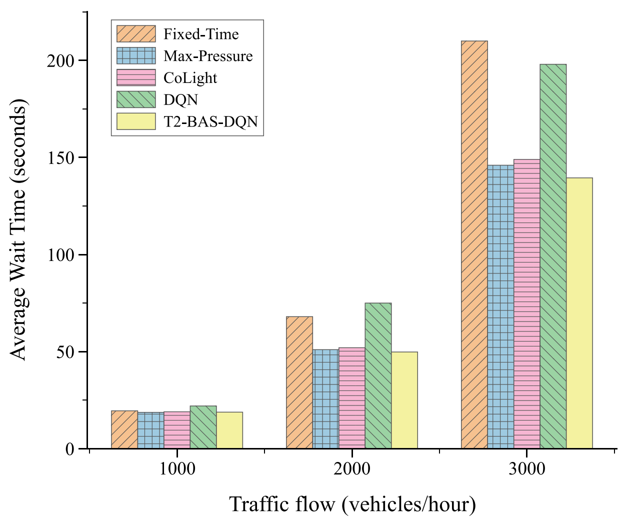


Figure 11: Average waiting time under different traffic flows.

As shown in Fig. 10, under the low-demand condition of 1000 veh/h, the queue lengths of all methods are relatively small, indicating that the road network is not heavily congested. When the traffic demand increases to 2000 and 3000 veh/h, the differences among different control methods become more significant. Fixed-Time and DQN show larger queue accumulation because they either lack real-time adaptability or have limited coordination ability. Max-Pressure reduces queue accumulation by considering queue imbalance and pressure differences among competing traffic movements. CoLight improves regional coordination through multi-agent information interaction. Compared with Fixed-Time, Max-Pressure, CoLight, and DQN, T2-BAS-DQN achieves lower or competitive queue length under medium and high traffic demands, indicating that the hybrid framework can maintain demand adaptability in the tested scenarios.

As shown in Fig. 11, the average waiting time increases as the traffic demand rises for all compared methods. Fixed-Time shows the fastest increase because it cannot adapt signal timing to real-time traffic fluctuations. DQN reduces waiting time compared with Fixed-Time, but its performance becomes less stable under high demand due to limited coordination and uncertainty handling. Max-Pressure and CoLight achieve better waiting-time control by using pressure-based phase selection and multi-agent coordination, respectively. Compared with these main baselines, T2-BAS-DQN shows lower or competitive waiting time under medium and high traffic demands. This result suggests that the integration of Type-2 fuzzy uncertainty modeling, BAS-based offline parameter optimization, and regional coordination can help improve the adaptability of the controller under increasing traffic demand.

Overall, the results under 1000, 2000, and 3000 veh/h indicate that the proposed T2-BAS-DQN method maintains competitive performance under different demand levels. Under low traffic demand, the performance differences among the methods are relatively small because the road network is not saturated. As the traffic demand increases, the limitations of Fixed-Time and the basic DQN become more obvious, while CoLight, Type-2-FDQN, and T2-BAS-DQN show better adaptability to increasing traffic pressure. In particular, T2-BAS-DQN maintains lower or competitive queue length and waiting time under medium and high traffic demands, suggesting that the proposed hybrid framework can improve demand adaptability in the tested scenarios. Nevertheless, these results mainly verify the adaptability of the proposed method under different traffic volumes within the same network structure, and further validation on larger and more heterogeneous networks is still required.

Therefore, the demand-generalization experiment should be interpreted from two perspectives. The main comparison evaluates T2-BAS-DQN against Fixed-Time, Max-Pressure, CoLight, and DQN. The component contribution of Type-2-FDQN is discussed separately in Section 4.2 and is not included in this demand-generalization figure set.

4.5 Performance on the Real-Geometry Road Network

To further evaluate the applicability of the proposed method under more realistic road topology and turning-movement conditions, a SUMO simulation scenario was constructed based on real-world road geometry and field-calibrated traffic demand. It should be emphasized that this experiment is not a field deployment test, but a simulation with real-world geometry and demand calibration. The road network contains nine key signalized intersections located in Jiangning District, Nanjing, China. The distribution of these intersections is shown in Fig. 12.

The traffic flow data used for demand calibration were collected through field surveys on 21 February 2025. The survey recorded the through, left-turn, and right-turn vehicle volumes at each approach of the nine intersections. These field-count data were then used to calibrate the turning ratios and route inputs in SUMO, so that the simulation could better reflect realistic traffic demand patterns rather than relying only on synthetic traffic flows. The detailed field traffic counts are summarized in Table 10.

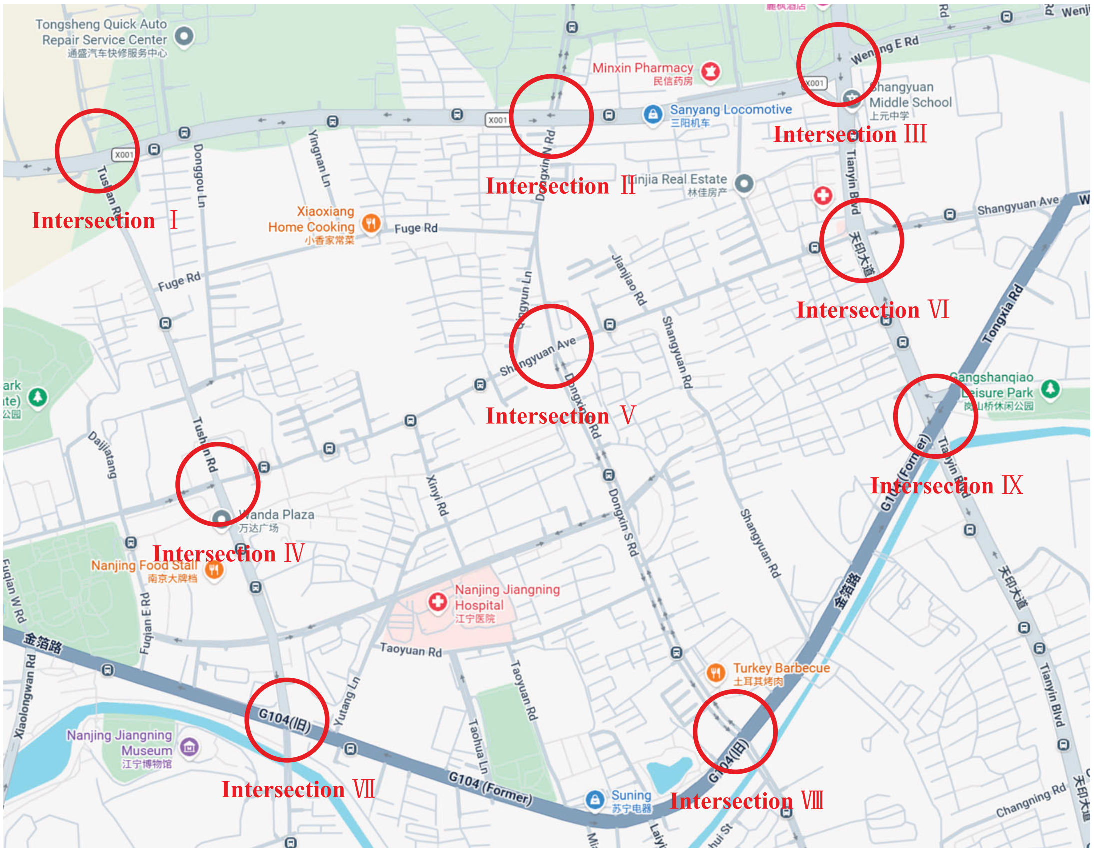


Figure 12: Distribution of key signalized intersections in the real-geometry road network.

Table 10: Field-count traffic volumes used for real-geometry demand calibration.

Intersection Name	Import Lanes	Number of Left Turns (vehicles)	Number of Proceed Straight (vehicles)	Number of Right Turns (vehicles)
Wenjing Road–Tushan Road (Intersection I)	South import	88	203	115
	North import	97	162	171
	West import	121	137	106
	East import	91	125	118
Shangyuan Street–Zhushan Road (Intersection II)	South import	201	402	159
	North import	187	396	105
	West import	132	471	160
	East import	159	526	172

(Continued)

Table 10 (continued)

Intersection Name	Import Lanes	Number of Left Turns (vehicles)	Number of Proceed Straight (vehicles)	Number of Right Turns (vehicles)
Gold Leaf Road–Zhushan Road (Intersection III)	South import	97	210	71
	North import	72	149	106
	West import	60	114	55
	East import	83	131	102
Wenjing Road–Dongxin South Road (Intersection IV)>	South import	61	327	72
	North import	59	283	51
	West import	48	155	57
	East import	64	130	56
Shangyuan Street–Dongxin South Road (Intersection V)	South import	77	315	58
	North import	56	321	47
	West import	74	508	73
	East import	82	552	76
Gold Leaf Road–Dongxin South Road (Intersection VI)	South import	76	303	60
	North import	69	319	77
	West import	81	102	79
	East import	98	164	92
Wenjing Road–Tianyin Avenue (Intersection VII)	South import	66	895	81
	North import	47	806	66
	West import	96	132	43
	East import	102	179	52
Shangyuan Street–Tianyin Avenue (Intersection VIII)	South import	130	818	71
	North import	135	775	53
	West import	131	482	96
	East import	127	451	132
Gold Leaf Road–Tianyin Avenue (Intersection IX)	South import	77	798	57
	North import	62	856	49
	West import	93	147	90
	East import	89	156	83

The same reward function, training protocol, and hyperparameter settings as those used in the synthetic-network experiments were adopted in this real-geometry simulation, unless otherwise specified. This setting ensures that the comparison focuses mainly on the influence of road topology and field-calibrated demand rather than changes in algorithm configuration.

The performance comparison in the real-geometry simulation scenario is shown in Table 11. To keep the statistical reporting consistent with the synthetic-network experiment, each method was evaluated over $N = 10$ independent runs using random seeds 1–10. The results are reported as mean ± standard deviation.

This setting allows the robustness of different control methods to be compared under the same real-geometry road network and field-calibrated demand.

Table 11: Statistical performance comparison in the real-geometry simulation scenario.

Evaluation Indicator	DQN	Type-2-FDQN	CoLight	T2-BAS-DQN
Average speed (m/s)	10.95 ± 0.23	11.77 ± 0.18	11.95 ± 0.16	12.15 ± 0.14
Average queue length (vehicles)	22.98 ± 1.18	16.03 ± 0.91	13.35 ± 0.76	12.08 ± 0.68
Average number of stops	1.56 ± 0.07	1.45 ± 0.05	1.42 ± 0.04	1.39 ± 0.04
Average waiting time (s)	39.05 ± 1.72	31.22 ± 1.21	30.41 ± 0.98	29.53 ± 0.89

In terms of mean values, compared with DQN, the proposed T2-BAS-DQN increases the average speed from 10.95 to 12.15 m/s, reduces the average queue length from 22.98 vehicles to 12.08 vehicles, decreases the average number of stops from 1.56 to 1.39, and reduces the average waiting time from 39.05 to 29.53 s. These results correspond to an average speed improvement of approximately 10.96%, a queue length reduction of approximately 47.43%, a stop reduction of approximately 10.90%, and a waiting-time reduction of approximately 24.38%.

Compared with CoLight, T2-BAS-DQN also shows slightly better mean values in the real-geometry simulation scenario. Specifically, the average speed increases from 11.95 to 12.15 m/s, the average queue length decreases from 13.35 vehicles to 12.08 vehicles, the average number of stops decreases from 1.42 to 1.39, and the average waiting time decreases from 30.41 to 29.53 s. However, the margins over CoLight are relatively modest. Therefore, these results should be interpreted as simulation evidence under the tested setting rather than a statistically conclusive demonstration of general superiority.

As shown in Table 11, the real-geometry simulation results further indicate that the proposed method is not limited to the regular synthetic grid network. Compared with DQN, T2-BAS-DQN achieves clear improvements in all four indicators, including higher average speed, shorter average queue length, fewer stops, and lower average waiting time. Compared with CoLight, T2-BAS-DQN also shows slightly better mean values, with the average speed increasing from 11.95 to 12.15 m/s, the average queue length decreasing from 13.35 vehicles to 12.08 vehicles, the average number of stops decreasing from 1.42 to 1.39, and the average waiting time decreasing from 30.41 to 29.53 s. These results suggest that the integration of Type-2 fuzzy uncertainty modeling, BAS-based parameter optimization, and regional coordination can provide competitive simulation performance under a more complex road topology and field-calibrated traffic demand.

However, the performance margins over strong baselines such as CoLight are relatively modest. Therefore, the results should be interpreted as simulation evidence under the tested real-geometry scenario rather than a statistically conclusive demonstration of general superiority. It should also be noted that this experiment has several limitations. First, the experiment remains a SUMO-based simulation rather than an on-road deployment. Second, the demand calibration is based on field-count data collected on a single day, which does not fully reflect day-of-week, weather, seasonal, or incident-related variations. Third, the currently deployed signal timing plans at these intersections were not available, and therefore a direct comparison with the actual field control system was not conducted. These limitations will be addressed in future work by collecting multi-day traffic data, conducting statistical significance analysis, and cooperating with traffic management agencies to obtain deployed signal timing plans.

4.6 Discussion on Scalability, Generalization and Deployment Limitations

The experimental results in Sections 4.3–4.5 demonstrate the effectiveness of the proposed T2-BAS-DQN method from three perspectives: performance comparison under the same traffic flow, adaptability under different traffic demand levels, and simulation evaluation on a real-geometry road network. The results under 1000, 2000, and 3000 veh/h show that the proposed method can maintain competitive performance when traffic demand changes. In addition, the real-geometry simulation calibrated with field traffic counts further indicates that the proposed method is not limited to a regular synthetic grid network.

From the perspective of scalability, the proposed framework adopts a regional multi-intersection control structure. Each intersection agent mainly uses local traffic states and neighboring-intersection information for decision-making. Therefore, when the network size increases, additional intersections can be incorporated by adding corresponding local agents and neighbor-based coordination links. This decentralized observation and local coordination mechanism provides a basis for extending the method to larger networks.

However, several limitations should also be acknowledged. First, the current experiments are mainly conducted on a 3 × 3 regional network. Although different traffic demand levels are considered, these experiments mainly verify demand adaptability within the same network structure and do not fully prove scalability to large-scale urban networks with more complex topology. Second, the real-geometry experiment remains a SUMO-based simulation rather than an on-road deployment. The field-count data were collected on a single day, which cannot fully reflect day-of-week, weather, seasonal, or incident-related variations. Third, the currently deployed signal timing plans at the selected intersections were not available, and therefore a direct comparison with the actual field control system was not conducted.

In addition, the traffic signal model in this study adopts a simplified four-phase structure, while practical urban intersections may involve more complex phase schemes, such as dual-ring control, leading/lagging left-turn phases, pedestrian phases, and bus priority control. Multimodal road users, including pedestrians, cyclists, buses, and emergency vehicles, are not explicitly modeled in the current simulation. Moreover, although BAS optimization is performed offline and does not significantly increase the online control burden, it still introduces additional training and parameter optimization cost compared with standard DQN-based methods.

From the perspective of fair comparison, although traffic performance and computational cost are reported separately in this study, a strict matched-budget comparison has not yet been conducted. Specifically, DQN, CoLight, and T2-BAS-DQN are not trained under exactly the same total computational budget because T2-BAS-DQN includes an additional BAS-based offline parameter optimization stage. Future work will conduct matched-budget experiments and compare BAS with other metaheuristic optimizers, such as PSO, differential evolution, genetic algorithms, and DNA evolutionary algorithms, under the same fitness-evaluation budget.

Future work will further evaluate the proposed method on larger and more heterogeneous urban road networks, such as 5 × 5 or city-level networks. More diverse traffic scenarios, multi-day field data, multimodal road users, and actual deployed signal timing plans will be considered to improve the practical applicability of the proposed framework. In addition, graph-based information aggregation and parameter-sharing multi-agent learning can be introduced to reduce communication and training costs in large-scale regional traffic signal control.

5 Conclusion

This study proposed a hybrid bio-inspired Type-2 fuzzy reinforcement learning framework, named T2-BAS-DQN, for regional traffic signal coordination control under uncertain and dynamic traffic conditions.

The proposed framework integrates DQN-based adaptive decision-making, Type-2 fuzzy uncertainty-aware correction, BAS-based offline parameter optimization, and regional coordination information. In the proposed method, the DQN network remains the online signal decision-making module, while the Type-2 fuzzy module is used to smooth uncertain traffic-state fluctuations and provide correction for phase selection. BAS is employed only in the offline parameter-optimization stage to tune a low-dimensional parameter vector related to fuzzy correction, reward adjustment, and coordination pressure. Therefore, the proposed method does not directly optimize the high-dimensional DQN weight space using BAS, but combines learning-based control, fuzzy uncertainty modeling, and lightweight parameter tuning in a unified traffic signal control framework.

Simulation experiments were conducted on a 3 × 3 regional SUMO road network under different traffic demand levels. The results indicate that T2-BAS-DQN achieves competitive mean performance in the tested scenarios. Compared with Fixed-Time and the basic DQN baseline, the proposed method can reduce queue accumulation and waiting time more effectively, especially under medium and high traffic demand conditions. Compared with Type-2-FDQN, the results show that the BAS-based offline parameter optimization module can further improve the adaptability of the hybrid fuzzy-reinforcement learning controller. The demand-level experiments also suggest that the proposed framework can maintain relatively stable performance when traffic volume increases from low to high levels. Nevertheless, these experiments are conducted within the same synthetic network structure, and the demand-generalization results should therefore be interpreted as simulation evidence under controlled traffic conditions rather than a complete validation across all possible network types.

In addition to the synthetic network experiments, a real-geometry SUMO simulation scenario calibrated with field traffic counts from Jiangning District, Nanjing, was constructed to further examine the applicability of the proposed method under a more complex road topology and field-calibrated traffic demand. Compared with the vanilla DQN baseline in this calibrated scenario, T2-BAS-DQN reduces the average queue length by 47.4%, decreases the average waiting time by 24.4%, and increases the average travel speed by 11.0%. These results indicate that the proposed hybrid framework can provide useful simulation-level improvements over the basic learning baseline. Compared with stronger baselines such as Max-Pressure and CoLight, T2-BAS-DQN also obtains slightly better or comparable mean values in several indicators. However, the performance margins over these strong baselines are relatively modest and may be of the same order as the reported standard deviations. Since no formal statistical significance test is conducted in this study, the results should be regarded as simulation evidence of the effectiveness of the proposed hybrid pipeline rather than a statistically conclusive demonstration of general superiority.

It should also be noted that T2-BAS-DQN includes an additional offline BAS parameter-optimization stage, whereas some baseline methods do not receive an equivalent tuning budget. Therefore, the reported improvements reflect the performance of the complete hybrid pipeline, including offline parameter optimization, rather than superiority under an equal total computational budget. In online deployment, BAS is no longer executed, and the controller only performs traffic state observation, Type-2 fuzzy inference, one DQN forward pass, and fused action selection. This design keeps the online computational burden acceptable for real-time traffic signal control, while the extra optimization cost is mainly introduced during the offline training and validation stage.

This study still has several limitations. First, all experiments are SUMO-based simulations, and no on-road deployment or closed-loop field test has been conducted. Second, the real-geometry simulation is calibrated using field-count data collected on a single day, which cannot fully represent day-of-week, weather, seasonal, or incident-related traffic variations. Third, the actual deployed signal timing plans at the studied intersections were not available, and therefore a direct comparison with the field control system could not be performed. Fourth, although repeated-run statistics are reported, formal significance tests are not included. Finally, the comparison with learning-based baselines is not conducted under a strictly matched total computational budget, because the proposed method includes an additional offline BAS optimization stage.

Future work will focus on several aspects. First, multi-day and multi-period traffic data will be collected to improve demand calibration and evaluate the robustness of the proposed method under more diverse traffic conditions. Second, cooperation with traffic management agencies will be pursued to obtain deployed signal timing plans and support comparisons with actual field control strategies. Third, larger and more heterogeneous road networks will be used to further examine the scalability and transferability of the proposed framework. Fourth, formal statistical significance tests and matched-budget comparisons will be conducted to provide a more rigorous evaluation of the proposed method against strong analytical and learning-based baselines. Finally, different metaheuristic optimizers, such as differential evolution, genetic algorithms, particle swarm optimization, and DNA evolutionary algorithms, may be compared under the same computational budget to further clarify the specific contribution of BAS in the proposed Type-2 fuzzy reinforcement learning framework.

Acknowledgement: None.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 62303214) and Key Project of Basic Science Research in Universities of Jiangsu Province (No. 23KJA460008).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Yunrui Bi and Qiliang Yang; methodology, Qinglin Ding and Qiliang Yang; software, Qinglin Ding; validation, Qinglin Ding and Qiliang Yang; formal analysis, Mingjie Zhang and Qiliang Yang; investigation, Qinglin Ding; data curation, Mingjie Zhang; writing—original draft preparation, Qiliang Yang; writing—review and editing, Qiliang Yang, Yunrui Bi, Bin Ran and Kun Liu; supervision, Yunrui Bi, Bin Ran and Kun Liu; project administration, Yunrui Bi. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The SUMO network files, route files, controller scripts, configuration files, random seeds, evaluation scripts, and field-collected traffic count data used in this study have not been deposited in a public repository at the time of submission. These materials and data can be provided by the corresponding author upon reasonable request for academic verification. Exact numerical reproduction of the reported simulation results requires these simulation artifacts, implementation files, random seeds, and field-collected traffic count data.

Ethics Approval: Not applicable.

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

References

1. Wu Y. Enhancing urban traffic flow through fuzzy logic-based signal light control optimization. Int J e-Collab. 2024;20(1):1–13. doi:10.4018/ijec.358746.
2. Singh AR, Ashraf M.W. A, Rathore RS, Li B, Sujatha MS. Real-time traffic flow optimization using large language models and reinforcement learning for smart urban mobility. Appl Soft Comput. 2025;185(1):113917. doi:10.1016/j.asoc.2025.113917.
3. Li M, Pan X, Liu C, Li Z. Federated deep reinforcement learning-based urban traffic signal optimal control. Sci Rep. 2025;15(1):11724. doi:10.1038/s41598-025-91966-1.
4. Fu T, Wang L, Garg S, Zhu Z, Zhang J. Adaptive signal light timing for regional traffic optimization based on graph convolutional network empowered traffic forecasting. Inf Fusion. 2024;103(4):102072. doi:10.1016/j.inffus.2023.102072.

5. Wang T, Zhu Z, Zhang J, Tian J, Zhang W. A large-scale traffic signal control algorithm based on multi-layer graph deep reinforcement learning. Transp Res Part C Emerg Technol. 2024;162(3):104582. doi:10.1016/j.trc.2024.104582.
6. Su W, Mu C, Xue L, Yang X, Zhu S. An improved traffic coordination control integrating traffic flow prediction and optimization. Eng Appl Artif Intell. 2025;143(4):109969. doi:10.1016/j.engappai.2024.109969.
7. Moreno-Malo J, Posadas-Yagüe JL, Cano JC, Calafate CT, Conejero JA, Poza-Lujan JL. Improving traffic light systems using deep Q-networks. Expert Syst Appl. 2024;252:124178.
8. Li Y, Zhang Y, Li X, Sun C. Regional multi-agent cooperative reinforcement learning for city-level traffic grid signal control. IEEE/CAA J Autom Sinica. 2024;11(9):1987–98. doi:10.1109/jas.2024.124365.
9. Tan T, Bao F, Deng Y, Jin A, Dai Q, Wang J. Cooperative deep reinforcement learning for large-scale traffic grid signal control. IEEE Trans Cybern. 2019;50(6):2687–700. doi:10.1109/tcyb.2019.2904742.
10. 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(19):3894. doi:10.3390/electronics13193894.
11. Chala TD, Kóczy LT. Agent-based intelligent fuzzy traffic signal control system for multiple road intersection systems. Mathematics. 2024;13(1):124. doi:10.3390/math13010124.
12. Wu Q, Wu J, Shen J, Du B, Telikani A, Fahmideh M, et al. Distributed agent-based deep reinforcement learning for large scale traffic signal control. Knowl-Based Syst. 2022;241(15):108304. doi:10.1016/j.knosys.2022.108304.
13. Qian Q, Deng Y, Sun H, Pan J, Yin J, Feng Y, et al. Enhanced beetle antennae search algorithm for complex and unbiased optimization. Soft Comput. 2022;26(19):10331–69. doi:10.1007/s00500-022-07388-y.
14. Haddad TA, Hedjazi D, Aouag S. A deep reinforcement learning-based cooperative approach for multi-intersection traffic signal control. Eng Appl Artif Intell. 2022;114(3):105019. doi:10.1016/j.engappai.2022.105019.
15. Kang L, Huang H, Lu W, Liu L. A dueling deep Q-network method for low-carbon traffic signal control. Appl Soft Comput. 2023;141(9):110304. doi:10.1016/j.asoc.2023.110304.
16. Bi Y, Srinivasan D, Lu X, Sun Z, Zeng W. Type-2 fuzzy multi-intersection traffic signal control with differential evolution optimization. Expert Syst Appl. 2014;41(16):7338–49. doi:10.1016/j.eswa.2014.06.022.
17. Bi Y, Sun Z, Lu X, Sun Z, Liu D, Liu K. Adaptive type-2 fuzzy traffic signal control with on-line optimization. J Intell Fuzzy Syst. 2018;35(2):1889–904. doi:10.3233/JIFS-171405.
18. Xiao F, Lu J, Li L, Tu W, Li C. Advances in reinforcement learning for traffic signal control: a review of recent progress. Intell Transp Infrastruct. 2025;4:liaf009. doi:10.1093/iti/liaf009.
19. Ahmad A, Al-Sumaiti AS, Byon YJ, Alhosani K. Multiple intelligent control strategies for travel-time reduction of connected emergency vehicles. IEEE Trans Intell Transp Syst. 2024;26(1):337–53. doi:10.1109/tits.2024.3484771.
20. Nautiyal K, Gangodkar D, Diwakar M, Singh P, Bijalwan A. Intelligent traffic light management using predictive and dynamic traffic flow analysis. Sci Rep. 2025;15(1):37188. doi:10.1038/s41598-025-13694-w.
21. Almomany A, Eedi E, Sutcu M. Real-time traffic signal optimization for urban mobility: a reinforcement learning-enhanced framework with application to Kuwait City. Front Robot AI. 2025;12:1669952. doi:10.3389/frobt.2025.1669952.
22. Zhu Y, Lv Y, Lin S, Xu J. A stochastic traffic flow model-based reinforcement learning framework for advanced traffic signal control. IEEE Trans Intell Transp Syst. 2025;26(1):714–23. doi:10.1109/tits.2024.3494251.
23. Fan S, Lu K, Wang Y, Tian X, Zhang M. Action masking-based proximal policy optimization with the dual-ring phase structure for adaptive traffic signal control. IEEE Trans Intell Transp Syst. 2025;26(2):2422–33. doi:10.1109/tits.2024.3510379.
24. Wan L, Yu C, Lo HK. Traffic signal coordination under stochastic demands and turning ratios considering spatial-temporal dependencies. IEEE Trans Intell Transp Syst. 2025;25(11):18236–51. doi:10.1109/tits.2024.3453495.
25. Mittal K, Jain A, Vaisla KS, Castillo O, Kacprzyk J. A comprehensive review on type 2 fuzzy logic applications: past, present and future. Eng Appl Artif Intell. 2020;95(1):103916. doi:10.1016/j.engappai.2020.103916.
26. Jovanović A, Kukić K, Stevanović A, Teodorović D. Restricted crossing U-turn traffic control by interval Type-2 fuzzy logic. Expert Syst Appl. 2023;211(2):118613. doi:10.1016/j.eswa.2022.118613.
27. Chala TD, Kóczy LT. Intelligent fuzzy traffic signal control system for complex intersections using fuzzy rule base reduction. Symmetry. 2024;16(9):1177. doi:10.3390/sym16091177.

28. Panchal M, Prajapati P. FPA-DQN: a fairness-and pressure-aware dueling deep Q-network for adaptive traffic signal control using UAV-based trajectory data. Indian J Sci Technol. 2025;18(28):2257–72.
29. Kumar A, Raj R, Mohammadzadeh A. Recent advancements in type-3 fuzzy logic systems: a comprehensive review. IEEE Trans Emerg Top Comput Intell. 2024;9(4):2676–89. doi:10.1109/tetci.2024.3433514.