# Heuristic Knowledge-Driven Spatio-Temporal Forecasting via Multigraph

Xiao Xiao, *Member, IEEE*, Xufeng Xiang, Xinyue Yang, Zhiling Jin, Jing Xu, Shuo Wang, Guoqiang Mao, *Fellow, IEEE*, and Wei Shao, *Member, IEEE*

***Abstract*—The importance of precise long-term forecasting in practical applications continues to rise. Extensive scenarios, including parking resource prediction and environmental quality monitoring, rely significantly on LSTF's accurate spatio-temporal forecasting capabilities. This technology strengthens prediction effectiveness by combining interaction relationships between spatial-temporal dimensions with contextual data integration. Over time, graph neural networks (GNNs) have proven highly effective in capturing spatial interdependencies. Recent advances have introduced multi-GNNs (MGNNs), which incorporate more contextual insights to improve predictive accuracy. However, when MGNNs are applied to long-term spatio-temporal forecasting (LSTF), they encounter challenges such as limited generality, under-utilization of context, static graph merging methods, and overlooking dynamic interrelations. To address these issues, we propose novel graph structures that encode each node's contextual information while fully exploiting long-term spatio-temporal dependencies. Furthermore, this research designs a dynamic multigraph fusion architecture that integrates spatial dimensions, temporal features, and graph attention mechanisms to simultaneously capture intragraph node correlations and cross-graph interactions. To strengthen relational analysis, trainable weight tensors are employed for quantitative evaluation of node importance across graphs. Systematic experiments on three large-scale benchmark datasets confirm that this approach achieves significant performance enhancement for existing GNNs in LSTF tasks.**

***Index Terms*—Attention mechanisms, deep learning, graph neural networks (GNNs), long-term spatio-temporal forecasting (LSTF).**

## I. Introduction

LONG-TERM spatio-temporal forecasting (LSTF) has become increasingly critical in addressing real-world challenges, such as traffic flow prediction [1], [2], [3], parking space forecasting [4], [5], and air quality estimation [6], [7], [8]. Accurate forecasting enhances urban life by improving commuting efficiency [9] and supporting policymakers in resource distribution [10], [11]. Despite its potential, LSTF faces significant challenges due to the inherent nonlinearity and complexity of long-term spatio-temporal patterns, which often exceed the capabilities of traditional knowledge-driven methods [11]. Addressing these challenges requires innovative approaches that can effectively capture and leverage these intricate dependencies.

Data-intensive methodologies have progressed significantly for LSTF tasks. Traditional time-series models (e.g., RNNs) effectively capture temporal patterns but often neglect spatial correlations [12], [13], [14], leading to suboptimal accuracy. In contrast, spatio-temporal graph neural networks (ST-GNNs) [2], [3], [15], [16], [17], [18] have addressed this limitation by integrating spatial and temporal correlations, significantly enhancing predictive performance. However, many existing GNN-based approaches rely on a single graph to represent node relationships, which restricts their ability to capture complex and dynamic interconnections, limiting their overall efficacy.

To overcome this limitation, multi-GNNs (MGNNs) [19], [20], [21], [22], [23] have been proposed to model relationships from multiple perspectives. As illustrated in Fig. 1, an urban environment can be represented using diverse graphs: a distance graph (physical proximity), a neighbor graph (direct adjacency), and a functionality graph (similar activity patterns). By integrating these perspectives, MGNNs can capture richer contextual features and provide a more comprehensive model of interregional dependencies.

However, existing MGNNs still face significant challenges in LSTF, as follows.

1) *Limited Focus on Spatial Node Similarity:* Existing MGNN primarily emphasizes spatial node similarities, such as distance or neighborhood correlations. While Wu et al. [24] introduced an adaptive adjacency matrix leveraging node embedding inner products to uncover latent spatial dependencies, these approaches often underutilize the knowledge encoded in existing adjacency matrices, potentially overlooking critical spatial insights [20].
2) *Model-Specific Multigraph Fusion:* Current MGNNs often employ fusion techniques and graph construction methods tailored to specific GNN architectures. This limits the flexibility and adaptability of MGNNs, making it difficult to generalize across different tasks or optimize module combinations effectively.
3) *Neglect of Long-Term Spatio-Temporal Dependencies:* Many MGNNs project spatio-temporal dependencies from observed data to prediction horizons. However, graph types like distance graphs [3] or neighbor graphs [20] typically encode static spatial data, failing to

Received 20 May 2025; revised 26 October 2025; accepted 13 January 2026. This work was supported in part by the Key Project of the Joint Fund of the National Natural Science Foundation of China under Grant U21A20446 and in part by the Natural Science Basic Research Program of Shaanxi Province under Grant 2025JC-YBMS-716. *(Corresponding authors: Wei Shao; Xufeng Xiang.)*

Xiao Xiao, Xufeng Xiang, Xinyue Yang, Zhiling Jin, and Guoqiang Mao are with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: xufengxiang1@gmail.com).

Jing Xu and Shuo Wang are with the School of Systems Science, Beijing Normal University, Beijing 100875, China.

Wei Shao is with Data61, CSIRO, Melbourne, VIC 3806, Australia (e-mail: wei.shao@data61.csiro.au).

Digital Object Identifier 10.1109/TNNLS.2026.3656372

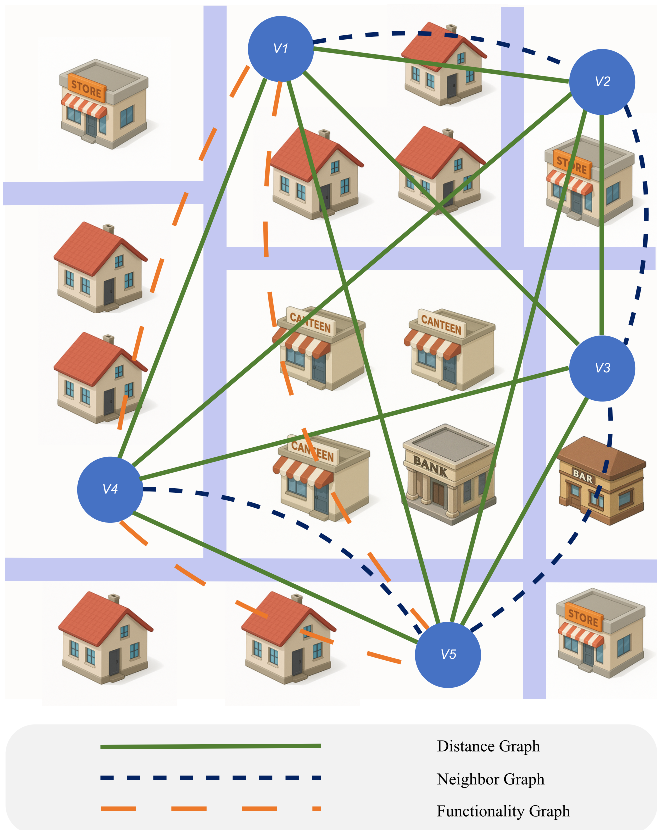



Fig. 1. Illustration of multigraph spatio-temporal forecasting.

capture evolving long-term dependencies inherent in dynamic spatio-temporal contexts.

4) *Inability to Model Time-Relationships:* Most MGNNs rely on static graphs to model node relationships [19], [20], [21], ignoring their temporal variability. For instance, during holidays, nodes near entertainment zones may exhibit stronger correlations. Static graphs cannot adapt to such dynamic changes, reducing prediction accuracy in time-sensitive scenarios.
5) *Static Graph Fusion Limitations:* Traditional static graph fusion methods combine multiple graphs into a single unified graph before training. Once fused, the graph structure remains fixed throughout the learning process, making it incapable of adapting to evolving data distributions over time. This rigidity hampers the model's ability to capture dynamic patterns and dependencies, leading to suboptimal performance in environments with significant temporal variations.

To resolve these issues, we propose a novel graph construction and fusion framework specifically designed for LSTF. Our core contribution is twofold: 1) the construction of a "heuristic graph" that integrates human insights and long-term spatio-temporal data distributions, and 2) a dynamic fusion mechanism to leverage it. This mechanism features two key components: the multigraph spatio-temporal embedding (MGSTE) for generating initial node representations, and the dynamic multigraph attention block (DMGAB). The DMGAB is central to our approach: it dynamically integrates graph weights into a trainable adaptive correlation tensor. This tensor evolves during training, allowing the model to capture contextual node information and dynamic temporal dependencies across multiple graphs, directly addressing the limitations of static fusion. Our framework employs a cascade of spatial, graph, and temporal attention mechanisms to aggregate node information both within and across graphs, ensuring robust modeling of complex spatio-temporal correlations. Finally, we validate the framework by applying it to various ST-GNN models across multiple datasets. Results demonstrate its versatility, practicality, and ability to improve prediction performance in diverse LSTF scenarios.

In summary, our contributions includethe following.

1) Propose a heuristic graph tailored for LSTF, capturing long-term spatio-temporal dependencies from historical or heuristic data. This graph is adaptable to various GNN architectures.
2) Construct a dynamic graph fusion module that combines multiple graph models through graph-based, spatial, and temporal attention mechanisms.
3) Develop a multigraph attention mechanism to capture intrinsic node relationships. By combining spatial, temporal, and graph-level attention, the framework effectively models node correlations across different times and locations, significantly improving LSTF prediction accuracy.
4) Perform comprehensive experimental assessments on three authentic large-scale spatio-temporal datasets. Demonstrate the efficacy of the proposed graph-structured models and fusion approaches via controlled ablation studies.

## II. Related Work

### A. Graph Convolutional Networks

Spatio-temporal data prediction has gained significant attention in recent years, with applications in air quality monitoring, parking availability prediction, and traffic flow analysis. Unlike traditional convolutional neural networks (CNNs), which are unsuitable for processing data in non-Euclidean spaces, graph convolutional networks (GCNs) excel at capturing spatial correlations in such contexts. As a result, GCN-based methods have become increasingly popular for spatio-temporal prediction tasks.

The foundation of GCNs was laid by Bruna et al. [25], who introduced CNNs on graphs. This approach was later enhanced by Defferrard et al. [26], who proposed fast localized convolutions to improve efficiency. Since then, GCNs have been extensively applied in spatio-temporal domains. For example, Zhang et al. [5] introduced a hierarchical semi-supervised recurrent graph neural network-*X* architecture specifically designed for parking availability forecasting. Their method incorporated a contextual graph convolution block and a multiresolution soft clustering graph convolution block to capture both local and global spatial correlations, alongside a hierarchical attentive recurrent network for temporal dependencies. Xiao et al. [10] introduced a dual-path dynamic directed GCN for air quality prediction. By modeling air quality monitoring stations as a directed dynamic graph, their approach combined two directed GCNs with gated recurrent units (GRUs) to capture spatio-temporal dependencies effectively. This design leveraged wind and meteorological data to enhance prediction performance significantly.

### B. Multi-GCNs

While the superiority of GCNs in spatio-temporal data prediction tasks is well established, such data often contains prior knowledge that is underutilized. For instance, in parking availability prediction, functional relationships, such as the distribution of hospitals and schools, may significantly influence parking patterns beyond mere spatial proximity. To incorporate this prior knowledge, researchers have developed multi-GCNs (MGCNs), which leverage multiple graphs to enhance spatio-temporal data modeling.

MGCN-based approaches for spatio-temporal prediction typically involve three key steps: 1) graph or embedding fusion, 2) temporal feature extraction (often via RNNs or CNNs), and 3) multigraph convolution operations. The sequence of these steps may vary depending on the specific application.

For example, Chai et al. [27] introduced a model utilizing three distinct graphs—the distance graph, interaction graph, and correlation graph—to represent node relationships for bike flow prediction. They merged the graphs and applied a multigraph convolution layer followed by an LSTM-based prediction network. Geng et al. [20] proposed the spatio-temporal MGCN (ST-MGCN) for ride-hailing demand prediction. Their approach used three graphs: the neighborhood graph, the function similarity graph, and the transportation connectivity graph. Unlike Chai et al. [27], they encoded these graphs using contextual gated recurrent neural networks (CGRNNs) and GCNs, followed by a fully-connected neural network to generate predictions. Lv et al. [19] designed a temporal MGCN for traffic flow prediction. Their model performed multigraph convolution first and then utilized two GRU layers to capture temporal dependencies for final predictions. Li et al. [28] proposed BLoG, a novel method for graph representation learning that integrates local and global regularization. Local regularization captures fine-grained relationships within local neighborhoods, while global regularization ensures consistency across the entire graph. This approach enhances robustness and accuracy in graph-based recommendation systems, highlighting the potential of multigraph techniques for diverse applications.

Compared to previous works, our approach addresses several key limitations. In particular, we introduce a heuristic graph to effectively capture long-term spatio-temporal dependencies often overlooked by existing methods. In addition, our framework incorporates a dynamic multigraph attention mechanism, enabling adaptive fusion of multiple graphs during training. This overcomes the rigidity of static graph fusion techniques, enhances the representation of dynamic node relationships, and improves adaptability across diverse datasets and scenarios.

### C. Graph Construction

The construction of an effective graph is crucial for the performance of GNNs, particularly when handling heterogeneous and dynamic data. Traditional methods often rely on domain knowledge to construct graphs based on predefined relationships, such as physical proximity or functional similarity. While effective in some contexts, these static approaches often fail to capture the dynamic and evolving nature of spatio-temporal data.

*1) Static Graph Construction:* Static graph construction methods typically rely on fixed criteria, such as $k$-nearest neighbors (KNNs) or threshold-based connections. For instance, Qiu et al. [29] employed KNN to construct graphs from high-dimensional data for semi-supervised learning tasks. These methods are straightforward and computationally efficient but may struggle to represent temporal dynamics, which are critical for spatio-temporal prediction tasks.

*2) Dynamic Graph Construction:* To address the limitations of static graphs, dynamic graph construction methods update graph structures in response to new data, ensuring that the graph reflects the current state of the system. For example, Shi et al. [30] utilized dynamic graph construction techniques to adapt to real-time changes in data distributions, enabling more accurate and timely predictions. By capturing evolving relationships, dynamic methods are better suited for spatio-temporal data.

*3) Multigraph Construction:* Multigraph construction further enhances graph representation by creating multiple graphs, each focusing on distinct aspects of the data. This approach has been widely adopted in various domains. For example, Wang et al. [31] applied multigraph construction in recommendation systems, modeling different types of user-item interactions with separate graphs. By leveraging multiple graphs, this method captures richer and more complex relationships, leading to improved predictive performance across diverse applications.

While existing methods provide valuable insights, they often lack flexibility in integrating heuristic knowledge or adapting to dynamic spatio-temporal contexts. Our approach addresses these gaps by introducing a heuristic graph, which incorporates human insights and long-term spatio-temporal patterns that are typically overlooked by static and dynamic graph construction methods.

### D. Attentional Mechanisms

Attention mechanisms have demonstrated strong effectiveness and flexibility in identifying relevant features within input data, particularly for modeling potential dependencies. In recent years, numerous graph-based models have incorporated attention mechanisms for spatio-temporal prediction tasks.

The use of attention mechanisms in graph-structured data was pioneered by Veličkovi'c et al. [32]. Building on this, Guo et al. [33] proposed the attention-based spatial-temporal GCN (ASTGCN) for traffic flow prediction, introducing a spatio-temporal attention mechanism to capture dynamic dependencies. Huang et al. [1] further

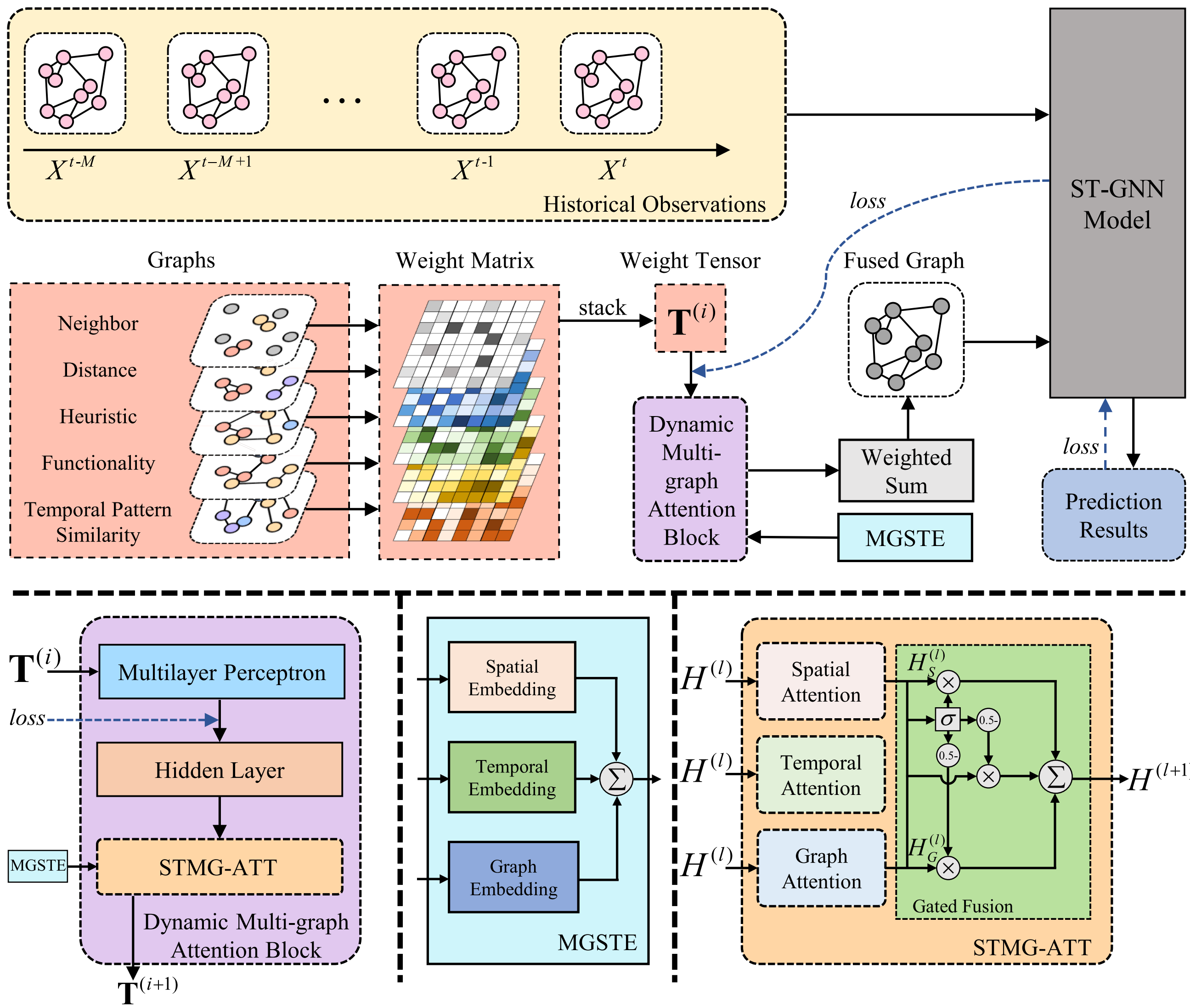



Fig. 2. Overview of the LSTF system.

developed the long short-term GCN (LSGCN), which incorporated a novel attention network called cosATT. This network not only captured dynamic spatio-temporal correlations but also delivered more stable prediction results compared to prior models. Zheng et al. [34] proposed the graph multiattention network (GMAN) for traffic forecasting, integrating multiple attention mechanisms—temporal attention, spatial attention, and transform attention—to comprehensively model spatio-temporal dependencies. These works showcase the potential of attention mechanisms to enhance prediction performance by effectively capturing complex correlations.

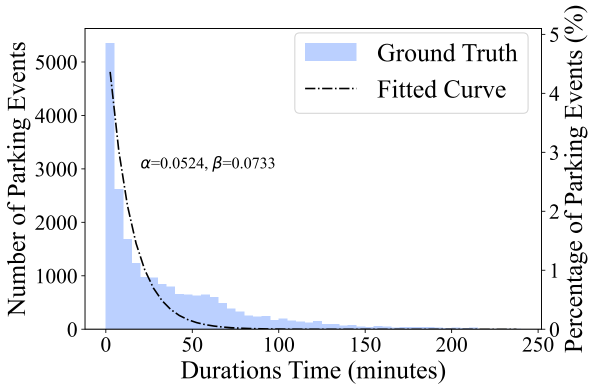
While the above models leverage attention mechanisms to capture spatio-temporal dependencies, they are often limited to single-graph frameworks, which can overlook crucial prior knowledge encoded in multiple graphs. Our approach addresses this limitation by introducing a DMGAB, which integrates multiple attention mechanisms—spatial, temporal, and graph-level—across diverse graphs. By incorporating multiple graphs and dynamically fusing them during training, our framework not only captures richer spatio-temporal relationships but also adapts to evolving data patterns, resulting in improved prediction performance.

## III. Methodologies

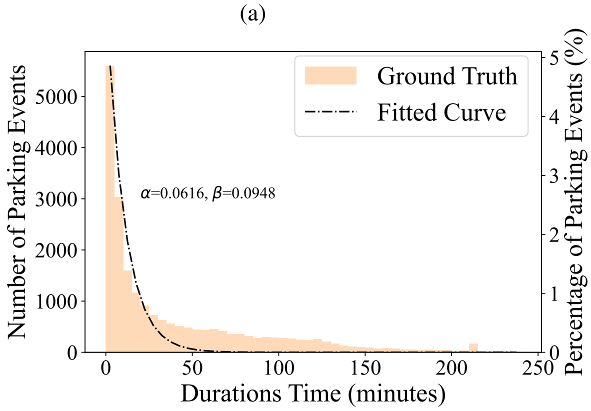
The proposed architecture, depicted in Fig. 2, comprises three key elements: 1) graph construction module, 2) dynamic multigraph integration component, and 3) ST-GNN. The graph construction module generates five distinct graphs to capture the multidimensional characteristics of spatio-temporal data. The fusion module employs adaptable trainable tensors for aligning spatio-temporal dependencies, coupled with spatial, temporal, and graph attention mechanisms to explore intergraph node relationships, ultimately producing forecasts through established ST-GNN frameworks.

### A. Graph Construction

In this part, we present two recently introduced graph models: the heuristic graph $G^H = \{V, E, W^H\}$ and the functionality graph $G^F = \{V, E, W^F\}$. These graphs are designed to capture additional dimensions of spatio-temporal information that are not addressed by traditional graph models. These two graphs are combined with three existing graph models to form a




(a)




(b)

Fig. 3. Parking duration distribution of two randomly selected vertices and their fit curves (a) Chinatown. (b) Mint.

comprehensive multiple graph set $\mathbf{G} = \{G^D, G^N, G^F, G^H, G^T\}$, including the distance graph $G^D = \{V, E, W^D\}$, neighbor graph $G^N = \{V, E, W^N\}$, and temporal pattern similarity graph $G^T = \{V, E, W^T\}$.

*1) Heuristic Graph:* We introduce a new graph model, the heuristic graph $G^H$, which integrates heuristic knowledge and human insights. This graph is designed to capture long-term spatio-temporal data characteristics that may not be evident in traditional graph representations.

The heuristic graph construction begins by analyzing the spatio-temporal training data to extract patterns. In particular, we process the data to generate histograms that represent the overall data distribution for each vertex. For example, in the parking dataset, we calculate histograms of parking durations, with each bin representing a 5-min interval and a maximum duration of 240 min. This step reflects domain knowledge, as it aligns with real-world constraints on parking times.

Next, we fit the histogram data using the function $f(x) = \alpha e^{-\beta x}$, where $\alpha$ and $\beta$ are parameters that describe the data distribution for each vertex $v_i$. These parameters are obtained through least-squares fitting, which ensures an accurate representation of the distribution. By fitting the histogram, we capture the unique data characteristics of each vertex in terms of parking duration patterns. To illustrate the distribution fitting process, Fig. 3 presents the parking duration distributions of two representative vertices along with their exponential fitting curves. The results demonstrate that the proposed fitting function can effectively capture the long-term temporal characteristics of parking behaviour.

To measure the differences between vertices, we compute the Euclidean distance $d_{ij}^H$ based on their respective distribution parameters $\alpha$ and $\beta$

$$d_{ij}^H = \sqrt{(\alpha_1 - \alpha_2)^2 + (\beta_1 - \beta_2)^2}. \tag{1}$$

The similarity between two vertices is then defined in the weight matrix $W^H$ using a Gaussian kernel

$$W_{ij}^H := \begin{cases} \exp\left(-\dfrac{\left\| d_{ij}^H \right\|^2}{\sigma_H^2}\right), & \text{for } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\sigma_H^2$ is a parameter that controls the distribution of edge weights, ensuring that vertices with similar parking patterns are assigned higher weights.

To further enhance robustness, we compute the Kullback–Leibler (KL) divergence between the probability distributions of vertices, offering an additional measure of similarity. The combination of Euclidean distance and KL divergence ensures that $G^H$ captures both structural and probabilistic differences between vertices.

*2) Functionality Graph:* In practical settings, locations with comparable functions or services, like factories, schools, and hospitals, often show significant correlations. These correlations can reflect the global contextual relationships between locations, which are crucial for modeling spatio-temporal dependencies.

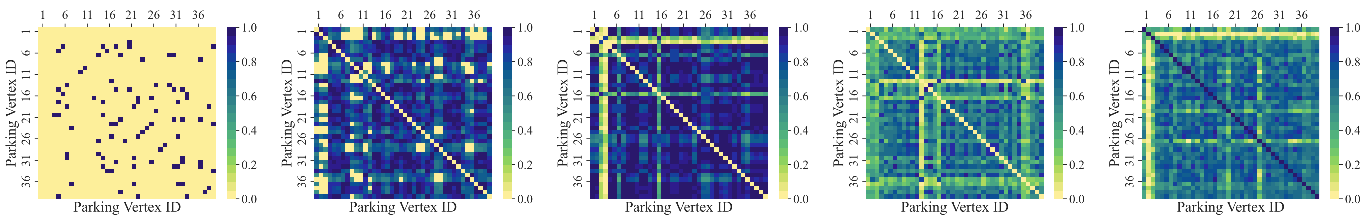
In this article, we propose a novel functionality graph $G^F = \{V, E, W^F\}$, which differs from previous functionality graph models, such as the one proposed by Geng et al. [20]. We use Pearson correlation coefficients [21] in our functionality graph to assess the global contextual similarity of functions between nodes.

Let $K$ be the total number of functions (e.g., factories, schools, and hospitals) considered in the dataset. For a given vertex $v_i$, its functionality vector $F_i = \{f_{i,1}, f_{i,2}, \ldots, f_{i,k}, \ldots, f_{i,K}\}$ represents the number of each function type present within the region associated with $v_i$. Using these vectors, we compute the functionality matrix $W^F$ as follows:

$$W_{ij}^F := \begin{cases} \dfrac{\sum_{k=1}^{K} \left(f_{i,k} - \overline{F_i}\right)\left(f_{j,k} - \overline{F_j}\right)}{\sqrt{\sum_{i=1}^{k} \left(f_{i,k} - \overline{F_i}\right)^2} \sqrt{\sum_{j=1}^{k} \left(f_{j,k} - \overline{F_j}\right)^2}}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $f_{i,k}$ represents the count of function $k$ for vertex $v_i$, $\overline{F_i}$ denotes the mean value of $F_i$, and $W_{ij}^F$ quantifies the functional similarity between vertices $v_i$ and $v_j$ using Pearson correlation coefficients.

This method ensures that all function types contribute equally to the relationships between nodes, reflecting the global contextual similarities. The functionality graph captures these relationships comprehensively, making it particularly effective in domains where functional attributes significantly impact node interactions.




(a) (b) (c) (d) (e)

Fig. 4. Illustrations of five weight matrices on the parking dataset (a) Neighbor. (b) Distance. (c) Heuristic. (d) Functionality. (e) Temporal Pattern Similarity.

*3) Distance Graph:* Distance matrix $W^D$ elements are derived through a threshold-based Gaussian kernel function, illustrated in the following equation [35]:

$$W_{ij}^D := \begin{cases} \exp\left(-\dfrac{d_{ij}^2}{\sigma_D^2}\right), & \text{for } i \neq j \text{ and } \exp\left(-\frac{d_{ij}^2}{\sigma_D^2}\right) \geq \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $d_{ij}$ represents the Euclidean distance between $v_i$ and $v_j$. The parameters $\varepsilon$ and $\sigma_D^2$ are used to regulate the sparsity and distribution of $W^D$.

*4) Neighbor Graph:* Each entry in the neighbor matrix $W^N$ is specified by the following formulation:

$$W_{ij}^N := \begin{cases} 1, & \text{if } v_i \text{and } v_j \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

*5) Temporal Pattern Similarity Graph:* For vertex $v_i$, the training time series vector is defined as $T_i = \{t_{i,1}, \ldots, t_{i,P}\}$, where $P$ denotes sequence length and $t_{i,p}$ corresponds to the temporal data value of $v_i$ at step $p$. The temporal similarity matrix $W^T$ is constructed using Pearson correlation coefficients [21] with element definitions

$$W_{ij}^T := \begin{cases} \dfrac{\sum_{p=1}^{P}\left(t_{i,p} - \overline{T_i}\right)\left(t_{j,p} - \overline{T_j}\right)}{\sqrt{\sum_{i=1}^{p}\left(t_{i,p} - \overline{T_i}\right)^2}\sqrt{\sum_{j=1}^{p}\left(f_{j,p} - \overline{T_j}\right)^2}}, & \text{if } i \neq j \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

In our work, we also propose a new attention-based graph fusion mechanism to merge all these different graphs into a single fused graph $\mathcal{G}^*$. Fig. 4 provides visualizations of the five types of weight matrices used in our model, illustrating how each graph captures unique relationships and dependencies between parking vertices (nodes) in the parking dataset. Each subfigure highlights a distinct aspect of the data through its corresponding graph representation.

1) *Neighbor Graph:* Captures the adjacency relationships between nodes, emphasizing direct spatial proximity. The weight matrix is sparse, with nonzero values indicating immediate neighbors.
2) *Distance Graph:* Represents the physical distances between nodes, with weights inversely proportional to the distance. Shorter distances result in higher weights, indicating stronger spatial connections.
3) *Heuristic Graph:* Encodes long-term spatio-temporal dependencies based on heuristic knowledge, integrating domain-specific insights to model latent interactions.
4) *Functionality Graph:* Reflects functional similarities between nodes, such as their roles in the urban environment (e.g., proximity to schools or hospitals), capturing potential impact on parking behavior.
5) *Temporal Pattern Similarity Graph:* Measures the similarity of historical temporal patterns between nodes, revealing nodes with synchronized or correlated behaviors over time.

These weight matrices together form the backbone of the graph construction process, allowing our model to comprehensively capture spatial, functional, and temporal relationships. The visualizations clearly demonstrate the diversity in graph representations, which is further leveraged by our attention-based graph fusion mechanism to dynamically integrate these relationships into a unified graph $\mathcal{G}^*$, enhancing the overall spatio-temporal modeling.

**Algorithm 1** Dynamic Multigraph Fusion

**Input:** Weight matrices: $W^D$, $W^N$, $W^F$, $W^H$, $W^T$
**Parameter:** Number of batches: Bt
**Output:** Fused weight matrix $W^*$

```
Stack weight matrices to tensor T^(0) ∈ R^{|G|×N×N}.
Train T^(0) while training ST-GNN models.
for i ∈ [0, Bt − 1] do
    T^(i+1) ⟵ T^(i)
    i = i + 1
    W*_jk = Σ_{i=1}^{|G|} T^(i)(i, j, k), where W*_jk is the element of the
    weight matrix of the fused graph.
end for
return W*
```

## B. Dynamic Multigraph Fusion

Effective graph fusion becomes critical in MGNNs due to the inadequacy of elementary operations (e.g., arithmetic averaging or linear combinations) for integrating multiple graph structures [36]. To overcome this limitation, we introduce a dynamic graph fusion approach, with the full process depicted in Fig. 2 and described in detail in Algorithm 1.

The "dynamic" nature of our fusion approach is reflected in two key aspects.

1) *Parameter-Level Dynamics:* The fusion weights $\lambda^D$, $\lambda^N$, $\lambda^F$, $\lambda^H$, and $\lambda^T$ are learnable parameters that continuously evolve during training through backpropagation.
2) *Forward-Pass Dynamics:* When the attention mechanism is enabled, each forward pass recalculates edgewise

attention weights based on current embeddings, allowing adaptive fusion responses to input variations.

The process initiates with the creation of a learnable weight tensor, functioning as the input for the DMGAB. This tensor is generated by stacking the five weight matrices: $W^D$, $W^N$, $W^F$, $W^H$, and $W^T$, corresponding to the distance, neighbor, functionality, heuristic, and temporal pattern similarity graphs, respectively. During the training process, a new graph $W$ is generated by $W_{\text{fuse}} = \lambda^D W^D + \lambda^N W^N + \lambda^F W^F + \lambda^H W^H + \lambda^T W^T$, where $\lambda^D$, $\lambda^N$, $\lambda^F$, $\lambda^H$, and $\lambda^T$ are trainable parameters. These parameters are adjusted iteratively during training, allowing the model to learn the optimal contribution of each graph.

*1) Multigraph Spatio-Temporal Embedding:* To further capture the intricate spatio-temporal correlations across multiple graphs, we introduce an MGSTE. This embedding integrates spatial, temporal, and graph-level information, serving as an additional input to the DMGAB. By incorporating MGSTE, the DMGAB can better model complex dependencies between nodes in different graphs, improving the representation of dynamic relationships.

The spatial embedding $E^S_{v_i} \in \mathbb{R}^D$ is derived using node2vec [37], which preserves the structural information of the graph. For each vertex $v_i, E^S_{v_i} \in \mathbb{R}^D$ represents its high-dimensional spatial features, capturing the connectivity and topology of the graph.

To encode temporal information, we generate a temporal embedding $E^T_{t_i} \in \mathbb{R}^D$ from historical observations. The time series data is first transformed using one-hot encoding and then processed through fully-connected multilayer perceptrons (MLPs). In this study, we use the past 24 time steps to predict the next 24 time steps, ensuring that the embedding captures essential temporal patterns and dependencies.

To model relationships across multiple graphs, we introduce a multigraph embedding $E^{\text{MG}}_{G_i} \in \mathbb{R}^D$. Each of the five graphs ($W^D$, $W^N$, $W^F$, $W^H$, and $W^T$) is encoded into a high-dimensional space $\mathbb{R}^{|\mathbf{G}|}$. A two-layer fully-connected neural network is employed to map this information into a unified vector representation. The final MGSTE for each vertex $v_i$, time step $t_i$, and graph $G_i$ is obtained by fusing the spatial, temporal, and multigraph embeddings

$$E_{v_i,t_i,G_i} = E^S_{v_i} + E^T_{t_i} + E^{\text{MG}}_{G_i}. \tag{7}$$

This fused embedding captures the critical spatial structure, temporal dynamics, and intergraph relationships, providing a comprehensive representation for each node in the multigraph setting. The MGSTE provides a robust foundation for spatio-temporal modeling in complex environments. By unifying information from spatial, temporal, and graph dimensions, it enables the model to fully exploit the richness of multigraph structures and spatio-temporal dependencies, supporting accurate and flexible forecasting in dynamic and heterogeneous datasets.

*2) Dynamic Multigraph Attention Block:* The condition of a node in a graph is influenced by both its historical observations and interactions with other nodes. For instance, in parking scenarios, if a parking area is highly occupied over a period,

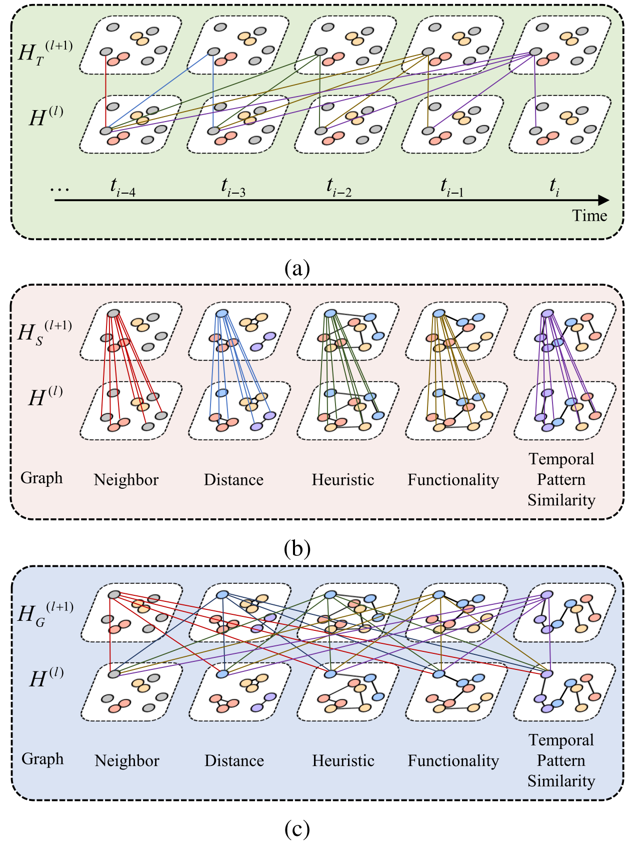



Fig. 5. Attention mechanisms adopted in this article. The neighbor graph is selected for illustration in temporal attention. (a) Temporal attention. (b) Spatial attention. (c) Graph attention.

drivers are likely to shift to nearby areas, which affects the original area's occupancy over time.

To model these dependencies, we present the DMGAB, which combines temporal, spatial, and graph attention mechanisms to model intricate spatio-temporal dependencies across multiple graphs.

The DMGAB implements forward-pass dynamics through its attention mechanisms. Unlike static fusion that uses fixed weights, DMGAB recalculates attention scores at each forward pass based on current node embeddings, enabling adaptive responses to varying spatio-temporal patterns.

Fig. 2 illustrates the architecture of the multigraph attention module incorporating temporal, spatial, and graphwise attention components. The $l$th layer receives input $H^{(l)}$, where $h^{(l)}_{v_i,t_i}$ encodes vertex $v_i$'s temporal state at $t_i$, while $h^{(l)}_{v_i,G_i}$ captures its graph-specific representation in $G_i$. The corresponding outputs are structured as $H^{(l+1)}_T$ (temporal), $H^{(l+1)}_S$ (spatial), and $H^{(l+1)}_G$ (graph-level).

*a) Spatial attention:* This component models internode contextual dependencies within individual graphs. Drawing inspiration from [34], we propose a spatial attention mechanism [visualized in Fig. 5(b)] to capture contextual associations between nodes. Diverging from conventional approaches that process temporal batch hidden states, our method operates on weight tensor hidden states. The subsequent hidden state for graph $G_i$ is computed through the following formulation:

$$hs_{v_i,G_i}^{(l+1)} = \sum_{v_k \in \mathcal{V}_i} \alpha_{v_i,v_k} \cdot h_{v_k,G_i}^{(l)}. \tag{8}$$

The set $\mathcal{V}_i$ corresponds to all graph vertices excluding $v_i$, while the attention coefficient $\alpha_{v_i,v_k}$ quantifies the relevance of vertex $v_k$ to $v_i$.

In practical applications, vertex interactions extend beyond intragraph relationships to include intergraph influences. A concrete example is parking space utilization, where a location's occupancy rate depends on both geographical proximity to other sites and their functional characteristics. To comprehensively model these spatial and structural patterns, we integrate hidden states with MGSTE and employ scaled dot-product computation to quantify vertex correlations between $v_i$ and $v_k$

$$s_{v_i,v_k} = \frac{\left\langle h_{v_i,G_i}^{(l)} \middle\| E_{v_i,G_i}, h_{v_k,G_i}^{(l)} \middle\| E_{v_k,G_i} \right\rangle}{\sqrt{2D}} \tag{9}$$

where $\|$ denotes vector concatenation and $\langle\cdot|\cdot\rangle$ corresponds to the dot-product computation. The raw similarity score $s_{v_i,v_k}$ undergoes softmax normalization: $\alpha_{v_i,v_k} = (\exp(s_{v_i,v_k}))/\left(\sum_{v_k\in\mathcal{V}_i}\exp(s_{v_i,v_k})\right)$, producing attention weights across neighboring vertices.

To enhance training stability, we integrate $M$ parallel attention units into a multihead attention architecture [34] with

$$s_{v_i,v_k}^{(m)} = \frac{\left\langle f_{s,1}^{(m)}\left(h_{v_i,G_i}^{(l)} \| E_{v_i,G_i}\right), f_{s,2}^{(m)}\left(h_{v_k,G_i}^{(l)} \| E_{v_k,G_i}\right)\right\rangle}{\sqrt{d}} \tag{10}$$

$$hs_{v_i,G_i}^{(l+1)} = \Big\|_{m=1}^{M} \left\{\sum_{n=1}^{N} \alpha_{v_i,v_n}^{(m)} \cdot f_{s,3}^{(m)}\left(h_{v_n,G_i}^{(l)}\right)\right\} \tag{11}$$

where $f_{s,1}^{(m)}(\cdot)$, $f_{s,2}^{(m)}(\cdot)$, and $f_{s,3}^{(m)}(\cdot)$ represents distinct ReLU functions applied as nonlinear transformations in the $m$th attention head. The attention weight $\alpha_{v_i,v_n}^{(m)}$ is derived using a softmax function in the $m$th head, and $hs_{v_i,G_i}^{(l+1)}$ denotes the updated hidden representation of node $v_i$ in graph $G_i$.

*b) Temporal attention:* It measures the influence of historical observations on the current time step. To capture the complex potential temporal correlations, we apply the temporal attention to measure the impacts of historical observations on the current time step. Following the spatial attention paradigm, hidden states are combined with MGSTE through concatenation, while multihead attention calculates temporal relationships. In particular, node $v_i$'s temporal linkage across $t_j$ and $t_k$ is mathematically defined as:

$$o_{t_j,t_k}^{(m)} = \frac{\left\langle f_{t,1}^{(m)}\left(h_{v_i,t_j}^{(l)} \| E_{v_i,t_j}\right), f_{t,2}^{(m)}\left(h_{v_i,t_k}^{(l)} \| E_{v_i,t_k}\right)\right\rangle}{\sqrt{d}} \tag{12}$$

$$ht_{v_i,t_j}^{(l+1)} = \Big\|_{m=1}^{M} \left\{\sum_{k=1}^{T} \beta_{t_j,t_k}^{(m)} \cdot f_{t,3}^{(m)}\left(h_{v_i,t_k}^{(l)}\right)\right\} \tag{13}$$

where $\beta_{t_j,t_k}^{(m)}$ represents the attention weight computed via a softmax function in the $m$th head, capturing the relevance of time step $t_k$ to $t_j$. The functions $f_{t,1}^{(m)}(\cdot)$, $f_{t,2}^{(m)}(\cdot)$, and $f_{t,3}^{(m)}(\cdot)$ are distinct ReLU activation functions applied within the $m$th attention head. Since there are five graphs, we calculate each correlation on each graph parallelly and then use $\lambda^D$, $\lambda^N$, $\lambda^F$, $\lambda^H$, and $\lambda^T$ mentioned in Section III-B as coefficients on the weighted summation to obtain the final correlation between time step $t_j$ and $t_k$.

*c) Graph attention:* It models the interactions between nodes across different graphs. To capture the self-correlations of a node across multiple graphs, we utilize a graph attention mechanism (see Fig. 5). Building upon the temporal-spatial attention framework, we fuse hidden states with MGSTE and employ multihead computations to establish cross-graph dependencies. In particular, the cross-graph interaction governing node $v_i$ between graphs $G_j$ and $G_k$ is formulated as

$$u_{G_j,G_k}^{(m)} = \frac{\left\langle f_{G,1}^{(m)}\left(h_{v_i,G_j}^{(l)} \| E_{v_i,G_j}\right), f_{G,2}^{(m)}\left(h_{v_i,G_k}^{(l)} \| E_{v_i,G_k}\right)\right\rangle}{\sqrt{d}} \tag{14}$$

$$hg_{v_i,G_j}^{(l+1)} = \Big\|_{m=1}^{M} \left\{\sum_{k=1}^{|\mathbf{G}|} \gamma_{G_j,G_k}^{(m)} \cdot f_{G,3}^{(m)}\left(h_{v_i,G_k}^{(l)}\right)\right\} \tag{15}$$

where $\gamma_{G_j,G_k}^{(m)}$ represents the attention score computed via a softmax function in the $m$th attention head, reflecting the relevance of graph $G_k$ to $G_j$. The functions $f_{G,1}^{(m)}(\cdot)$, $f_{G,2}^{(m)}(\cdot)$, and $f_{G,3}^{(m)}(\cdot)$ are ReLU activations applied within the $m$th head.

*d) Gated fusion:* To combine the outputs from temporal, spatial, and graph attention mechanisms, we adopt a gated fusion strategy as proposed in [34]. In the $l$th block, the temporal output $H_T^{(l)}$, spatial output $H_S^{(l)}$, and graph output $H_G^{(l)}$ are integrated using the following approach:

$$\begin{aligned} H^{(l)} &= z \odot H_S^{(l)} + (0.5 - 0.5z) \odot H_T^{(l)} \\ &\quad + (0.5 - 0.5z) \odot H_G^{(l)} \end{aligned} \tag{16}$$

where the gate $z$ is calculated by

$$z = \sigma\left(H_S^{(l)}\mathbf{W}_{z,1} + H_T^{(l)}\mathbf{W}_{z,2} + H_G^{(l)}\mathbf{W}_{z,3} + \mathbf{b}_z\right) \tag{17}$$

where $\mathbf{W}_{z,1}, \mathbf{W}_{z,2}, \mathbf{W}_{z,3} \in \mathbb{R}^{D\times D}$, and $\mathbf{b}_z \in \mathbb{R}^D$ are trainable parameters. The symbol $\odot$ denotes the elementwise Hadamard product, while $\sigma(\cdot)$ represents the sigmoid activation function. By integrating temporal, spatial, and graph attention mechanisms, we design a spatio-temporal multigraph attention block (STMG-ATT), as illustrated in Fig. 2.

## IV. Experiments

### A. Datasets

1) *Parking:* The Melbourne parking dataset, provided by the city council and accessible via https://data.melbourne.vic.gov.au/, includes 42 672 743 parking records collected at 5-min intervals by in-ground sensors in Melbourne's CBD during 2019 [38]. For our study, we used data from June 1 to July 31, 2019. The sensors are grouped into 40 regions, each represented by its geometric center as a node. The temporal resolution is 5 min.
2) *Air Quality:* A large-scale air quality dataset published by China's Ministry of Ecology and Environment (available at https://english.mee.gov.cn/) contains hourly $PM_{2.5}$ measurements from 92 monitoring stations across

TABLE I

PERFORMANCE COMPARISON OF SEVEN ST-GNN MODELS, WITH AND WITHOUT THE MULTIGRAPH MODULE, ACROSS THREE DATASETS. (“*” DENOTES THE MODEL INCORPORATING THE PROPOSED DYNAMIC MULTIGRAPH FUSION MECHANISM)

| Datasets | Metric | Methods | STGCN | STGCN* | ST-MGCN | ST-MGCN* | ASTGCN | ASTGCN* | MSTGCN | MSTGCN* | MegaCRN | MegaCRN* | STPGNN | STPGNN* | Graph WaveNet | Graph WaveNet* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Parking* | RMSE | 3 | 0.0611 | **0.0510** | 0.0552 | **0.0521** | 0.0519 | **0.0476** | 0.0607 | **0.0478** | **0.0489** | 0.0494 | 0.0549 | **0.0491** | 0.0478 | **0.0465** |
| | | 6 | 0.0753 | **0.0650** | 0.0677 | **0.0641** | 0.0643 | **0.0601** | 0.0726 | **0.0612** | 0.0640 | **0.0619** | 0.0672 | **0.0600** | 0.0609 | **0.0581** |
| | | 9 | 0.0871 | **0.0774** | 0.0794 | **0.0741** | 0.0748 | **0.0694** | 0.0834 | **0.0709** | 0.0781 | **0.0717** | 0.0767 | **0.0681** | 0.0709 | **0.0666** |
| | | 12 | 0.0993 | **0.0884** | 0.0900 | **0.0830** | 0.0843 | **0.0770** | 0.0939 | **0.0788** | 0.0923 | **0.0806** | 0.0868 | **0.0747** | 0.0801 | **0.0739** |
| | | 15 | 0.1108 | **0.0981** | 0.0999 | **0.0911** | 0.1115 | **0.0846** | 0.1041 | **0.0868** | 0.1063 | **0.0894** | 0.0948 | **0.0797** | 0.0883 | **0.0803** |
| | | 18 | 0.1210 | **0.1071** | 0.1092 | **0.0985** | 0.1228 | **0.0912** | 0.1140 | **0.0935** | 0.1198 | **0.0980** | 0.1021 | **0.0855** | 0.0965 | **0.0857** |
| | | 21 | 0.1301 | **0.1153** | 0.1178 | **0.1052** | 0.1308 | **0.0976** | 0.1229 | **0.0995** | 0.1330 | **0.1062** | 0.1098 | **0.0916** | 0.1040 | **0.0905** |
| | | 24 | 0.1393 | **0.1229** | 0.1259 | **0.1112** | 0.1410 | **0.1035** | 0.1318 | **0.1053** | 0.1458 | **0.1139** | 0.1172 | **0.0953** | 0.1114 | **0.0950** |
| | MAE | 3 | 0.0429 | **0.0357** | 0.0375 | **0.0357** | 0.0363 | **0.0324** | 0.0445 | **0.0327** | **0.0334** | 0.0337 | 0.0396 | **0.0342** | 0.0324 | **0.0318** |
| | | 6 | 0.0532 | **0.0452** | 0.0467 | **0.0446** | 0.0452 | **0.0420** | 0.0533 | **0.0429** | 0.0445 | **0.0430** | 0.0494 | **0.0420** | 0.0423 | **0.0409** |
| | | 9 | 0.0617 | **0.0547** | 0.0556 | **0.0522** | 0.0531 | **0.0493** | 0.0614 | **0.0505** | 0.0551 | **0.0511** | 0.0547 | **0.0483** | 0.0499 | **0.0475** |
| | | 12 | 0.0712 | **0.0628** | 0.0638 | **0.0590** | 0.0601 | **0.0559** | 0.0692 | **0.0568** | 0.0656 | **0.0586** | 0.0613 | **0.0543** | 0.0571 | **0.0532** |
| | | 15 | 0.0799 | **0.0701** | 0.0715 | **0.0652** | 0.0675 | **0.0618** | 0.0769 | **0.0630** | 0.0759 | **0.0658** | 0.0688 | **0.0591** | 0.0635 | **0.0581** |
| | | 18 | 0.0882 | **0.0771** | 0.0788 | **0.0710** | 0.0742 | **0.0671** | 0.0846 | **0.0681** | 0.0859 | **0.0726** | 0.0735 | **0.0640** | 0.0698 | **0.0623** |
| | | 21 | 0.0956 | **0.0836** | 0.0855 | **0.0761** | 0.0810 | **0.0720** | 0.0918 | **0.0728** | 0.0959 | **0.0790** | 0.0791 | **0.0680** | 0.0755 | **0.0659** |
| | | 24 | 0.1036 | **0.0899** | 0.0919 | **0.0809** | 0.0867 | **0.0767** | 0.0995 | **0.0774** | 0.1057 | **0.0851** | 0.0892 | **0.0718** | 0.0812 | **0.0693** |
| *Air Quality* | RMSE | 3 | 6.6847 | **6.6251** | 6.7811 | **6.7417** | **6.3954** | 6.9839 | 7.0431 | **6.3711** | 6.0567 | **5.9162** | 6.9777 | **6.7885** | 6.4879 | **6.2105** |
| | | 6 | 8.3992 | **8.1668** | 8.7089 | **8.4812** | **7.7518** | 8.1622 | 8.4451 | **7.6542** | 7.6486 | **7.1699** | 7.8064 | **7.4038** | 8.2324 | **7.8326** |
| | | 9 | 9.7763 | **8.8673** | 10.3525 | **9.7767** | **9.0521** | 9.4711 | 9.3189 | **8.7710** | 9.6258 | **8.1311** | 9.2542 | **7.9729** | 10.3233 | **9.4309** |
| | | 12 | 10.8080 | **9.7566** | 11.5616 | **10.5443** | 10.7794 | **10.2950** | 10.7145 | **9.6745** | 11.6202 | **9.0879** | 11.0781 | **8.6102** | 12.9487 | **11.1324** |
| | | 15 | 11.7172 | **10.3794** | 12.3341 | **10.8383** | 11.9669 | **10.9197** | 11.4235 | **10.7136** | 13.2844 | **9.8540** | 12.3174 | **9.1599** | 15.7093 | **12.6535** |
| | | 18 | 11.9014 | **10.9515** | 12.7944 | **10.8020** | 13.2015 | **11.8594** | 12.3950 | **11.1139** | 14.7477 | **10.4557** | 11.7271 | **9.4606** | 19.2235 | **12.9247** |
| | | 21 | 12.5268 | **11.4529** | 13.1333 | **10.5942** | 14.4416 | **11.6752** | 13.1675 | **10.7628** | 15.9183 | **10.9855** | 11.8508 | **10.1532** | 21.1240 | **13.3891** |
| | | 24 | 12.9587 | **12.3110** | 13.4853 | **10.5584** | 14.6537 | **10.7631** | 13.3226 | **11.3826** | 16.5032 | **11.5430** | 12.5560 | **10.8380** | 21.2758 | **13.4559** |
| | MAE | 3 | **4.8976** | 4.9423 | 5.2829 | **5.1224** | **4.7641** | 5.4472 | 5.3351 | **4.7268** | 4.3330 | **4.1492** | **4.8923** | 5.2385 | 4.9827 | **4.3737** |
| | | 6 | 6.9109 | **6.4199** | 7.1791 | **6.7221** | **6.0110** | 6.5113 | 6.7292 | **5.9743** | 5.8606 | **5.2163** | 5.6401 | **5.3272** | 6.6243 | **5.9126** |
| | | 9 | 7.7050 | **7.1422** | 8.8039 | **7.9352** | **7.2883** | 7.8158 | 7.6317 | **7.0492** | 7.8082 | **6.1374** | 7.3854 | **5.9768** | 8.6868 | **7.4941** |
| | | 12 | 8.8756 | **8.1219** | 9.9873 | **8.6429** | 9.1761 | **8.6062** | 9.1154 | **7.9511** | 9.6118 | **7.0437** | 9.0175 | **6.4796** | 11.1945 | **9.1035** |
| | | 15 | 9.3082 | **8.7745** | 10.7280 | **8.9142** | 10.3348 | **9.1871** | 9.8296 | **9.0198** | 11.0007 | **7.6683** | 10.1933 | **6.7700** | 13.6732 | **10.5456** |
| | | 18 | 10.2937 | **9.3613** | 11.1683 | **8.8919** | 11.5575 | **9.0799** | 10.7857 | **9.3872** | 12.2164 | **8.0551** | 9.6992 | **7.2732** | 16.5516 | **10.7753** |
| | | 21 | 10.7669 | **9.8619** | 11.4962 | **8.7456** | 12.7541 | **8.8653** | 11.5581 | **8.9661** | 13.2064 | **8.3511** | 9.7229 | **8.2450** | 18.0054 | **11.2592** |
| | | 24 | 10.9152 | **10.6965** | 11.8230 | **8.7487** | 12.9317 | **8.9079** | 11.6787 | **9.6143** | 13.7204 | **8.6844** | 10.4175 | **9.0186** | 18.1966 | **11.3203** |
| *Ride Hailing* | RMSE | 3 | 29.5662 | **29.3236** | **33.0120** | 32.0153 | 33.1289 | **32.3012** | 32.6449 | **30.5524** | 27.6364 | **26.3274** | 26.6563 | **26.0476** | 27.2799 | **27.0863** |
| | | 6 | 32.1893 | **32.1633** | **33.0120** | 32.0153 | 33.1289 | **32.3012** | 32.6449 | **30.5524** | 30.5202 | **28.7158** | 29.0564 | **27.7674** | **28.8068** | 29.1252 |
| | | 9 | 34.2533 | **33.5350** | 35.8524 | **34.2897** | 35.6106 | **34.2772** | 35.8217 | **32.7692** | 33.0281 | **30.6759** | 30.5179 | **29.6031** | **30.1752** | 30.9078 |
| | | 12 | 37.5943 | **35.8843** | 40.2453 | **38.0918** | 39.5084 | **38.3022** | 39.2747 | **34.2789** | 37.2533 | **34.3742** | 32.3281 | **31.3148** | 32.4654 | **31.8965** |
| | | 15 | 41.6075 | **39.8592** | 44.5880 | **41.6047** | 44.2874 | **41.9079** | 44.1031 | **37.4167** | 41.9931 | **38.7010** | 35.9525 | **33.6783** | 34.7654 | **34.1141** |
| | | 18 | 44.4370 | **41.7588** | 47.3992 | **43.9012** | 47.1648 | **45.6327** | 47.8783 | **40.1539** | 45.3727 | **41.6331** | 37.7235 | **34.3853** | 36.4093 | **35.3749** |
| | | 21 | 46.1910 | **43.2324** | 49.6615 | **45.7239** | 48.8270 | **46.7077** | 49.3727 | **41.3584** | 47.9148 | **43.4837** | 39.7585 | **35.0915** | 37.7469 | **35.9011** |
| | | 24 | 47.2149 | **44.5805** | 52.2225 | **48.1538** | 51.6441 | **49.0229** | 51.7495 | **43.0147** | 50.4866 | **45.6240** | 41.8183 | **35.3215** | 38.9829 | **36.6612** |
| | MAE | 3 | 16.4803 | **15.2362** | 16.8174 | **16.4679** | 15.8601 | **15.7588** | 18.0352 | **15.0494** | 14.6972 | **14.2256** | **13.7401** | 13.7651 | 13.6751 | **13.5505** |
| | | 6 | 17.2791 | **17.1172** | 18.2547 | **17.7313** | **17.9090** | 17.9096 | 18.1003 | **16.3884** | 16.3873 | **15.9569** | 15.4038 | **14.7751** | 15.0251 | **15.0209** |
| | | 9 | 18.9824 | **18.8224** | 20.1815 | **19.3857** | 19.4710 | **19.4389** | 20.2173 | **17.9841** | 18.3034 | **17.8651** | 16.6977 | **15.6326** | **15.9874** | 16.2062 |
| | | 12 | 21.1205 | **20.8666** | 22.9625 | **21.7645** | 22.3462 | **22.2024** | 22.4222 | **19.0954** | 20.8166 | **20.3529** | 17.9838 | **16.7234** | 17.2701 | **17.0103** |
| | | 15 | 23.8585 | **23.6034** | 25.6956 | **23.9382** | 25.7334 | **24.7974** | 26.1905 | **21.4476** | 23.7264 | **23.0537** | 20.2448 | **17.9678** | 18.7764 | **18.4398** |
| | | 18 | 26.1226 | **25.2285** | 27.5798 | **25.5239** | 27.8473 | **27.4657** | 28.4511 | **23.6879** | 25.7781 | **24.9648** | 21.3881 | **18.3419** | 19.5001 | **19.1254** |
| | | 21 | 27.8849 | **26.7217** | 29.4320 | **27.1973** | 29.6007 | **29.3231** | 30.5508 | **24.8186** | 27.6307 | **25.5898** | 22.8561 | **18.9501** | 20.2884 | **19.6949** |
| | | 24 | 28.7715 | **27.7837** | 31.2099 | **28.8214** | 31.8545 | **31.2421** | 32.6702 | **26.3013** | 29.4129 | **27.3251** | 24.2627 | **19.1533** | 21.1472 | **20.1057** |
| Count | | | 1 | 47 | 2 | 46 | 7 | 41 | 0 | 48 | 2 | 46 | 2 | 46 | 3 | 45 |

Jiangsu province for the year 2020 [6]. The data is sampled at a 3-h interval.

3) *Ride Hailing:* This dataset is collected from [20], ranging from May 1, 2017 to December 31, 2017.

## *B. Experimental Details*

*1) Baselines:* We used seven spatio-temporal GNNs as baselines: STGCN [2], ASTGCN [33], MSTGCN [33], ST-MGCN [20], Graph WaveNet [24], MegaCRN [39], and STPGNN [40].

1) *STGCN:* The spatio-temporal GCN combines 1-D CNN with gated linear units (GLUs) and graph convolution to capture spatio-temporal correlations [2].
2) *ASTGCN:* The ASTGCN adds spatio-temporal attention to the convolutional backbone to capture dynamic spatio-temporal correlations [33].
3) *MSTGCN:* The multicomponent spatio-temporal GCN follows the ASTGCN settings without the spatio-temporal attention mechanism [33].
4) *ST-MGCN:* The ST-MGCN leverages multigraph convolution to model non-Euclidean relations across regions and employs a CGRNN for temporal dynamics [20].
5) *Graph WaveNet:* utilizes a self-adaptive adjacency matrix to preserve hidden spatial dependencies and assembles graph convolution with dilated *causal* convolution to capture spatio-temporal dependencies [24].
6) *MegaCRN:* Implements spatio-temporal *meta-graph learning* by plugging a meta-graph learner into a GCRN encoder–decoder, enabling adaptive graph structure learning under heterogeneity and nonstationarity [39].
7) *STPGNN:* The spatio-temporal *Pivotal* graph neural network identifies and exploits pivotal spatio-temporal patterns via pivotal-aware aggregation to enhance long-range dependencies while reducing redundancy [40].

*2) Implementation Details:* We standardize inputs with *StandardScaler*. After the conv/linear projections that produce $Q, K, V$, we apply *BatchNorm2d* to stabilize intermediate feature scales. In the dynamic-graph module, learned adjacency scores are passed through *ReLU* and then *softmax*-normalized along the neighbor axis. For multigraph fusion, we 1) *softmax*-normalize scalar per-graph weights across graphs; or 2) apply *ReLU* followed by *softmax* along the graph axis for elementwise fusion. These normalizations keep magnitudes comparable across temporal, spatial, and graph attention while preserving a unified scaled dot-product kernel.

*3) Platform:* All experiments were conducted on a Linux machine equipped with an Intel[1] Core[2] Ultra 7 265K processor running at a base frequency of 3.90 GHz, and an NVIDIA GeForce RTX 5090 GPU.

[1]Registered trademark.

[2]Trademarked.

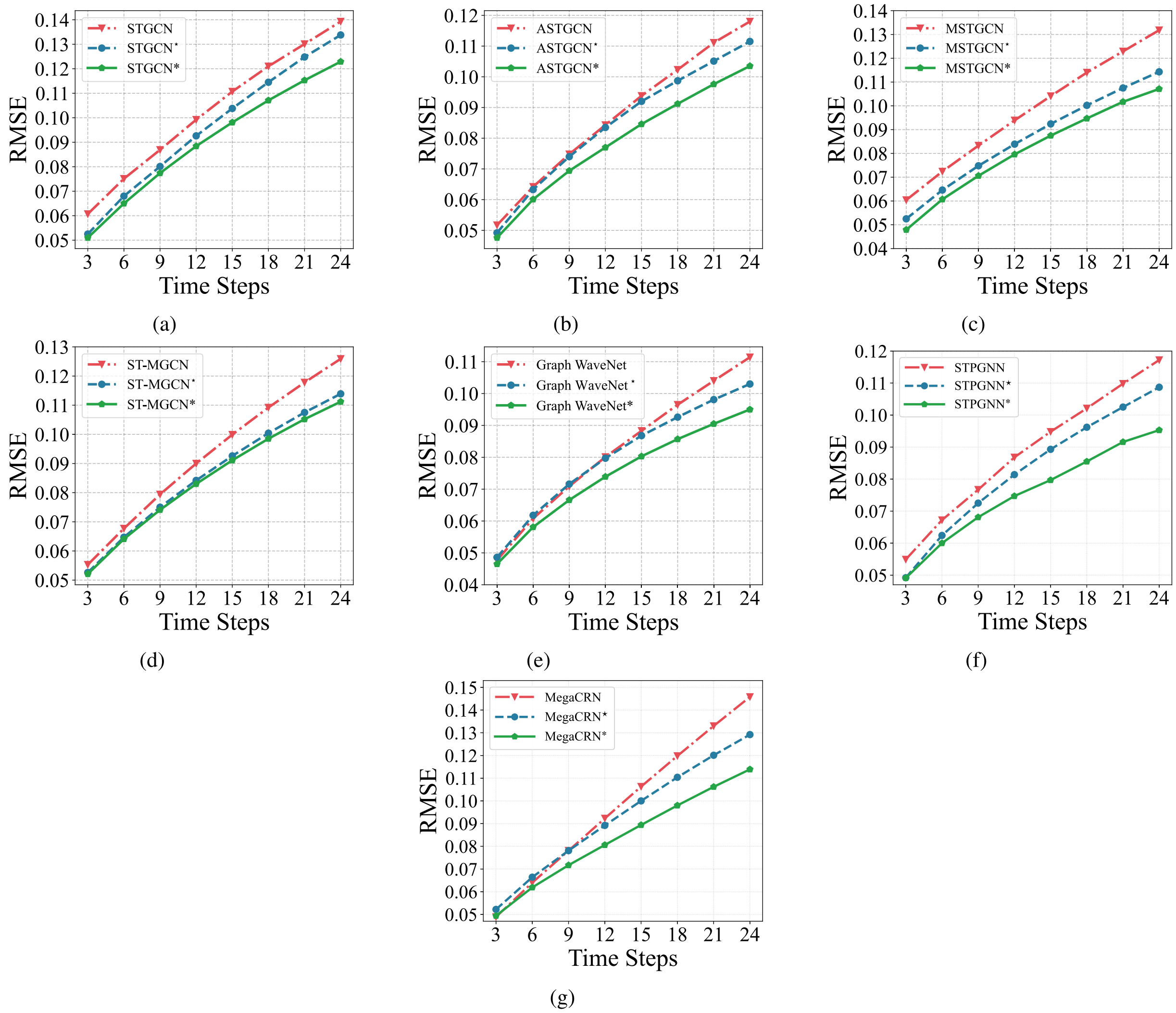



Fig. 6. Temporal evolution of RMSE performance on parking dataset: baseline ST-GNNs (red) versus static graph fusion (blue⋆) versus dynamic multigraph fusion (green*). (a) STGCN. (b) ASTGCN. (c) MSTGCN. (d) ST-MGCN. (e) Graph WaveNet. (f) STPGNN. (g) MegaCRN.

*4) Other Settings:* Each experiment used a 24-step historical window, with prediction horizons varying between 3 and 24 steps. The dataset was split into 70% for training, 10% for validation, and the remaining 20% for testing. The models were trained using the Adam optimizer with a fixed learning rate of $1e^{-4}$. An L1 loss function was applied to evaluate performance. We used a batch size of 32 and set the random seed to 0 to ensure reproducibility. For the attention mechanism, the number of heads $M$ and dimension per head $d$ were configured as follows: $(M = 8, d = 8)$ for the parking dataset, $(M = 24, d = 6)$ for the air quality dataset, and $(M = 8, d = 6)$ for the ride-hailing dataset.

*5) Evaluation Metrics:* In our study, mean absolute error (MAE) and root-mean-square error (RMSE) were used for performance comparison.

### C. Results and Analysis

Table I shows the performance of each ST-GNN model on the three datasets, with prediction horizons ranging from 3 to 24 time steps. The best results are highlighted in bold, and the number of bold entries is counted to enable a quantitative comparison of model performance.

The primary experimental phase focused on assessing the comprehensive efficacy of graph constructions and fusion methodologies. This involved benchmarking baseline ST-GNN performance (without multigraph integration) against our proposed multigraph framework across identical evaluation protocols.

Table I reveals three key observations.

1) The proposed dynamic multigraph fusion method (denoted by "*") demonstrated notable performance enhancements. For instance, when integrated with Graph WaveNet, our approach achieved a 9.6% reduction in average RMSE across all prediction horizons.
This confirms the method's capability to capture richer information and its effectiveness across different ST-GNN architectures.
2) Under identical ST-GNN configurations, our methodology consistently surpassed their original counterparts

TABLE II

PREDICTED RMSE FOR EACH MODEL IN THE PARKING DATASET. "†" AND "‡" REPRESENT THE ST-GNN MODELS THAT USE MULTIGRAPH FUSION WITH THE FUNCTIONALITY GRAPH PROPOSED BY [20] OR THE PROPOSED FUNCTIONALITY GRAPH, RESPECTIVELY

| Model | | | **STGCN** | **ASTGCN** | **MSTGCN** | **ST-MGCN** | **STPGNN** | **MegaCRN** | **Graph WaveNet** |
|---|---|---|---|---|---|---|---|---|---|
| MAE | 12 | † | **0.0622** | 0.0579 | 0.0581 | 0.0604 | 0.0570 | 0.0612 | 0.0544 |
| | | ‡ | 0.0628 | **0.0559** | **0.0568** | **0.0590** | **0.0543** | **0.0586** | **0.0532** |
| | 24 | † | 0.0939 | 0.0847 | 0.0810 | 0.0831 | 0.0753 | 0.0911 | 0.0711 |
| | | ‡ | **0.0899** | **0.0767** | **0.0774** | **0.0809** | **0.0718** | **0.0851** | **0.0693** |
| RMSE | 12 | † | **0.0876** | 0.0910 | 0.0884 | 0.0852 | 0.0796 | 0.0864 | 0.0755 |
| | | ‡ | 0.0884 | **0.0770** | **0.0788** | **0.0830** | **0.0747** | **0.0806** | **0.0739** |
| | 24 | † | 0.1283 | 0.1114 | 0.1099 | 0.1269 | 0.1012 | 0.1334 | 0.0969 |
| | | ‡ | **0.1229** | **0.1035** | **0.1053** | **0.1112** | **0.0953** | **0.1139** | **0.0950** |

in win rates, highlighting its robust generalization capacity.

3) The experimental outcomes suggest superior adaptability of our framework for LSTF tasks. Notably, the performance disparity between conventional ST-GNN models and our proposed solution continued to expand with longer prediction horizons. This pattern is visually corroborated in Fig. 6, where the proposed model (green trajectory) maintains stable performance with extended horizons, contrasting with the deteriorating accuracy of baseline ST-GNN models (red trajectory) in long-term predictions.

### D. Ablation Study

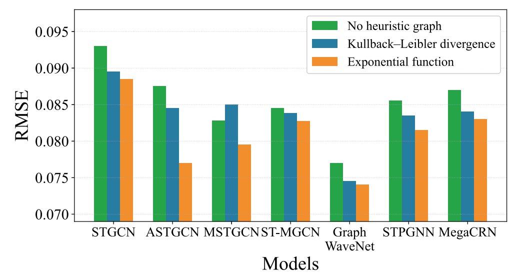
To verify the individual contributions of framework components, systematic ablation experiments were performed on the parking dataset. These experiments examine the effectiveness of the proposed functionality graph, heuristic graph, and the STMG-ATT mechanism.

*1) Performance of Functionality Graphs:* Table II highlights the impact of using the proposed functionality graph $G^F$ compared to the functionality graph from [20]. We observe the following key trends.

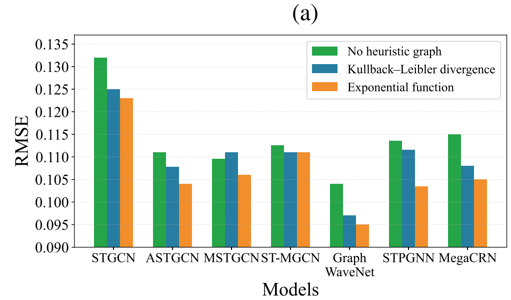
1) Models incorporating the proposed functionality graph (marked with "‡") consistently outperform those using the prior functionality graph, demonstrating its ability to better capture global contextual function similarities.
2) The performance drop is significantly smaller when prediction horizons increase from 12 to 24 time steps, indicating that the proposed functionality graph provides more stable relationships overextended temporal periods. This robustness highlights its suitability for LSTF tasks.

*2) Performance of Heuristic Graph:* Fig. 7 evaluates the heuristic graph $G^H$ generated using different methods, including the exponential approximation function and KL divergence. Key observations include the following.

1) Graphs constructed using the exponential approximation function demonstrate superior performance across 12-times and 24-times step prediction intervals, proving the method's capability to effectively model long-term spatio-temporal correlations.
2) While slightly less effective than exponential approximation, heuristic graphs generated using the KL divergence still show improved performance compared to models without heuristic graphs. This suggests that KL divergence provides complementary information that enhances spatio-temporal modeling.
3) Models without heuristic graphs exhibit significantly lower performance, underscoring the importance of incorporating human insights and heuristic knowledge in graph construction.




(a)

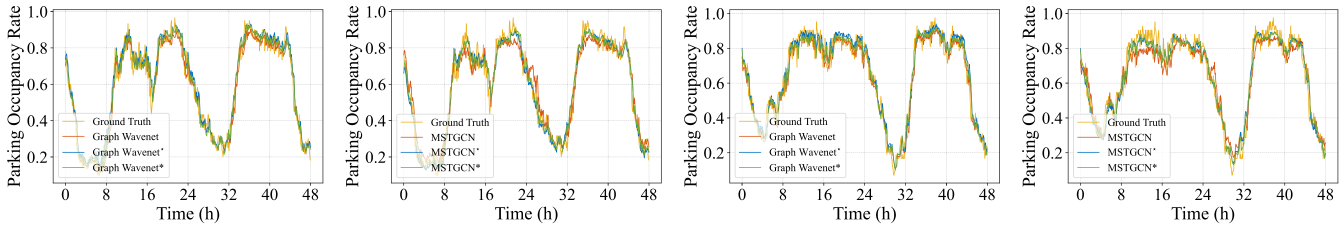



(b)

Fig. 7. Model performance on the parking dataset. Each model is evaluated with or without heuristic graphs, which are generated using the KL divergence or the exponential approximation function. (a) Prediction horizon 12. (b) Prediction horizon 24.

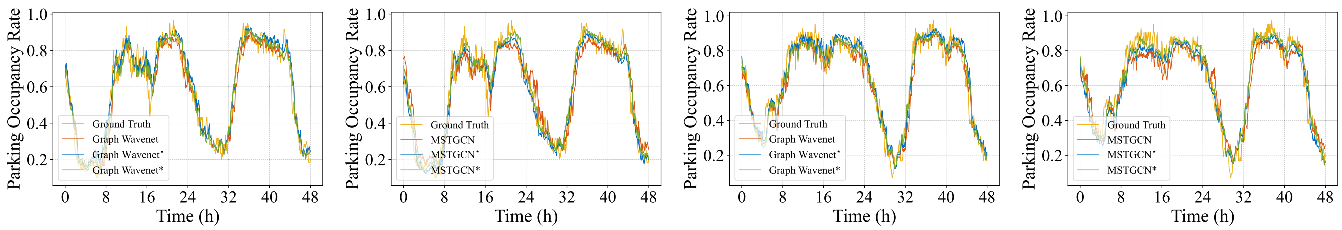
*3) Performance of STMG-ATT:* Fig. 6 displays comparative experimental results between the framework with STMG-ATT mechanism (denoted by $^*$) and without it (denoted by $^\star$). The analytical findings demonstrate the following critical observations.

1) The inclusion of STMG-ATT yields notable performance improvements, particularly in long-term forecasting tasks, due to its ability to dynamically model complex spatio-temporal dependencies across multiple graphs.



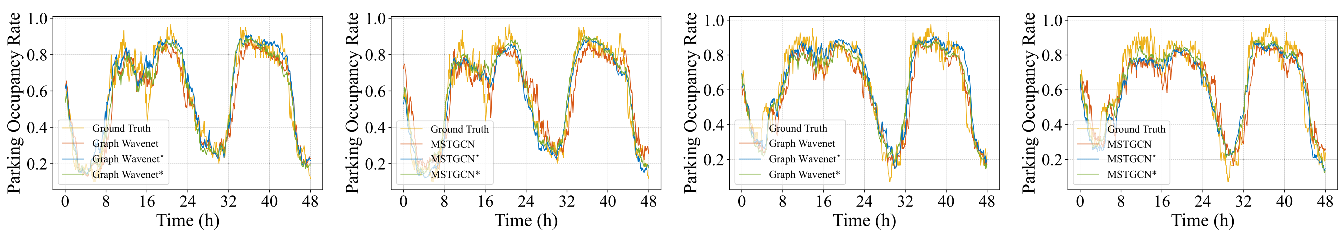

(a) (b) (c) (d)

Fig. 8. Prediction result curves on the parking dataset with 15 min prediction horizon (three time steps). (a) and (b) Hardware. (c) and (d) McKillop.



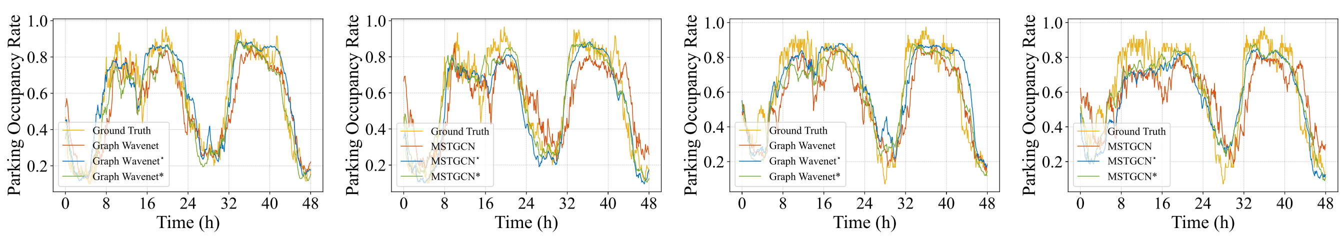

(a) (b) (c) (d)

Fig. 9. Prediction result curves on the parking dataset with 30 min prediction horizon (six time steps). (a) and (b) Hardware. (c) and (d) McKillop.




(a) (b) (c) (d)

Fig. 10. Prediction result curves on the parking dataset with 60 min prediction horizon (12 time steps). (a) and (b) Hardware. (c) and (d) McKillop.




(a) (b) (c) (d)

Fig. 11. Prediction result curves on the parking dataset with 120 min prediction horizon (24 time steps). (a) and (b) Hardware. (c) and (d) McKillop.

2) Figs. 8 to 11 show the prediction curves for two selected areas (Hardware and Mckillop) using Graph WaveNet and MSTGCN models. As the prediction horizon increases, models with STMG-ATT (green lines) produce predictions closer to the ground truth compared to models without STMG-ATT, confirming the effectiveness of this mechanism in reducing error propagation over time.
3) The STMG-ATT mechanism enhances the predictive performance of multiple ST-GNN models, demonstrating its versatility and compatibility with different neural architectures.

These results validate the design choices of the proposed framework and highlight the complementary roles of its individual components in achieving promising performance for LSTF tasks.

*4) Leave-One-Out on STPGNN:* We conduct a leave-one-out ablation on the STPGNN backbone to assess component necessity. The full five-graph setting (*Full-5*) is compared with variants that remove one graph at a time ($-D/-N/-T/-H/-F$). Table III summarizes RMSE and MAE over horizons 3–24. Overall, *Full-5* consistently achieves the best performance, and dropping any single graph leads to a clear degradation that becomes more pronounced at longer horizons. Minor short-horizon fluctuations may appear, but do not alter this conclusion.

TABLE III

PREDICTED RMSE AND MAE FOR THE STPGNN LEAVE-ONE-OUT ABLATION ON THE FIVE GRAPHS AND THE ATTENTION BLOCK ON THE PARKING DATASET. HERE, *−D*/*−N*/*−T*/*−H*/*−F* DENOTE REMOVING THE DISTANCE/NEIGHBOR/TEMPORAL-SIMILARITY/HEURISTIC/FUNCTIONALITY GRAPHS

| | Step | Full-5 | −D | −N | −T | −H | −F |
|---|---|---|---|---|---|---|---|
| RMSE | 3 | 0.0491 | **0.0472** | 0.0474 | 0.0476 | 0.0484 | 0.0474 |
| | 6 | **0.0600** | 0.0603 | 0.0608 | 0.0609 | 0.0602 | 0.0604 |
| | 9 | **0.0681** | 0.0685 | 0.0688 | 0.0689 | 0.0685 | 0.0683 |
| | 12 | **0.0747** | 0.0761 | 0.0768 | 0.0752 | 0.0755 | 0.0762 |
| | 15 | **0.0797** | 0.0826 | 0.0835 | 0.0816 | 0.0823 | 0.0829 |
| | 18 | **0.0855** | 0.0884 | 0.0900 | 0.0881 | 0.0881 | 0.0892 |
| | 21 | **0.0916** | 0.0938 | 0.0964 | 0.0933 | 0.0932 | 0.0949 |
| | 24 | **0.0953** | 0.0985 | 0.1021 | 0.0978 | 0.0972 | 0.1002 |
| MAE | 3 | 0.0342 | 0.0355 | 0.0351 | 0.0361 | **0.0339** | 0.0343 |
| | 6 | 0.0420 | 0.0421 | 0.0424 | **0.0412** | 0.0423 | 0.0421 |
| | 9 | **0.0483** | 0.0491 | 0.0494 | 0.0496 | 0.0485 | 0.0486 |
| | 12 | **0.0543** | 0.0553 | 0.0559 | 0.0544 | 0.0548 | 0.0551 |
| | 15 | **0.0591** | 0.0608 | 0.0613 | 0.0597 | 0.0604 | 0.0607 |
| | 18 | **0.0640** | 0.0668 | 0.0665 | 0.0652 | 0.0665 | 0.0659 |
| | 21 | **0.0680** | 0.0711 | 0.0714 | 0.0693 | 0.0696 | 0.0704 |
| | 24 | **0.0718** | 0.0739 | 0.0776 | 0.0746 | 0.0742 | 0.0742 |

TABLE IV

COMPLEXITY AND EFFICIENCY ON PARKING (UNIFIED PROTOCOL: SAME SPLITS, NORMALIZATION, AND SCHEDULE). TRAIN/EPOCH = TOTAL/40; TEST TIME = FULL TEST SET; LATENCY IS PER-SAMPLE AT BATCH SIZES 1 AND 32

| Metric | Full | No-Attn | Backbone |
|---|---|---|---|
| Params (M) | 0.906 | 0.906 | 0.790 |
| Train/epoch (s) | 4.73 | 3.47 | 3.20 |
| Test (s) | 9.23 | 7.46 | 7.31 |
| Peak GPU (GB) | 0.414 | 0.296 | 0.244 |
| CPU RSS (GB) | 2.478 | 2.457 | 2.233 |
| Latency (ms, bs=1) | 8.872 | 8.527 | 8.181 |
| Latency (ms, bs=32) | 0.281 | 0.264 | 0.251 |

Attention submodules are registered at initialisation; the attention switch bypasses their computation in the forward pass, hence identical parameter counts for *Full* and *No-Attn*. Timing is measured in `eval` mode with GPU synchronisation on [GPU model / CUDA / framework]; latency values are per-sample medians after warm-up.

### *E. Complexity and Efficiency*

Under a unified protocol (same data splits, normalization, and schedule), we profile the computational cost on the STPGNN backbone. Table IV reports parameter counts, wall-clock training time, test-time, peak GPU memory, CPU RSS, and per-sample latency.

*1) Observation:* Disabling attention reduces total training/test time and peak memory while keeping parameter counts identical (attention layers are registered but bypassed in the forward pass). The **Backbone** (no FusionGraph/Attn) further slightly lowers parameters and latency; accuracy impacts are shown in the ablations.

## V. CONCLUSION

This article focuses on addressing LSTF through multi-GNNs with hybrid attention mechanisms. An innovative graph-based framework is designed to acquire context-aware features and knowledge discovery in spatio-temporal systems, with specialized heuristic graph construction for modeling long-range temporal dependencies. A dynamic multigraph fusion component is created to achieve cross-graph node alignment and temporal synchronization, with outputs fed into diverse GNNs. An innovative cross-graph attention mechanism is proposed to analyze intrinsic node relationships in multigraph environments, integrating spatio-temporal and graph attention to identify latent node couplings. Empirical studies on three real-world datasets demonstrate significant improvements in forecasting performance for LSTF tasks.

## REFERENCES


[1] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "LSGCN: Long short-term traffic prediction with graph convolutional networks," in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2355–2361.

[2] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[4] W. Shao, Y. Zhang, B. Guo, A. K. Qin, J. Chan, and F. D. Salim, "Parking availability prediction with long short term memory model," in *Green, Pervasive, and Cloud Computing*. Cham, Switzerland: Springer, 2019, pp. 124–137.

[5] W. Zhang, H. Liu, Y. Liu, J. Zhou, T. Xu, and H. Xiong, "Semi-supervised city-wide parking availability prediction via hierarchical recurrent graph neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3984–3996, Aug. 2022.

[6] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, "Pm2.5-GNN: A domain knowledge enhanced graph neural network for pm2.5 forecasting," in *Proc. 28th Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2020, vol. 2020, no. 1, pp. 163–166.

[7] C. Liu et al., "Emulation of an atmospheric gas-phase chemistry solver through deep learning: Case study of Chinese mainland," *Atmos. Pollut. Res.*, vol. 12, no. 6, Jun. 2021, Art. no. 101079. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1309104221001458

[8] Y. Huang, J. Yu, X. Dai, Z. Huang, and Y. Li, "Air-quality prediction based on the EMD–IPSO–LSTM combination model," *Sustainability*, vol. 14, no. 9, p. 4889, Apr. 2022, doi: 10.3390/su14094889.

[9] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[10] X. Xiao et al., "A dual-path dynamic directed graph convolutional network for air quality prediction," *Sci. Total Environ.*, vol. 827, Jun. 2022, Art. no. 154298.

[11] H. Lin, R. Bai, W. Jia, X. Yang, and Y. You, "Preserving dynamic attention for long-term spatial–temporal prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 36–46.

[12] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114–119, Jun. 2019.

[13] A. Ceschini, A. Rosato, and M. Panella, "Hybrid quantum-classical recurrent neural networks for time series prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.

[14] J. Siłka, M. Wieczorek, and M. Woźniak, "Recurrent neural network model for high-speed train vibration prediction from time series," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13305–13318, Aug. 2022.

[15] F. Li, J. Feng, H. Yan, D. Jin, and Y. Li, "Crowd flow prediction for irregular regions with semantic graph attention network," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 13, no. 5, pp. 1–4, 2022, doi: 10.1145/3501805.

[16] G. Jin, F. Li, J. Zhang, M. Wang, and J. Huang, "Automated dilated spatio-temporal synchronous graph modeling for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8820–8830, Aug. 2023.

[17] W. Jiang, Y. Xiao, Y. Liu, Q. Liu, and Z. Li, "Bi-GRCN: A spatio-temporal traffic flow prediction model based on graph neural network," *J. Adv. Transp.*, vol. 2022, pp. 1–12, Feb. 2022.

[18] X. Ta, Z. Liu, X. Hu, L. Yu, L. Sun, and B. Du, "Adaptive spatio-temporal graph neural network for traffic forecasting," *Knowl.-Based Syst.*, vol. 242, Apr. 2022, Art. no. 108199.
[19] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 4946–4956, Dec. 2020.
[20] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 3656–3663.
[21] X. Zhang, R. Cao, Z. Zhang, and Y. Xia, "Crowd flow forecasting with multi-graph neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.
[22] X. Wang, L. Chen, H. Zhang, P. Wang, Z. Zhou, and Y. Wang, "A multi-graph fusion based spatiotemporal dynamic learning framework," in *Proc. 28th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 2365–2375, doi: 10.1145/3539597.3570396.
[23] F. Li, H. Yan, G. Jin, Y. Liu, Y. Li, and D. Jin, "Automated spatio-temporal synchronous modeling with multiple graphs for traffic prediction," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 1084–1093.
[24] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.
[25] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.
[26] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 3844–3852.
[27] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 397–400.
[28] M. Li, L. Zhang, L. Cui, L. Bai, Z. Li, and X. Wu, "BLoG: Bootstrapped graph representation learning with local and global regularization for recommendation," *Pattern Recognit.*, vol. 144, Dec. 2023, Art. no. 109874.
[29] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec," in *Proc. 11th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017, pp. 459–467.
[30] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2021, pp. 1548–1554.
[31] H. Wang et al., "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 968–977.
[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lip, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[33] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 922–929.
[34] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.
[35] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[36] S. Wang, H. Miao, H. Chen, and Z. Huang, "Multi-task adversarial spatial–temporal networks for crowd flow prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manag.*, 2020, pp. 1555–1564.
[37] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
[38] W. Shao, F. D. Salim, T. Gu, N.-T. Dinh, and J. Chan, "Traveling officer problem: Managing car parking violations efficiently using sensor data," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 802–810, Apr. 2018.
[39] R. Jiang et al., "Spatio-temporal meta-graph learning for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, pp. 8078–8086.
[40] W. Kong, Z. Guo, and Y. Liu, "Spatio-temporal pivotal graph neural networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 8, pp. 8627–8635.

**Xiao Xiao** (Member, IEEE) received the bachelor's degree in control technology and instrumentation from Xidian University, Xi'an, China, in 2004, and the Ph.D. degree in measuring and testing technologies and instruments from Zhejiang University, Hangzhou, China, in 2009.

He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. He joined the Min H. Kao Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA, in 2019, as a Research Fellow. His research interests include wireless network architecture, intelligent transportation systems, computer vision, and deep learning.

**Xufeng Xiang** received the B.E. degree in communication engineering from the School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan, China, in 2024. He is currently pursuing the master's degree with the School of Telecommunications Engineering, Xidian University, Xi'an, China.

His research interests include deep reinforcement learning, intelligent transportation systems, graph neural networks, and autonomous driving.

**Xinyue Yang** received the B.E. degree in communication engineering from the School of Computer and Information Engineering, Hubei University, Wuhan, China, in 2022, and the M.E. degree from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2025.

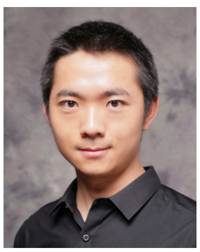
Her research interests include deep learning, intelligent transportation systems, computer vision, and graph neural networks.

**Zhiling Jin** received the B.E. degree from Xidian University, Xi'an, China, in 2021, and the M.E. degree from the School of Telecommunications Engineering, Xidian University, in 2024.

His research interests include deep learning, graph neural networks, intelligent transportation systems, big data, and spatio-temporal data mining.

**Jing Xu** received the B.E. degree in computer science and engineering from Northwest Normal University, Lanzhou, China, in 2019. She is currently pursuing the M.S. degree with the School of System Science, Beijing Normal University, Beijing, China.

Her research interests include spatio-temporal modeling with graph neural networks and time-series predictions.



**Shuo Wang** received the M.E. degree in intelligent systems and pattern recognition from Northeastern University, Shenyang, China, in 2018. He is currently pursuing the Ph.D. degree with Beijing Normal University, Beijing, China.

He is a Visiting Ph.D. Student at ETH Zürich, Zürich, Switzerland. His research interests include spatio-temporal prediction, computer vision, and AI for science, particularly focusing on Earth science.

**Guoqiang Mao** (Fellow, IEEE) has authored or co-authored more than 200 papers in international conferences and journals, which have been cited more than 9000 times. His research interests include intelligent transport systems, applied graph theory and its applications in telecommunications, the Internet of Things, wireless sensor networks, wireless localization techniques, network modeling, and performance analysis.

Dr. Mao received the Top Editor Award for Outstanding Contributions to IEEE Transactions on Vehicular Technology in 2011, 2014, and 2015, respectively. He was a Co-Chair of the IEEE Intelligent Transport Systems Society Technical Committee on Communication Networks. He has been an Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, since 2018, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, since 2010. He was an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, from 2014 to 2019.

**Wei Shao** (Member, IEEE) received the M.S. degree (Hons.) in computer science (CS) from The University of Hong Kong, Hong Kong, in 2011, and the Ph.D. degree from RMIT University, Melbourne, VIC, Australia, in 2018.

He was a Research Assistant at the Computer Science Department, The University of Hong Kong, from 2012 to 2014. He was a Post-Doctoral Researcher at UC Davis, Davis, CA, USA, under the mentorship of Professor Junshan Zhang, following a similar collaboration at Arizona State University, Tempe, AZ, USA. He is a Senior Research Scientist at Data61, CSIRO, Sydney, NSW, Australia. He is currently engrossed in exploring AI security, network security, graph neural networks, spatio-temporal data mining, reinforcement learning, and the Internet of Things.