

Multi-Intersection Traffic Signal Controls in Smart Cities Based on Graph Convolution Collaborative Communication

Wenfang Jing¹ and Hengli Duan²

¹ Associate Professor, School of Artificial Intelligence and Electronic Engineering, Sichuan Technology and Business University, Chengdu, 611745, China

² Associate Professor, School of Artificial Intelligence and Electronic Engineering, Sichuan Technology and Business University, Chengdu, 611745, China, E-mail: hengliudian11@outlook.com (corresponding email).

Project Management

Received February 8, 2026; revised June 1, 2026; accepted August 27, 2026

Available online September 10, 2026

Abstract: To solve the poor control effect of the traffic signal control model due to low resource sharing and weak node connections in the collaborative communication module of the traffic signal control model, this study first constructed a multi-intersection graph topology to enhance the connections between nodes. Then, the Graph Convolutional Network (GCN) and Long Short-Term Memory (LSTM) are used to extract features from the graph topology and predict changes in traffic flow. Finally, deep reinforcement learning is applied to autonomously control traffic signals. This study tested these control methods. The results showed that after using this method to optimize the traffic signal control model at four multi-intersections in the Traffic Flow Forecasting dataset, the average waiting time of vehicles at these intersections was reduced by 56.4%, 53.0%, 65.6%, and 61.3%, respectively. Moreover, it could reduce the average queue length by 56.2%, 61.9%, 43.3%, and 52.4%, while improving traffic efficiency. The research method can accurately control traffic signals at multiple intersections, reduce average vehicle waiting time, and reduce traffic congestion.

Keywords: Collaborative communication; graph convolutional network; long short-term memory network; multi-intersection; traffic signal control.

Copyright © Journal of Engineering, Project, and Production Management (EPPM-Journal).
DOI 10.32738/JEPPM-2026-203

1. Introduction

With the development of the social economy, more families can travel by car, leading to a gradual increase in the number of vehicles on the road and more frequent traffic congestion (Li, 2024). Traffic Signal Control (TSC) can manage traffic signals at intersections to regulate vehicle flow and prevent congestion (Zhang et al., 2024). Due to the large number of lanes at multiple intersections, road congestion is more pronounced, so these intersections have higher requirements for TSC (Zhai et al., 2025). At present, many scholars have studied TSC methods at multiple intersections. For example, Shi et al. (2024) applied the optimal signal control method to the multi-intersection tram problem to enhance tram system reliability. The results showed that this approach was beneficial for the tram to adopt flexible signal control strategies. To address traffic congestion at multiple intersections, Du et al. (2024) used sample-efficient reinforcement learning to build a TSC model. After the model controlled traffic signals, it can provide relatively fair traffic signal control without affecting performance. Furthermore, Yoon et al. (2025) employed multi-agent reinforcement learning to control traffic signals. The results showed that this control method was capable of re-balancing traffic demands, thereby enhancing the efficiency of the overall signal control system. Moreover, to address traffic congestion at multiple intersections and across multiple scenarios, Yu et al. (2025) used the rapid diffusion model to control traffic signals. It could alleviate the traffic congestion problem to a great extent. Based on the aforementioned related studies, current traffic signal control methods can regulate vehicle traffic. However, due to weak connections between nodes at intersections, inefficient collaborative communication, and poor coordination, challenges such as low control accuracy, long average vehicle waiting times, and long average queue lengths will arise. Therefore, this research aims to design a traffic signal control method that improves the efficiency of collaborative communication and strengthens connections between nodes.

The graph structure can connect various nodes, share node information, and improve the efficiency of collaborative communication (Cheng et al., 2024). GCN is a core method in Deep Learning (DL) for processing graph data, enabling the extraction of spatial features within graph structures (Mokayed et al., 2023). LSTM can extract temporal features from the graph structure and predict future trends in the data based on these features (Qin et al., 2025). The aforementioned

technologies are widely used across various fields. To predict the falling behavior of the human body during movement, Guendoul et al. (2025) used GCN to extract the features of human movement. The feature extraction accuracy could reach 98.5%. Zhao et al. (2024) used the LSTM algorithm to position and track the intelligent track system, thereby improving its tracking accuracy. The positioning accuracy in the test could reach 92.4%. Deep Reinforcement Learning (DRL) can achieve end-to-end control from image input to action output through interaction between the agent and the environment, and is widely used across various fields (Liang et al., 2024). Wei et al. (2024) utilized DRL technology to optimize the Internet of Vehicles task migration and resource offloading allocation and scheduling scheme, improving optimization and scheduling efficiency and ultimately achieving a 21.3% improvement.

To sum up, although the current TSC model can control vehicles at multiple intersections, the low efficiency of cooperative communication makes its effect not obvious, and its improvement in vehicle congestion is limited. To address the poor control model effect and low efficiency of cooperative communication, this study employs GCN-LSTM and DRL to construct a multi-intersection TSC model that improves traffic efficiency and reduces congestion. The innovation of the research is to first construct a multi-intersection graph structure, and then use GCN-LSTM to extract the features. The purpose is to predict traffic flow at the intersection, optimize signal light allocation and resource allocation strategies and dynamically adjust the power, frequency band, and other communication parameters of the communication lights at the intersection to achieve efficient collaborative communication. The research model then uses DRL to autonomously control traffic signals at multiple intersections, thereby reducing the Average Waiting Time (AWT) and AQL of vehicles at multiple intersections.

The structure of this research is as follows: Section 1 introduces the current state of traffic signal control and the problems to be addressed, and discusses related work on the research method. Section 2 details the graph structure, GCN algorithm, LSTM, DRL algorithm, and traffic signal control model used in the paper. Section 3 analyzes the test results. Finally, section 4 summarizes the main findings of this research, compares them with prior research results to assess performance, and proposes future research directions.

2. Methods

2.1. Collaborative Communication Technology Based on Graph Structure and GCN

With accelerating urbanization, urban traffic congestion has become increasingly severe. Especially at multi-intersection locations, unreasonable TSC will seriously affect transportation, increase travel costs and increase environmental pollution (Dai et al., 2025). Collaborative communication is a technology that improves communication efficiency by sharing resources and coordinating among multiple communication nodes or devices. Using this technology to optimize the TSC method at multiple intersections can improve the reliability of the TSC system. However, current cooperative communication technology still faces challenges such as weak node connectivity and limited resource sharing, which result in poor TSC performance. The graph structure can connect nodes and share information among them. This study constructs a graph topology for multi-intersections, as shown in Fig. 1.

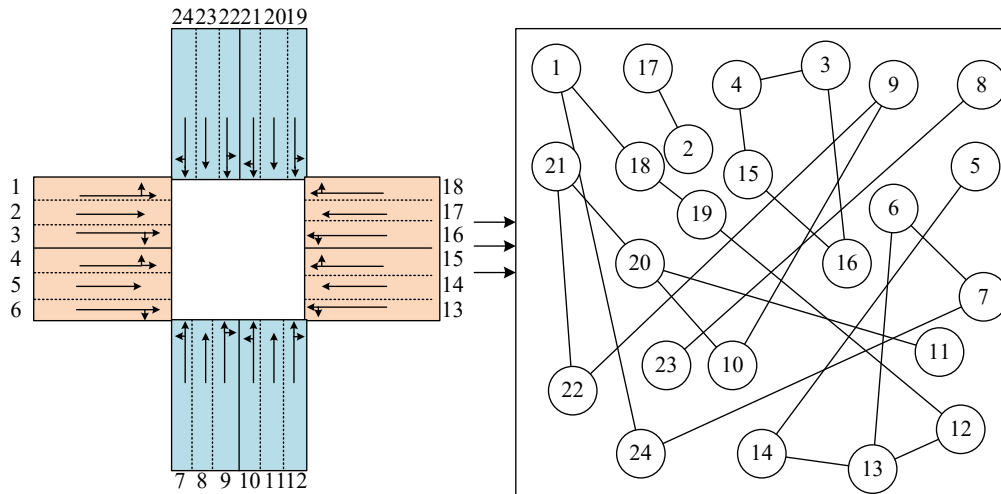


Fig. 1. Topology structure of multi-intersection diagram

In Fig. 1, the graph structure models each intersection in the multi-intersection as a node and the traffic road connection relationships between intersections as edges. The real-time traffic flow, average vehicle speed, and vehicle queuing information at the intersection can serve as node attributes to comprehensively reflect traffic status across multiple intersections. After the graph structure is established, the feature information within it needs to be extracted. GCN is a DL model that specializes in processing graph-structured data and can effectively capture local and global features in images (Wei et al., 2024). Through this GCN, traffic information features such as vehicle information and pedestrian information in multiple intersections can be extracted, and the traffic flow of the multiple intersections can be analyzed through the extracted features. GCN can extract graph-level features through its multi-layer graph convolution operations. The data extraction process is shown in Fig. 2.

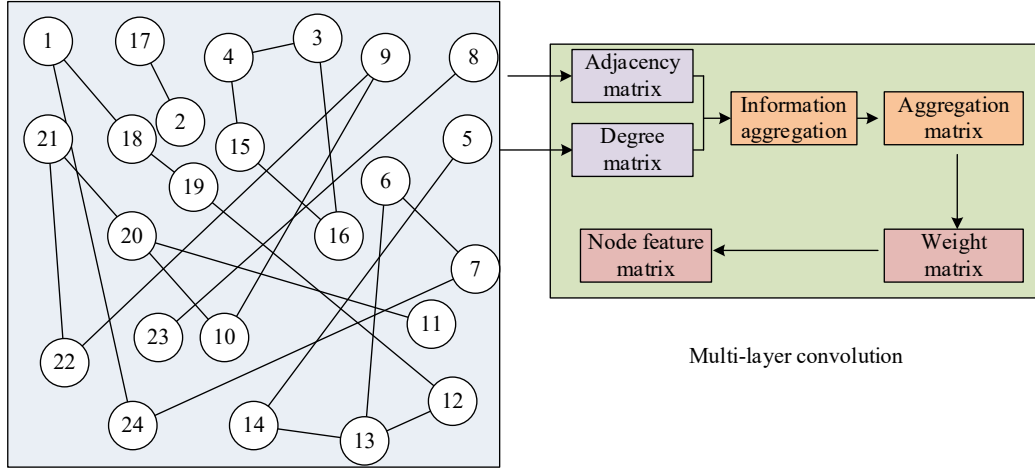


Fig. 2. Basic process of GCN

In Fig. 2, the constructed graph structure is used as input to the GCN, and its node features, adjacency, degree, and weight matrices are obtained from it. When GCN performs graph convolution operations, it first uses the adjacency matrix to aggregate the current node's neighborhood information. Each node's characteristics and those of its neighbors are weighed and averaged to obtain the aggregated characteristics, as shown in Eq. (1).

$$h_i^{(l+1)} = \sigma(\sum_{j \in N(i)} \frac{1}{\sqrt{d_i d_j}} W^{(l)} h_j^{(l)}) \quad (1)$$

In Eq. (1), $h_i^{(l+1)}$ and $h_i^{(l)}$ are the feature vector of node i in the next layer and layer l . $N(i)$ means the set of neighbor nodes of i . d_i and d_j are the degrees of i and node j , that is, the number of edges connecting the two nodes. $W^{(l)}$ is the weight matrix of layer l . σ denotes a nonlinear activation function. In the next step, the aggregated features are linearly transformed by the weight matrix to extract more useful feature representations. The linear transformation is shown in Eq. (2).

$$h_i^{(l+1)} = W^{(l)} (\sum_{j \in N(i)} \frac{1}{\sqrt{d_i d_j}} h_j^{(l)}) \quad (2)$$

Afterward, it uses a σ to map the linearly transformed features and maps the features to a nonlinear space, enhancing the model's expression ability and enabling it to learn more complex feature relationships. The nonlinear mapping is shown in Eq. (3).

$$h_i^{(l+1)} = \sigma(W^{(l)} (\sum_{j \in N(i)} \frac{1}{\sqrt{d_i d_j}} h_j^{(l)})) \quad (3)$$

The eigenvectors are transformed through Equations (1)-(3). Finally, the nonlinear mapping expression of the eigenvectors is obtained. Then, the activated features are used as the current output feature matrix for the next layer's graph convolution operation. The above operations are repeated until the set number of network layers is reached or the convergence conditions are met. The multilayer graph convolution process is given by Eq. (4).

$$H^{(l+1)} = \sigma(\hat{A} H^{(l)} W^{(l)}) \quad (4)$$

In Eq. (4), $H^{(l+1)}$ is the node characteristic matrix of layer l . $\hat{A}$ is the normalized adjacency matrix. In each layer of graph convolution, node features are updated by aggregating the adjacency matrix and applying feature transformations, gradually integrating more neighborhood information and learning more global and abstract features. After finally passing through multi-layer graph convolutions, the resulting node feature matrix contains information about each node's own characteristics and the adjacency matrices. However, in intersections, traffic flow has significant temporal dynamics, and the LSTM algorithm is a time-cyclic network that can capture long-term dependencies in traffic flow data (Chen et al., 2025). Therefore, this study uses an LSTM to perform time-series modeling on the spatial features extracted by a GCN, and then performs subsequent collaborative communication based on the modeling results. The basic process of collaborative communication based on LSTM-GCN is shown in Fig. 3.

In Fig. 3, the collaborative communication technology extracts spatial features from the graph structure using a GCN model, and then inputs the resulting spatial feature matrix into the LSTM. This module uses the memory unit and gating mechanism of LSTM to perform temporal modeling of spatial features and to capture the changing patterns and long-term dependencies of traffic flow at different time steps. In this process, the input gate, forget gate, and output gate of the LSTM are given by Eq. (5).

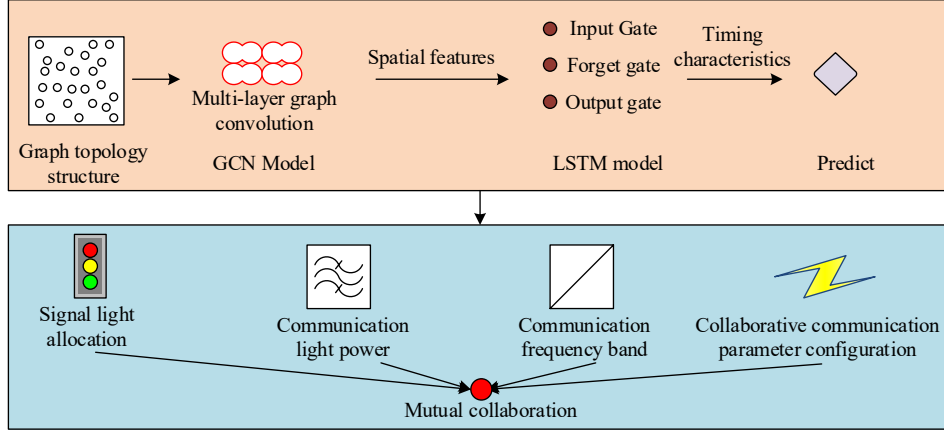


Fig. 3. GCN-LSTM collaborative communication process

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \end{cases} \quad (5)$$

In Eq. (5), x_t is the input feature at the t -th moment, that is, the spatial feature extracted by GCN. h_{t-1} is the hidden state at the previous moment. $W_i/W_f/W_o$ and $b_i/b_f/b_o$ are the weight matrices and bias terms of the input gate, forget gate, and output gate. Finally, based on the output results of LSTM, the future status of the intersection is predicted, the signal light configuration, resource allocation cooperation strategy, and communication parameters, such as power and frequency band of communication lights in the intersection, are dynamically adjusted to achieve efficient collaborative communication. The traffic state prediction is shown in Eq. (6).

$$y_{pred} = \text{Soft}(W_y \cdot h_t + b_y) \quad (6)$$

In Eq. (6), y_{pred} is the predicted distribution of future traffic states. h_t is the hidden state at the current moment. W_y and b_y are the weights and bias terms of the LSTM output matrix.

2.2. Multi-Intersection TSC Method

After optimizing traffic communication at multiple intersections using GCN-LSTM, traffic signals are controlled based on this optimized collaborative communication technology to alleviate congestion and improve traffic flow. DRL can connect the intelligent agent to the collaborative communication vector provided by GCN-LSTM, enabling autonomous decision-making and control of traffic signals (Bi et al., 2025). Fig. 4 shows the basic framework of the DRL-based TSC model.

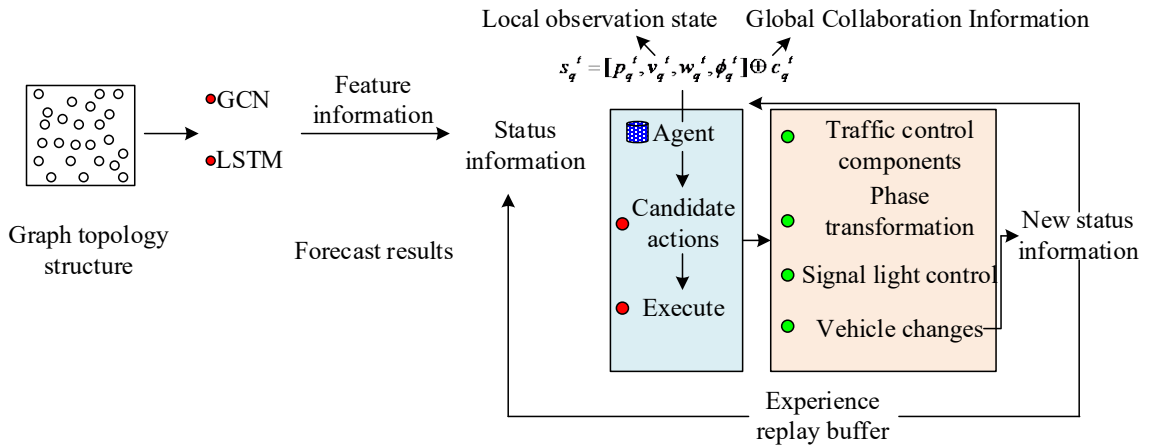


Fig. 4. Basic framework of TSC model

In Fig. 4, the control model first converts the multi-intersection spatial features proposed by GCN and the temporal features obtained by LSTM into the state information required by DRL. The agent in DRL then selects an action to execute based on GCN-LSTM predictions to determine the optimal traffic signal phase for the current traffic state. The agent in DRL is composed of the local observation state and global collaboration information at that moment, as shown in Eq. (7).

$$s_q^t = [p_q^t, v_q^t, w_q^t, \phi_q^t] \oplus c_q^t \quad (7)$$

In Eq. (7), s_q^t is the observation state of agent q , that is, vehicle q at time t . p_q^t is the vehicle queue length. v_q^t is the average speed of vehicles. w_q^t is the AWT. ϕ_q^t is the current signal phase. p_q^t , v_q^t , w_q^t , and ϕ_q^t are all local observation states of the vehicle. c_q^t is the global cooperation vector. The global collaboration vector c_q^t is formed by concatenating the spatial features of GCN with the temporal features of LSTM. With this information, the TSC component in the frame switches to the new signal phase to control the traffic light. After the traffic lights are controlled, the vehicle will drive on a specific road according to the changes in the traffic lights. After the vehicle moves, position changes will occur, and new status information will be generated in the traffic environment. The returned status information is the reward for the previous time-step action and is also the input to the next time step of the control model. This information is stored in an experience replay pool, which is used to regularly update the control model as the agent learns. The interaction trajectory in the experience replay pool is shown in Eq. (8).

$$T_q^t = [s_q^t, a_q^t, r_q^t, s_q^{t+1}, done] \in M \quad (8)$$

In Eq. (8), T_q^t is the transfer sample of q of agent t at time moment. s_q^{t+1} is the new state of the agent at time $t + 1$ after executing action a_q^t . $done$ is 0 or 1, which is the end of the round. M is the experience replay pool of DRL. The agent in DRL is shown in Fig. 5.

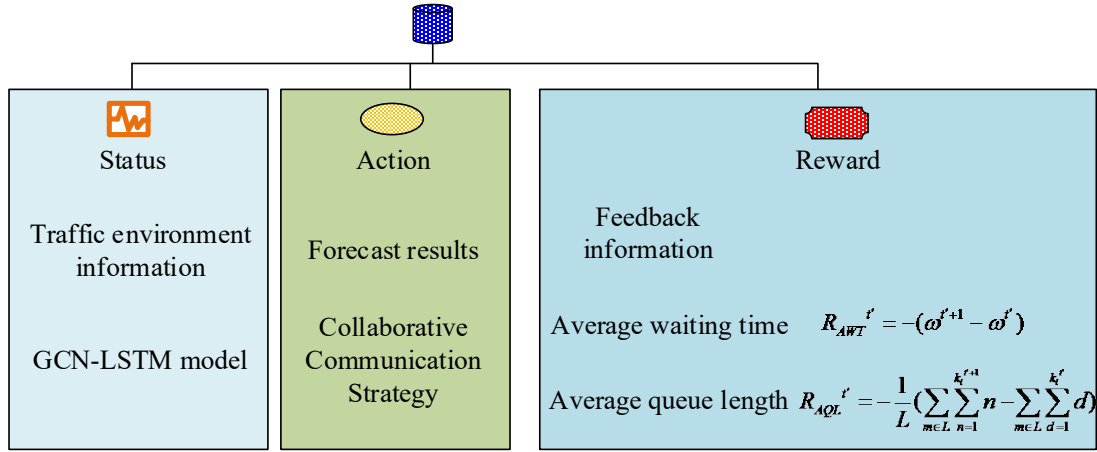


Fig. 5. Intelligent agent design

In Fig. 5, the DRL agent comprises states, actions, and rewards. The state is the traffic-environment information obtained via GCN-LSTM. The action is the TSC strategy and collaborative communication strategy predicted by the GCN-LSTM model based on traffic information in the action space. The final reward is feedback on various data obtained by the DRL agent after it performs an action, including the vehicle's AWT and AQL. The AWT refers to the time a vehicle waits in the waiting area before passing the intersection. If the waiting time is too long, the vehicle consumes extra fuel. The reward function of the AWT is shown in Eq. (9).

$$R_{AWT}^{t'} = -(\omega^{t'+1} - \omega^{t'}) \quad (9)$$

In Eq. (9), $R_{AWT}^{t'}$ is the reward expression function of AWT. ω^t is the vehicle's AWT in the state matrix at time step t' . The AQL denotes the vehicles at the intersection. Queues with higher AQL can receive higher priority to coordinate vehicle waiting times at the intersection. The reward function of AQL is shown in Eq. (10).

$$R_{AQL}^{t'} = -\frac{1}{L}(\sum_{m \in L} \sum_{n=1}^{k_l^{t'+1}} n - \sum_{m \in L} \sum_{d=1}^{k_l^{t'}} d) \quad (10)$$

In Eq. (10), $R_{AQL}^{t'}$ is the reward function of vehicle AQL. L is a multi-intersection lane set. l is a lane in the set L . $k_l^{t'}$ is the number of waiting vehicles in lane l in the state matrix at time step t' . $\sum_{d=1}^{k_l^{t'+1}} d$ and $\sum_{d=1}^{k_l^{t'}} d$ are the accumulation of relevant quantities of vehicles waiting on lane l at time steps $t' + 1$ and t' , that is, the accumulation of n . In DRL, after each action, the fluctuation range of the reward function value is less than 5%, and the fluctuation range remains below this threshold in consecutive rounds. In this case, DRL can be regarded as converging, and the final traffic signal control strategy can be obtained. The training process of the TSC model based on GCN-LSTM and DRL is shown in Fig. 6.

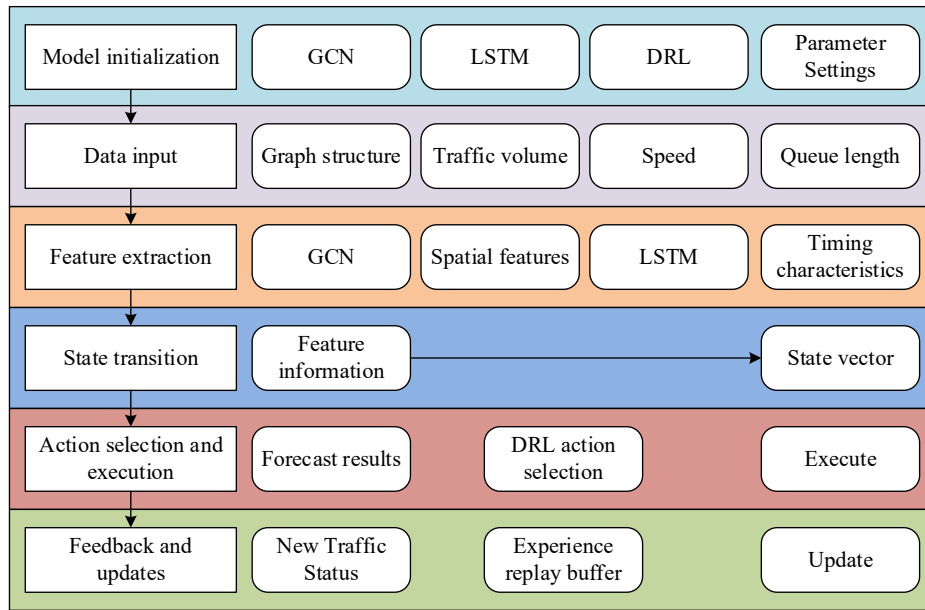


Fig. 6. Training steps of TSC model

In Fig. 6, the training process consists of six steps: model initialization, data input, feature extraction, state transition, action selection and execution, and feedback and update. During model initialization, the GCN parameters, the number of LSTM hidden layers, and the DRL learning rate and discount factor are set. The agent's experience replay pool is initialized. Data input is real-time traffic data, such as traffic volume, speed, and vehicle queue length at intersections, represented as a graph structure for the GCN-LSTM model. In this model, GCN extracts spatial features from the graph structure, and LSTM extracts temporal relationships from the spatial feature matrix, yielding temporal features. State transition converts the features extracted by GCN-LSTM into the state vector required by DRL. This vector includes traffic flow, speed, vehicle queue length at the current intersection, and historical traffic trends and traffic status at adjacent intersections. Action selection and execution involve DRL selecting an action based on the current state and GCN-LSTM predictions, and applying it to the TSC system by switching the corresponding signal lights and adjusting the signal timing. It calculates rewards based on the traffic status after the action is executed, then stores the status, actions, and rewards in the experience replay pool, and updates the DRL policy network using the replay pool to achieve adaptive signal control.

3. Results

3.1. Performance analysis of GCN-LSTM

To analyze the actual effect of the multi-intersection TSC model based on GCN-LSTM and DRL, the feature-extraction capability of GCN-LSTM in the multi-intersection traffic environment and its predictive performance are evaluated. The experimental environment configuration is shown in Table 1.

Table 1. Environmental parameters

Project	Index	Allocation
Hardware environment	CPU	Intel Core i7-12700K
	GPU	NVIDIA RTX 309
	RAM	64GB DDR4
	OS	Windows 10
Software environment	DL framework	PyTorch 1.12
	ML library	Scikit-learn 1.1
	Data analysis software	Python 3.9

In the experiment, the GCN layer parameters are set to 2. The first-layer hidden dimension is 64; the second-layer feature dimension is 128, and the ReLU activation function is used. The LSTM hidden layer dimension is 128, the step size is 10, the optimizer is Adam, the learning rate is 0.001, the batch size is 32, and the training rounds are 100. Moreover, the size of DRL's experience replay buffer is 10,000, the initial exploration rate of DRL is 0.9, and the attenuation factor is 0.9. The experiment then selects the Traffic Flow Forecasting Dataset (TFFD) from the UCI machine learning repository, which contains intersection traffic flow, signal light status, and other data from multiple cities (the dataset URL is

<https://archive-beta.ics.uci.edu/dataset/734/traffic%2Bflow%2Bforecasting-1>). This dataset was automatically collected and generated by the traffic management Department of the Northern Virginia/Washington, D.C., metropolitan area in the United States, using 36 Inductive Loop Detectors deployed on two major highways (I-66 and I-495). The collection period was from January 1, 2015, to December 31, 2015, with a sampling frequency of 15-minute aggregation and a total of 35,040 time steps. The dataset contains 47 input features: 10 dimensions of historical traffic flow, 10 dimensions of time-encoding attributes, 10 dimensions of road spatial attributes, 1 dimension of lane number, and 1 dimension of road name. The overall missing rate in the dataset is 2.3%, which is randomly distributed. Linear interpolation is used to fill in short-term missing data. The historical mean method for the same period is used to fill in missing data over a long period. The 3- σ principle and the box-and-whisker method are used to detect and eliminate outliers. The data is normalized using the Min Maxb normalization method. The experiment selects 300 records for each of 4 intersections (A, B, C, and D) from TFFD, and divides them into a training set, test set, and verification set in a ratio of 8:1:1. During testing, the data-level feature extraction effect of GCN-LSTM in four intersections is shown in Fig. 7.

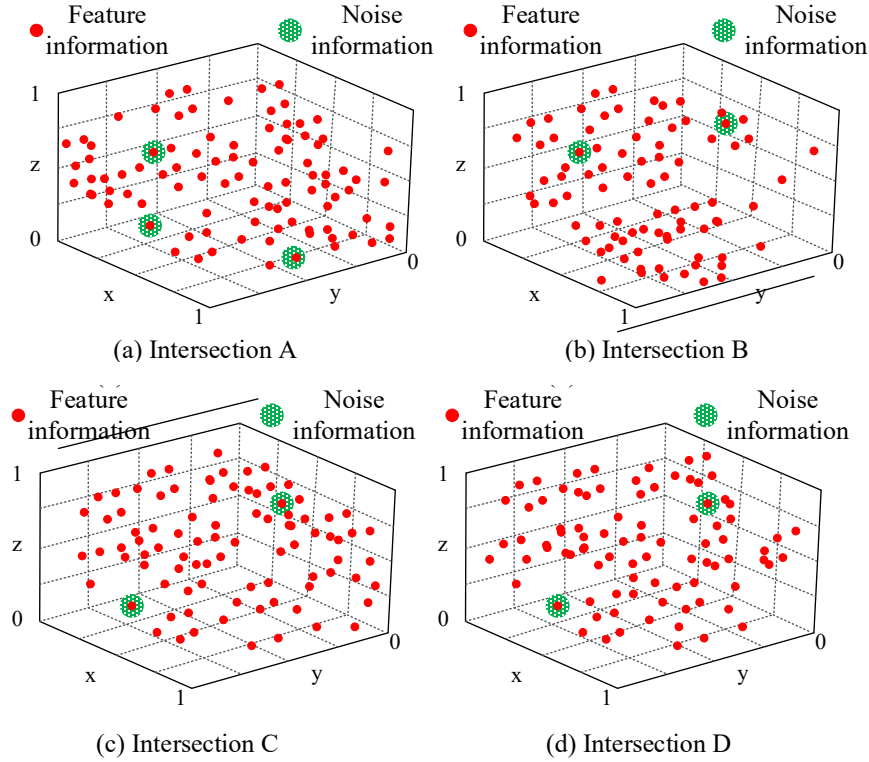


Fig. 7. Data feature extraction effect

In Fig. 7(a), when GCN-LSTM is used to extract feature information in intersection A, the extraction effect is better, and only a very small amount of noise information is extracted. Combined with Figs. In 7(b)-(d), when GCN-LSTM extracts features at intersections B, C, and D, its performance is good, and it extracts only a very small amount of irrelevant information. GCN-LSTM can accurately extract features from intersection data. Fig. 8 presents the prediction accuracy and prediction time consumption of GCN-LSTM for traffic flow in multiple intersections.

In Fig. 8(a), the prediction accuracy of GCN-LSTM for the traffic flow in intersections A, B, C, and D is 92.1%, 94.3%, 92.8%, and 94.2%. The prediction accuracy exceeds 90%, and the performance is better. In Fig. 8(b), when GCN-LSTM makes predictions, the prediction time is also shorter, both below 50ms. GCN-LSTM can accurately predict traffic flow at intersections over short time horizons and provide data support for subsequent TSC.

3.2. TSC Effect Analysis

After verifying the prediction performance of GCN-LSTM, the control effect of the TSC model based on GCN-LSTM-DRL is analyzed. During the test, the dataset and experimental environment configuration are the same as those of the GCN-LSTM performance test. First, after passing TSC, the vehicle AQL at four intersections is analyzed, as shown in Fig. 9.

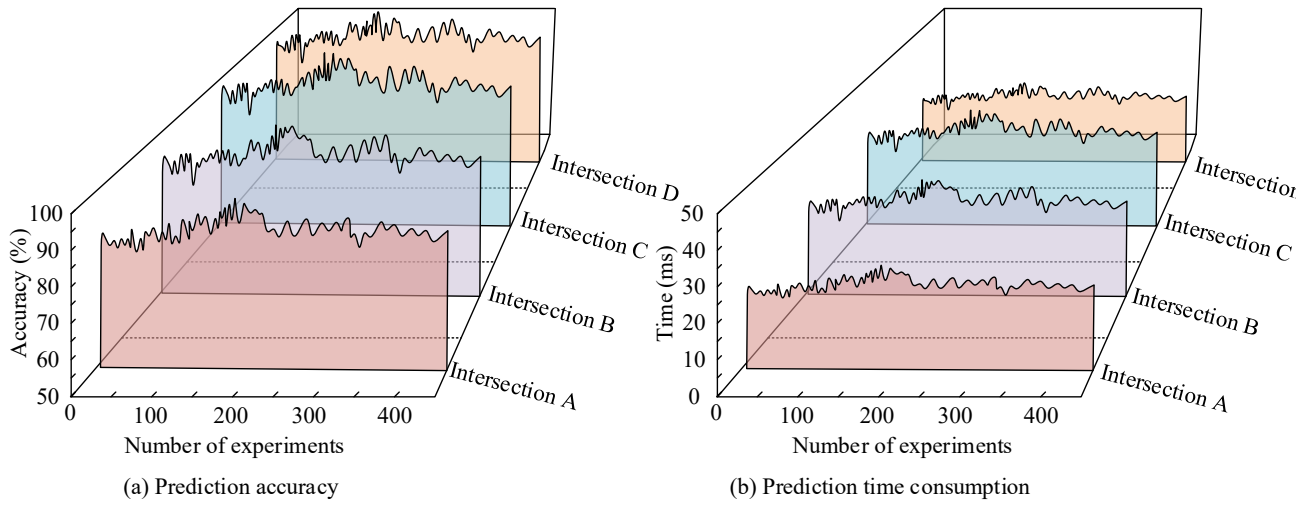


Fig. 8. Prediction accuracy and prediction time

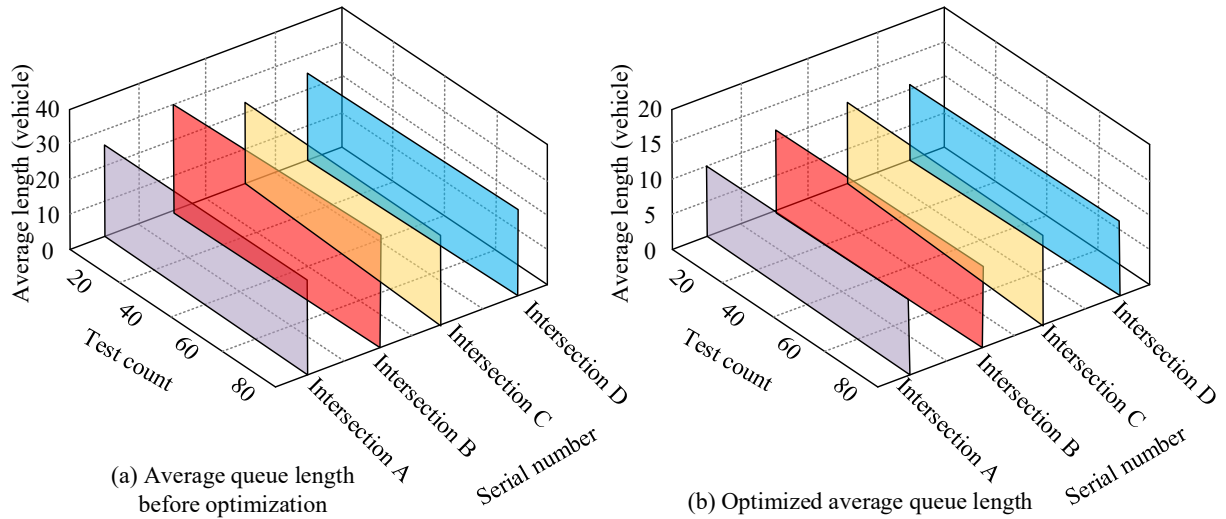
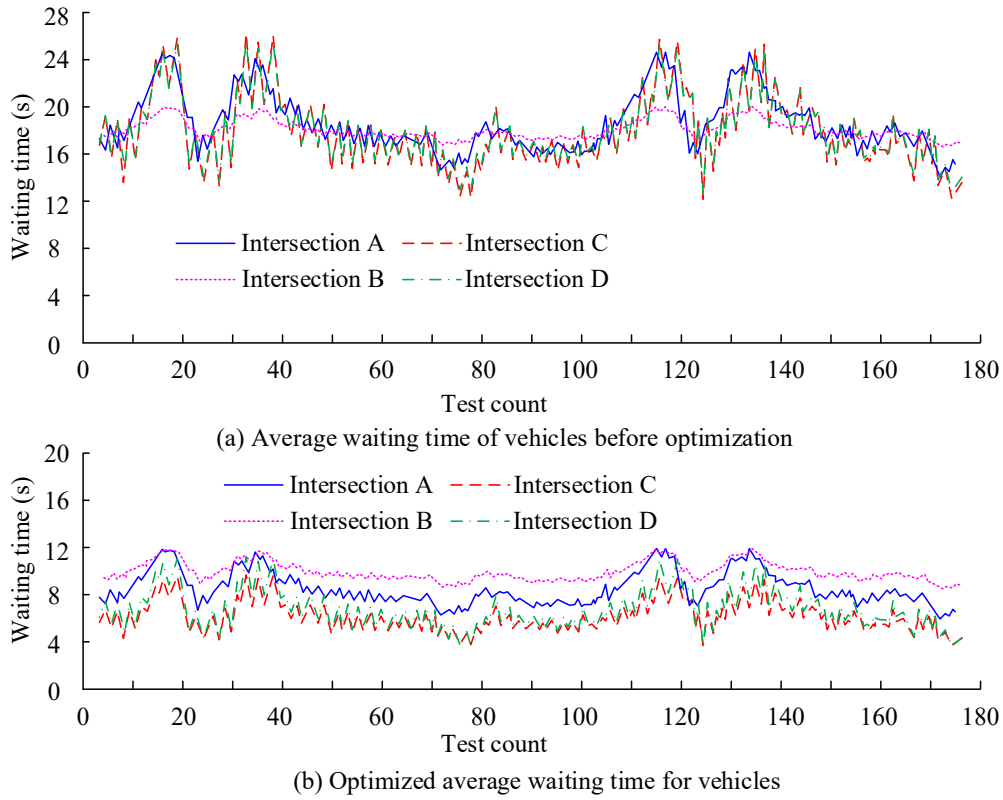


Fig. 9. AWT of vehicles before and after optimization

In Fig. 9(a), before optimization, the vehicle AQLs at intersections A, B, C, and D are 23.5, 30.2, 21.7, and 24.6 vehicles, respectively, and the vehicle AQL is longer. In Fig. 9(b), after using GCN-LSTM-DRL to control the traffic signal at four multi-intersections, the AWT of vehicles in the four multi-intersections is reduced. In Intersection A, the AWT is reduced to 10.3 vehicles, a 56.2% reduction. The AWT at intersection B decreases from 30.2 vehicles to 11.5 vehicles, a 61.9% reduction. At intersections C and D, the AWT of vehicles is reduced to 12.3 and 11.7 vehicles, a decrease of 43.3% and 52.4%, respectively. After optimizing TSC, vehicle AWT will decrease, and traffic efficiency will improve. The changes in vehicle waiting time before and after optimization are analyzed, as shown in Fig. 10.

In Fig. 10(a), before optimization, the AWTs at multi-intersections A, B, C, and D are 23.4s, 19.8s, 24.1s, and 21.7s, respectively. After many tests, it was found that the vehicle's AWT varies widely, and the curve fluctuates significantly. In Fig. 10(b), after GCN-LSTM-DRL controls traffic signals, the AWT of vehicles in the four intersections is reduced to 10.2 s, 9.3 s, 8.3 s, and 8.4 s, which is reduced by 56.4%, 53.0%, 65.6%, and 61.3%. After optimizing the control model, the change amplitude of vehicle AWT is relatively low in the four multi-intersections. In summary, GCN-LSTM-DRL can reduce the AWT of vehicles by controlling traffic signals at four multi-intersections, thereby optimizing traffic efficiency and reducing road traffic congestion. Finally, the performance of the GCN-LSTM-DRL-based TSC is compared with the Fixed-Time (FT), the Actuated Control (AC), and the Transformer-based model. When testing the four control models, select intersection A from the dataset. To ensure fairness and reproducibility of experimental comparisons, all comparison models are set in the same hardware environment and with the same dataset partitioning. During the test, the signal period in FT is set to 120 seconds, and the number of phases is set to 4. The minimum green light time of AC is set to 15s, the maximum green light time is 60s, and the detector position is set 50m upstream of the parking line. The Transformer's encoder layer number is 4, the number of attention heads is 8, and the learning rate is 0.001. After parameter verification, the four models are completely consistent in input dimension (47 dimensions), test sample size (30 items), evaluation index calculation method, and running frequency (5 independent experiments). The test results are shown in Table 2.

**Fig. 10.** AWT of vehicles before and after optimization**Table 2.** Control model performance test results

Model	Improvement in traffic efficiency	Decreased waiting time	Reduction in queue length
FT	12.3%	10.2%	9.4%
AC	18.1%	17.3%	12.3%
Transformer	21.5%	21.4%	19.6%
GCN-LSTM-DRL	62.1%	56.4%	56.2%

In Table 2, compared with other TSC models, the TSC model GCN-LSTM-DRL achieves the largest improvement in traffic efficiency, averaging 62.1%, which is higher than the 12.3% of FT, 18.1% of AC, and 21.5% of Transformer. Moreover, after GCN-LSTM-DRL controls traffic signals, vehicle waiting time on the road can be reduced by 56.4%, a higher reduction than that of the other models. In addition, GCN-LSTM-DRL can also reduce the queue length of vehicles on the road by 56.2%, while the reduction rates of FT, AC, and GCN-LSTM-DRL is 9.4%, 12.3%, and 19.6%, respectively, all lower than that of GCN-LSTM-DRL. The test is conducted at an actual multiple intersection, and the schematic diagram of the test intersection is shown in Fig. 11.

According to Table 3, after applying different methods to control traffic signals at multiple intersections, the GCN-LSTM-DRL model reduces vehicle delay to 10.2 seconds, which is higher than FT's 23.4 seconds, the AC model's 19.2 seconds, and the Transformer model's 15.8 seconds, and it achieves the best performance. During peak periods, the GCN-LSTM-DRL model reduces carbon emissions by only 391kg, which is much lower than that of other models. And after using this method, the number of manual interventions at multiple intersections is reduced to only 2 per day, significantly reducing the commander's workload. This shows that GCN-LSTM-DRL can better control traffic signals, thereby improving traffic efficiency and reducing congestion.

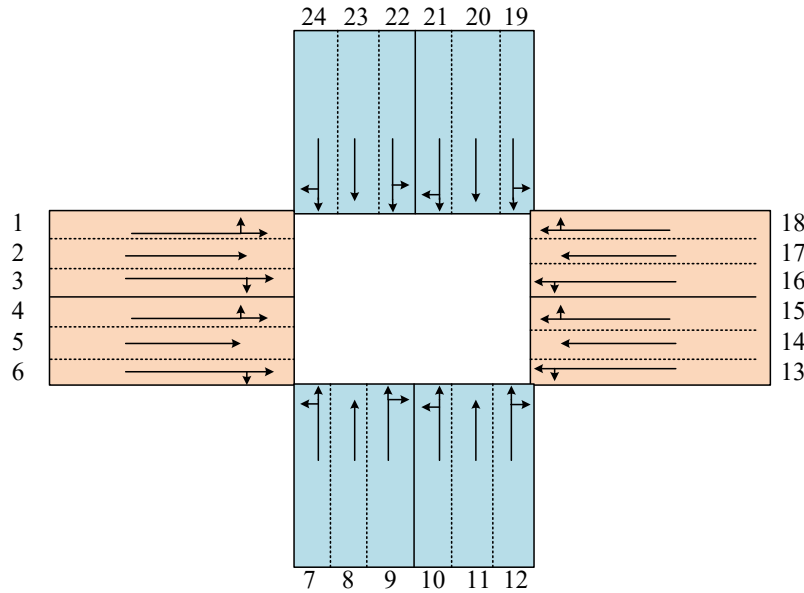


Fig. 11. Schematic diagram of multiple intersections

Based on Fig. 11, the actual control effects of the four models were analyzed and are shown in Table 3.

Table 3. Actual performance analysis

Model	FT	AC	Transformer	GCN-LSTM-DRL
Vehicle delay duration (s)	23.4	19.2	15.8	10.2
Regional Equity Index	0.62	0.68	0.71	0.89
Peak carbon emissions (kg)	874	713	598	391
Plan generation time (h)	72	48	24	2
Human intervention demand (times/day)	12	8	6	2

4. Conclusion

To address poor TSC performance due to low resource sharing and weak connections among collaborative nodes in the traffic signal cooperative communication module, this study first constructed a graph structure of multiple intersections. It increased each node's connectivity, improved the efficiency of collaborative communication, and used GCN-LSTM to extract graph-level features. By predicting traffic flow at intersections, parameters for communication lights at each intersection were determined, thereby improving the efficiency of collaborative communication. In addition, DRL technology was used to control traffic signals with optimized collaborative communication information, enabling autonomous decision-making. This study first analyzed the feature extraction and information prediction effects of GCN-LSTM. Using GCN-LSTM had a better effect on the data features in the four multi-intersections in TFFD. When predicting traffic flow-related information at intersections A, B, C, and D, the prediction accuracy was 92.1%, 94.3%, 92.8%, and 94.2%. The prediction accuracy was high, and when this method was used, the prediction time was less than 50ms. Compared with the above results, when Arif T et al. (2025) used YOLOv8 and LSTM algorithms to predict traffic flow on traffic roads, the prediction accuracy of this study was higher. The method proposed by Arif T et al. only used LSTM to extract the time series information in traffic flow when predicting information, and predict future traffic flow. This ignores spatial features, resulting in lower prediction accuracy than the GCN-LSTM used this time.

After verifying the performance of GCN-LSTM, the effect of the TSC model optimized with GCN-LSTM-DRL was analyzed. After optimization, the model could reduce vehicle AQL at four multi-intersections, A, B, C and D, by 56.2%, 61.9%, 43.3% and 52.4%, respectively. Moreover, it could reduce waiting time by 56.4%, 53.0%, 65.6%, and 61.3%, respectively. This result was compared with the results of Liu et al. (2025) on the control effect of traffic signals. After using GCN-LSTM-DRL to optimize the TSC model, its AWT and vehicle AQL reductions were larger. Although the multi-agent reinforcement learning control model used by Liu W et al. could predict traffic flow, its prediction accuracy was low, only 82.3%. This leads to low accuracy of traffic flow analysis in traffic control models, reducing their control effectiveness. Comparison with FT, AC, and Transformer models showed that GCN-LSTM-DRL achieved the highest improvements in communication efficiency, waiting time reduction, and vehicle AQL at multi-intersections, reaching 62.1%, 56.4%, and 56.2%, respectively.

Based on the above analysis, urban management departments can transform traffic signal control from "experience driven" to "data model dual driven". When controlling traffic signals at multiple intersections, management personnel can first deploy GCN-LSTM-DRL in the traffic light intersection controller and connect it to the 120/119 dispatch system to

achieve priority passage for emergency vehicles. They can establish a district level collaborative control unit to eliminate the imbalance of the main branch, set up an "algorithm trainer" position to be responsible for model iteration and anomaly labeling, incorporate the regional fairness index into performance evaluation, and promote cross departmental data sharing to break information silos. The decision-making of traffic signal control has shifted from lagging governance of "perception response" to proactive governance of "prediction prevention". The analysis shows that the GCN-LSTM-DRL can optimize the efficiency of collaborative communication, thereby improving the control effect on traffic signals, reducing the AWT and AQL of vehicles at multi-intersections, and improving road traffic efficiency. However, this study does not account for external data, such as daytime brightness and temperature, across multiple intersections. In the future, relevant data can be obtained in collaboration with the meteorological department to more comprehensively optimize the control model.

Author Contributions

Wenfang Jing contributed to conception and design, material preparation, and the draft of the manuscript. Hengli Duan contributed to conception and design, data collection and analysis, manuscript review and revision.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Declaration of Artificial Intelligence (AI) Tools

The authors confirm that no AI tools were used in the preparation of this manuscript.

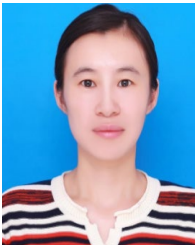
References

- Arif, T., Muhammad Mussa, A. S., Rahman, M. and Joarder, M. A. (2025). Vision-based real-time traffic flow monitoring system for road intersections in Dhaka city. *Applied Intelligence*, 55(11), 800. <https://doi.org/10.1007/s10489-025-06631-x>
- Bi, Z., Ma, F., Guan, J., Wu, J., Li, J., and Li, F. (2025). Apple leaf disease severity grading based on deep learning and the DRL-watershed algorithm. *Scientific Reports*, 15(1), 1–19. <https://doi.org/10.1038/s41598-025-15246-8>
- Chen, X., Wang, B., Zhang, L., Lai, Y., Shi, T., and Zhu, M. (2025). Adaptive network slicing and LSTM-based resource allocation for real-time industrial robot control in 6G networks. *IET Communications*, 19(1), 80–93. <https://doi.org/10.1049/cmu2.70080>
- Cheng, T., Zeng, G., and Sun, Z. (2024). A big graph clustering method to support parallel processing by perceiving graph's application algorithm semantics. *Journal of Supercomputing*, 80(2), 2838–2861. <https://doi.org/10.1007/s11227-023-05572-x>
- Du, X., Li, Z., Long, C., Xing, Y., Yu, P. S., and Chen, H. (2024). FELight: Fairness-aware traffic signal control via sample-efficient reinforcement learning. *IEEE Transactions on Automatic Control*, 36(9), 4678–4692. <https://doi.org/10.1109/TKDE.2024.3376745>
- Dai, J., Tian, X., Liu, L., Zhang, H., Fu, J., and Yu, M. (2025). The intelligent traffic safety system based on 6G technology and random forest algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 26(10), 17244–17254. <https://doi.org/10.1109/TITS.2024.3451184>
- Guendoul, O., Zobi, M., Ait Abdelali, H., Tabii, Y., Rachid, O. H. T., and Bourja, O. (2025). Capturing spatio-temporal patterns of falls individuals using efficient graph convolutional network model. *Applied Intelligence*, 55(11), 1–16. <https://doi.org/10.1007/s10489-025-06316-5>
- Li, Y. (2024). Constructing the intelligent expressway traffic monitoring system using the internet of things and inspection robot. *Journal of Supercomputing*, 80(7), 8742–8766. <https://doi.org/10.1007/s11227-023-05794-z>
- Liang, G., Hu, J., Zhao, Y., and Yang, K. (2024). Intelligent link adaptation for integrated data and energy transfer: An enhanced DRL approach for long-term constraints. *IEEE Transactions on Communications*, 72(11), 6956–6972. <https://doi.org/10.1109/TCOMM.2024.3407204>
- Liu, W., Zhang, C., Fang, W., Zhou, K., Li, Y., and Zhan, F. (2025). Vehicle-level fairness-oriented constrained multi-agent reinforcement learning for adaptive traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 26(4), 4878–4890. <https://doi.org/10.1109/TITS.2025.3544223>
- Mokayed, H., Quan, T. Z., Alkhaled, L., and Sivakumar, V. (2023). Real-time human detection and counting system using deep learning computer vision techniques. *Artificial Intelligence and Applications*, 1(4), 221–229. <https://doi.org/10.47852/bonviewAIA2202391>
- Qin, Q., Li, Y., Mi, Y., Shen, J., Wu, K., and Wang, Z. (2025). SQL injection detection algorithm based on Bi-LSTM and integrated feature selection. *Journal of Supercomputing*, 81(4), 1–32. <https://doi.org/10.1007/s11227-025-07109-w>
- Shi, L., Zhou, S., Chang, Y., and Zhang, Q. (2024). Study on tram travel-time reliability characteristics for optimal traffic signal control. *Transportation Research Record*, 2678(4), 43–55. <https://doi.org/10.1177/03611981231182957>
- Wei, M., Wen, M., and Zhang, Y. (2024). A novel spatial electric load forecasting method based on LDTW and GCN. *IET Generation, Transmission and Distribution*, 18(3), 491–505. <https://doi.org/10.1049/gtd2.13088>
- Wei, R., Qin, T., Huang, J., Yang, Y., Ren, J., and Yang, L. (2024). Resource allocation scheduling scheme for task migration and offloading in 6G cybertwin internet of vehicles based on DRL. *IET Communications*, 18(18), 1244–1265. <https://doi.org/10.1049/cmu2.12826>

- Yoon, J., Ahn, K., Lee, K., Park, J., and Yeo, H. (2025). Decentralized and communication-based multi-agent traffic signal control model employing a graph representation for the state. *IEEE Transactions on Intelligent Transportation Systems*, 26(5), 5947–5960. <https://doi.org/10.1109/TITS.2025.3548945>
- Yu, J. L., Wang, Z., and Zhang, R. (2025). DiffusionLight: A multi-agent reinforcement learning approach for traffic signal control based on shortcut-diffusion model. *Applied Intelligence*, 55(7), 527–539. <https://doi.org/10.1007/s10489-025-06359-8>
- Zhai, Z., Hao, R., Cui, B., and Wang, S. (2025). HGAT and multi-agent RL-based method for multi-intersection traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 26(5), 6848–6864. <https://doi.org/10.1109/TITS.2025.3526223>
- Zhang, Y., Zhou, Y., and Fujita, H. (2024). Distributed multi-agent reinforcement learning for cooperative low-carbon control of traffic network flow using cloud-based parallel optimization. *IEEE Transactions on Intelligent Transportation Systems*, 25(12), 20715–20728. <https://doi.org/10.1109/TITS.2024.3452430>
- Zhao, X., Tian, F., and Shao, Z. (2024). Positioning and tracking with ODE-LSTM algorithm for emerging smart rail systems. *Journal of Supercomputing*, 80(15), 21975–21995. <https://doi.org/10.1007/s11227-024-06296-2>



Wenfang Jing obtained her Master of Engineering degree in electronic and communication engineering from Sichuan University in 2013. She is currently an Associate Professor in the Communication Engineering program at Sichuan Technology and Business University. She has published academic papers in more than ten internationally renowned journals and conference proceedings, with research interests including machine learning, image processing, mobile communications and electronic technology.



Hengli Duan, Master of Engineering in Electronic and Communication Engineering from Southwest Jiaotong University, is currently an Associate Professor at Sichuan Technology and Business University. Her research focuses on artificial intelligence, network planning, optimization, and data communication networks, with over ten published papers, three co-authored monographs, two patents and more than ten software copyrights.