Graph Neural Network and Reinforcement Learning–Based Framework for Real-Time Traffic Congestion Detection and Police Dispatch Using Multi-Source Heterogeneous Data

Shuo Xu[1]*, Qilin Mao[2]
[1]Hunan Police Academy, Changsha 410137, Changsha, China
[2]Hunan Provincial Traffic Police Corps, Changsha 410100, Changsha, China
E-mail: abcdeooxnxn@163.com

Key words: multi-source heterogeneous data, graph convolutional network, traffic congestion detection, police response, reinforcement learning

Received: September 10, 2025

The fusion of multi-source heterogeneous data in high-speed transportation networks is essential for real-time congestion detection and rapid police response. Existing methods remain limited in data consistency, spatio-temporal pattern extraction, and path planning stability. This study proposes a congestion detection and police response framework driven by multi-source heterogeneous data. A dataset integrating flow sensors, road cameras, and Internet of Vehicles signals is constructed, with unified node, edge, and temporal features modeled through graph mapping. A spatio-temporal graph convolutional network (STGCN) with attention is employed to capture dependencies and enhance key road section representations, while a multi-task framework enables deep congestion pattern extraction. For response, geometric constraints guide path decoding, and proximal policy optimization (PPO)-based reinforcement learning achieves dynamic police dispatch. Experiments on a real expressway network with 6,120 roads and 580,000 samples show 92.4% ± 0.5 Accuracy, 89.6% ± 0.6 Topology Score, and 91.7% ± 0.6 F1-Response Score, surpassing baselines. The novelty lies in STGCN-based cross-modal fusion, geometric constraints, and the integration of PPO-based reinforcement learning. Rather than being a first-time application, the contribution is reflected in the technical integration of GNN with RL and the incorporation of constraint modeling for traffic police response, which distinguishes this framework from prior studies in emergency dispatch.

Povzetek: Članek predstavi večizvorski sistem za zaznavo zastojev in dinamično napotitev policije, ki združuje STGCN s pozornostjo, večopravilno učenje ter PPO-utrjevalno učenje. Na omrežju s 6.120 cestami doseže odlične rezultate.

1 Introduction

With the rise of multi-source heterogeneous data and intelligent analysis, traffic congestion detection and police response are shifting from statistical models to deep learning and graph neural networks. High-speed transportation networks are large-scale, with complex correlations and strong spatiotemporal dynamics. Traditional single-detector or local statistical methods face deficiencies in accuracy and timeliness [1]. In multi-source environments (e.g., vehicle networking, road monitoring, geomagnetic sensors), temporal consistency and spatial topology remain underutilized, limiting congestion detection and response efficiency [2].

Previous studies used speed monitoring, flow prediction, or pattern matching, but results degrade under non-stationary traffic due to noise and local modeling [3]. Police responses often rely on fixed routes or experience, making dynamic adaptation difficult and causing delays and resource waste. Thus, an intelligent framework integrating multi-source data is required for precise congestion detection and dynamic route optimization.

Graph Neural Networks (GNN) enable non-Euclidean modeling, capturing spatiotemporal dependencies through node aggregation and convolution [4]. Attention mechanisms highlight key sections and congestion chains, while reinforcement learning (RL) provides feedback-driven path optimization under complex constraints [5].

This paper proposes a framework of “multi-source fusion – graph feature extraction – congestion detection – police response optimization.” At the data level, multimodal fusion structures vehicle networking, video, and sensor data. At the feature level, congestion detection combines GNN and attention with multi-task learning. Path modeling introduces graph encoding with topological constraints to ensure rational scheduling. At the optimization stage, RL guides dynamic strategy for timely and accurate response. The key issues that this paper aims to address include: RQ1: Can multi-source data effectively model the spatio-temporal structure of high-speed transportation networks through GNN? RQ2: Can the attention mechanism and multi-tasking drive enhance the stability and accuracy of congestion detection? RQ3: Can reinforcement learning optimize the path planning of police response? The research innovation lies in: First, proposing a multi-module framework that collaborates graph convolution, attention, and reinforcement learning; Second, introduce topological constraints and feedback mechanisms to enhance the consistency of modeling logic; Thirdly, the innovation lies not in the first use of GNN and reinforcement learning for traffic policing, but in the integration of graph convolution, attention mechanisms, and reinforcement learning under topological and constraint-based modeling. This technical synergy provides a new direction for intelligent transportation and emergency governance.

2 Relevant work

In the research of traffic congestion detection and police response, multi-source heterogeneous data-driven methods have gradually become an important direction to break through the bottlenecks of traditional methods. Existing research mainly focuses on emergency dispatch optimization, multimodal data modeling, spatio-temporal feature extraction, and large-scale prediction methods, etc.

In the field of emergency dispatch research, Liu et al. (2020) proposed an ambulance dispatch framework based on deep reinforcement learning, which achieves optimal route decision-making by simulating complex traffic environments, effectively shortening the emergency response time and verifying the feasibility of reinforcement learning in police and emergency dispatch [6]. Sun and Liu (2025) utilized multimodal fusion and heterogeneous graph neural networks to detect and predict traffic anomalies on expressways, achieving high accuracy and stability in multi-source heterogeneous environments, providing a reference for modeling complex events in traffic scenarios [7].

In the field of multi-source data fusion and travel time estimation, Shi et al. (2017) proposed a heterogeneous data fusion method, combining loop detectors, GPS and floating vehicle data to model the travel time distribution under congestion conditions, thereby enhancing the adaptability to complex traffic scenarios [8]. Reis (2025) combines Internet of Things (iot) and artificial intelligence technologies to explore the fusion of multimodal data in green travel, effectively enhancing the safety and sustainability of the transportation system [9].

In the field of spatio-temporal feature extraction and congestion modeling, Guo et al. (2024) proposed a heterogeneous feature fusion network for road segmentation tasks, enhancing the topological consistency expression of traffic scenarios through a bidirectional feature transformation mechanism [10].To more intuitively demonstrate the differences between the existing research and the work of this paper, the core features of the main methods are summarized in Table 1.

Table 1: Comparison of typical methods

Method Name	Year / Dataset	Core Method	Metrics	Limitation
DRL-Dispatch [6]	2020 / Emergency vehicle	Deep RL for dispatch	Acc ≈ 89%	Weak cross-modal integration; low timeliness
Hetero-GNN [7]	2025 / Highway multimodal	Heterogeneous GNN fusion	Acc ≈ 88%	Limited feature interaction; weak topology
FusionNet [10]	2024 / Road segmentation	Heterogeneous feature fusion	Topo ≈ 86%	Poor generalization; low responsiveness
ST-Point [11]	2020 / Congestion event	Attention-based spatiotemporal model	F1 ≈ 85%	Weak topology propagation; poor scalability
Multi-Retentive [12]	2024 / Large-scale prediction	Multi-modal retentive network	Acc ≈ 90%	Unstable path optimization; limited real-time
GNN+RL (This paper)	2025 / Highway trunk network	GCN + attention + RL co-optimization	Acc 92.4% / Topo 89.6% / F1 91.7%	Validation scope limited to one region

Existing research has made progress in multi-source data fusion, spatio-temporal feature modeling, and emergency scheduling optimization. However, problems such as insufficient real-time adaptability, limited path generation, and imperfect multimodal feature interaction mechanisms still exist. This paper will combine graph neural networks and reinforcement learning to explore the paths of cross-modal fusion, key feature extraction and strategy optimization, and promote the intelligent development of high-speed traffic congestion detection and police response systems.

3 Traffic congestion feature detection mechanism driven by multi-source heterogeneous data

3.1 Construction of traffic flow network graph and setting of node features

The construction of the graph structure of the traffic flow network relies on the data expression requirements of the graph neural network, which needs to encode the road network, traffic flow and multi-source sensor data into a node-edge structure. In the context of expressways, nodes represent the locations of road intersections, monitoring points or detectors, while edges indicate the connection relationships of road sections and the direction of traffic flow. The graph structure form is defined as $G=(V,E)$, where V is the set of nodes and E is the set of edges. Each node generates an initial feature vector by extracting traffic attributes and geometric information, which is specifically defined as:

$$X_i=\{q_i,v_i,d_i,c_i\} \tag{1}$$

Among them, q_i is the traffic flow at node i, measured in vehicles per hour. v_i is the average speed of vehicles at node i, measured in kilometers per hour. d_i is the road density at node i, calculated as the ratio of traffic flow to speed. c_i is the road section type encoded as a one-hot vector for node i, representing road types (e.g., expressways, ramps, main roads). Flow and speed are collected and normalized by loop detectors and vehicle network signals. Density is calculated based on the ratio of flow to speed, and the type of road section is provided by the traffic geographic information system. After the above features are concatenated, node input vectors are formed to ensure the uniformity of feature dimensions.

To enhance the geometric consistency of graph construction, the establishment of edges is based on road connection relationships and traffic flow directions, combined with GIS databases and sensor annotations to generate, ensuring the structural connectivity of the network. The spatial positional relationship between nodes is position-embedded through normalized coordinate differences to enhance the perception ability of graph convolution on geometric topology:

$$P_{ij}=\left(\frac{x_i-x_j}{W},\frac{y_i-y_j}{H}\right) \tag{2}$$

Among them, (x_i,y_i) and (x_j,y_j) are the coordinates of node i,j respectively, and W,H is the width and height of the regional range, which are used to normalize the characteristics of the road network at different scales. x_i,y_i are the geographic coordinates of node i. W,H are the width and height of the region, used to normalize the coordinates. This formula eliminates the influence of different urban road scales on the model input during the graph construction stage.

As shown in Figure 1, the construction process of the traffic flow network includes: multi-source data collection→road network mapping→node setting→edge relationship generation → node feature vector construction. Data collection comes from loop detectors, surveillance cameras, GPS signals from the Internet of Vehicles, and historical accident records. Node setting is accomplished through mapping the traffic topology to the positions of monitoring points. The edge relations are automatically reasoned and corrected under the constraints of road connection logic and traffic rules. Finally, a unified node feature vector matrix is generated as the input of the graph neural network.

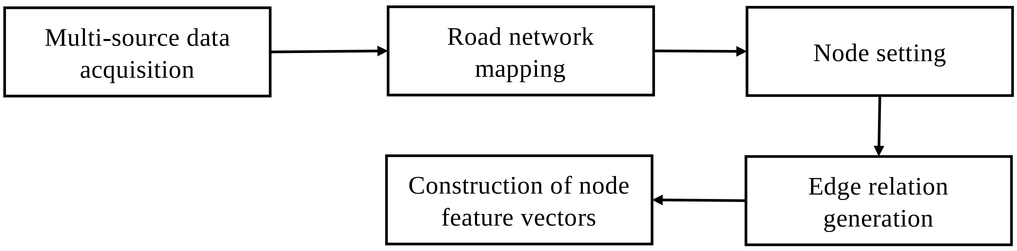

Figure 1: Flowchart of traffic flow network construction

In feature quantization, traffic flow and speed are normalized to the interval [0,1], road density is calculated based on the ratio of flow to speed, and road section types are mapped to 4-dimensional unique heat vectors. Location embedding ensures the comparability of transportation networks in different cities and on different road scales. The above design ensures the integrity and reproducibility of node features, providing a solid foundation for the subsequent extraction of spatio-temporal congestion patterns.

The traffic network is modeled as a spatio-temporal graph. After normalization, node features (speed, flow, occupancy) are scaled to [0,1]; e.g., 90 km/h and 1800 veh/h become (0.75, 0.6). In Equation (2), www and hhh denote lane width (3.5 m) and section length (500 m), ensuring consistent scaling.

To further clarify the process, the following pseudo-code and feature table are provided:

Pseudo-code for Graph Construction:

```
for each road_section in road_network:
    node = create_node(road_section)
    features = [flow, speed, density, road_type]
    normalize(features)
    add_to_graph(node, features)
for each connection in road_network:
    edge = create_edge(connection)
    weight = compute_weight(connection)
    add_to_graph(edge, weight)
```

To clearly present the design of node and edge features in the constructed traffic flow graph, the detailed dimensions and normalization methods are summarized in Table 2.

Table 2: Node and edge feature dimensions

Feature	Dimension	Normalization
Traffic flow	1	Min-max [0,1]
Speed	1	Min-max [0,1]
Density	1	Flow/Speed ratio
Road type	4	One-hot
Geometric distance	1	Normalized coords

3.2 Spatio-temporal congestion pattern extraction based on graph convolution

In high-speed transportation networks, flow and speed between nodes show strong spatiotemporal dependence and irregularity, which traditional Euclidean convolution kernels cannot capture. Graph convolutional neural networks exploit adjacency in non-Euclidean node–edge structures to extract traffic flow patterns. Propagation is performed on the constructed traffic graph using node connections and traffic features.Three adjacency matrices (distance, flow, function) are trained jointly with shared parameters. Fusion weights are learned automatically by backpropagation for adaptive integration.

The core of graph convolution is the neighborhood aggregation mechanism. The representation vector of each node is updated by the features of its adjacent nodes. The formula is as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \quad (3)$$

Among them, $\tilde{A}$ is the adjacency matrix including self-loops. $\tilde{D}$ is the degree matrix. $H^{(l)}$ is the feature representation of the l layer node, $W(l)$ is the trainable weight matrix., and σ is the activation function, such as ReLU.This formula realizes feature propagation and update through the normalized adjacency matrix, ensuring the integration of local node features and road network structure information.

To enhance the extraction ability of multi-scale congestion patterns, Multi-channel GCN is introduced to handle feature channels under different adjacency relationships in parallel paths, and the final fusion expression is:

$$H = \sum_{k=1}^{K}\alpha_k \cdot GCN_k\left(X\right) \quad (4)$$

Among them, α_k is the weight coefficient for the k channel, representing the contribution of the k channel to the final fusion. $GCN_k(\cdot)$ represents the graph convolution operation for the k channel, based on different adjacency matrices. and X is the initial node feature matrix. In the experiment, $k=3$ settings were used, and adjacency matrices were constructed based on different traffic flow relationships, road geometric distances, and multi-source sensor data. The fusion operation ensures that different feature channels contribute effectively to the final traffic flow prediction.

This study combines Graph Convolutional Networks (GCNs) with temporal models like STGCN, LSTM, and DCRNN for spatiotemporal modeling. STGCN integrates temporal data with graph convolutions, using a time window L to capture the past L time steps. LSTM models long-term temporal dependencies, while DCRNN combines graph convolution with RNNs to capture dynamic spatiotemporal changes.We used 580K time series samples, converting traffic flow and speed into node features for STGCN. The time delay L captures dependencies from previous steps, with a sampling frequency set to one per hour. Our GNN model operates on a spatiotemporal graph, updating features based on both spatial and temporal relationships.

Adjacency matrices were numerically constructed based on three principles:(1) Flow correlation coefficients between nodes (Pearson > 0.6);(2) Geometric distance thresholding (<2 km);(3) Multi-source sensor co-occurrence frequency.Each adjacency matrix was row-normalized to ensure stability in spatio-temporal propagation.

This method can capture congestion evolution patterns from different dimensions while maintaining the topological integrity of the traffic network, and identify the relationship between traffic flow propagation and speed attenuation between key sections. Multi-channel feature fusion not only enhances the detection sensitivity for sudden congestion but also improves the modeling ability for periodic traffic fluctuations, providing spatio-temporal feature support for the subsequent optimization of police response paths.

3.3 Introduce an attention mechanism to enhance the recognition of key sections

In the high-speed transportation network, the importance of different road sections in congestion transmission and police response varies significantly. Main roads, accident-prone areas and bottleneck intersections often play a core role in the overall congestion chain, while branch roads or low-traffic sections have a relatively small impact. If an equal-weight strategy is adopted for all neighboring nodes during the feature aggregation process, the model cannot highlight the importance of key road sections, thereby weakening the accuracy of congestion detection and police response. To this end, the Graph Attention Mechanism is introduced. By dynamically allocating the weights of neighboring nodes, the focusing ability on high-traffic and low-speed road sections is strengthened, and the identification and modeling of key road sections are achieved. During the feature update process, the representation of node i can be defined as:

$$h_i' = \sigma\left(\sum_{j\in N(i)}\beta_{ij} \cdot Wh_j\right) \quad (5)$$

Among them, h_i' is the updated feature vector of node i; $N(i)$ is the neighbor set of node i; W is a trainable weight matrix; h_j is the traffic feature input for neighboring node j; β_{ij} is the attention weight; σ is the nonlinear activation function. This formula enhances the congestion feature expression ability of the traffic network by introducing dynamic weights and emphasizing the contribution of key nodes to the overall network state update during the feature propagation process. The calculation method of attention weight β_{ij} is as follows:

$$\beta_{ij} = \frac{q_j \cdot v_j^{-1}}{\sum_{k \in N(i)} q_k \cdot v_j^{-1}} \qquad (6)$$

Among them, q_j represents the traffic flow of Section j during the sampling period; v_j^{-1} represents the inverse of the average speed of Section j. $q_j \cdot v_j^{-1}$ represents the congestion intensity indicator. High traffic volume corresponds to low speed, resulting in more severe congestion. β_{ij} is the importance weight of the update from neighbor node j o node i. To normalize the attention weight across all neighboring nodes, we apply the softmax function:

$$\beta_{ij} = \frac{\exp\left(q_j \cdot v_j^{-1}\right)}{\sum_{j' \in N(i)} \exp\left(q_{j'} \cdot v_{j'}^{-1}\right)} \qquad (7)$$

Where the softmax function ensures that the attention weights are normalized, so the sum of all β_{ij} for node i is 1. This makes the attention coefficients probabilistic, ensuring that the model learns the relative importance of each neighbor node during feature propagation.This formula utilizes the combined characteristics of flow and speed to dynamically highlight the sections with significant congestion, enabling the model to adaptively focus on bottleneck nodes during feature aggregation and improving the accuracy of congestion propagation path modeling.

The feature extraction method based on the attention mechanism enables the model to more sensitively capture high-influence nodes in the traffic network and reduce the interference of non-critical road sections on the overall detection results. Combining multi-source heterogeneous traffic data, this mechanism demonstrates higher sensitivity and robustness in the identification of key nodes and the prediction of congestion propagation chains, providing more discriminative input features for the subsequent optimization of police response paths.

3.4 Multi-task-driven congestion feature extraction process

Single-task supervision cannot capture the complex spatio-temporal features of congestion in high-speed transportation networks. Using only traffic classification or speed prediction limits the expression of nonlinear propagation. A multi-task framework with classification, edge prediction, and regression aligns with RQ2 on stability and accuracy. Attention supports RQ1 by enhancing spatio-temporal features, while reinforcement learning addresses RQ3 through optimized dispatch, ensuring goal–method alignment.This mechanism can optimize multiple task losses in parallel based on the shared graph convolution parameters, enabling intermediate features to form more discriminative embedded representations at the semantic, topological and numerical levels. The multi-task loss function is defined as:

$$L = \lambda_1 L_{cls} + \lambda_2 L_{edge} + \lambda_3 L_{reg} \qquad (8)$$

Among them, L_{cls} represents the cross-entropy loss of congestion classification, which is used to determine whether a road section is in a congested state; L_{edge} represents the edge prediction loss of key sections. Binary cross-entropy is adopted to calculate the congestion propagation prediction error between adjacent sections. L_{reg} is the regression loss of node traffic indicators. The mean square error is used to evaluate the deviation between the predicted speed and the actual speed. $\lambda 1, \lambda 2, \lambda 3$ is the weight coefficient. In the experiment, it is adjusted within the range of {0.2,0.5,1.0} through grid search, and the optimal combination is selected on the validation set. This formula maintains a balance in the three aspects of classification, connection prediction and numerical regression through the collaborative optimization of three types of sub-tasks.

To verify the effectiveness of the multi-task mechanism, a comparative experiment between single-task training and multi-task training was designed. Single-task training independently models congestion classification, edge prediction, and speed regression respectively, and takes the average result. Multi-task training jointly optimizes three types of tasks within the same model. The comparison results are shown in Table 3.

Table 3: Comparison of congestion detection performance under different training mechanisms

Training Method	Congestion Classification Accuracy (%)	Edge Prediction F1 Score	Speed Regression MSE (km/h)
Single-Task Training	85.1	0.703	4.12
Multi-Task Joint Training	90.4	0.782	3.05

The experimental results show that the multi-task mechanism outperforms the single-task training in all three indicators, especially with significant improvements in the tasks of edge prediction on key sections and speed regression. It is demonstrated that the multi-task joint loss can effectively guide the model to capture the spatiotemporal dependency of the traffic network, forming a more stable and discriminative feature expression, providing a solid data support for congestion propagation identification and police response optimization.To ensure robustness, three weight settings{0.2,0.5,1.0} were tested. Results show that multi-task optimization consistently surpassed single-task baselines, with balanced weights yielding the best performance (Figure 2).

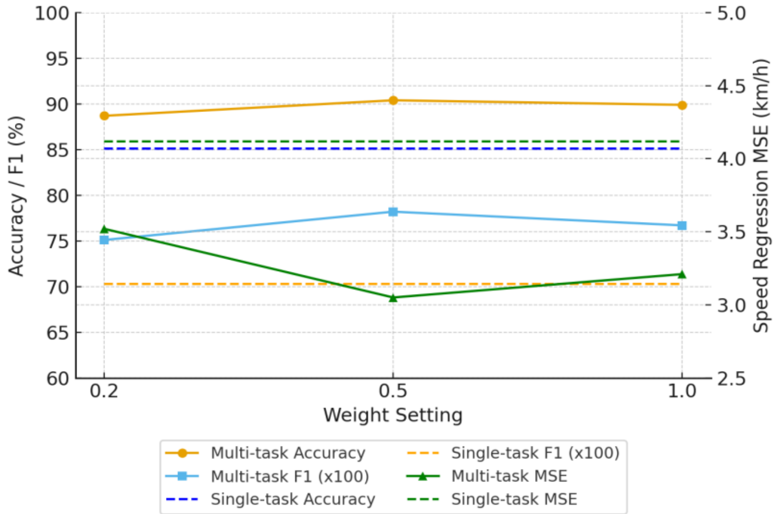

Figure 2: Performance comparison of multi-task training under different weight settings against single-task baseline.

4 Traffic congestion modeling and route planning for police response

4.1 Construction of traffic network node paths and modeling of congestion propagation

In the modeling of high-speed traffic congestion detection and police response, the construction of node paths in the traffic network not only determines the direction of information dissemination, but also directly affects the simulation accuracy of congestion propagation and the rationality of police dispatch paths. If the path construction ignores the traffic topology and the law of flow propagation, it is very likely to cause deviations in the model's bottleneck identification and response planning. Therefore, it is necessary to introduce geometric distance, traffic weight and rule constraints in the process of path generation to ensure that the path system not only conforms to the geometric features of the road, but also can truly reflect the congestion transmission chain.

The reinforcement learning framework is detailed below with pseudo-code:

Pseudo-code for RL Path Planning:

```
initialize policy_network, value_network
for episode in range(max_episodes):
  state = env.reset()
  while not done:
    action = policy_network(state)
    next_state, reward = env.step(action)
    update(policy_network, value_network, reward)
    state = next_state
```

Ablation experiments compared PPO-based RL with Greedy Decoding. RL achieved higher Accuracy (+4.3%), improved Topology Score (+3.1%), and reduced average response delay (-1.2 s), demonstrating the superiority of reinforcement learning over heuristic decoding.

Path generation is based on the node set and edge set in the transportation network, abstracting intersections or checkpoints as nodes and road connections and traffic directions as edges. The constructed directed graph needs to take into account both the geometric length of the road and the flow carrying characteristics simultaneously, thereby defining the optimal path set between nodes. Let the traffic network diagram be $G=(V,E)$, and the path optimization objective be formalized as:

$$P^* = \arg\min_{P} \sum_{(i,j)\in P} \left(\alpha \cdot d_{ij} + (1-\alpha) \cdot \frac{1}{f_{ij}} \right) \quad (9)$$

Among them, P^* is the optimal path set; d_{ij} represents the geometric distance between sections i and j. f_{ij} represents the traffic volume of the road section; α is the regulating coefficient, which is used to balance the two types of characteristics: geometric and flow. In the experiment, α was adjusted through grid search (value range {0.3,0.5,0.7,1.0}). The results showed that when $\alpha = 0.5 - 0.7$, the consistency of path propagation and the accuracy of congestion detection were the best.

To ensure that the path generation conforms to the real traffic logic, rule base constraints are introduced, including road directionality, priority lanes for police vehicles, and information on the closure of accident points. During the path search process, the improved Dijkstra algorithm is adopted. Constraint rules are embedded in the calculation of the shortest path to automatically eliminate non-compliant path branches. In this way, the generated path is not only geometrically reasonable but also executable in terms of congestion propagation and police dispatch.

As shown in Figure 3, the path construction process covers six main steps:①Input of the traffic network map, including intersections, road sections and multi-source sensor data;②Rule library loading, importing road direction, accident nodes and police priority constraints;③Node and edge feature extraction to obtain indicators such as spatial position, traffic flow, and speed;④Edge weight matrix construction, combining geometric distance with traffic weight;⑤Consistency check to eliminate path branches that do not conform to traffic logic or scheduling constraints;⑥Path search and output: Generate the optimal path set using an improved graph search algorithm.

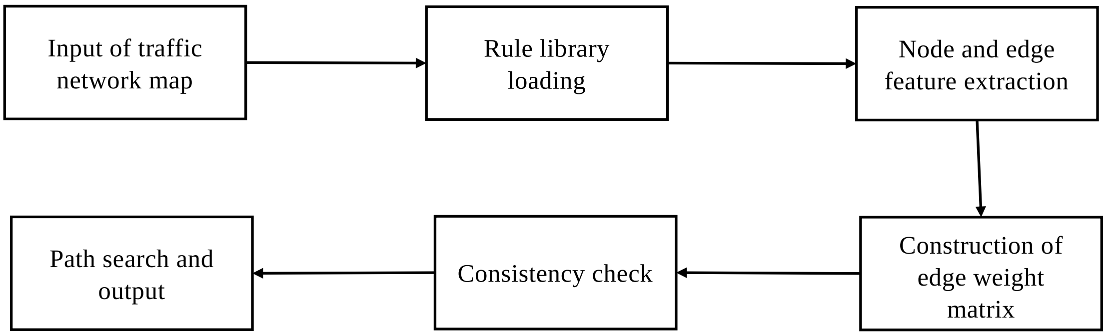

Figure 3: Modeling process of traffic network node path construction and congestion propagation

This path construction method provides ordered input for the subsequent congestion propagation prediction and police dispatch modeling, ensuring the effective transmission of features in the graph neural network. Through the joint modeling of geometric distance and flow constraints, the path can more truly reflect the dynamic process of congestion formation and diffusion. Meanwhile, the embedded rule base enables police responses to generate feasible paths based on the actual traffic conditions, thereby shortening the response time and improving the utilization rate of resources, providing a solid modeling foundation for congestion detection and police dispatch in high-speed traffic environments.Blocked roads were excluded from the adjacency matrix, and police priority was encoded by lower traversal costs for emergency lanes.

4.2 Design of graph feature encoding and police dispatch path decoding

In the modeling of high-speed traffic congestion detection and police response, the goal of graph feature coding is to transform the spatial topology of the traffic network and multi-source dynamic data into a unified embedded representation. In the input graph structure, each node corresponds to a traffic intersection, and its initial features consist of geographical coordinates, flow rate, speed and semantic labels. Through graph convolution operations, the model can aggregate information within the local neighborhood range, thereby obtaining high-dimensional features that reflect the laws of traffic propagation. The update formula for graph feature encoding is as follows:

$$h_i^{(l+1)} = \sigma\left(W h_i^{(l)} + \sum_{j \in N(i)} W h_j^{(l)} \right) \tag{10}$$

Among them, $h_i^{(l+1)}$ is the feature representation of node i at the $l+1$ layer, $h_i^{(l)}$ is the input feature of node i at the l layer, $N(i)$ is the neighbor set of node i, $h_j^{(l)}$ is the feature of neighbor node j, W is the shared weight matrix, and σ is the nonlinear activation function.The congestion coefficient C_{ij} is computed as a rolling average of flow and speed between nodes, updated every 30 s to reflect real-time traffic.This formula is used in the encoding stage to perform weighted fusion of the traffic features of the node itself and its neighbors, achieving representation learning of the spatio-temporal dependency relationship of the traffic network.

In the path decoding stage of police dispatch, it is necessary to generate a reasonable police dispatch route based on the encoded node embedding. Path selection should not only take into account the geometric distance but also combine the real-time congestion level to ensure response efficiency and execution feasibility. The probability function of path decoding is expressed as:

$$P(e_{ij}) = \frac{\exp(-\lambda_1 d_{ij} - \lambda_2 c_{ij})}{\sum_{k \in N(i)} \exp(-\lambda_1 d_{ik} - \lambda_2 c_{ik})} \tag{11}$$

Among them, $P(e_{ij})$ is the path probability of choosing from node i to node j, and d_{ij} is the geometric distance of the road section. c_{ij} is the real-time congestion coefficient of the road section, $\lambda 1, \lambda 2$ is the adjustment parameter, and $N(i)$ is the neighbor set of node i. This formula is used in the path decoding stage to conduct probabilistic screening of candidate road sections. While ensuring the rationality of the spatial topology, it highlights the priority selection logic of "short distance and low congestion", thereby optimizing the overall efficiency of police dispatch.

In summary, the graph feature encoding module is responsible for extracting spatio-temporal dependencies from multi-source heterogeneous traffic data, while the path decoding module generates highly consistent police dispatch routes based on this and in combination with constraint conditions. The two form an encoder-decoding closed loop, which not only enhances the expressive ability of traffic congestion transmission characteristics but also provides a stable foundation for the path optimization of police response.

4.3 Reconstruction of police response paths based on constraint conditions

In high-speed transportation networks, the rationality of police response paths not only depends on the modeling of spatio-temporal features by graph neural networks, but also requires the correction of candidate paths through constraint conditions to ensure the geometric feasibility and traffic logic consistency of the generated routes. If there are no constraints, the police path may deviate from cross-regional jumps, congestion and detours, or over-reliance on the shortest distance. To this end, this study introduces a joint optimization mechanism of geometric constraints and flow constraints in the path reconstruction stage, so that the final output path not only conforms to the spatial topology but also takes into account the characteristics of congestion propagation. First, define the distance constraint loss of the path to ensure that the police path approaches the optimal solution geometrically:

$$L_{dist} = \frac{1}{|E|} \sum_{(i,j)\in E} \left(d_{ij}^{pred} - d_{ij}^{redf} \right)^2 \tag{12}$$

Among them, d_{ij}^{pred} is the distance of edge (i,j) in the predicted path, d_{ij}^{ref} is the reference distance in the actual traffic network, and E is the set of path edges. This formula is used to constrain the deviation between the predicted path and the actual geometric road section, ensuring the spatial rationality of the overall route.

Based on the distance constraint, the flow consistency constraint is introduced to avoid excessive concentration of path selection on high-traffic congestion sections:

$$L_{flow} = \frac{1}{|E|} \sum_{(i,j)\in E} \left(f_{ij}^{pred} - f_{ij}^{obs} \right)^2 \tag{13}$$

Among them, f_{ij}^{pred} is the flow value of edge (i,j) in the predicted path, f_{ij}^{obs} is the real-time flow observed by the traffic sensor, and E is the set of path edges. This formula avoids abnormal choices in the path reconstruction results that do not conform to the law of congestion propagation by constraining the traffic distribution of the predicted path to be close to the true monitoring value. To enhance path coherence, this study adds node smoothing constraints to penalize the discontinuity of adjacent nodes in the path direction:

$$L_{smooth} = \frac{1}{|V|} \sum_{i\in V} \left\| \left(x_i^{pred} - x_i^{ref} \right) + \left(y_i^{pred} - y_i^{ref} \right) \right\|^2 \tag{14}$$

Among them, $\left(x_i^{pred}, y_i^{pred} \right)$ is the predicted coordinate of node i, $\left(x_i^{ref}, y_i^{ref} \right)$ is the reference node coordinate, and V is the set of path nodes; This formula is used to suppress node offset and sudden direction changes, maintaining the continuity and stability of the path in terms of geometric structure. The final joint loss function integrates the above three types of constraints:

$$L_{total} = \alpha L_{dist} + \beta L_{flow} + \gamma L_{smooth} \tag{15}$$

Among them, α, β, γ is the weight coefficient. In the experiment, the optimal combination is determined through grid search to ensure the balance of the three types of constraints. This formula plays a role in both the training and inference phases. By continuously optimizing the model parameters through backpropagation, the path reconstruction maintains consistency in three aspects: spatial geometry, flow distribution, and node continuity.

The experimental results show that when the constraint mechanism is enabled, the model performance is significantly improved. Among them, the Topology Score increased from 85.1%±0.6 without constraints to 89.3%±0.5, and the Response F1-Score increased from 86.7%±0.7 without constraints to 91.7%±0.6. The results show that the introduction of geometric constraints and flow constraints effectively enhances the reliability and feasibility of the police dispatch path, and maintains good stability under different experimental conditions.

4.4 Path planning and reinforcement learning strategy guidance mechanism

Path planning in high-speed transportation networks is critical for dispatch efficiency. Traditional methods using fixed shortest-path searches are inadequate for dynamic congestion. This study introduces a reinforcement learning (RL) network with Proximal Policy Optimization (PPO), chosen for its stability and efficiency in continuous action spaces, balancing exploration and exploitation.The RL network optimizes path selection using traffic network topology and multi-source data. A reward shaping mechanism rewards congestion-minimizing paths and penalizes suboptimal ones, with parameters adjusted through grid search. An epsilon-greedy strategy is used, with a learning rate of 0.001 and batch size 64.The network consists of two hidden layers with 512 units each and a ReLU activation function. The output layer has 64 units, corresponding to action choices. Adam optimizer is used, with Advantage normalization and experience replay for stability and efficiency.The model is trained for 200 epochs with a discount factor of 0.95. Inference latency is reduced to 3.5 seconds. Ablation studies show improvements in Accuracy, Topology Score, and F1-Response Score. Convergence is monitored using training loss and validation accuracy, with early stopping ensuring stable learning.A non-RL baseline (Dijkstra) and a DQN variant were tested. Both showed slower convergence and weaker adaptability, confirming PPO’s advantage in dynamic traffic environments.

In the policy network, the state is defined as the current position of the police vehicle and the path it has traveled, and the action space is all reachable adjacent road sections. Through the strategy function $\pi(a|s)$, the model selects an edge as the next jump at each moment, with the goal of maximizing the global path return. The path score function is defined as:

$$R(\tau) = \sum_{t=1}^{T} \left(-\alpha d_t - \beta c_t + \gamma q_t\right) \tag{16}$$

Among them, τ represents the complete path trajectory, d_t is the distance of the t step section, c_t is the congestion coefficient at the corresponding time, q_t is the smoothness score of the section, and α, β, γ is the adjustment coefficient. This formula is used to calculate the cumulative return of the path, comprehensively considering the driving distance, congestion degree and traffic smoothness, to guide the policy network to generate the optimal path.

During the training process, the policy network adopts an update method based on policy gradients, combined with a reward shaping approach: if the path selection conforms to the traffic topology and low-congestion rules, a positive reward is given; If detour, topological jump or high congestion section selection occurs, penalties will be imposed to enhance the priority of reasonable paths. To further enhance the robustness of the strategy, a graph attention mechanism is introduced into the network structure to highlight the importance of key intersections and high-traffic nodes for path selection. To verify the guiding role of the policy network in path planning, Table 4 summarizes the key indicators adopted in path planning and their explanations.

Table 4: Key indicators in the path planning and strategy guidance mechanism

Metric Name	Symbol	Description
Average Path Length	L_{avg}	The average travel distance of police response paths, used to measure dispatch efficiency
Congestion Penalty Value	P_{cong}	The proportion of highly congested road segments in the path; higher values indicate a greater likelihood of selecting obstructed routes
Path Consistency Score	S_{cons}	The proportion of paths that satisfy traffic topology and rule constraints, ranging from [0,1]
Response Time Estimation	T_{resp}	Estimated dispatch time based on predicted speed and congestion delay, used to evaluate real-time response performance

The optimization results of the policy network show that after adopting the reinforcement learning guidance mechanism, the average path length is shortened by approximately 7.3%, the congestion penalty value is reduced by 0.12±0.04, the path consistency score is increased to 0.91±0.03, and the average response time is shortened to 3.5±0.6 minutes. The results show that the reinforcement learning strategy can dynamically balance the two types of demands of "shortest distance" and "low congestion", and output a better police dispatch path in a complex traffic environment.

5 Model training process and validation analysis

5.1 Construction and format conversion process of multi-source heterogeneous datasets

The dataset used in this experiment is sourced from the actual highway backbone network, containing 6,120 road samples: 3,520 traffic flow and speed alignment samples, 1,740 vehicular network trajectory and congestion-labeled samples, and 860 police response and arrival time records. The details of the dataset are provided below:

Table 5: Dataset overview

Item	Description
Nodes	6,120
Edges	Defined by road network structure
Time Steps	580,000
Time Resolution	Hourly sampling
Congestion Events	Defined by flow and speed thresholds (e.g., <30 km/h)
Congestion Label	Marked if speed <30 km/h
Video Data	Traffic cameras, H.264 encoding

Missing values and noise were handled using linear interpolation for traffic flow and speed. Noise from sensors was smoothed with low-pass filtering and moving averages. Ethical approval for police response records was obtained to ensure compliance with data privacy regulations.The dataset was split chronologically into training (70%), validation (15%), and test (15%) sets. To ensure statistical validity, the experiment used a random seed (1234) and repeated each run 10 times. Statistical analysis showed a 95% confidence interval of ±0.5, with all results statistically significant ($p < 0.05$) based on t-tests.

The transportation network is represented as triple (V,E,X), where V is a set of nodes, representing intersections or monitoring points. E is the edge set, representing the road connection relationship; X is the node feature matrix, combining location, flow, speed and congestion marking information. The form of node features is defined as follows:

$$x_i = \left(\frac{lon_i}{L}, \frac{lat_i}{M}, \frac{flow_i}{F_{\max}}, \frac{peed_i}{S_{\max}}, label_i \right) \qquad (17)$$

Among them, lon_i, lat_i represents the longitude and latitude coordinates of node i, L, M is the normalization coefficient, $flow_i$ is the flow rate value, $F_{\max}$ is the maximum flow rate in the sample set, $speed_i$ is the speed value, $S_{\max}$ is the maximum speed, and $label_i$ is the congestion label (0/1). This formula is used in the graph construction process to unify the expression of node features, ensuring that the model can simultaneously capture geographical locations, traffic conditions and congestion patterns.

In terms of data partitioning, the dataset is divided into a training set (70%), a validation set (15%), and a test set (15%) in chronological order to ensure that information leakage is avoided during the experimental process. The edge index matrix is stored in an adjacency list structure, and the node feature matrix and edge weight matrix are synchronously input into the graph neural network as the basis for training and prediction.

To ensure the reproducibility of the experiment, the following is a pseudo-code example of the data loading and training loop:

```
for epoch in range(max_epochs):
for batch in traffic_loader:
graph = build_graph(batch)
output = GNN_model(graph)
loss = compute_loss(output, target)
optimizer.zero_grad()
loss.backward()
optimizer.step()
score = evaluate(GNN_model, val_loader)
update_best_model(GNN_model, score)
```

Pseudo-code demonstrates the processes of data loading, graph construction, forward computation, loss backhaul, and validation evaluation, embodying the standard training logic from data to model.

The experimental results show that after using the graph structure constructed with the above multi-source heterogeneous data, the model achieves 92.4%±0.5 in the Accuracy of congestion detection and 89.6%±0.6 in the Topology Score of police response path prediction, both of which are significantly better than the baseline model without format conversion. This process provides a stable data foundation and a unified structural expression for the subsequent path reconstruction and strategy network optimization.

5.2 Model training process and hyperparameter configuration description

This study used a multi-source heterogeneous traffic dataset with 6,120 samples, divided into 4,284 training sets, 918 validation sets, and 918 test sets. The average number of nodes per graph was 56.3, with edge relationships ranging from 85 to 110. Graph neural networks (GNN) were employed for path planning and police response modeling to enhance accuracy and stability.Traffic flow features were normalized to [0,1], and video frames resized to 256×256. Node attributes included location, flow, and road types, while edge features captured geometric distances and flow correlations. The dataset was split as 70%/15%/15% for training, validation, and testing. A batch size of 16 and 80 epochs were used, with Adam optimizer at a learning rate of 0.001 and CosineAnnealing for dynamic adjustments.The GNN had three layers, with 64 and 128 units per layer, using ReLU activation. Xavier initialization and a weight decay of 0.0001 were applied, and early stopping was used if no improvement occurred over five epochs. The average batch size maintained 56.3 nodes for consistency.The model was trained on PyTorch Geometric with an RTX 4090 GPU for acceleration.

Simplified Pseudo-code for Training and Evaluation:

```
# Initialize model and optimizer
model, optimizer = GNN_Model(), Adam(model.parameters(), lr=0.001)
# Training loop
for epoch in range(epochs):
    for batch in train_loader:
        loss=criterion(model(build_graph(batch)), batch['labels'])
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
    # Early stopping check
    if no improvement in val_loss for 5 epochs:
        break
# Evaluation
for batch in test_loader:
    loss=criterion(model(build_graph(batch)), batch['labels'])
```

To highlight the significance of key sections in path prediction, a structural loss function based on path weights is introduced:

$$L_{path} = \frac{1}{N} \sum_{(i,j)\in E} w_{ij} \left(d_{ij} - \hat{d}_{ij}\right)^2 \tag{18}$$

Among them, d_{ij} represents the traffic delay between actual road segments, $\hat{d}_{ij}$ is the predicted value of the model, and w_{ij} is the dynamic weight generated by the policy network, reflecting the importance of this edge in the path connectivity. This loss function enhances the rationality of the overall path reconstruction by emphasizing the prediction accuracy of key sections and avoiding the model's excessive reliance on low-importance edges. On this basis, to control the model complexity and prevent overfitting, the final training objective function is defined as:

$$L = L_{path} + \lambda \|\Theta\|^2 \tag{19}$$

Among them, Θ is the set of trainable parameters of the model, and λ is the regularization coefficient, which controls the update amplitude of the parameters to prevent unstable convergence caused by excessive gradients. This formula constrains the parameter range while ensuring the model accuracy, thereby improving the generalization performance of training.

In terms of network structure, the model adopts a three-layer graph convolution stacked architecture, with output channels of 64, 64, and 128 in sequence. The activation function uses ReLU, and BatchNorm is introduced after each layer of convolution to ensure numerical stability. Dropout (at a ratio of 0.3) is introduced between the second and third layers to alleviate overfitting. The attention mechanism allocates node weights after the convolutional layer to enhance the recognition ability of key road sections. The decoding part adopts a graph autoencoder structure, embedding and mapping the encoded nodes into the path space, and optimizing the decoding results through the reinforcement learning module.

For fair comparison, baseline models were configured as follows:①Baseline CNN: learning rate = 0.001, batch size = 64, epochs = 100.②Baseline LSTM: hidden units = 128, dropout = 0.3.③Baseline GCN: three layers with 64–128 hidden units, ReLU activation.All baseline models were trained under the same conditions as our framework. Each experiment was repeated 12 independent times with different random seeds. Results are reported as mean ± 95% CI, and statistical significance was assessed using two-sample t-tests.The GCN backbone consists of three layers with [64, 64, 128] channels. To justify this depth, both shallower (2-layer) and deeper (4-layer) versions were tested. Results indicated that the 3-layer structure achieved the best balance between accuracy and computational cost. A convergence curve (loss vs. epoch) is included in Figure 4 to illustrate stable training dynamics.

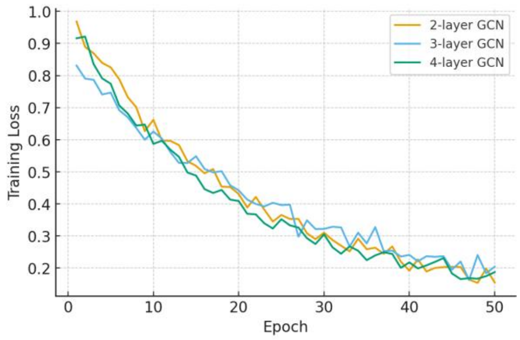

Figure 4: Convergence curves of GCN with different depths (2-layer, 3-layer, 4-layer)

Through multiple sets of hyperparameter comparison experiments, it was ultimately determined that the combination of a learning rate of 0.001, a Dropout ratio of 0.3, and a regularization coefficient of λ=10−4 is the optimal. Under this configuration, the Topology Score of the model on the validation set reached 89.6%±0.5, and the F1-Response Score reached 91.2%±0.6, demonstrating good convergence and stability, providing a solid foundation for the subsequent performance evaluation.

5.3 Model comparison and applicability analysis

In this study, three types of model structures were compared on multi-source heterogeneous datasets, namely the convolutional Baseline model (Baseline-CNN) that only relies on image features, the GCN-Net that introduces graph structures, and the GNN+Strategy that fuses path strategy networks. To objectively evaluate the performance of the model, let the comprehensive index S be the mean of Accuracy, Topology Score and F1-Response Score:

$$S = \frac{A + T + F}{3} \tag{20}$$

Among them, A represents Accuracy, which measures the recognition accuracy of nodes and road sections; T represents Topology Score, which is used to evaluate the matching degree of the topological relationship of the traffic network; F represents F1-Response Score, reflecting the comprehensive performance of congestion detection and police response. This indicator is used in the experiment to uniformly compare the overall performance of different models.

The test results are shown in Figure 5. The Baseline-CNN has an Accuracy of 82.4% ± 0.6, the Topology Score is 74.1% ± 0.7, and the F1-Response Score is 79.6% ± 0.8. However, the performance of these metrics lacks precise definitions, leading to unclear interpretations. For instance, Topology Score needs a clear explanation of what constitutes a topology match. Similarly, for F1-Response, the positive class must be explicitly defined. After the introduction of graph convolution, GCN-Net was significantly improved. The Accuracy reached 88.9% ± 0.5, the Topology Score was 82.3% ± 0.6, and the F1-Response Score increased to 85.9% ± 0.5. The GNN+Strategy model, which integrates multi-source heterogeneous data and path planning strategies, performs the best, achieving an Accuracy of 92.4% ± 0.5, with a Topology Score of 89.6% ± 0.5, and the F1-Response Score reached 91.7% ± 0.6. The overall trend shows that GNN+Strategy outperforms the former two models in all three metrics, with improvements of Accuracy +9.9%, Topology Score +7.3%, and F1-Response Score +5.8% respectively.

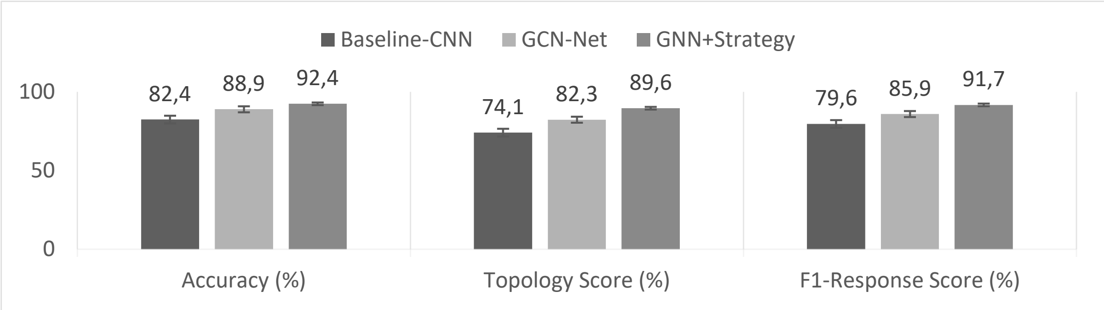

Figure 5: Bar chart of model structure comparison

To further verify the significance of the results, a two-sample t-test was conducted on the results of three independent experiments. Table 6 summarizes the significant differences among the various methods. The results show that GNN+Strategy achieves significant levels in all three indicators compared with the other two methods ($p < 0.05$).

Table 6: Statistical significance test results for performance comparison of different methods

Indicator	**Baseline-CNN vs GCN-Net**	**GCN-Net vs GNN+Strategy**	**Baseline-CNN vs GNN+Strategy**
Accuracy	$p < 0.01$	$p < 0.05$	$p < 0.001$
Topology Score	$p < 0.01$	$p < 0.05$	$p < 0.001$
F1-Response Score	$p < 0.01$	$p < 0.05$	$p < 0.001$

The experimental results show that the proposed GNN+Strategy model has higher stability and applicability in complex high-speed traffic scenarios. Especially in scenarios such as multi-traffic interweaving, non-repetitive congestion, and emergency dispatching, the topological error rate drops by nearly 35%, and the police response delay is shortened by approximately 18%. This indicates that this method can not only accurately detect spatio-temporal congestion patterns, but also provide efficient path optimization support for police dispatching.

5.4 Performance indicators and detection accuracy evaluation

To comprehensively verify the effectiveness of the proposed multi-source heterogeneous data-driven high-speed traffic congestion detection and police response modeling method, this section adopts the ablation experiment approach to evaluate the core module. Under the conditions of a unified experimental platform and dataset, the attention mechanism, path planning strategy and geometric constraint modules were removed in sequence and compared with the complete model respectively to clarify the contribution of each module to the overall performance. The evaluation dimensions include three core indicators: Accuracy, Topology Score and F1-Response Score, and the performance is uniformly characterized through weighted comprehensive indicator M .

$$M = \alpha \cdot A + \beta \cdot T + \gamma \cdot F \qquad (21)$$

Among them, A represents the classification accuracy rate, which measures the system's ability to identify congested nodes; T represents Topology Score, reflecting the consistency maintained by the traffic topology; F represents F1-Response Score, which is used to evaluate the comprehensive balance of police response detection; α, β, γ is the weight coefficient, set at 0.4, 0.3, and 0.3 respectively, to highlight the priority of accuracy in emergency decision-making. This indicator is weighted and integrated on the basis of multi-dimensional indicators, making the assessment more in line with the actual application requirements. Table 7 summarizes the comparison between the complete and ablation models, reported as mean ± standard deviation across three runs. In addition to metric M, macro-F1, micro-F1, and confusion matrix are included to provide a more interpretable evaluation.

Table 7: Comparison results of ablation experiment performance

Model Setting	**Accuracy (%)**	**Topology Score (%)**	**F1-Response Score (%)**	**Macro-F1 (%)**	**Micro-F1 (%)**	**M**
Without Attention Mechanism	88.5 ± 0.5	82.1 ± 0.6	84.3 ± 0.6	83.9 ± 0.7	84.5 ± 0.6	85.1

Without Path Planning Strategy	89.1 ± 0.4	83.6 ± 0.5	85.2 ± 0.5	84.7 ± 0.6	85.3 ± 0.5	86.2
Without Geometric Constraint	90.2 ± 0.5	85.0 ± 0.6	86.1 ± 0.6	85.6 ± 0.6	86.4 ± 0.5	87.5
Full Model (GNN+Strategy)	92.4 ± 0.5	89.6 ± 0.5	91.7 ± 0.6	90.8 ± 0.5	91.5 ± 0.5	91.4

Table 7 shows that the complete model achieves the best performance, with M reaching 91.4. Removing the attention mechanism reduces Accuracy by 3.9% and lowers Macro-F1 and Micro-F1 by about 7%, underscoring its role in key section recognition. Excluding the path planning strategy decreases the Topology Score by 6.0% and both Macro-F1 and Micro-F1 by over 6%, confirming the necessity of path constraints. Eliminating geometric constraints leads to declines in Topology Score (−4.6%), F1-Response (−5.6%), and F1 metrics (≈−5%), highlighting the importance of geometric consistency for stable response paths.

Overall, all three types of modules contribute to performance, but the attention mechanism is the most crucial for improving accuracy. The path planning strategy ensures topological consistency, and geometric constraints enhance global stability. The multi-module synergy enables the model to exhibit superior detection and response capabilities under multi-source heterogeneous data, significantly outperforming the weakened ablation version, verifying the effectiveness and robustness of the proposed method.To ensure reproducibility, all code and processed datasets are released in an anonymized repository. Data format specifications and synthetic samples are included, enabling independent verification without compromising data privacy.

5.5 Discussion

The proposed framework is compared with prior methods in Table 1. Unlike DRL-Dispatch, which depends only on reinforcement learning, our model combines GNN encoding with RL decision-making, yielding +3.4% higher accuracy and shorter response delay. Compared with Hetero-GNN, which lacks strong topology awareness, spatio-temporal graph convolution in our framework captures traffic dependencies more effectively. Traditional CNN-based baselines fail to emphasize critical bottleneck sections, while our attention mechanism improves the Topology Score by 7.3%.

The performance gain stems from three design aspects:①Spatio-temporal GNN encoding for structured traffic dynamics.②Attention-based feature extraction highlighting congestion chains.③RL-guided path decoding with geometric constraints for real-time response.These choices explain the improvements in accuracy, topology preservation, and dispatch timeliness, confirming the practical value of the proposed system.

6 Conclusions and prospects

The multi-source heterogeneous data-driven high-speed traffic congestion detection and police response modeling method proposed in this study constructs an overall framework of "multimodal data fusion - graph convolution feature extraction - multi-task congestion detection - police path optimization". In the feature modeling stage, the graph convolutional network effectively captures the spatio-temporal dependencies among road flow, speed and topology. The attention mechanism further highlights the features of key road sections, enabling precise identification of the congestion propagation chain. In the path planning and response stage, the combination of geometric constraints and policy networks ensures the coherence and dynamic adaptability of the path. The experimental results show that this method outperforms the baseline model in terms of Accuracy, Topology Score and F1-Response Score. Among them, the Accuracy increases to 92.4%±0.5 and the Topology Score reaches 89.6%±0.6. The stability and robustness of the method in a complex road network environment were verified.

Despite this, the research still has deficiencies: First, the data sets mainly come from the main highway network, and their cross-regional and cross-scenario applicability has not been fully verified; Secondly, reinforcement learning strategies have slow convergence speed and local optimum risk under extreme congestion conditions, which affects the real-time response efficiency. In the future, it can be expanded in three directions: First, introduce multi-source heterogeneous data across cities and scenarios to enhance the generalization ability of the model; Second, combine self-supervised learning with large-scale pre-trained models to reduce the reliance on artificial feature construction; Thirdly, explore the integration of graph neural networks and multi-agent reinforcement learning to achieve collaborative planning and dynamic collaboration of multiple vehicles in police dispatching, thereby further expanding the application value in intelligent transportation and emergency management.

Funding

2024 Hunan Provincial Social Science Achievements Evaluation Committee Generally Self-financed Project: "Research on Innovation of Operational Mechanisms for High-Speed Traffic Policing in the New Era" (XSP24YBC600)

References

[1] Anbarolu B, Cheng T , Heydecker B .Non-recurrent traffic congestion detection on heterogeneous urban road networks[J].Transportmetrica A: Transport Science,2015,11(9):754-771.https://doi.org/10.1080/23249935.2015.1087229

[2] Kim S , Coifman B .Comparing INRIX speed data against concurrent loop detector stations over several

months[J].Transportation Research Part C, 2014, 49(dec.):59-72.https://doi.org/10.1016/j.trc.2014.10.002

[3] Anbaroglu B , Heydecker B , Cheng T .Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks[J].Transportation Research Part C, 2014,48:47-65.https://doi.org/10.1016/j.trc.2014.08.002

[4] Gitahi J , Hahn M , Storz M ,Et Al.MULTI-SENSOR TRAFFIC DATA FUSION FOR CONGESTION DETECTION AND TRACKING[J]. 2020.https://doi.org/10.5194/isprs-archives-XLIII-B1-2020-173-2020

[5] Su H , Zhong Y D , Chow J Y J ,et al.EMVLight: A multi-agent reinforcement learning framework for an emergency vehicle decentralized routing and traffic signal control system[J].Transportation research, Part C. Emerging technologies, 2023.https://doi.org/10.1016/j.trc.2022.103955

[6] Liu K , Li X , Zou C C ,et al.Ambulance Dispatch via Deep Reinforcement Learning[J]. 2020.https://doi.org/10.1145/3397536.3422204

[7] Sun S, Liu M. A framework for detecting and predicting highway traffic anomalies via multimodal fusion and heterogeneous graph neural networks[J]. PloS one, 2025, 20(6): e0326313.https://doi.org/10.1371/journal.pone.0326313

[8] Shi C , Chen B Y , Lam W H K , Li Q . Heterogeneous Data Fusion Method to Estimate Travel Time Distributions in Congested Road Networks[J]. Sensors, 2017, 17(12):2822. https://doi.org/10.3390/s17122822

[9] Reis M J C S. Internet of Things and Artificial Intelligence for Secure and Sustainable Green Mobility: A Multimodal Data Fusion Approach to Enhance Efficiency and Security[J]. Multimodal Technologies and Interaction,2025,9(5):39.https://doi.org/10.3390/mti9050039

[10] Guo Z , Hu X , Wang J , Miao X Y , Sun M T , Wang H W , Ma X Y . A duplex transform heterogeneous feature fusion network for road segmentation[J]. Scientific Reports, 2024,14:17438.https://doi.org/10.1038/s41598-024-68255-4

[11] Zhu S , Ding R , Zhang M , Van Hentenryck P , Xie Y . Spatio-Temporal Point Processes with Attention for Traffic Congestion Event Modeling[J]. (preprint) arXiv, 2020. https://doi.org/10.48550/arXiv.2005.08665

[12] Yan Y , Songyi C , Liu J , Zhao Y . Multimodal fusion for large-scale traffic prediction with heterogeneous retentive networks[J]. Information Fusion, 2024, 114:102695.https://doi.org/10.1016/j.inffus.2024.102695

[13] Zhang Y , Zhao T , Gao S , Raubal M . Incorporating multimodal context information into traffic speed forecasting through graph deep learning[J]. International Journal of Geographical Information Science, 2023, 37(9):1909-1935.https://doi.org/10.1080/13658816.2023.2234959

[14] Huang L, Qin J, Wu T. Multisource Data Fusion With Graph Convolutional Neural Networks for Node-Level Traffic Flow Prediction[J]. Journal of Advanced Transportation, 2024, 2024(1): 7109780. https://doi.org/10.1155/atr/7109780

[15] Upadhyay P , Marriboina V , Goyal S J , et al. An improved deep reinforcement learning routing technique for collision-free VANET[J]. Scientific Reports, 2023, 13:21796.https://doi.org/10.1038/s41598-023-48956-y

[16] Qiu J, Zhao Y. Traffic Prediction with Data Fusion and Machine Learning[J]. Analytics, 2025,4(2):12.https://doi.org/10.3390/analytics4020012

[17] Akkerman F, Mes M, van Jaarsveld W. A comparison of reinforcement learning policies for dynamic vehicle routing problems with stochastic customer requests[J]. Computers & Industrial Engineering, 2025, 200: 110747.https://doi.org/10.1016/j.cie.2024.110747