EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Proxy-Cost-Aware Explainable Graph Neural Network for Multi-Domain O&M Decision Support in Digital Twin-Enabled Smart-City IoT Systems

Fujing Qin[1,*]

[1]Shandong University of Finance and Economics, Jinan, 250002, China

Abstract

INTRODUCTION: Digital twin-enabled smart-city IoT systems may interact with interconnected transportation, parking, environmental, communication, and edge-computing infrastructures. Failures or congestion in one domain may propagate across service chains and increase operational burden, yet existing spatiotemporal graph models rarely represent cost semantics, risk propagation, and maintenance actions jointly.

OBJECTIVES: This study develops a cost-aware explainable graph neural network for multi-domain operation and maintenance decision support, with emphasis on proxy-cost prediction, risk identification, rule-based action ranking, and interpretable attribution.

METHODS: A multi-layer digital twin graph integrates sensing, communication, computing, service, maintenance, and proxy-cost nodes. A state–risk–cost three-channel propagation mechanism separately models operational states, abnormal-risk diffusion, and cost-impact transmission. Rule-based graph interventions, including maintenance, task migration, link switching, and sampling adjustment, are used to estimate action-induced proxy-cost changes. CityPulse is used as a heterogeneous multi-domain proxy scenario, while Caltrans PeMS provides a homogeneous traffic-sensor reference. Performance is evaluated using MAE, F1, NDCG@5, and Fidelity, where NDCG@5 measures consistency with predefined action-ranking rules.

RESULTS: On CityPulse, the proposed method achieved lower MAE and higher NDCG@5 and Fidelity than the strongest corresponding baselines, although its F1 was slightly lower than STAEformer. On PeMS, it achieved the highest Fidelity but remained below the best traffic-specific models in MAE, F1, and NDCG@5. Ablation results showed the respective contributions of cost nodes, explicit three-channel propagation, rule-based intervention evaluation, and explanation constraints.

CONCLUSION: The method is more effective for heterogeneous multi-domain O&M scenarios than for homogeneous traffic forecasting. Because both datasets provide proxy rather than real financial, energy-system, and intervention records, the results support proxy-based methodological validation but do not constitute direct physical-energy-system validation or evidence of realized cost savings and causal intervention effectiveness.

Keywords: smart-city IoT, digital twin, explainable graph neural network, multi-domain O&M, proxy operational cost

Received on 15 July 2026, accepted on 07 August 2026, published on 18 August 2026

doi: 10.4108/ ew.14036

*Corresponding author. Email: 17861320256@163.com

1. Introduction

Digital twin-enabled smart-city IoT systems are evolving into cross-domain cyber-physical infrastructures composed of sensing devices, communication gateways, edge-computing nodes, urban-service platforms, and maintenance resources [1,2]. These systems integrate transportation, parking, environmental sensing, communication, computing, and service domains. The present study uses intelligent transportation IoT as its primary empirical setting, while environmental monitoring, smart lighting, and energy-aware services are treated as potential extension scenarios rather than experimentally validated applications. In such systems, device failures, link congestion, or edge-node overload may propagate along data flows and service chains, resulting in changes in proxy operational costs, including service latency, resource consumption, maintenance pressure, and degraded service quality [3,4]. Therefore, characterizing cross-layer dependencies in a digital twin environment and using graph neural networks to model relationships among state changes, risk propagation, and proxy costs constitute an important problem for explainable O&M analysis [5,6].

Existing studies provide a useful foundation. Digital twin models for intelligent transportation can predict indicators such as queue length, waiting time, and traffic volume. Network digital twin methods can model network topology, device states, and QoE indicators [7]. Fault diagnosis and explainable graph neural network studies mainly focus on anomaly identification and model interpretation. However, in proxy O&M cost modeling for smart-city IoT systems, traffic-oriented models provide only limited coverage of communication, computing, and maintenance resources; network digital twins do not explicitly characterize the relationship between performance degradation and cost semantics; and fault diagnosis models provide insufficient joint modeling of fault consequences, cost impacts, and predicted action effects [8]. If proxy cost is incorporated into existing spatiotemporal models only as a regression target, the model may struggle to distinguish state diffusion, risk propagation, and cost formation, or to explain how predefined interventions influence proxy-cost rankings [9].

To address these challenges, this paper develops the method from three perspectives. First, a cost-aware multi-layer digital twin graph incorporates sensing devices, communication links, edge computing nodes, urban services, maintenance resources, and proxy-cost nodes into a unified representation. Second, a state–risk–cost three-channel propagation GNN separately models operational-state diffusion, fault-risk propagation, and cost-impact transmission. Third, a rule-based explainable action-evaluation mechanism is developed. Candidate actions, including maintenance, task migration, link switching, and sampling adjustment, are implemented through predefined interventions to estimate their predicted effects on future proxy costs and provide attribution explanations at the levels of nodes, edges, service chains, and cost types.

This study extends multi-domain O&M decision support for digital twin-enabled smart-city IoT systems from single-purpose state prediction to the joint modeling of proxy-cost semantics, risk propagation, and rule-based action evaluation. CityPulse assesses heterogeneous multi-domain representation, whereas PeMS provides a homogeneous traffic-sensor reference for examining the method's boundary. Neither dataset contains real financial costs, maintenance records, computing-task logs, communication-path logs, or observed intervention outcomes; therefore, the experiments provide proxy-based methodological validation within smart-city IoT settings. They do not constitute direct validation of physical energy-system operation or energy-related cost savings. Transfer to Energy Web applications would require physical energy measurements, equipment-health records, maintenance logs, and observed operational costs.

2. Related Work

2.1. Application Scenarios and Challenges

Multi-domain O&M analysis in digital twin-enabled smart-city IoT systems involves intelligent transportation, environmental monitoring, smart lighting, communication infrastructure, and edge-service scheduling [10]. Typical tasks include state prediction, fault identification, resource-consumption estimation, service-quality assessment, maintenance-priority ranking, and O&M strategy selection. Unlike conventional single-system optimization, smart-city IoT infrastructure exhibits cross-layer coupling: sensing devices collect data, communication links transmit information, edge nodes perform computation, service platforms support control and decision-making, and maintenance resources determine system recovery capability [11,12]. As a result, local device anomalies, link congestion, or computational overload may propagate along service chains, leading to increased latency, greater resource consumption, and degraded service quality. The research value therefore lies not only in prediction accuracy but also in characterizing relationships among state changes, risk diffusion, and proxy operational costs.

Existing studies often use traffic flow, mobility traces, network measurements, energy-consumption records, and fault logs [13,14]. However, these data usually originate from individual subsystems, while integrated data spanning the "sensing–communication–computation–service–maintenance–cost" chain remain limited. Real maintenance work orders, financial expenditures, SLA-violation records, and observed action outcomes are particularly difficult to obtain from public datasets. Therefore, indicators such as latency, resource consumption, congestion, anomaly risk, and service-quality degradation may be used to construct proxy costs [15]. These indicators represent relative operational burden rather than actual financial cost. Common metrics, including MAE, RMSE, F1, AUC, latency, and resource utilization, evaluate local performance but do not directly characterize proxy-cost consequences, risk diffusion, service-chain impacts, or explanation credibility.

2.2. Review of Mainstream Methods

Digital twins and spatiotemporal graph learning for intelligent transportation are relatively mature. Existing methods model roads, intersections, and traffic flows through temporal convolution, recurrent structures, graph convolution, or graph attention. However, their graphs mainly rely on road topology or traffic correlations and serve traffic-efficiency tasks [16,17]. Communication, edge-computing, maintenance, and cost entities are rarely represented explicitly.

Network digital twins integrate topology, device states, and application states for monitoring, prediction, and optimization [18,19]. Zhang et al. combined digital twins, reinforcement learning, and reliability constraints for edge-caching optimization [20]. Subsequent studies examined twin creation, adaptation, deployment, and security [21], while VH-Twin supported wireless-network mapping through vertical construction, horizontal updating, and regional clustering [22]. These studies support twin construction and network optimization but do not represent performance degradation as propagable proxy-cost nodes or separate state, risk, and cost transmission.

Energy-focused digital-twin studies have examined equipment monitoring, predictive maintenance, energy-efficiency management, and operational optimization in renewable and building energy systems [23,24]. Energy-SLA-aware resource management further considers energy consumption, operating costs, and service constraints in edge–cloud task allocation [25]. These studies identify the equipment states, energy measurements, SLA constraints, and cost records required for possible transfer to Energy Web applications. However, they are reviewed as related application references rather than evidence that the CityPulse and PeMS experiments validate physical energy-system performance. They generally focus on individual assets, energy-efficiency objectives, or specific resource-allocation tasks, while cross-layer propagation among sensing, communication, computing, service, maintenance, risk, and proxy-cost entities in heterogeneous smart-city IoT systems remains insufficiently modeled.

Fault diagnosis and predictive maintenance have been studied in digital-twin and O&M settings [5,6,9]. Explainable GNN methods further identify influential graph structures and features [26]. However, these studies do not directly address proxy-cost simulation and rule-based action ranking.

2.3. Most Closely Related Studies

The first related category concerns spatiotemporal graph-based digital twins for intelligent transportation corridors [27]. These methods predict queue length, waiting time, traffic volume, and travel time to support traffic control. The present study differs by incorporating sensing, communication, computing, maintenance, and proxy-cost nodes into a shared graph that links operational states with IoT cost semantics.

The second category concerns network-twin construction and optimization [20–22,28]. Digital-twin-assisted caching uses virtual-network states to train and evaluate resource-allocation policies. Other studies address the life cycle of digital network twins, while VH-Twin focuses on physical-to-digital mapping and asynchronous synchronization. These works explain how network twins are constructed, updated, and optimized.

The present study builds on these foundations rather than claiming digital-twin construction as a new concept. Its distinction lies in embedding proxy-cost nodes and propagation edges into a cross-layer graph, mapping congestion, delay, anomalies, and service degradation to traceable cost semantics, and separating state, risk, and cost propagation for rule-based action ranking. The third category includes fault diagnosis and predictive maintenance, whereas this study focuses on proxy-cost consequences along service chains and predicted changes under predefined candidate actions.

Table 1 compares the proposed method with closely related digital-twin and graph-learning studies.

Table 1. Comparison with closely related digital-twin and graph-learning studies

Study	Primary objective	Twin or graph representation	Proxy-cost nodes	State–risk–cost separation	Action or optimization mechanism
Transportation digital twins [27]	Traffic prediction and control	Roads, intersections, and traffic relations	No	No	Traffic-control evaluation
Ngo et al. [7]	GNN-enabled network digital twins	Network entities, states, and GNN architectures	No	No	Monitoring, prediction, and optimization
Guo et al. [9]	Service-function-chain failure localization	Service functions and failure dependencies	No	Partial	Failure localization
D-REC [20]	Reliable edge-caching optimization	Network twin with caching and load states	No	No	RL-based caching with reliability constraints
Digital Network Twins [21]	Twin creation, adaptation, and optimization	Physical–virtual network models	No	No	Traffic forecasting and edge caching

VH-Twin [22]	Wireless-network mapping and synchronization	Vertical and horizontal twinning	No	No	Twin construction and updating
Fault diagnosis, predictive maintenance, and explainable GNNs [5,6,9,26]	Fault detection and explanation	Devices, faults, and dependencies	No	Partial	Generally not considered
Proposed method	Proxy-cost modeling and rule-based ranking	Cross-layer graph with proxy-cost nodes	Yes	Explicit three-channel propagation	Rule-based proxy-cost comparison

As shown in Table 1, the distinction of this study is not merely the addition of a prediction target. Proxy-cost items are represented as propagable semantic nodes, while state, risk, and cost information is modeled through explicitly separated channels. In addition, the rule-based intervention transformation ranks predefined O&M actions by comparing predicted proxy-cost changes between baseline and intervention graphs. Therefore, the contribution lies in explicit proxy-cost semantic representation, state–risk–cost propagation, and rule-based action evaluation rather than the general construction of digital twins or application of GNNs to network optimization.

2.4. Summary and Research Gap

Overall, existing studies have advanced intelligent-transportation state prediction, network digital-twin modeling, fault diagnosis, and explainable graph learning. Nevertheless, systematic characterization of relationships among state changes, risk diffusion, and proxy-cost semantics across coupled smart-city IoT domains remains insufficient. Prior studies have examined traffic-state variations, network-performance changes, device faults, and resource-consumption changes. However, how local anomalies are transformed into system-level proxy operational costs through communication, computing, and service chains remains insufficiently modeled.

This study addresses this gap through a cost-aware multi-layer digital-twin graph, a state–risk–cost three-channel propagation GNN, and a rule-based action-evaluation mechanism. Proxy operational costs are transformed from external indicators into learnable semantic entities, while relationships among state diffusion, risk propagation, and cost formation are modeled explicitly. Congestion, latency, anomaly risk, and service degradation are mapped to proxy costs for methodological evaluation; they are not treated as real financial or physical energy-system costs. Accordingly, the framework covers cross-layer proxy-cost modeling, risk–cost propagation, and explainable rule-based action evaluation within the empirically supported smart-city IoT scope.

3. Methodology

3.1. Problem Formulation

This study investigates proxy operational-cost modeling and candidate-action evaluation for multi-domain infrastructure in digital twin-enabled smart-city IoT systems. Given a dynamic system comprising sensing devices, communication links, edge nodes, urban services, and maintenance resources, observations at time slices $t = 1, \dots, T$ are represented as:

$$G_t = (V_t, E_t, X_t, R_t) \tag{1}$$

where V_t is the entity set; E_t contains physical connections, data flows, service dependencies, and maintenance associations; $X_t \in \mathbb{R}^{|V_t|\times d}$ contains node features; d is the feature dimension; and R_t is the relation-type set. A historical input window is defined as:

$$X = \{G_{t-L+1}, G_{t-L+2}, \dots, G_t\} \tag{2}$$

where L is the window length. X may contain device states, link performance, edge load, service quality, resource consumption, fault alarms, and maintenance-resource states. The output is:

$$Y = (C_{t+1:t+H}, S_{t+1:t+H}, \Delta C_{t+1:t+H}(a)) \tag{3}$$

where H is the prediction horizon. $C_{t+1:t+H}$ denotes future normalized proxy costs constructed from congestion, latency, anomaly risk, and service degradation. In future Energy Web applications with real O&M records, the cost variable could instead be calibrated using observed energy consumption, maintenance expenditure, downtime loss, SLA penalties, or computing costs; these quantities are not available in the current experiments. $S_{t+1:t+H}$ denotes future risk states; a denotes a candidate action; and $\Delta C_{t+1:t+H}(a)$ is its cost change relative to the baseline. Without observed action outcomes, $\Delta C(a)$ is approximated through rule-based digital-twin interventions only to evaluate ranking consistency.

The training set is:

$$\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^{N} \tag{4}$$

where N is the number of training samples. A mapping $f: X \mapsto Y$ estimates proxy costs, risk states, and predicted action-induced changes. The objective is:

$$f^* = \arg\min_f \mathbb{E}_{(X,Y)\sim P(X,Y)}[\mathcal{L}(f(X), Y)] \tag{5}$$

where $\mathcal{L}$ is the overall loss, $P(X,Y)$ is the joint sample distribution, and f^* is the loss-minimizing function. The objective models risk propagation, proxy-cost changes, and action ranking rather than directly minimizing financial expenditure or estimating causal intervention effects

3.2. Overall Framework

The overall framework is shown in Figure 1. It takes multi-domain operational data from digital twin-enabled smart-city IoT systems as input, including transportation sensing, communication links, edge-computing loads, service quality, resource-related operational burden, fault alarms, and maintenance resources. These data are organized into dynamic cross-layer graph sequences to characterize associations among sensing devices, communication nodes, edge-computing nodes, urban services, maintenance resources, and proxy operational costs. The framework then performs cost-semantic graph construction, state–risk propagation modeling, rule-based candidate-action evaluation, and attribution-based explanation.

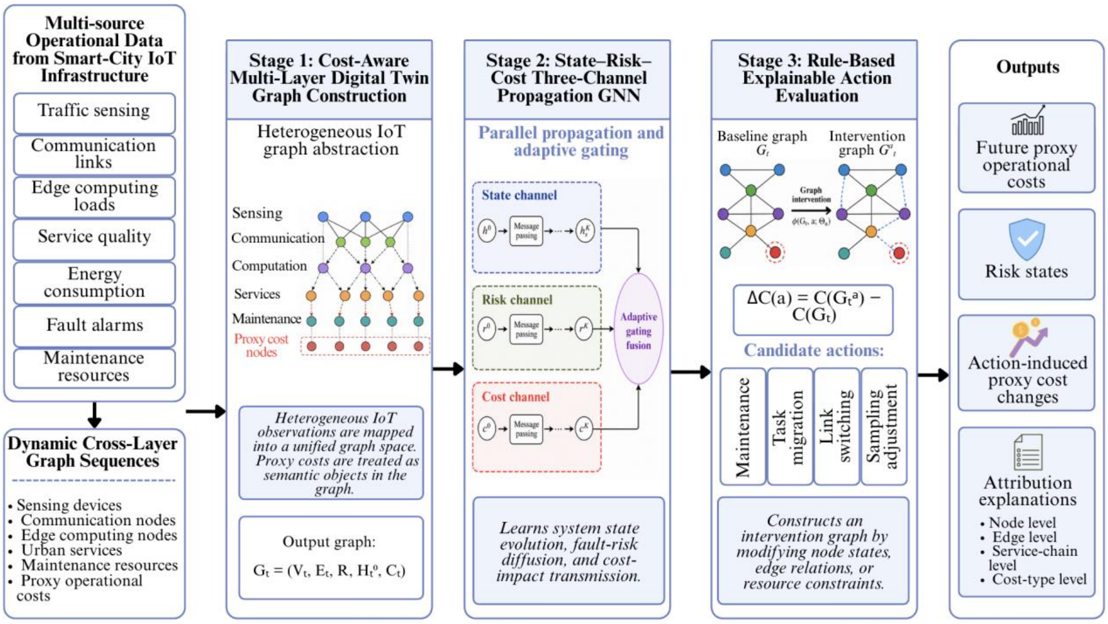

Figure 1. Overall framework of the proposed method

The first stage is the cost-aware multi-layer digital-twin graph construction module. It maps heterogeneous IoT observations into a unified graph where sensing, communication, computing, service, maintenance, and proxy-cost entities are represented jointly. Proxy costs are treated as semantic graph objects, providing a structural basis for cost tracing.

The second stage is the state–risk–cost three-channel propagation GNN module. It models system-state evolution, fault-risk diffusion, and cost-impact transmission, thereby characterizing how local anomalies affect system-level proxy costs through communication links, edge-computing nodes, and urban-service chains.

The third stage is the rule-based explainable action-evaluation module. Based on the state, risk, and cost representations, it estimates predicted proxy-cost changes under predefined candidate actions, including maintenance, task migration, link switching, and sampling adjustment, and produces corresponding attribution explanations.

The three modules are not simply connected as an independent sequential pipeline. Cost-node representations provide semantic constraints for cost propagation, while the three-channel outputs establish the state, risk, and cost baselines for rule-based intervention evaluation. The final outputs include future proxy costs, risk states, predicted proxy-cost changes for candidate actions, and attribution explanations at the node, edge, service-chain, and cost-type levels. The framework therefore establishes a modeling process from multi-source smart-city IoT data to proxy-cost prediction, rule-based action ranking, and explanation output.

3.3. Module Descriptions

3.3.1. Cost-Aware Multi-Layer Digital Twin Graph Construction Module

Smart-city IoT infrastructure consists of heterogeneous entities such as traffic sensing devices, communication links, edge computing resources, urban services, and maintenance resources. If the graph is constructed solely on the basis of physical topology or traffic correlations, it is difficult to represent the cross-layer dependencies among device states, service quality, maintenance resources, and proxy costs. Therefore, this module is designed to organize multi-source observations in a unified manner and introduce cost semantics into the graph representation.

The module constructs a cost-aware heterogeneous dynamic graph in which entities are represented as nodes, while physical connections, data flows, service dependencies, fault propagation, and cost attribution are represented as multi-relational edges. Type-specific projections align heterogeneous features within a shared space. With real O&M data, cost nodes may represent energy consumption, maintenance, downtime, and SLA violations. In the public-data experiments, they represent congestion, latency, anomaly risk, and service degradation. Edges connecting cost nodes with device, service, and maintenance nodes are initialized from operational dependencies, training-set correlations, and predefined proxy-cost rules, and are subsequently updated through relation-aware attention.

As shown in Figure 2, the original feature of node v, whose type is $\tau(v)$, is projected as:

$$h_v^0 = W_{\tau(v)} x_v + b_{\tau(v)} \tag{6}$$

where x_v is the original feature vector, $\tau(v)$is the node type, h_v^0 is the projected representation, and $W_{\tau(v)}$ and $b_{\tau(v)}$ are type-specific learnable parameters. The resulting cost-aware graph is:

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{R}_t, H_t^0, \mathcal{C}_t) \tag{7}$$

where $\mathcal{V}_t$, $\mathcal{E}_t$, and $\mathcal{R}_t$ denote the augmented node, edge, and relation-type sets; H_t^0 contains the projected node representations; and $\mathcal{C}_t \subseteq \mathcal{V}_t$ is the cost-node set.

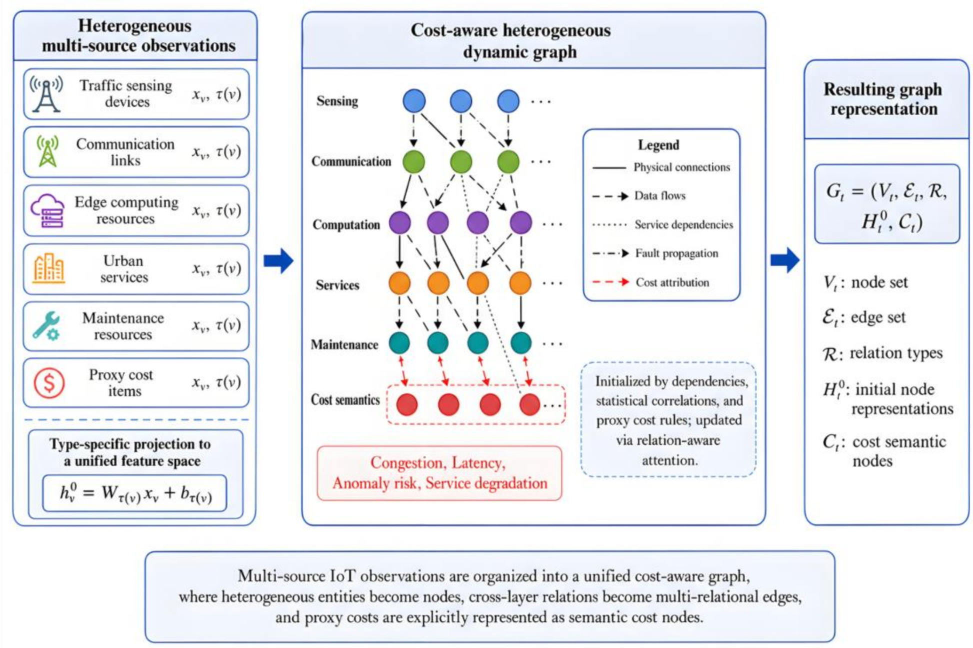

Figure 2. Cost-Aware Multi-Layer Digital Twin graph construction module

3.3.2. State–Risk–Cost Three-Channel Propagation GNN Module

In smart-city IoT O&M, operational states, fault risks, and proxy-cost formation are related but semantically distinct. Operational states describe observed conditions, risks characterize derived abnormalities, and proxy costs represent downstream consequences. A unified aggregation mechanism may mix congestion, failures, overload, and proxy-cost changes into general feature variations. Therefore, the three-channel mechanism is introduced as an inductive bias to reduce cross-task interference. The state channel models operational variables, the risk channel models fault, congestion, and service-degradation diffusion, and the cost channel models their proxy-cost consequences. The channels are jointly trained through the risk-identification, cost-prediction, action-impact, and explanation objectives defined in Section 3.4.

To separate the benefit of semantic partitioning from increased model capacity, the ablation study includes a parameter-matched implicit-channel variant that retains three latent branches but learns their assignments automatically through attention.

As shown in Figure 3, the channels propagate information in parallel and are fused through adaptive gating. For node $i$at layer l, the channel-specific message is:

$$m_i^{q,l} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{q,l} W_q^l h_j^l, q \in \{s, r, c\}. \tag{8}$$

where $\mathcal{N}(i)$ is the neighbor set of node i, while s, r, and $c$denote the state, risk, and cost channels. The layer-specific gating weights are:

$$[g_i^{s,l}, g_i^{r,l}, g_i^{c,l}] = \mathrm{softmax}(W_g^l h_i^l + b_g^l) \tag{9}$$

The node representation is updated as:

$$h_i^{l+1} = \sigma(g_i^{s,l} m_i^{s,l} + g_i^{r,l} m_i^{r,l} + g_i^{c,l} m_i^{c,l} + W_0^l h_i^l) \tag{10}$$

where $\alpha_{ij}^{q,l}$ is the relation-aware attention weight, $g_i^{q,l}$ is the channel gate, $\sigma(\cdot)$ is the activation function, and W_q^l, W_g^l, b_g^l, and W_0^l are learnable parameters.

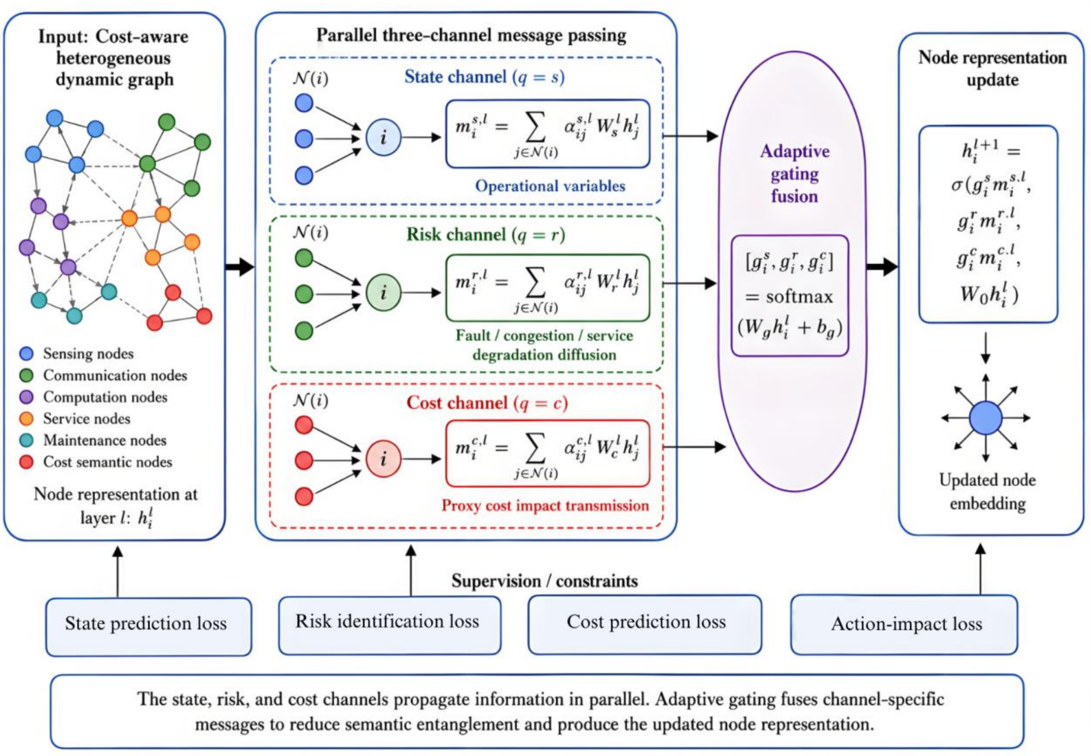

Figure 3. State–Risk–Cost Three-Channel propagation GNN module

3.3.3. Rule-Based Simulation and Explainable Action Evaluation Module

Predicting future costs or risks alone is insufficient for action selection. This module therefore simulates predefined candidate actions, including maintenance, task migration, link switching, and sampling adjustment, and estimates their proxy-cost differences while providing explanations at the node, edge, cost-channel, and service-chain levels. As shown in Figure 4, it compares the baseline graph with rule-modified graphs to rank actions and identify components associated with the predicted differences.

For candidate action a, the module modifies relevant node states, edge relations, or resource constraints. Maintenance changes device-health states and maintenance-resource edges; task migration changes computational loads and service-dependency edges; link switching replaces communication-path edges; and sampling adjustment changes sensing loads and data-flow edges. Because the datasets contain no action-execution logs, these modifications follow predefined proxy-cost rules. The rankings therefore measure rule consistency rather than observed action effectiveness.

The intervention graph is represented as: :

$$\tilde{G}_t^a = \phi(G_t, a; \Theta_a) = \begin{cases} T_a(G_t; \Theta_a), & F_a(G_t) = 1, \\ G_t, & F_a(G_t) = 0, \end{cases} \quad (11)$$

where $T_a(\cdot)$ denotes the action-specific graph transformation and $F_a(G_t)$ is its feasibility indicator. For task migration, $F_a(G_t) = 1$ only when the destination is adjacent in the service-dependency graph and all normalized post-migration node loads remain no greater than 1.0. For link switching, it additionally requires that the replacement edge has a normalized post-switch load no greater than 1.0 and preserves source-to-destination reachability. Actions with $F_a(G_t) = 0$ are excluded from the feasible action set A_i and are not ranked. Because actual capacity and routing logs are unavailable, these limits represent normalized training-range capacity proxies rather than measured hardware capacities.

The simulated proxy-cost difference is:

$$\widehat{\Delta C}(a) = \hat{C}(\tilde{G}_t^a) - \hat{C}(G_t). \quad (12)$$

where $\hat{C}(G_t)$ and $\hat{C}(\tilde{G}_t^a)$ denote the model-predicted baseline and post-intervention proxy costs, respectively. To distinguish the model output from the rule-generated training reference, the predicted difference is denoted by $\widehat{\Delta C}(a)$, whereas $\Delta C^{rule}(a)$ is independently obtained by recomputing the predefined proxy-cost index from the rule-transformed features. Therefore, the rule-generated reference does not depend on the model output. Neither quantity should be interpreted as an observed treatment effect, realized action benefit, or causal effect. For sample i, the raw importance score of explanation unit $j$is defined as:

$$u_{i,j} = \begin{cases} \frac{1}{d_h} \parallel \tilde{\mathbf{h}}_{i,j}^{a} - \mathbf{h}_{i,j} \parallel_1, & j \text{ is a node}, \\ \mid \tilde{\alpha}_{i,j}^{a} - \alpha_{i,j} \mid, & j \text{ is an edge}, \\ \mid \hat{C}_{i,j}^{a} - \hat{C}_{i,j} \mid, & j \text{ is a cost channel}, \end{cases} \quad (13)$$

where $\mathbf{h}_{i,j}$ and $\tilde{\mathbf{h}}_{i,j}^{a}$ are the baseline and simulated node representations, while $\alpha_{i,j}$ and $\tilde{\alpha}_{i,j}^{a}$ are the corresponding cost-channel attention weights. A service-chain score is the mean score of its constituent nodes and edges. For a node or edge present in only one graph, its value in the other graph is set to zero.

The normalized explanation weights are:

$$z_{i,j} = 1 - \exp\left(-\frac{u_{i,j}}{\bar{u}_i + \varepsilon}\right), z_{i,j}' = 1 - \exp\left(-\frac{u_{i,j}'}{\bar{u}_i' + \varepsilon}\right), \quad (14)$$

where $\bar{u}_i = K_i^{-1} \sum_{j=1}^{K_i} u_{i,j}$, $\bar{u}_i' = K_i^{-1} \sum_{j=1}^{K_i} u_{i,j}'$, K_i is the number of explanation units, and $\varepsilon = 10^{-8}$. The perturbed score $u_{i,j}'$ is obtained by repeating Equation (13) after applying $\mathbf{X}_i' = \mathbf{X}_i + \boldsymbol{\epsilon}_i$, where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, 0.01^2\mathbf{I})$. The graph topology and action remain fixed.

The perturbation, transformation rules, and feasibility conditions are treated as constants during optimization. Gradients from $z_{i,j}$ and $z_{i,j}'$ propagate through the representations, attention weights, and proxy-cost predictions using first-order automatic differentiation. The resulting weights are used in the explanation constraint in Equation (19).

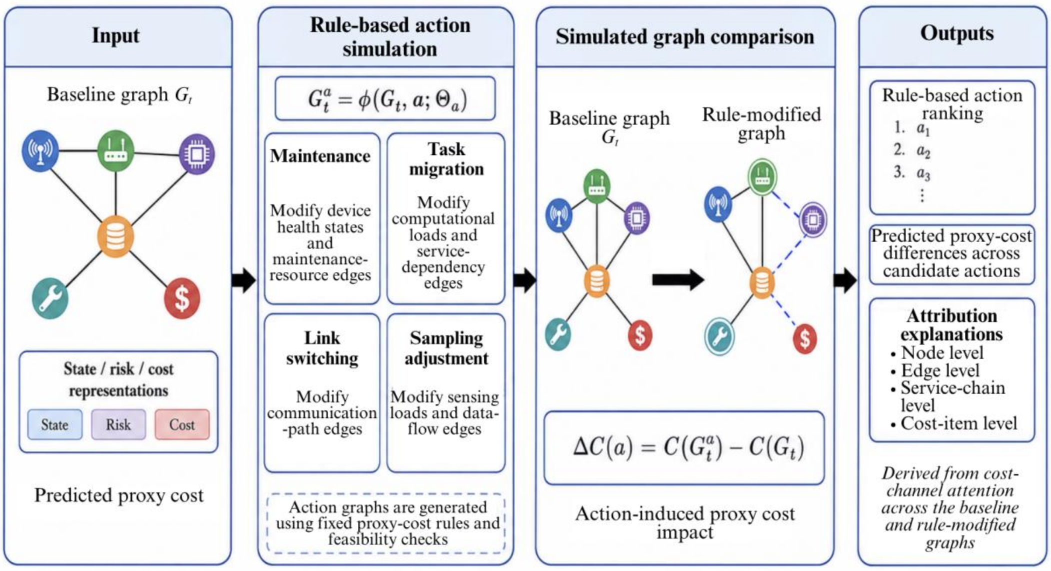

Figure 4. Rule-based action simulation and explainable proxy-cost comparison

3.4. Objective Function and Optimization

To jointly optimize proxy-cost prediction, risk identification, action-impact estimation, and explanation quality, the following multi-task objective is used:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cost}} + \lambda_r \mathcal{L}_{\text{risk}} + \lambda_a \mathcal{L}_{\text{act}} + \lambda_e \mathcal{L}_{\text{exp}} + \lambda_2 \parallel \theta \parallel_2^2, \quad (15)$$

where θ contains all trainable parameters, while λ_r, λ_a, and λ_e control the corresponding task losses. Their candidate values are {0.1, 0.3, 0.5, 0.7, 1.0}, and the final combination is selected using the validation set. The regularization coefficient λ_2 is $10^{-4}$and is implemented through AdamW weight decay.

Let $\hat{C}_{i,t}$ and $C_{i,t}$ denote the predicted and reference proxy costs of sample i at forecast step t, respectively. The cost-prediction loss is:

$$\mathcal{L}_{\text{act}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|A_i^{feas}|} \sum_{a \in A_i^{feas}} \left(\widehat{\Delta C}_i(a) - \Delta C_i^{rule}(a)\right)^2$$

(16)

where $\Delta C_i^{rule}(a) < 0$ indicates a rule-generated proxy-cost reduction, while $\Delta C_i^{rule}(a) > 0$ indicates an increase.

Risk identification is formulated as a multi-label task because anomaly, congestion, and service degradation may occur simultaneously. Let M be the number of risk types, $y_{i,t,m} \in \{0,1\}$ the label of risk type m, and $\hat{p}_{i,t,m} \in [0,1]$ its predicted probability. The weighted binary cross-entropy loss is:

$$\mathcal{L}_{\text{risk}} = -\frac{1}{NH} \sum_{i=1}^{N} \sum_{t=1}^{H} \sum_{m=1}^{M} \bar{\omega}_m \left[y_{i,t,m} \log \hat{p}_{i,t,m} + (1 - y_{i,t,m}) \log(1 - \hat{p}_{i,t,m})\right] \quad (17)$$

where $\omega_m = \frac{1}{\log(1+n_m)}$, $\bar{\omega}_m = \frac{\omega_m}{\sum_{k=1}^{M}\omega_k}$, and n_m is the number of positive training labels for risk type m. The normalized weights assign greater importance to less frequent risks while controlling the overall loss scale.

Let A_i denote the feasible action set for sample i. For action $a \in A_i^{feas}$, $\widehat{\Delta C}_i(a)$ is the model-predicted proxy-cost change defined in Equation (12), while $\Delta C_i^{rule}(a)$ is the rule-generated reference obtained by recomputing the proxy-cost index after the fixed graph transformation. The action-impact loss is:

$$\mathcal{L}_{\text{act}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|A_i|}\sum_{a\in A_i}\left[\widehat{\Delta C}_i(a) - \Delta C_i^{\text{rule}}(a)\right]^2. \quad (18)$$

A negative reference value indicates a rule-generated proxy-cost reduction, whereas a positive value indicates an increase. Because the datasets contain no observed action outcomes, $\Delta C_i^{\text{rule}}(a)$ is simulation-based and does not represent a realized or causal action effect.

The explanation constraint promotes sparse and perturbation-stable attribution weights. Using $z_{i,j}$ and $z'_{i,j}$defined in Equation (14), the loss is:

$$\mathcal{L}_{\text{exp}} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{K_i}\sum_{j=1}^{K_i} z_{i,j} + \frac{1}{K_i}\sum_{j=1}^{K_i} | z_{i,j} - z'_{i,j} |\right], (19)$$

where K_i is the number of node, edge, cost-channel, and service-chain explanation units. The first term penalizes diffuse attribution, while the second penalizes changes under the Gaussian perturbation defined in Section 3.3.3. The original and perturbed inputs are processed through two forward passes with shared model parameters. The sampled noise, rule-based graph transformations, and feasibility masks are treated as constants, while gradients from $z_{i,j}$ and $z'_{i,j}$ propagate through the node representations, attention layers, and proxy-cost heads using first-order automatic differentiation.

The model is trained using AdamW with a learning rate of 10^{-3}, weight decay of 10^{-4}, batch size of 32, and a maximum of 200 epochs. Training stops when the total validation loss does not improve for 20 consecutive epochs, and the gradient norm is clipped at 5.0. Continuous inputs are standardized using training-set statistics, while proxy-cost labels are normalized to $[0, 1]$. All hyperparameters are selected using the validation set and remain fixed during testing.

Algorithm 1 summarizes the training, intervention, and action-ranking procedure.

Algorithm 1. Training and action-ranking procedure

Input: Dynamic graph sequences G, labels Y, candidate actions A, maximum epochs R
Output: Proxy-cost predictions, risk scores, action rankings, and explanations

```
Initialize model parameters θ and best validation loss L_best.
for epoch = 1to R do
    for each mini-batch Bfrom (G, Y)do
        Construct cost-aware graphs and project node features.
        Compute state, risk, and cost messages.
        Predict proxy costs and risk scores.
        for each feasible action a ∈ A do
            Generate the rule-modified intervention graph.
            Estimate the proxy-cost change and z_{i,j}.
        end for
        Perturb continuous inputs and calculate z'_{i,j}.
        Compute the multi-task objective.
        Update θ using AdamW.
    end for
    Evaluate the total validation loss.
    if validation loss <L_best then
        Save the checkpoint and update L_best.
    end if
end for
Load the checkpoint with the lowest validation loss.
Output predictions, risk scores, rankings, and explanations.
```

4. Experiments and Results

4.1. Experimental Settings

Two public datasets evaluate the cost-aware graph, three-channel propagation, and rule-based action-simulation mechanism: CityPulse Smart City Datasets and the Caltrans Performance Measurement System (PeMS), as summarized in Table 2. Consistent with the revised smart-city IoT scope, CityPulse combines traffic, parking, and weather observations to examine heterogeneous cross-domain representation, whereas PeMS provides a homogeneous traffic-sensor reference for assessing the method's boundary. Neither dataset is presented as a physical energy-system dataset. They do not contain energy-system measurements, maintenance work orders, financial expenditures, SLA-loss records, or observed interventions. The experiments therefore assess proxy-cost prediction, risk identification, and rule-ranking consistency within smart-city IoT settings rather than direct energy-system performance, financial optimization, observed action effectiveness, or causal effects.

Table 2. Key information of the experimental datasets

Dataset	Main purpose	Task type	Data modality	Experimental split	Key features / target variables	Source / access
CityPulse Smart City Datasets	Heterogeneous smart-city IoT proxy validation and ablation	Multi-source spatiotemporal graph prediction, risk identification, and proxy cost prediction	Vehicle traffic, parking occupancy, weather, etc.	Chronological split	Traffic flow, parking occupancy, and weather states are used as inputs; congestion risk and normalized proxy cost are used as target variables	Public smart-city IoT datasets from the CityPulse project
Caltrans PeMS	Homogeneous traffic-sensor reference and boundary analysis	Traffic-state prediction, congestion-risk identification, and delay-based proxy cost prediction	Freeway detectors, incidents, lane closures, vehicle classification, etc.	Chronological split	Detector states are used as node features; speed reduction, congestion level, and delay penalties are used to construct risk and proxy cost labels	Official data source of the California Department of Transportation

Accordingly, CityPulse and PeMS serve different methodological roles within the empirically supported smart-city IoT scope and are not used to claim direct validation of Energy Web operation. Records are chronologically divided into training, validation, and test sets at 70%/10%/20%. The split is fixed across runs, and no input window crosses a boundary. Standardization uses training-set statistics. CityPulse records are aligned to their coarsest common resolution; continuous missing values are interpolated, while discrete states are forward-filled. PeMS detector measurements are interpolated, and recorded incident and lane-closure states are forward-filled. Traffic, parking, weather, and detector entities form nodes, while road connectivity, spatial proximity, data-flow dependencies, and cost attribution form directed edges. After preprocessing, CityPulse contains 128 nodes and 512 edges, while PeMS contains 207 nodes and 621 edges. Both topologies remain fixed across all splits.

Proxy operational cost is a dimensionless composite index. All components are min–max normalized to $[0,1]$ using training statistics. For CityPulse, c_t, d_t, r_t, and s_t denote traffic intensity, travel-time increase, the proportion of traffic, parking, and weather variables exceeding two training standard deviations, and parking saturation, respectively. For PeMS, they denote detector occupancy, relative speed deficit, $d_t = \mathrm{clip}\left(1 - \frac{v_t}{v_{\mathrm{ff}}}, 0,1\right)$, the proportion of detector variables exceeding two training standard deviations, and incident or lane-closure severity. Here, $v_{\mathrm{ff}}$is the training-set 85th-percentile speed. The index is: $C_t^{\mathrm{proxy}} = \alpha_1 c_t + \alpha_2 d_t + \alpha_3 r_t + \alpha_4 s_t$, $\sum_{j=1}^{4} \alpha_j = 1$, with $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.25,0.25,0.25,0.25)$. Equal weighting is retained as a transparent reference because the datasets provide no monetary evidence for estimating component-specific economic importance. It does not imply that congestion, delay, risk, and service degradation have equal real-world financial consequences. To evaluate dependence on this assumption, four severely skewed configurations are additionally tested: congestion-dominant $(0.70, 0.10, 0.10, 0.10)$, delay-dominant $(0.10, 0.70, 0.10, 0.10)$, risk-dominant $(0.10, 0.10, 0.70, 0.10)$, and degradation-dominant $(0.10, 0.10, 0.10, 0.70)$. For each configuration, the proxy-cost labels are reconstructed and the proposed model is retrained using the same preprocessing, data splits, hyperparameters, and five random seeds. The sensitivity analysis reports MAE, NDCG@5, and Fidelity in Section 4.7. All weight configurations are fixed before training and are not tuned on validation or test data. The index therefore measures relative smart-city IoT operational burden rather than observed expenditure, realized savings, component-level economic importance, or physical energy-system cost.

Risk identification uses three non-exclusive labels: anomaly, congestion, and service degradation. Anomaly is activated when at least one standardized observation exceeds two training standard deviations. CityPulse congestion is derived from traffic intensity and travel-time increase, while degradation reflects parking saturation and abnormal environmental conditions. PeMS congestion uses occupancy and speed deficit, while degradation uses incidents and lane closures. The same training-derived rules are applied to all splits.

Action-simulation rules are fixed before training and use deterministic rather than stochastic transformations.

Equivalently, the migration and sampling ratios follow point-mass distributions $\rho_{\text{mig}} \sim \delta_{0.20}$ and $\rho_{\text{samp}} \sim \delta_{0.20}$; maintenance replacement is fixed at the historical-window median, and link selection is fixed at the feasible edge with the lowest delay. No Monte Carlo sampling or fitted distribution is used.

Maintenance replaces abnormal sensing features with historical-window medians and resets the associated risk indicator. For task migration from node $u$to adjacent node v, $\tilde{l}_u = (1-\rho_{\text{mig}})l_u$and $\tilde{l}_v = l_v + \rho_{\text{mig}} l_u$. Migration is feasible only when both loads remain within 1.0. Link switching selects the lowest-delay alternative from the predefined topology only if its post-switch load does not exceed 1.0 and source-to-destination reachability is preserved. Sampling adjustment reduces sensing load and outgoing edge weights by 20%. Actions without a feasible destination or path are excluded. These limits are normalized proxies rather than measured computing or communication capacities.

In CityPulse, maintenance and sampling adjustment use observed variables, whereas task migration and link switching are simulated because task and path logs are unavailable. PeMS evaluates only detector maintenance and sampling adjustment. Each ranking item is an action–target pair, and all feasible pairs are included. For each pair, the rule-generated reference score $\Delta C_i^{\text{rule}}(a)$ is the proxy-cost difference obtained after applying the corresponding fixed graph transformation. The pairs are ordered from the largest predicted proxy-cost reduction to the smallest. These scores are simulation-derived references rather than observed action outcomes. The pairs are ordered from the largest predicted proxy-cost reduction to the smallest. These scores are simulation-derived references rather than ground-truth action outcomes. NDCG@5 is calculated for the five highest-ranked pairs, excluding samples with fewer than five feasible pairs. It therefore measures agreement with rule-generated reference rankings rather than post-action system rewards, real-world effectiveness, or causal benefit.

Experiments use one NVIDIA RTX 4090 GPU, Python 3.10, PyTorch 2.x, and CUDA 12.x. All models use an input window of $L = 12$ and horizon of $H = 3$. The proposed GNN contains two relation-aware layers with a hidden dimension of 64. Each channel uses four 16-dimensional attention heads, followed by concatenation, linear projection, ReLU, 0.1 dropout, and a residual connection. AdamW uses a learning rate of 10^{-3}, weight decay of 10^{-4}, batch size of 32, and at most 200 epochs. Early stopping uses validation loss, and the test set is reserved for final reporting. The total loss follows Section 3.4. Robustness tests add Gaussian noise at 5%, 10%, and 15% of each feature's standard deviation. These evaluation-time robustness levels are separate from the 1% standardized Gaussian perturbation used to calculate the explanation-stability loss in Equation (19). Five seeds, $\{13, 21, 42, 87, 102\}$, affect only initialization, dropout, and mini-batch order. Results are reported as mean ± standard deviation. Two-sided paired t-tests use the five paired runs, and Cohen's d_{av} provides a descriptive effect-size estimate.

Four metrics are used. MAE evaluates proxy-cost prediction: $\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i |$. F1 evaluates multi-label risk identification: $\text{F1} = \frac{2PR}{P+R}$. NDCG@5 measures consistency with rule-generated action rankings, excluding samples with fewer than five feasible action–target pairs. As a rule-independent plausibility check, two IoT O&M reviewers blindly rank feasible actions for 30 randomly selected CityPulse test windows without accessing the rules or model outputs. The proposed method achieves an NDCG@3 of 0.713 and Kendall's $\tau$of 0.526 against their averaged rankings, while inter-reviewer weighted Cohen's κ is 0.684. This assessment does not measure realized action outcomes. Fidelity measures the action-score change after removing the top-ranked explanation units. For action–target pair (i, a) : $q_i(a; G_i) = -\widehat{\Delta C}_i(a; G_i), S_i^{(k)} = \text{TopK}\left(\{z_{i,j}\}_{j=1}^{K_i}\right), k = \lceil 0.1 K_i \rceil$. Fidelity is defined as:

$$\text{Fidelity} = \frac{1}{|Q|}\sum_{(i,a)\in Q} | q_i(a; G_i) - q_i\left(a; G_i \setminus S_i^{(k)}\right) |, \quad (20)$$

where Q contains all evaluated pairs. Selected node features are replaced by their training-set means, edges are deleted, cost-channel activations are masked, and selected service chains mask their constituent nodes and edges. All explainers use the same top-10% removal budget; higher Fidelity indicates greater dependence on the identified units.

GNNExplainer is applied to the trained model as an external post-hoc baseline [29]. Its structural and feature masks $\mathbf{M}_i$ are optimized by:

$$\mathcal{L}_{\text{GE}} = [q_i(a; G_i) - q_i(a; G_i \odot \sigma(\mathbf{M}_i))]^2 + \beta_1 \parallel \sigma(\mathbf{M}_i) \parallel_1 + \beta_2 \mathcal{H}(\sigma(\mathbf{M}_i)) \quad (21)$$

where $\mathcal{H}(\cdot)$ is mask entropy. Each instance is optimized for 100 iterations using Adam with a learning rate of 0.01, $\beta_1 = 0.005$, and $\beta_2 = 0.1$. Node and service-chain scores are averaged from their constituent masks, while cost-channel scores use the corresponding cost-node masks. Both explainers are evaluated on the same test pairs using Equation (20).

4.2. Baselines

The baselines include classical and recent spatiotemporal models and heterogeneous GNNs, adapted from their original single-task versions. Their encoders are retained, while identical proxy-cost, risk-identification, and action-scoring heads provide consistent outputs. All models receive the same multimodal observations, with heterogeneous features aligned through identical type-specific projections. All models use the same data split, input window, prediction horizon, labels, loss functions, training epochs, and early-stopping rule. Each baseline is independently tuned on the validation set using the same search space: hidden dimensions $\{32, 64, 128\}$, learning rates $\{10^{-4}, 3\times 10^{-4}, 10^{-3}\}$, dropout rates $\{0, 0.1, 0.3\}$, and weight decay values $\{10^{-5}, 10^{-4}\}$. The configuration with the lowest validation multi-task loss is selected, without using the test set for model selection.

The classical models include DCRNN and STGCN [30,31]. DCRNN combines diffusion convolution with recurrent structures, while STGCN integrates graph convolution and gated temporal convolution. Both implementations use two graph layers and 64-dimensional hidden representations. These models capture traffic dependencies but lack cost semantic nodes and explicit risk–cost propagation.

The recent baselines include PDFormer, MegaCRN, and STAEformer [32–34]. PDFormer models propagation delays and long-range dependencies, MegaCRN captures non-stationary patterns through meta-graph learning, and STAEformer uses spatiotemporal adaptive embeddings. Although competitive in traffic forecasting, they do not explicitly model cost nodes and propagation relationships.

HAN and HGT are added as heterogeneous graph baselines [35,36]. HAN applies node- and semantic-level attention over relation-derived meta-path views, whereas HGT uses node-type- and relation-type-dependent transformations. Both receive the same heterogeneous graph schema, node and relation types, proxy-cost nodes, multimodal features, historical windows, feature projections, output heads, candidate actions, and labels as the proposed method. Their selected implementations use two graph layers, 64-dimensional representations, and four attention heads. HAN and HGT use the same graph schema, input features, multimodal projections, prediction heads, labels, data splits, and training protocol as the proposed method. They replace the proposed encoder and do not include explicit state–risk–cost propagation or rule-modified graph comparison. These comparisons therefore test whether the proposed architecture provides benefits beyond general heterogeneous projection and aggregation.

Accordingly, the traffic models serve as spatiotemporal prediction references, while HAN and HGT provide heterogeneous graph-encoding references. The experiments examine proxy-cost modeling, rule-ranking consistency, and explanation credibility under both baseline categories.

4.3. Quantitative Results

The proposed method is compared with DCRNN, STGCN, PDFormer, MegaCRN, and STAEformer, as well as the heterogeneous graph baselines HAN and HGT, on CityPulse and PeMS. Results are reported as mean ± standard deviation over five runs using the same seeds. Unless otherwise stated, the main comparisons use the equal-weight proxy-cost configuration (0.25, 0.25, 0.25, 0.25) defined in Section 4.1. For each metric, a two-sided paired t-test compares the proposed method with the strongest corresponding baseline among all seven alternatives. Cohen's d_{av} is reported as a standardized effect-size estimate, with positive values favoring the proposed method. Given the limited number of runs, significance and effect sizes are interpreted cautiously. The results are shown in Table 3. Sensitivity to severely skewed proxy-cost weights is examined separately in Section 4.7.

Table 3. Main performance comparison and external Fidelity reference on CityPulse and PeMS

Method	CityPulse MAE ↓	CityPulse F1 ↑	CityPulse NDCG@5 ↑	CityPulse Fidelity ↑	PeMS MAE ↓	PeMS F1 ↑	PeMS NDCG@5 ↑	PeMS Fidelity ↑
DCRNN	0.169±0.011	0.724±0.018	0.689±0.034	0.428±0.026	0.137±0.006	0.754±0.012	0.708±0.020	0.414±0.022
STGCN	0.157±0.009	0.746±0.015	0.706±0.025	0.433±0.030	0.134±0.005	0.768±0.014	0.719±0.018	0.427±0.019
PDFormer	0.146±0.008	0.771±0.013	0.724±0.021	0.466±0.031	0.119±0.004	0.792±0.009	0.751±0.016	0.446±0.021
MegaCRN	0.149±0.014	0.763±0.017	0.742±0.026	0.472±0.028	0.127±0.007	0.779±0.014	0.764±0.019	0.461±0.025
STAEformer	0.142±0.007	0.779±0.012	0.735±0.018	0.487±0.022	0.122±0.005	0.787±0.011	0.756±0.017	0.468±0.020
HAN	0.141±0.009	0.769±0.014	0.749±0.024	0.499±0.027	0.130±0.007	0.775±0.013	0.737±0.021	0.489±0.026
HGT	0.137±0.008	0.777±0.013	0.758±0.022	0.517±0.029	0.123±0.006	0.785±0.012	0.748±0.020	0.501±0.028
GNNExplainer (post-hoc)	—	—	—	0.529±0.032	—	—	—	0.509±0.029
Proposed method	0.130±0.010* ($d_{av}=0.78$)	0.772±0.016 ($d_{av}=-0.50$)	0.776±0.027* ($d_{av}=0.73$)	0.552±0.033* ($d_{av}=1.13$)	0.125±0.006 ($d_{av}=-1.20$)	0.783±0.013 ($d_{av}=-0.82$)	0.741±0.026 ($d_{av}=-1.02$)	0.526±0.031* ($d_{av}=0.85$)

Note: Table 3 reports the equal-weight proxy-cost reference configuration. For the proposed method, * indicates a significant improvement, † indicates a significant decrease, and no symbol indicates no significant difference at $p < 0.05$. All tests use paired results from the same five seeds. Cohen's d_{av} is calculated relative to the strongest predictive baseline for each metric, excluding GNNExplainer, with positive values favoring the proposed method. GNNExplainer is applied to the trained proposed model as an external post-hoc explanation reference and therefore reports only Fidelity; "—" indicates that the metric is not applicable. Both explanation methods use the same test pairs, Fidelity definition, and top-10% removal budget. Alternative weight configurations are reported in Section 4.7.

Table 3 shows scenario- and metric-specific results. On CityPulse, HGT is the strongest predictive baseline, with an MAE of 0.137, an F1 of 0.777, an NDCG@5 of 0.758, and a Fidelity of 0.517. Compared with HGT, the proposed method improves MAE to 0.130, NDCG@5 to 0.776, and Fidelity to 0.552, with effect sizes of 0.78, 0.73, and 1.13. Because both methods use the same graph schema, proxy-cost nodes, projections, and task heads, these differences support the contribution of explicit state–risk–cost propagation. However, the proposed F1 of 0.772 remains below STAEformer's 0.779 and HGT's 0.777.

On PeMS, PDFormer achieves the best MAE and F1, while MegaCRN achieves the best NDCG@5. The proposed method remains lower on these metrics, with effect sizes of

-1.20, -0.82, and -1.02. It nevertheless achieves the highest Fidelity of 0.526, compared with HGT's 0.501 ($d_{av} = 0.85$). Thus, its benefits are less consistent in the homogeneous detector network, precluding overall superiority on PeMS.

GNNExplainer achieves Fidelity values of 0.529±0.032 on CityPulse and 0.509±0.029 on PeMS, compared with 0.552±0.033 and 0.526±0.031 for the proposed explanation module. Under the same test pairs and top-10% removal budget, the proposed module yields moderately higher mean Fidelity. However, this comparison assesses attribution faithfulness rather than the causal or operational correctness of the identified units.

Table 4. Efficiency and model complexity comparison under a fixed CityPulse input configuration

Method	Parameters / M	FLOPs / G	Training time per epoch / s	Inference time / ms	Convergence epoch
DCRNN	2.1	5.8	18.4	6.2	74
STGCN	1.8	4.9	15.7	5.5	68
PDFormer	5.6	12.4	31.8	10.9	52
MegaCRN	4.9	10.7	28.6	9.8	57
STAEformer	6.2	13.1	34.5	11.6	49
HAN	3.7	8.5	24.6	8.3	61
HGT	4.8	10.9	30.1	10.0	58
Proposed method	4.3	9.6	27.2	9.1	55

Table 4 reports efficiency under the same RTX 4090 GPU, fixed CityPulse topology, $L = 12$, $H = 3$, batch size of 32, and training protocol. The proposed method has 4.3 M parameters, requires 27.2 s per epoch, and converges near epoch 55. Its computational requirements are lower than those of HGT, PDFormer, MegaCRN, and STAEformer but higher than those of HAN, DCRNN, and STGCN. It therefore has intermediate complexity rather than a general efficiency advantage.

Figure 5 presents normalized training and validation loss curves for the proposed method and the representative spatiotemporal baselines. DCRNN and STGCN stabilize after approximately 65–75 epochs, MegaCRN after 55–60 epochs, and PDFormer and STAEformer after approximately 50 epochs. The proposed method fluctuates moderately between epochs 40 and 60 before stabilizing. The convergence epochs of HAN and HGT are reported separately in Table 4 and are omitted from Figure 5 for visual clarity. Validation loss follows training loss without persistent divergence, showing no clear evidence of severe overfitting under the reported setting. Final checkpoints are selected at the minimum validation loss.

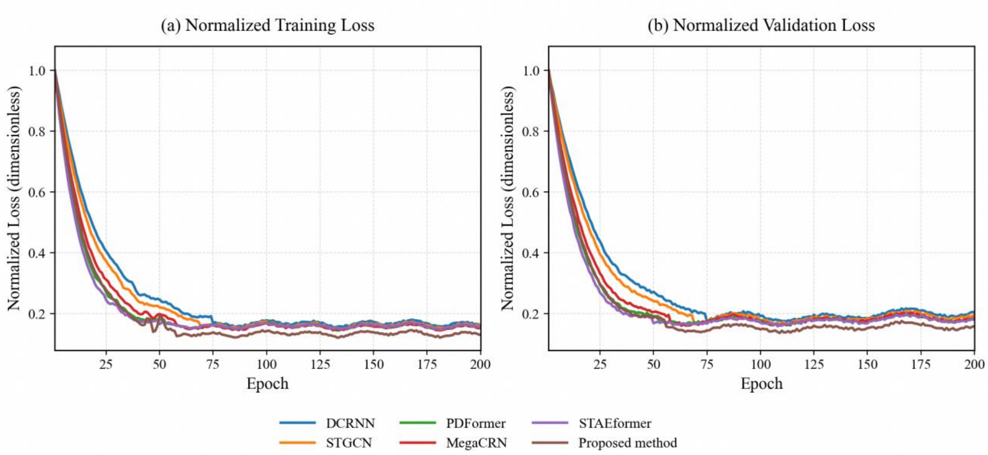

Figure 5. Normalized training and validation loss curves of the proposed method and representative spatiotemporal baselines. (a) Normalized training loss; (b) normalized validation loss

Overall, the results are scenario- and metric-dependent. The HAN and HGT comparisons show that heterogeneous graph encoding improves several CityPulse metrics but does not fully explain the performance of the proposed method. The proposed method improves CityPulse MAE, rule-ranking NDCG@5, and Fidelity but not F1. On PeMS, it improves Fidelity while remaining below the strongest traffic-specific and heterogeneous baselines in MAE, F1, and NDCG@5. Its contribution therefore lies in proxy-cost semantics, explicit state–risk–cost propagation, rule-ranking consistency, and attribution explanation for heterogeneous O&M settings rather than consistent superiority over all competing architectures. These conclusions refer to the equal-weight configuration; Section 4.7 evaluates whether they remain stable under severely skewed weights.

4.4. Case Analysis

To examine model behavior on specific samples, two representative CityPulse time windows are selected. Figures 6 and 7 show cases in which the predicted action ranking is more and less consistent with the rule-generated reference ranking, respectively. The visualizations include proxy-cost changes, risk intensity, candidate rankings, and attention to key nodes. These cases illustrate model behavior rather than real-world action effectiveness.

For a rule-independent plausibility check, 30 CityPulse test windows are randomly selected for blinded assessment. Two reviewers familiar with IoT O&M independently rank feasible actions using only traffic, parking, weather, and risk states, without access to rule-generated labels or model outputs. Agreement with the consensus ranking is measured by NDCG@3 and Kendall's τ, while weighted Cohen's κ evaluates inter-rater agreement. This assessment complements rule-based NDCG@5 but does not establish causal effectiveness.

Figure 6 corresponds to a window with increased traffic flow, fluctuating parking occupancy, and weather disturbance, followed by a higher proxy cost. The model attends mainly to traffic nodes, parking nodes, and related service edges, and ranks "sampling adjustment" and "task migration" among the top actions. This result is consistent with the predefined intervention ranking, suggesting that the model combines traffic, parking, and environmental information effectively in this sample.

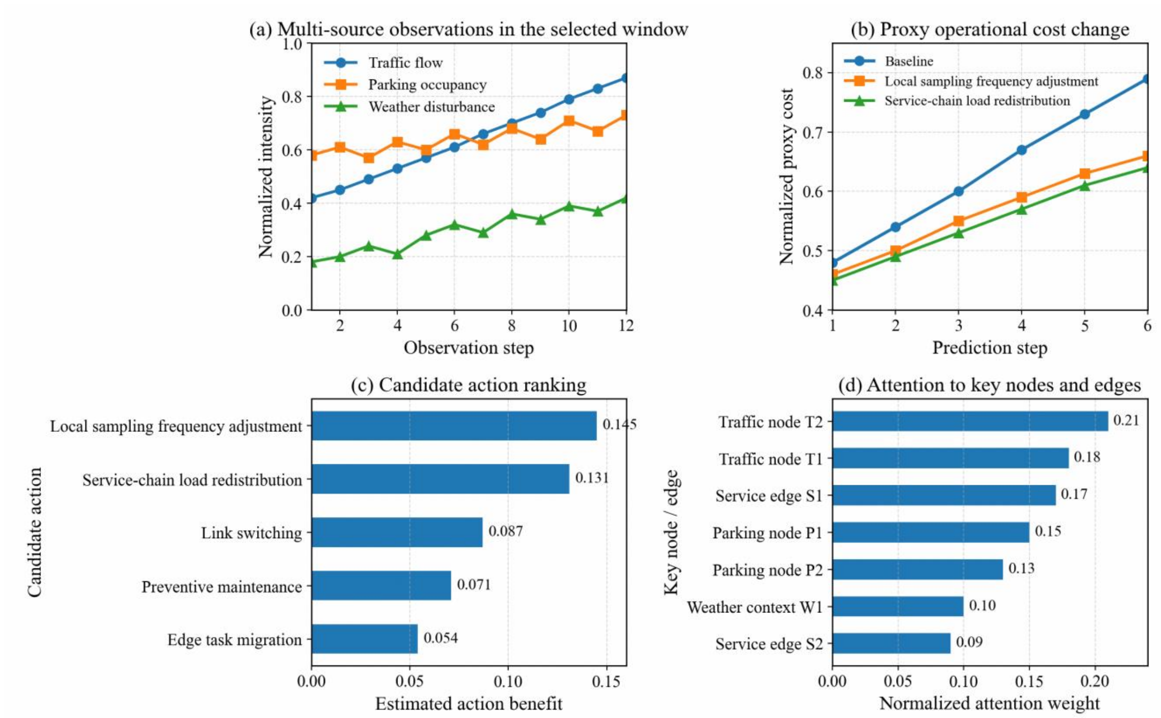

Figure 6. Visualization of action-benefit ranking in a cost-increasing scenario. (a) Multi-source observations. (b) Proxy operational cost change. (c) Candidate action ranking. (d) Attention to key nodes and edges

Figure 7 presents a higher-uncertainty window. Traffic changes are limited, while parking and weather variables fluctuate and some observations are imputed. The model ranks "link switching" highly among alternatives satisfying the normalized load and connectivity checks, although the predicted proxy-cost decrease is small. Meanwhile, the rule-based scores of "maintenance" and "sampling adjustment" are similar. This case indicates that missing modalities or similar action scores may cause excessive reliance on local edge-weight changes and biased rankings.

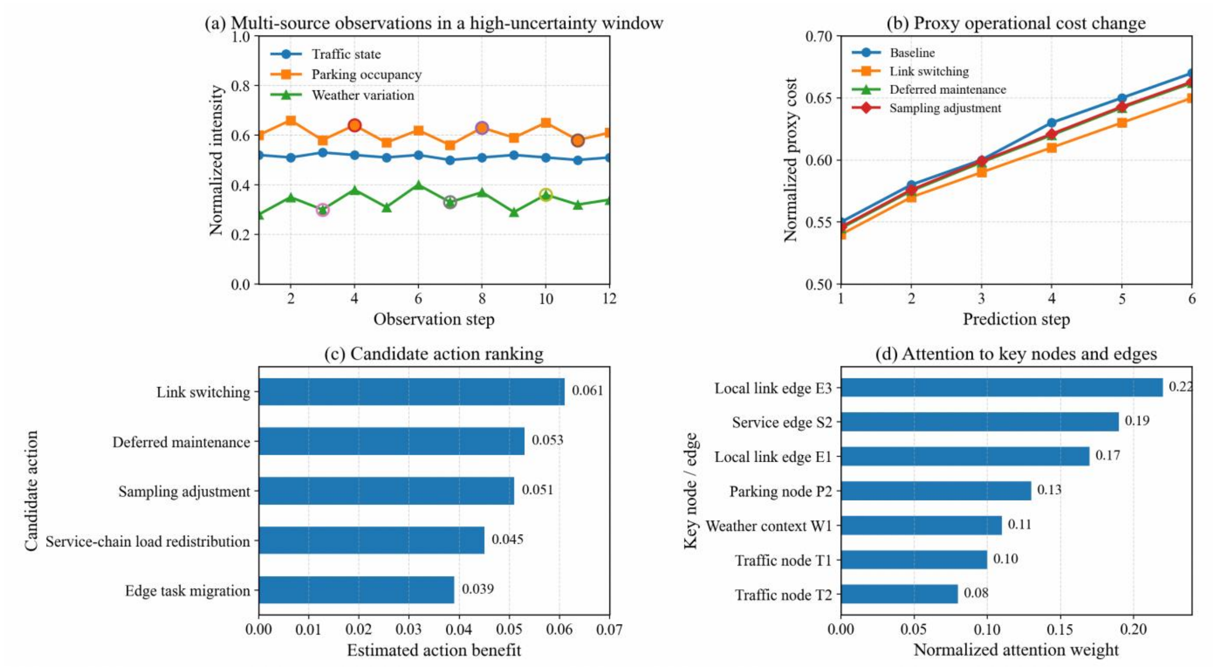

Figure 7. Visualization of action-ranking bias under insufficient modality information. (a) Multi-source observations in a high-uncertainty window. (b) Proxy operational cost change. (c) Candidate action ranking. (d) Attention to key nodes and edges

.Overall, Figure 6 shows a rule-consistent case, whereas Figure 7 illustrates a failure mode caused by missing information or small differences among rule-based action scores. The blinded assessment provides an independent plausibility reference, although neither evaluation verifies real-world causal effects. Future work should incorporate real maintenance records, service-quality logs, and uncertainty estimation to reduce proxy-label and local-noise effects.

4.5. Robustness

To evaluate the stability of the model under non-ideal inputs, this study conducts noise perturbation experiments on the CityPulse test set. Specifically, zero-mean Gaussian noise is injected into continuous input features, with the noise standard deviation set to 5%, 10%, and 15% of the feature standard deviation. The results are compared with those under the noise-free setting. Figure 8 shows the performance changes of different methods under different noise levels, where Figure 8(a) reports MAE and Figure 8(b) reports NDCG@5.

The results show that all methods experience performance degradation as the noise level increases. DCRNN and STGCN exhibit a relatively rapid increase in MAE when the noise level exceeds 10%. PDFormer, MegaCRN, and STAEformer remain relatively stable under low noise, but MegaCRN and STAEformer show more noticeable NDCG@5 degradation when the noise level increases to 15%. By comparison, the proposed method degrades more gradually within the 0%–10% noise range and consistently maintains the lowest MAE and the highest NDCG@5 across all noise levels. This suggests that the proposed method has better robustness under low- to moderate-intensity perturbations.

Specifically, under 10% noise, the proposed method achieves an MAE of 0.149, while its NDCG@5 decreases from 0.776 to 0.731, corresponding to a decline of approximately 5.8%. Under the same noise level, the NDCG@5 values of MegaCRN and STAEformer decrease by approximately 6.1%. Although the difference is not substantial, it indicates that the proposed method maintains action-ranking performance slightly better under moderate noise. When the noise level reaches 15%, the MAE of the proposed method increases to 0.174, and its NDCG@5 decreases to approximately 0.682, indicating that high-intensity perturbations still weaken model performance. Nevertheless, the proposed method still achieves the best performance among all compared methods under this high-noise setting.

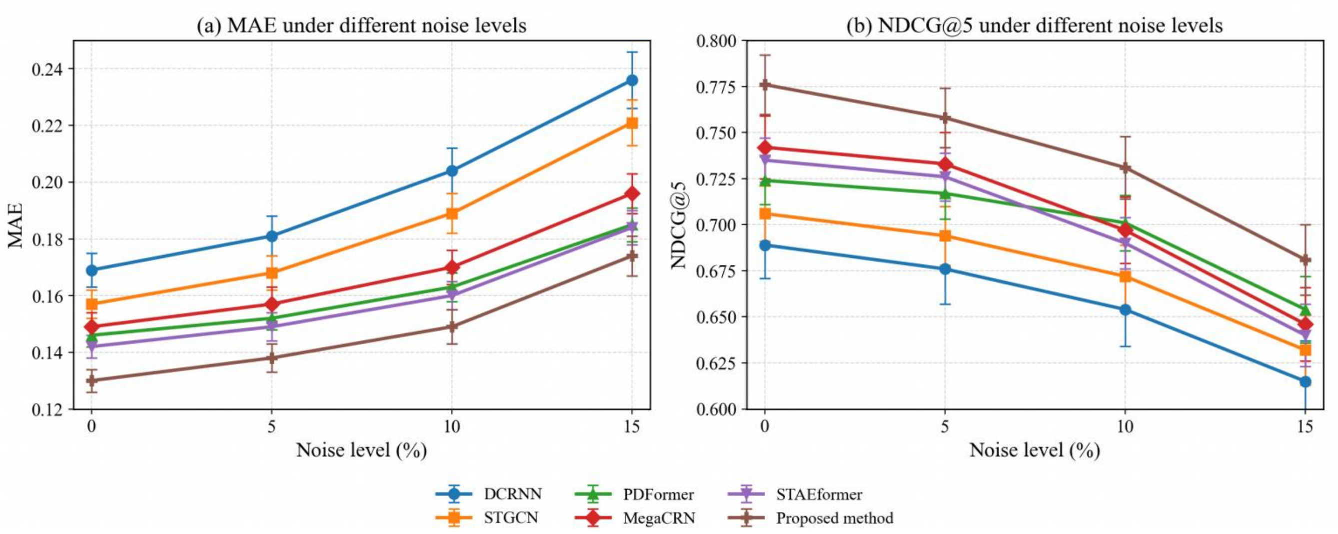

Figure 8. Robustness comparison under different noise levels: (a) MAE under different noise levels; (b) NDCG@5 under different noise levels

Overall, the proposed method maintains relatively stable proxy cost prediction and action-ranking capability under low to moderate noise levels. These results support its continued evaluation as an offline proxy O&M tool under mild missingness, measurement errors, or short-term fluctuations, but do not establish readiness for engineering deployment. For high-noise settings or severe modality missingness, further integration with uncertainty modeling and data-quality assessment mechanisms is still required.

4.6. Ablation Study

To evaluate the contribution of each module, this study conducts ablation experiments on CityPulse and PeMS. All results are reported as the mean ± standard deviation over five independent runs, as shown in Table 5. The variants include removing cost semantic nodes, removing the risk channel, replacing three-channel propagation with single-channel propagation, using a parameter-matched implicit-channel attention mechanism, removing rule-based action simulation, and removing the explanation constraint $\mathcal{L}_{\text{exp}}$.

In the w/o Rule-Based Simulation variant, the rule-modified action graphs and the before-and-after proxy-cost comparison in Equation (12) are removed. Candidate action–target pairs are instead ranked directly from the baseline graph representation using the same action-scoring head. The combined variant additionally removes $\mathcal{L}_{\text{exp}}$.

Table 5. Ablation study results

Model variant	CityPulse MAE ↓	CityPulse NDCG@5 ↑	CityPulse Fidelity ↑	PeMS MAE ↓	PeMS NDCG@5 ↑	PeMS Fidelity ↑
Full model	0.130±0.010	0.776±0.027	0.552±0.033	0.125±0.006	0.741±0.026	0.526±0.031
Implicit-channel attention	0.136±0.012	0.758±0.030	0.531±0.039	0.129±0.007	0.737±0.027	0.507±0.034
w/o Cost Nodes	0.144±0.013	0.739±0.031	0.506±0.038	0.132±0.008	0.735±0.026	0.491±0.029
w/o Risk Channel	0.138±0.011	0.752±0.034	0.519±0.041	0.129±0.007	0.749±0.025	0.505±0.034
Single-channel propagation	0.150±0.016	0.727±0.029	0.487±0.044	0.135±0.009	0.729±0.028	0.476±0.037
w/o Rule-Based Simulation	0.136±0.012	0.704±0.038	0.512±0.036	0.128±0.007	0.713±0.037	0.498±0.033
w/o L_{exp}	0.133±0.009	0.761±0.025	0.459±0.054	0.124±0.007	0.739±0.027	0.448±0.040
w/o Rule-Based Simulation + L_{exp}	0.141±0.015	0.686±0.041	0.428±0.052	0.131±0.010	0.698±0.034	0.421±0.045

The implicit-channel variant retains three latent branches and the same graph layers, gating mechanism, output heads, and training settings as the full model, but learns branch specialization automatically. The difference in trainable parameters is controlled below 1%.

As shown in Table 5, removing cost semantic nodes increases CityPulse MAE from 0.130 to 0.144 and decreases NDCG@5 from 0.776 to 0.739, indicating that cost nodes support proxy-cost prediction and rule-ranking consistency in multi-source scenarios. The smaller PeMS changes suggest more limited benefits in a homogeneous traffic network.

Removing the risk channel increases CityPulse MAE to 0.138 and decreases NDCG@5 to 0.752. The parameter-matched implicit-channel variant obtains a CityPulse MAE of 0.136, NDCG@5 of 0.758, and Fidelity of 0.531, while single-channel propagation further degrades all three metrics. These results suggest that multiple branches are beneficial and that predefined state–risk–cost assignments provide additional gains, particularly on CityPulse.

On PeMS, the full model achieves better MAE and NDCG@5 than the implicit-channel and single-channel variants: 0.125 and 0.741 versus 0.129 and 0.737, and 0.135 and 0.729, respectively. Thus, the multi-channel structure does not cause general information fragmentation. However, removing the risk channel increases NDCG@5 from 0.741 to 0.749 while worsening MAE from 0.125 to 0.129 and Fidelity from 0.526 to 0.505. This trade-off suggests that explicit risk separation may partially fragment ranking-relevant information in homogeneous traffic data, although it remains beneficial for proxy-cost prediction and attribution.

Removing rule-based simulation reduces NDCG@5 to 0.704 on CityPulse and 0.713 on PeMS. This indicates that baseline–intervention comparisons help reproduce preferences encoded in the predefined rules but does not demonstrate real-world action effectiveness. Removing $\mathcal{L}_{\text{exp}}$ has little effect on MAE but substantially reduces Fidelity, confirming that it primarily affects attribution rather than prediction accuracy.

When both rule-based simulation and $\mathcal{L}_{\text{exp}}$ are removed, CityPulse NDCG@5 and Fidelity decrease to 0.686 and 0.428. This supports their complementary roles in rule-ranking consistency and attribution fidelity rather than causal effectiveness. Overall, cost nodes and explicit channels provide clearer benefits on CityPulse, while PeMS shows a narrower trade-off between prediction and ranking. These ablations do not test propagation-delay modeling or adaptive temporal attention; whether traffic-specific encoders can close the PeMS accuracy gap remains for future evaluation.

4.7. Sensitivity to Proxy-Cost Weights

To examine dependence on equal weighting, the proposed model is retrained under four dominant-component configurations defined in Section 4.1. All settings use the same data splits, preprocessing, hyperparameters, and five seeds. Results are shown in Table 6.

Table 6. Sensitivity to proxy-cost weights

Weight configuration	CityPulse MAE ↓	CityPulse NDCG@5 ↑	CityPulse Fidelity ↑	PeMS MAE ↓	PeMS NDCG@5 ↑	PeMS Fidelity ↑
Equal (0.25, 0.25, 0.25, 0.25)	0.130± 0.010	0.776± 0.027	0.552± 0.033	0.125± 0.006	0.741± 0.026	0.526± 0.031
Congestion-dominant (0.70, 0.10, 0.10, 0.10)	0.134± 0.012	0.761± 0.030	0.541± 0.036	0.127± 0.007	0.736± 0.027	0.514± 0.034
Delay-dominant (0.10, 0.70, 0.10, 0.10)	0.132± 0.011	0.769± 0.028	0.547± 0.034	0.121± 0.006	0.750± 0.024	0.520± 0.032
Risk-dominant (0.10, 0.10, 0.70, 0.10)	0.140± 0.014	0.753± 0.035	0.526± 0.039	0.132± 0.009	0.728± 0.031	0.507± 0.038
Degradation-dominant (0.10, 0.10, 0.10, 0.70)	0.137± 0.013	0.746± 0.033	0.533± 0.037	0.129± 0.008	0.734± 0.029	0.511± 0.036

CityPulse MAE ranges from 0.130 to 0.140, while NDCG@5 and Fidelity remain between 0.746–0.776 and 0.526–0.552, respectively. Delay-dominant weighting produces results close to the equal-weight setting, whereas risk- and degradation-dominant settings affect different metrics unevenly. On PeMS, delay-dominant weighting improves MAE and NDCG@5, consistent with the dataset's traffic-delay characteristics, while risk-dominant weighting produces the largest overall variation. These non-monotonic changes show that the results are not determined solely by equal weighting, although proxy-cost definitions still influence prediction and ranking. Economically valid weights require real financial, downtime, and SLA-loss records.

5. Discussion

The proposed method performs better in heterogeneous proxy scenarios such as CityPulse but does not comprehensively outperform traffic-specific or heterogeneous graph models on PeMS. Its supported scope lies in cross-domain representation and O&M decision support for smart-city IoT rather than physical energy-system operation. DCRNN and STGCN capture traffic dependencies [30,31], while PDFormer, MegaCRN, and STAEformer use specialized spatiotemporal modeling [32–34]. HAN and HGT provide heterogeneous references [35,36]. Although the proposed method achieves higher Fidelity on PeMS, its weaker MAE, F1, and NDCG@5 preclude overall superiority. These gaps also preclude direct deployment in homogeneous traffic networks without further accuracy validation.

The comparisons clarify the contribution and boundary of this work. The weaker PeMS accuracy is consistent with the advantage of traffic-specific architectures in pure forecasting [30–34]. On CityPulse, HGT outperforms the traffic baselines in MAE, NDCG@5, and Fidelity. The proposed method further changes these metrics from 0.137, 0.758, and 0.517 to 0.130, 0.776, and 0.552. Because both methods use the same graph schema, proxy-cost nodes, projections, and task heads, these differences are not attributable solely to unequal inputs. However, the proposed method remains below HGT in F1 and several PeMS metrics, showing that semantic separation does not benefit every task [35,36]. Its contribution therefore lies in traceable proxy-cost modeling and heterogeneous O&M support rather than universal superiority [7,9,20–28]. The PeMS gap may arise from semantic partitioning and limited encoder specialization. Table 5 does not indicate general information fragmentation because the full model outperforms the parameter-matched implicit-channel and single-channel variants in MAE and NDCG@5. However, removing the risk channel increases PeMS NDCG@5 from 0.741 to 0.749 while worsening MAE from 0.125 to 0.129, suggesting partial fragmentation of ranking-relevant information. The current encoder also lacks the propagation-delay modeling and adaptive temporal attention used by PDFormer and STAEformer, which may explain why higher Fidelity does not produce stronger predictive metrics.

Cost nodes and three-channel propagation mainly benefit proxy-cost prediction, rule-ranking consistency, and explanation. Rule-based NDCG@5 measures agreement with simulated rankings rather than observed effectiveness. Blinded reviewers provide an independent plausibility assessment, but neither evaluation measures post-action rewards or causal effects. Under the same removal budget, the proposed explanation module achieves Fidelity values of 0.552 and 0.526 on CityPulse and PeMS, compared with 0.529 and 0.509 for GNNExplainer [29]. This comparison supports attribution faithfulness but not the causal correctness of the identified units.

Across equal and four skewed weight configurations, CityPulse MAE, NDCG@5, and Fidelity remain within 0.130–0.140, 0.746–0.776, and 0.526–0.552; the corresponding PeMS ranges are 0.121–0.132, 0.728–0.750, and 0.507–0.526. Delay-dominant weighting performs relatively well on PeMS, while other settings affect metrics unevenly. Thus, equal weighting does not solely determine the results, although proxy-cost definitions influence prediction and ranking. These tests do not identify economically correct weights.

Several limitations remain. Neither dataset contains physical energy measurements, maintenance orders, financial costs, SLA penalties, or closed-loop outcomes. The results therefore demonstrate proxy-cost modeling and rule-ranking performance only. The weight configurations remain analytical rather than economically calibrated. Some CityPulse actions are simulated, while PeMS supports only detector maintenance and sampling adjustment. Capacity and connectivity checks use normalized proxies rather than measured computing capacity, bandwidth, or routing logs; resulting cost differences remain predictions rather than physically validated outcomes. The modules also have intermediate complexity, while incomplete modalities or similar scores may produce misinterpreted service-chain risks. Higher Fidelity likewise does not guarantee semantically correct explanations.

Future work should calibrate proxy costs using financial and service-loss records, evaluate actions against observed outcomes, and incorporate measured capacity and path availability. A hybrid encoder combining data-dependent channel gating with propagation-delay modeling or adaptive temporal attention should be evaluated to reduce possible fragmentation and recover accuracy in homogeneous networks. Further evaluation should use expert-annotated explanations and additional post-hoc explainers. SUMO or NS-3 could independently measure travel time, packet delay, resource consumption, and SLA violations. Energy Web validation would require physical energy, equipment-health, price, and maintenance records.

Practical implementation would first synchronize telemetry, resource states, service quality, maintenance, SLA, and cost records within a smart-city digital twin. These data would be mapped to graph entities, and proxy-cost nodes would be calibrated using observed records. After offline training and shadow-mode evaluation, validated recommendations could be delivered through a human-in-the-loop interface. Automatic execution should follow only after prospective validation. The framework therefore remains an offline decision-support prototype.

6. Conclusion

This study proposed a cost-aware explainable GNN for proxy-cost modeling and O&M decision support in digital twin-enabled smart-city IoT. It integrates multi-source states, risk evolution, proxy-cost semantics, and candidate actions through cost nodes, state–risk–cost propagation, and rule-modified simulation graphs. Experiments show favorable performance in heterogeneous CityPulse scenarios, particularly in proxy-cost prediction, rule-ranking consistency, and Fidelity, but no comprehensive superiority over traffic-specific or heterogeneous graph models. Equal-input comparisons with HAN and HGT indicate that the CityPulse improvements are not explained solely by heterogeneous feature projection. However, weaker F1 and PeMS results confirm task- and scenario-dependent advantages. Under the same removal budget, the proposed explanation module also achieves moderately higher mean Fidelity than GNNExplainer, providing an external reference for attribution faithfulness. Sensitivity tests under congestion-, delay-, risk-, and degradation-dominant weights show that the main performance ranges persist under severely skewed configurations, although metric variation confirms dependence on the proxy-cost definition.

CityPulse and PeMS lack physical energy measurements, maintenance orders, financial expenditures, SLA penalties, and closed-loop actions. The findings therefore represent proxy-based smart-city IoT validation rather than evidence of energy-system performance, cost savings, or causal effectiveness. The sensitivity results do not establish economically valid weights, which require real cost, downtime, and SLA-loss records. Similarly, higher Fidelity does not establish the causal correctness of the explanations. Future work should use real O&M, financial, service-quality, and action-outcome records to calibrate proxy costs and evaluate post-action effects. Independent SUMO or NS-3 experiments are also needed before simulated rankings can be interpreted as operational recommendations.

Acknowledgement
Not applicable.

Declaration
Funding Statement: No funding.
Consent to Publish: Not applicable.
Consent to Participate： Not applicable.
Ethics declaration: Not applicable.
Data Availability Statement: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

[1] Choudhary A. Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things. 2024;4(1):31.

[2] Alourani A, Alam M, Ali A, Khan IR, Samal CK. Hybrid AI-IoT Framework with Digital Twin Integration for Predictive Urban Infrastructure Management in Smart Cities. CMC-COMPUTERS MATERIALS & CONTINUA. 2025;86(1).

[3] Wang Q, Li W, Yu Z, Abbasi Q, Imran M, Ansari S, et al. An overview of emergency communication networks. Remote Sensing. 2023;15(6):1595.

[4] Sowmya R, Nandhini M, Priyanga M. Enhancing Edge Node Resilience through SDN-Driven Proactive Failure Management. In: 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE). IEEE; 2024. p. 15-20.

[5] Jayadharshini P, Santhiya S, Vasuki C, Vanaja T, Archanaa S, Samyuktha K. The Impact of Digital Twin in Industry 4.0 Using Graph Neural Network: An Approach to Explainability in the Manufacturing Industry. In: Distributed Deep Learning and Explainable AI (XAI) in Industry 4.0. Cham: Springer Nature Switzerland; 2025. p. 185-212.

[6] Yu P, Zhang J, Fang H, Li W, Feng L, Zhou F, et al. Digital twin driven service self-healing with graph neural networks in 6G edge networks. IEEE Journal on Selected Areas in Communications. 2023;41(11):3607-3623.

[7] Ngo DT, Aouedi O, Piamrat K, Hassan T, Raipin-Parvédy P. Empowering digital twin for future networks with graph neural networks: Overview, enabling technologies, challenges, and opportunities. Future Internet. 2023;15(12):377.

[8] Mahmoudi Rashid S. Graph Neural Network-Based Digital Twin for Cyber-Resilient and Predictive Teleoperation Systems. Journal of AI and Data Mining. 2026;14(1):1-12.

[9] Guo K, Chen J, Dong P, Zou T, Zhu J, Huang X, et al. Dtfl: A digital twin-assisted graph neural network approach for service function chains failure localization. IEEE Transactions on Cloud Computing. 2023;11(4):3573-3590.

[10] Dahmane WM, Ouchani S, Bouarfa H. Smart cities services and solutions: A systematic review. Data and Information Management. 2025;9(2):100087.

[11] Zaman M, Puryear N, Abdelwahed S, Zohrabi N. A review of IoT-based smart city development and management. Smart Cities. 2024;7(3):1462-1501.

[12] Trigka M, Dritsas E. Edge and cloud computing in smart cities. Future Internet. 2025;17(3):118.

[13] Banerjee A, Costa B, Forkan ARM, Kang YB, Marti F, McCarthy C, et al. 5G enabled smart cities: A real-world evaluation and analysis of 5G using a pilot smart city application. Internet of Things. 2024;28:101326.

[14] Nikpour M, Yousefi PB, Jafarzadeh H, Danesh K, Shomali R, Asadi S, et al. Intelligent energy management with iot framework in smart cities using intelligent analysis: An application of machine learning methods for complex networks and systems. Journal of Network and Computer Applications. 2025;235:104089.

[15] Fereira RJ, Ranaweera C, Lee K, Schneider JG. Energy efficient resource management for real-time IoT applications. Internet of Things. 2025;30:101515.

[16] Fan G, Sabri AQM, Rahman SSA, Pan L, Rahardja S. Emerging Trends in Graph Neural Networks for Traffic Flow Prediction: A Survey. Archives of Computational Methods in Engineering. 2025;32(8):4811-4855.

[17] Hu N, Zhang D, Xie K, Liang W, Li KC, Zomaya AY. Dynamic multi-scale spatial–temporal graph convolutional network for traffic flow prediction. Future Generation Computer Systems. 2024;158:323-332.

[18] Qadir MA, Naeem M, Ejaz W. Digital twin-assisted multi-layer networks for low-latency and energy-efficient communication. Computer Communications. 2025;108219.

[19] Rostami M, Goli-Bidgoli S. An overview of QoS-aware load balancing techniques in SDN-based IoT networks. Journal of Cloud Computing. 2024;13(1):89.

[20] Zhang Z, Liu Y, Peng Z, Chen M, Xu D, Cui S. Digital twin-assisted data-driven optimization for reliable edge caching in wireless networks. IEEE Journal on Selected Areas in Communications. 2024;42(11):3306-3320.

[21] Zhang Z, Peng Z, Yu H, Chen M, Liu Y. Digital network twins for next-generation wireless: Creation, optimization, and challenges. IEEE Network. 2025.

[22] Zhang Z, Chen M, Yang Z, Liu Y. Mapping wireless networks into digital reality through joint vertical and horizontal learning. In: 2024 IFIP Networking Conference (IFIP Networking). IEEE; 2024. p. 359-367.

[23] Mbasso WF, Harrison A, Dagal I, Jangir P, Khishe M, Kotb H, et al. Digital twins in renewable energy systems: A comprehensive review of concepts, applications, and future directions. Energy Strategy Reviews. 2025;61:101814.

[24] Ba L, Tangour F, El Abbassi I, Absi R. Analysis of digital twin applications in energy efficiency: A systematic review. Sustainability. 2025;17(8):3560.

[25] Materwala H, Ismail L, Shubair RM, Buyya R. Energy-SLA-aware genetic algorithm for edge–cloud integrated computation offloading in vehicular networks. Future Generation Computer Systems. 2022;135:205-222.

[26] Serra G, Niepert M. L2XGNN: learning to explain graph neural networks. Machine Learning. 2024;113(9):6787-6809.

[27] Yousefzadeh N, Sengupta R, Dilmore J, Ranka S. TGDT: A Temporal Graph-based Digital Twin for Urban Traffic Corridors. arXiv preprint arXiv:2504.18008. 2025.

[28] Aykurt K, Stephan M, Ayvasik S, Zerwas J, Kellerer W. Digital Twin Opportunities with Leveraging Graph Neural Networks on Real Network Data. ITU Journal on Future and Evolving Technologies (ITU J-FET). 2024.
[29] Ying R, Bourgeois D, You J, Zitnik M, Leskovec J. Gnnexplainer: Generating explanations for graph neural networks. Advances in Neural Information Processing Systems. 2019;32:9240-9251.
[30] Li Y, Yu R, Shahabi C, Liu Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018). 2018.
[31] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018). 2018. p. 3634-3640.
[32] Jiang J, Han C, Zhao WX, Wang J. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37, No. 4. 2023. p. 4365-4373.
[33] Jiang R, Wang Z, Yong J, Jeph P, Chen Q, Kobayashi Y, et al. Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37, No. 7. 2023. p. 8078-8086.
[34] Liu H, Dong Z, Jiang R, Deng J, Deng J, Chen Q, et al. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023. p. 4125-4129.
[35] Wang X, Ji H, Shi C, Wang B, Ye Y, Cui P, et al. Heterogeneous graph attention network. In: The World Wide Web Conference. 2019. p. 2022-2032.
[36] Hu Z, Dong Y, Wang K, Sun Y. Heterogeneous graph transformer. In: Proceedings of the Web Conference 2020. 2020. p. 2704-2710.