MobiMixer: A Multi-Scale Spatiotemporal Mixing Model for Mobile Traffic Prediction

Jiaming Ma, Binwu Wang, Pengkun Wang, *Member, IEEE*, Zhengyang Zhou, *Member, IEEE*, Yudong Zhang, *Graduate Student Member, IEEE*, Xu Wang, and Yang Wang, *Senior Member, IEEE*

***Abstract*—Understanding mobile traffic data and predicting future trends are essential for wireless operators and service providers to allocate resources efficiently and manage energy effectively. Despite the strong performance of existing models, accurately forecasting mobile traffic remains a challenge due to limited spatial and temporal modeling capabilities and high computational complexity. This paper introduces MobiMixer, a lightweight and efficient multi-scale spatiotemporal mixing model. Its core concept is to integrate multi-scale information from both spatial and temporal dimensions to improve performance on mobile traffic data. We develop a hierarchical interaction module that incorporates super nodes to enable global high-level feature interactions among nodes with common patterns. Additionally, we employ a dynamic time warping strategy to decouple mobile traffic sequences into stable and seasonal components, which are then modeled at different scales using a multi-scale temporal mixing module. We conduct extensive experiments on mobile traffic datasets collected from four international cities. Compared with 21 state-of-the-art benchmark models, MobiMixer demonstrates highly competitive performance, achieving a maximum improvement of 48.49% on the Milan mobile dataset. The model achieves an improvement in training efficiency of up to 10.69 times and reduces memory usage by 33.01%.**

***Index Terms*—Mobility computing, mobile traffic, wireless network, mobile traffic prediction.**

I. Introduction

A. Background

RECENTLY, with the rapid advancement of mobile network technologies, a wide array of new network services have emerged, including internet of thing applications, as well as virtual and augmented reality. This dramatic increase in mobile traffic demand poses significant challenges for network operations and expenditures [1], [2]. To effectively allocate and optimize network resources, mobile service providers require accurate predictions of mobile traffic [3], [4]. By predicting future business loads, providers can dynamically allocate network resources, thereby improving the efficiency of spectrum and energy utilization. Furthermore, many emerging applications of the Internet of Things rely on precise device-level traffic predictions to improve the quality of service [5]. Consequently, mobile traffic prediction has emerged as a promising strategy to address these challenges, attracting considerable attention from both industry and academia [6].

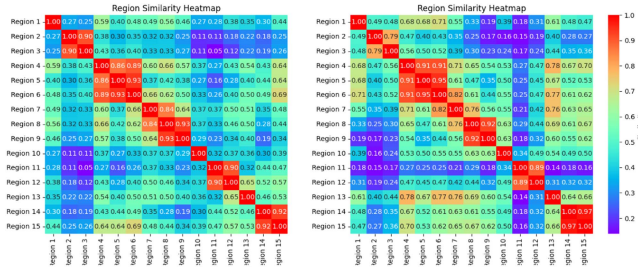

Fig. 1. Similarity heat maps of 15 regions at different times.

Mobile traffic prediction is a subfield of multivariate time series prediction that focuses on prediction future mobile traffic values based on observed sequences collected from various regions, such as network terminals or mobile access points. A critical aspect of this field is the ability to accurately capture the heterogeneous mobile traffic patterns inherent in traffic data. Mobile traffic exhibits varying dynamic distributions in different time steps, while correlations between traffic distributions in different regions are also present. Consequently, effectively modeling these spatiotemporal correlations is essential to achieve accurate predictions.

In recent years, driven by advances in deep learning technologies, researchers have introduced a variety of spatiotemporal learning models for mobile traffic prediction [7], [8], which have gradually become dominant in this field. These deep learning models typically consist of two key components: a spatial module designed to capture spatial correlations across regions and a temporal module aimed at modeling temporal dependencies across different time intervals.

Early deep learning approaches for mobile traffic prediction relied on Convolutional Neural Networks (CNNs) to identify spatial correlations between regions. However, their effectiveness in spatial modeling is constrained by the challenges associated with processing non-Euclidean data [9]. To address this limitation, researchers [10], [11], [12] introduced Graph Convolutional Networks (GCNs) into this domain. In GCN-based

Received 16 October 2024; revised 30 April 2025; accepted 17 June 2025. Date of publication 8 July 2025; date of current version 3 October 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 12227901 and in part by the Robotic AI-Scientist Platform of Chinese Academy of Sciences through the AI-driven Experiments, Simulations and Model Training. Recommended for acceptance by E. E. Tsiropoulou. *(Corresponding authors: Yang Wang; Binwu Wang.)*

The authors are with the University of Science and Technology of China, Heifei 230022, China (e-mail: JiamingMa@mail.ustc.edu.cn; wbw2024@ustc.edu.cn; pengkun@ustc.edu.cn; zzy0929@ustc.edu.cn; zyd2020@mail.ustc.edu.cn; wx309@ustc.edu.cn; angyan@ustc.edu.cn).

The source code is available at Anonymous Github.

Digital Object Identifier 10.1109/TMC.2025.3585007

models, mobile traffic data is represented as graph-structured data, where regions of interest or WiFi access points are modeled as nodes. By integrating GCNs with various temporal models, such as Recurrent Neural Networks (RNNs) or Transformers, these approaches can effectively capture complex spatiotemporal correlations. Leveraging the robust representational capabilities of GCNs for modeling spatiotemporal relationships, these models have achieved excellent performance in mobile traffic prediction tasks [13], [14], [15], [16], [17].

B. Motivation

Despite the promising success of current GCN-based models, we contend that specific limitations remain that need to be addressed to further enhance their performance:

❶ *Short-range and single-level spatial modeling:* Current GCN-based models primarily rely on predefined graphs to represent spatial dependencies between regions. However, these predetermined graphs may fail to fully capture the complex spatial relationships between regions. For instance, some models construct graphs based on geographical proximity, operating under the assumption that adjacent regions share similar traffic patterns. This assumption can be misleading, as functionally similar business districts may exhibit similar mobile traffic distribution patterns despite being geographically distant. To capture large-scale correlations between distant nodes, researchers need to stack multiple GCN layers, which encounters the over-smoothing challenge [18]. Furthermore, these static graphs remain unchanged throughout the model learning process, neglecting the dynamic nature of inter-regional correlations and potentially leading to misinterpretation of spatial dependencies. Furthermore, static graphs remain unchanged during the model training process, neglecting the dynamic nature of inter-regional correlations. Additionally, GCNs typically focus on modeling spatial features at a single microscopic level, overlooking the hierarchical structure of urban environments. Cities are composed not only of microscopic entities such as streets and blocks but also of macroscopic entities like business districts or large commercial areas [19], which encompass these smaller components. These macroscopic entities often experience more frequent internal movement interactions. Capturing higher-level macroscopic spatial features can significantly enhance the model's ability to analyze complex mobile traffic patterns and improve its prediction accuracy.

Using the mobile traffic dataset from Milan, Italy [20] as a case study, we divide Milan into multiple grids for analysis. Based on the dynamic time warping (DTW) method [21], we calculate the similarity of mobile traffic between 15 grids in two distinct time steps, as illustrated in Fig. 1. Interestingly, our analysis reveals dynamic and long-range correlations between regions. For instance, at the first time step, Region 12 and Region 13 exhibit a similarity of 0.65, indicating a strong correlation. However, at the second time step, the traffic correlation between these two adjacent regions diminishes significantly, with a similarity score dropping to only 0.31. In contrast, Region 13 demonstrates higher similarity with non-adjacent regions, such as Regions 4, 6, and 7. Furthermore, Regions 4, 5, and 6 consistently exhibit strong correlations across both time steps, as these three regions encompass a shopping mall, which likely contributes to their shared traffic patterns.

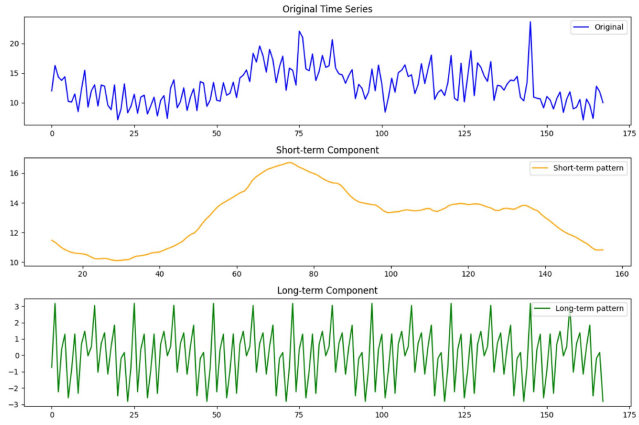

Fig. 2. Mobile traffic series decomposition. We decouple a mobile traffic series (top) of an area into the long-term pattern (bottom) and short-term pattern (middle).

❷ *Unified-scale temporal modeling:* Mobile traffic data typically exhibits multiple patterns, characterized by significant variations over time dimension, including increases, decreases, and fluctuations. These changes collectively form a complex mixture. Traditional mobile traffic prediction models often analyze these patterns at a unified time scale, which can lead to incorrect learning of mobile traffic patterns. For example, mobile traffic data recorded on an hourly basis may show noticeable fluctuations throughout the day. However, when aggregated into daily samples, these subtle variations may be smoothed out, while larger-scale fluctuations associated with holidays or weekends may become more prominent. Additionally, the unified-scale analysis approach struggles to address the non-stationary of mobile traffic sequences. Non-stationary in mobile traffic data refers to the gradual evolution of its temporal distribution characteristics (e.g., mean and variance) over time, posing a significant challenge for prediction.

In Fig. 2, we decompose the mobile traffic sequence of an area in Milan into two components: long-term patterns and short-term patterns. The long-term pattern displays relatively stable periodic features, whereas the short-term pattern depicts trends in sequence changes. These observations highlight the importance of employing a multi-scale analysis approach to uncover complex spatiotemporal variations. When modeled on a unified scale, the model struggles to capture these intricate patterns effectively.

❸ *Expensive computational complexity:* As the performance of the model improves, the computational complexity increases dramatically. The time complexity of spatial learners like GCNs grows quadratically with the number of nodes [22], [23], due to the message-passing mechanism in GCNs that propagates and aggregates features between nodes. Similarly, temporal learning modules such as RNNs or Transformers exhibit high computational complexity, with advanced architectures like Transformers showing quadratic scaling with input sequence length [24], [25]. In 5G networks, the dense deployment of

mobile cells, access points, and network devices poses significant challenges for these computationally intensive models, limiting their practical application in large-scale networks. The high computational requirements for prediction lead to unsustainable operational costs.

C. Contribution

To address these challenges, we design a multi-scale spatiotemporal mixing model, named MobiMixer. The detailed contributions are as follows.

❶ In the spatial dimension, we develop a novel hierarchical interaction module based on urban hierarchy theory: cities consist not only of microscopic nodes (such as roads or mobile access points) but also of macroscopic regions (such as business districts) with similar traffic patterns. To capture these dynamics, the module incorporates purification and diffusion processes. During the purification process, fine-grained nodes are clustered into a smaller number of coarse-grained super nodes, where the mapping relationships between nodes and super nodes are adaptively learned from data, enabling precise representation. Subsequently, these super nodes perceive shared macroscopic features from their fine-grained constituents. In the diffusion process, fine-grained nodes access their corresponding super nodes to extract useful macroscopic features. Through this hierarchical interaction, MobiMixer generates effective representations for each node. It further integrates node embedding techniques to learn personalized microscopic features for each fine-grained node. By modeling multi-scale spatial features at both microscopic and macroscopic levels, MobiMixer achieves comprehensive spatial learning without relying on predefined structures. Moreover, it facilitates correlation modeling between arbitrary (distant) nodes by grouping nodes with similar mobile traffic patterns into the same super-node, thereby avoiding the over-smoothing problem that commonly arises with stacked GCNs.

❷ In the temporal dimension, our method employs a multi-scale temporal fusion module. This module first uses Discrete Wavelet Transform (DWT) to extract high-frequency and low-frequency components from the frequency domain of mobile traffic sequences. These components are then transformed back into the time domain through a Multilayer Perceptron (MLP) layer, generating two components: long-term pattern and short-term pattern. Subsequently, we introduce a temporal prompt learning strategy to encode temporal prior information (e.g., time of day and day of the week). This prior knowledge helps the model analyze temporal dynamics more accurately. Following this, we apply a multi-scale modeling technique to extract and model information at different scales for both long-term and short-term components. Finally, the temporal fusion module integrates information from multiple temporal scales to analyze comprehensive temporal patterns. This multi-scale decoupling approach has dual advantages: the long-term patterns are relatively stable, which helps improve the model's robustness to non-stationary mobile data. In contrast, modeling short-term fluctuations enables the model to adapt more flexibly to the non-stationary characteristics of the data.

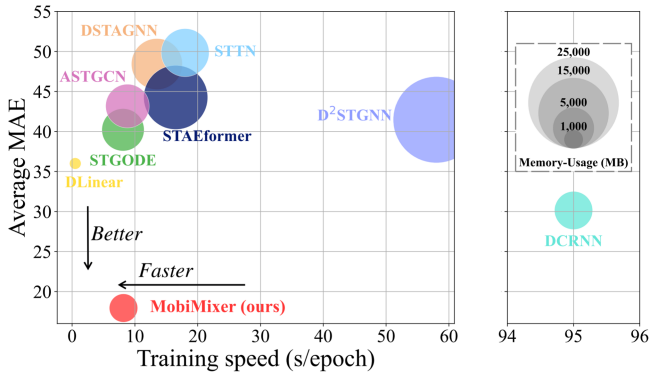

Fig. 3. Efficiency and performance comparison on the Milan SMS-IN dataset. A smaller bubble means a smaller memory usage.

❸ Deviating from computationally expensive graph convolution operators typically used for capturing spatial features, the proposed spatial interaction module is designed for linear computational complexity. Moreover, our model is based on a lightweight MLP architecture, providing modules with higher time efficiency. We validated the performance and efficiency of our model on a real mobile traffic dataset, and the results show that MobiMixer can achieve highly competitive performance and efficiency. As shown in Fig. 3, compared to state-of-the-art models, it achieves highly competitive performance on the Milan mobile dataset, with a maximum increase of 48.49%. Additionally, training efficiency is improved by 10.69 times, and memory usage is reduced by 33.01%.

Summary. The strength of MobiMixer lies in its effective integration of multi-scale information from both spatial and temporal dimensions. We have developed a hierarchical interaction module that utilizes super nodes, aggregating fine-grained nodes into macro-level super nodes through a refinement process to extract shared macroscopic features among nodes. This is followed by a diffusion process that enhances each node's representation by leveraging these macroscopic features. Additionally, we implement a dynamic time warping strategy to decouple mobile traffic sequences into stable and seasonal components, which are subsequently modeled at different scales using a multi-scale temporal mixing module.

D. Overview of the Paper

In the following sections, we will provide a detailed summary of existing research on mobile traffic prediction. Subsequently, we will formally define important variables and the mobile traffic prediction problem in Section III. In Section IV, we will then elaborate on the details of the designed model. Following that, we will evaluate the effectiveness of the model on a real mobile traffic dataset in Section V. Finally, we will conclude our work.

II. Related Work

A. Mobile Traffic Prediction

Early mobile traffic prediction methods relied on matching historical mobile traffic data with specific mathematical or statistical models to generate predictions based on probability

distributions. For instance, works such as [26], [27] employ the autoregressive integrated moving average or support vector regression to model short-term cellular traffic sequences [27], [28]. Beyond these approaches, ON-OFF models [29], Kalman filters [30], and Holt-Winters exponential smoothing models [31] have been applied to characterize the temporal and spatial attributes of mobile traffic loads. While these mathematical models simplified hyperparameter optimization, they struggled to capture complex non-linear correlations, leading to suboptimal performance.

In recent years, researchers have increasingly focused on designing deep learning techniques for mobile traffic prediction [3], [32]. DeepTP [33], an end-to-end approach, utilized a sequence-to-sequence model with attention mechanisms to capture spatial and temporal dependencies. LNTP [34] introduced an LSTM-based framework for timely and accurate network traffic prediction, incorporating wavelet transform and LSTM components. Several models have integrated GCN into mobile traffic prediction tasks [10], [12], [35], [36], achieving significant performance improvements due to the powerful representation capabilities of GCN. KGDA [37] proposed a graph-based model that decomposes the influence of static environmental factors and the dynamic autocorrelation of cellular traffic time series. SDGNet [15] introduced a switching-aware spatiotemporal graph neural network, leveraging dynamic graph convolution and gated linear units to predict short-term, medium-term, and long-term traffic consumption.

B. Multivariate Time Series Prediction

Mobile traffic prediction is a subset of multivariate time series prediction tasks that has witnessed significant advances in model development. One branch of multivariate time series prediction models focuses exclusively on modeling dependencies across multiple time steps. These models employ techniques such as multi-scale modeling and self-masking to enhance their temporal representation capabilities. For example, AMD [38] and TimeMixer [39] extract multi-scale representations through downsampling and mixing techniques for future data prediction. However, these models do not explicitly model spatial dependencies, which is essential for mobile traffic prediction tasks where modeling correlations between sites with similar traffic distributions is necessary. Our model proposes a novel hierarchical interaction mechanism to further capture multi-scale spatial information. Furthermore, our model novelly uses the DWT algorithm to decouple temporal structures and applies multi-scale modeling techniques for more precise temporal modeling.

Another branch of architecture for multivariate time series prediction tasks is the spatiotemporal learning model, which emphasizes the joint modeling of temporal and spatial aspects. Initially, CNN is used to extract local spatial features [40]. These models typically combine CNN with RNN or its variants to effectively capture temporal dependencies, improving prediction accuracy by integrating spatial and temporal patterns. For instance, ST-ResNet, based on CNN architecture with residual layers [41], has demonstrated promising results. Recent studies [35], [42], [43], [44] have explored the integration of GCN, GAT, and their variants with RNN, CNN, or Transformers to capture complex spatial and temporal dependencies in graph-structured spatiotemporal data. Models like DCRNN [45] utilize bidirectional random walks on graphs to handle spatial dependencies and adopt an encoder-decoder framework to address temporal dependencies. D ^{2} STGNN [46] introduces an estimation gate to separate traffic signals into diffusion and inherent components, thus improving the precision of multivariate time series prediction.

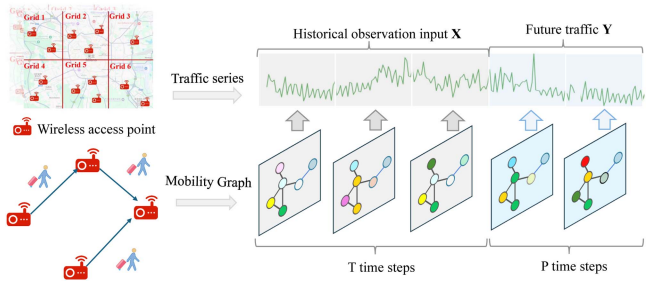

Fig. 4. Mobile graph and mobile traffic prediction. We consider access points or regions containing multiple access points as nodes, and characterize the mobile graph through its structure and traffic sequence.

TABLE I
SOME IMPORTANT VARIABLES WITH THEIR DEFINITIONS

Notation	Definition
$\mathbf{X}/\mathbf{Y}$	Historical observation input/Future values
T/P	History / Predicted time steps
N	The number of nodes
K	The number of super nodes
J	The number of temporal modules
U	The number of temporal scales
$\overline{\mathbf{H}}_s^0$ / $\overline{\mathbf{H}}_l^0$	The input representation of short-term / long-term components
$\overline{\mathbf{H}}_s^s$ / $\overline{\mathbf{H}}_l^s$	The representation of short-term / long-term components after hierarchical spatial interaction
$\overline{\mathcal{Z}}_l^J$ / $\overline{\mathcal{Z}}_s^J$	The representation after multi-scale temporal mixing of short-term / long-term components
$\mathcal{Z}$	Output representation
$\hat{\mathbf{Y}}$	Predicted value

III. Problem Formulation

In this section, we introduce specific notations and provide a formal definition of the mobile traffic prediction problem.

A. Mobile Network Unit of Interest

A mobile access point or a cell tower used to process and transmit mobile user data can be defined as a network unit of interest. Typically, considering the dense deployment of 5G or 6G base stations, we can also define a region containing multiple access points or cell towers as a network unit of interest to reduce the complexity of the research, as shown in Fig. 4. Some important variables are defined in Table I.

B. Mobile Graph

We use $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$ to represent a mobile graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ means N network units of interest (i.e., nodes) in the graph. $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the graph $\mathcal{G}$ to describe the connections between nodes. if there is a directed edge from node v_i to node v_j, the ith row and the jth column, $\mathcal{A}(i, j)$ are equal to 1. The vector of features of the node $X_t = \{x_t^i, \ldots, x_t^N\}$ denotes the mobile traffic volume of all the nodes at the time step t, where $x_t^i \in \mathbb{R}^d$ denotes the traffic volume

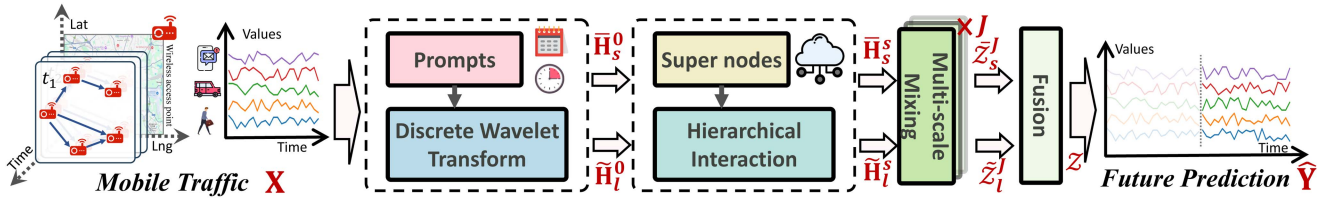

Fig. 5. Details of the proposed model. Our model first decomposes the input mobile traffic data into long-term and short-term components and utilizes embedding techniques. They are then sequentially input into the Hierarchical Interaction Module to capture spatial correlations and the multiple-scale temporal mixing module to model temporal dynamics.

of the node v_i at the time step t, and d means the number of observed traffic features.

C. Mobile Traffic Prediction

Given a mobile graph $\mathcal{G}$ with T past historical observed traffic volume matrix $\mathbf{X} = \{X_1, \ldots, X_t, \ldots, X_T\} \in \mathbb{R}^{T \times N \times d}$ and the graph $\mathcal{G}$, the mobile traffic prediction task aims to effectively predict the future P time steps traffic volume matrix $\mathbf{Y} = \{X_{T+1}, \ldots, X_{T+P}\} \in \mathbb{R}^{P \times N \times d}$.

IV. Method

A. Overview

As shown in Fig. 5, MobiMixer utilizes DWT to decompose mobile traffic time series into long-term pattern sequence $\mathbf{X}_l$ and short-term pattern sequence $\mathbf{X}_s$. Then, our spatiotemporal prompt embedding technique integrates prior knowledge into two sequences, with outputs denoted as $\overline{\mathbf{H}}_s^0$ and $\widetilde{\mathbf{H}}_l^0$. Subsequently, we use the hierarchical spatial interaction module to model the spatial features of mobile traffic, with the output denoted as $\overline{\mathbf{H}}_s^s$ and $\widetilde{\mathbf{H}}_l^s$. Following this, we employ the multi-scale temporal mixing module to model temporal dependencies across multiple time scales. The short-term component is denoted as $\overline{\mathcal{Z}}_s^J$ and the long-term component output is denoted as $\widetilde{\mathcal{Z}}_l^J$. Finally, we use a fusion module for the output representation $\mathcal{Z}$, and then we use MLP layers as decoder to generate predictions $\widehat{\mathbf{Y}}$.

B. Wavelet Transform for Mobile Decomposition

For complex data from multiple sources, decomposing them into different interpretable sources can enhance the resilience of the model to structurally rich variables. Drawing inspiration from temporal series structural decomposition techniques, we decompose mobile traffic sequences into stable patterns and fluctuating trend patterns. Mainstream approaches involve using mean kernel functions for decomposition [25], [39]. In this study, we propose an alternative approach by introducing the DWT to decompose the original mobile traffic data into multiple frequency sequences, enabling multi-resolution analysis. DWT involves two key processes [47], [48]. First, it transforms the time domain into the frequency domain and extracts different frequency components. Subsequently, in the second phase, it combines the extracted components and reconstructs them into

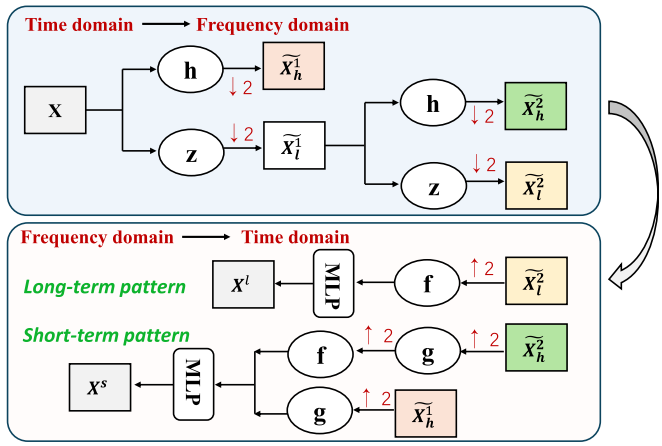

Fig. 6. DWT is used for mobile traffic decomposition.

long-term and short-term patterns in the time domain, as shown in Fig. 6.

Specifically, DWT is used to transform the mobile traffic series $\mathbf{X} \in \mathbb{R}^{T \times N \times d}$ into a low-frequency component $\widetilde{\mathbf{X}}_l^1 \in \mathbb{R}^{\lfloor \frac{T}{4} \rfloor \times N \times d}$ and two high-frequency components including $\widetilde{\mathbf{X}}_h^1 \in \mathbb{R}^{\lfloor \frac{T}{2} \rfloor \times N \times d}$ in the first process and $\widetilde{\mathbf{X}}_h^2 \in \mathbb{R}^{\lfloor \frac{T}{4} \rfloor \times N \times d}$ in the second process. Typically, low-frequency signal components capture slow-varying characteristics, whereas high-frequency components represent fine-grained changes. This process can be formulated as:

$$\begin{aligned} \widetilde{\mathbf{X}}_l^2 &= \left(\mathbf{z} * (\mathbf{z} * \mathbf{X})_{(\downarrow 2)}\right)_{(\downarrow 2)} \\ \widetilde{\mathbf{X}}_h^2 &= \left(\mathbf{h} * (\mathbf{z} * \mathbf{X})_{(\downarrow 2)}\right)_{(\downarrow 2)} \\ \widetilde{\mathbf{X}}_h^1 &= (\mathbf{h} * \mathbf{X})_{(\downarrow 2)} \end{aligned} \tag{1}$$

where $\mathbf{z}$ is low-pass filter and $\mathbf{h}$ represent high-pass filter of a wavelet. $*$ is the convolution operation and $\downarrow 2$ means that the output is down-sampled by 2. Next, the frequency components of the mobile traffic sequence are then mapped back to the time domain. In this process, we use upsampling techniques to align with the length of the input time series.

$$\begin{aligned} \mathbf{X}_l &= \mathrm{MLP}\left(\mathbf{f} * \left(\mathbf{f} * \left(\widetilde{\mathbf{X}}_l^2\right)_{\uparrow 2}\right)_{\uparrow 2}\right) \\ \mathbf{X}_s &= \mathrm{MLP}\left(\mathbf{f} * \left(\mathbf{g} * \left(\widetilde{\mathbf{X}}_h^2\right)_{\uparrow 2}\right)_{\uparrow 2} + \mathbf{g} * \left(\widetilde{\mathbf{X}}_h^1\right)_{\uparrow 2}\right) \end{aligned} \tag{2}$$

where $\mathbf{g}$ and $\mathbf{f}$ represent the up-sampled kernel. MLP$(\cdot)$ means a multilayer perceptron layer. $\mathbf{X}_l \in \mathbb{R}^{T\times N\times d}$ and $\mathbf{X}_s \in \mathbb{R}^{T\times N\times d}$ represent the long-term and short-term pattern components, respectively.

C. Spatiotemporal Prompt Embedding Learning

To enhance modeling capabilities, we incorporated the innovative concept of meta-learning, a strategy widely used in computer vision and natural language processing fields. This strategy utilizes various prior knowledge as supplements to improve the model's accuracy. To this end, we designed a series of embedding techniques specifically tailored for encoding time and space priors related to mobile traffic data. By integrating these embeddings into our framework, our aim is to enhance the model's ability to leverage prior knowledge and contextual cues, thereby facilitating a more detailed analysis of the inherent complex spatiotemporal dynamics in mobile traffic patterns.

To keep the native information in the raw data, we first utilize a MLP to map $\mathbf{X}_l$ and $\mathbf{X}_s$ into the high dimensional space and obtain the feature embedding:

$$\begin{aligned}\mathbf{E}_i^l &= \text{MLP}\left(\mathbf{X}_l\right) \in \mathbb{R}^{T\times N\times d_e}\\ \mathbf{E}_i^s &= \text{MLP}\left(\mathbf{X}_s\right) \in \mathbb{R}^{T\times N\times d_e}\end{aligned} \tag{3}$$

1) Temporal Embedding: We need to integrate temporal information into the model, which includes information about time step of day, days of week, and holiday data. This prior information is crucial for accurately mining temporal patterns.

Time step of day: We use a learnable embedding vector $\mathcal{E}_t \in \mathbb{R}^{N_t\times d_e}$ to encode the information of each time step in a weak, where N_d indicates the number of data points in a day, and $\mathcal{E}_t$ can adaptively represent fine-scale temporal information. For example, if the sampling frequency is one hour, then there will be 24 data points in a day, hence N_t is set to 24.

Denote $W^t \in \mathbb{R}^T$ to denote the time step-of-day data information from the first time step to the Tth time step in the input $\mathbf{X}$, then W^t is used as indices to extract the corresponding day-of-week embedding from $\mathcal{E}_t$, denoted as $\mathbf{E}_t \in \mathbb{R}^{T\times d_e}$.

Holiday information: We also use $W^h \in \mathbb{R}^T$ to denote the holiday information from the first time step to the Tth time step in the input $\mathbf{X}$. W^h is a 0-1 vector, and a value of 1 means that the location time step is on a holiday. Finally, we use onthot embedding method to encode W^h and get the output $\mathbf{E}_h \in \mathbb{R}^{N_d\times d_e}$ to integrate holiday information.

2) Spatial Embedding: Due to the fact that the geographical properties of different nodes may lead to variations in their traffic patterns, for instance, mobile traffic patterns in shopping malls differ from those in residential areas, it is essential to integrate these prior features into the model. However, acquiring these geographical attribute features may pose challenges due to data privacy or policy restrictions. To overcome this limitation, we employ an adaptive node embedding approach that analyzes data to capture the traffic distribution characteristics of different nodes. Specifically, we use a learnable node embedding $\mathbf{E}_s \in \mathbb{R}^{N\times d_e}$ to represent features of N nodes in graph. This embedding can be updated end-to-end with the model, which can capture the personalized traffic patterns of each node from the data, thereby enhancing generalization without relying on specific geographical location information.

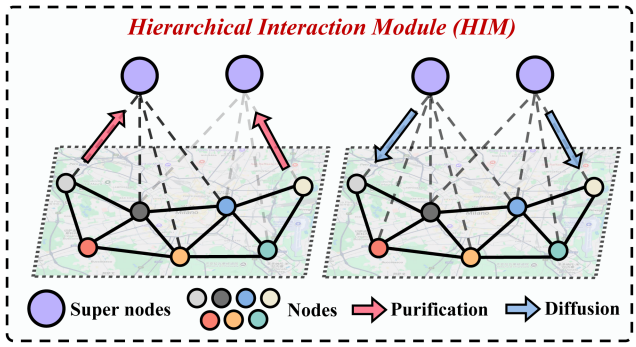

Fig. 7. The details of our hierarchical interaction module.

3) Output of Embedding Module: We extend the dimensions of $\mathbf{E}_t$, $\mathbf{E}_h$, and $\mathbf{E}_s$ to $(T\times N\times d_e)$. Finally, we splice three matrices with two input embedding $\mathbf{E}_i^l$to get the output:

$$\begin{aligned}\overline{\mathbf{H}}_s^0 &= \mathbf{E}_t||\mathbf{E}s||\mathbf{E}_i^s \in \mathbb{R}^{T\times N\times 3d_e}\\ \widetilde{\mathbf{H}}_l^0 &= \mathbf{E}_t||\mathbf{E}s||\mathbf{E}_i^l \in \mathbb{R}^{T\times N\times 3d_e}\end{aligned} \tag{4}$$

where $\overline{\mathbf{H}}_s^0$ and $\widetilde{\mathbf{H}}_l^0$ represent the short - and long-term output representation, respectively.

D. Hierarchical Interaction Module

GCNs have been widely used in spatiotemporal learning due to their powerful representation capabilities, primarily stemming from their execution of global message passing and aggregation mechanisms. However, their time complexity grows quadratically with the number of nodes. Additionally, the feature aggregation properties within the adjacency matrix limit their ability to capture short-range node relationships while also overlooking shared mobile traffic patterns between nodes.

To tackle these obstacles, we introduce a Hierarchical Interaction Module (HIM) for high-level spatial feature interaction. This novel concept takes inspiration from urban hierarchy principles. While the inter-dependencies among micro-nodes can be intricate, patterns tend to exhibit similarities between locations within a given region at a broader scale. HIM addresses this by defining coarse-scale super nodes that encapsulate sets of fine-grained nodes. The module incorporates a purification process to capture nuanced node-sharing features for refining super node representations and a diffusion process to diffuse super node features to fine-grained nodes, as illustrated in Fig. 7.

Specifically, HIM contains K super nodes, where K is a hyperparameter and $K \ll N$. then we also adopt a feature vector $\mathcal{M} \in \mathbb{R}^{K\times d_s}$ for these K super nodes, where d_s means the number of channels. $\mathcal{M}$ is a learnable parameter that adaptively represents macroscopic context features. Then if given the input embedding $\mathbf{H}_0$, we use it as query vector to calculate the affinity of each node and super nodes. This purification process can be denoted as follows:

$$\mathbf{A}_s(k,m) = \frac{\exp\left(\langle \mathcal{M}(k), Q(m)\rangle\right)}{\sum_{m'=1}^{N}\exp\left(\langle \mathcal{M}(k), Q(m')\rangle\right)} \tag{5}$$

where $\mathbf{A}(k,m)$ records the affinity between the mth node and kth super node. $\mathcal{M}(k)$ means the kth row in $\mathcal{M}$. $Q(m)$ means the mth row in the query vector $Q \in \mathbb{R}^{N\times d_q}$. $\langle\cdot\rangle$ represents the dot product of two matrices to calculate the attention coefficient. Finally, we can get an attention matrix $\mathbf{A}_s \in \mathbb{R}^{K\times N}$ that records the attention coefficients of N nodes and K super nodes.

Then we extract shared patterns from N nodes to update the features of super nodes, which represent coarse-scale macroscopic features shared among nodes:

$$\mathbf{H}_v = \mathbf{A}_s\left(\mathbf{H}_0\mathbf{W}_1\right) \in \mathbb{R}^{K\times d_q} \tag{6}$$

where $\mathbf{W}_1$ is the learnable parameter. Based on the extracted shared features $\mathbf{H}_v$, we adopt a diffusion process to diffuse these shared features to fine-scale nodes to achieve spatial feature interaction. Specifically, we compute the diffusion matrix:

$$\mathbf{A}_d = \frac{\exp\left(\langle Q(m), \mathcal{M}(k)\rangle\right)}{\sum_{k'=1}^{K}\exp\left(\langle Q(m), \mathcal{M}(k')\rangle\right)} \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{N\times K}$ represents the mapping coefficient from super nodes to fine-grained nodes, then we diffuse $\mathbf{H}_v$ to every node as follows,

$$\mathbf{H}_n = \mathbf{A}_d\left(\mathbf{H}_v\mathbf{W}_2\right) \in \mathbb{R}^{N\times d_q} \tag{8}$$

where $\mathbf{W}_2$ is learnable parameter. We add the output $\mathbf{H}_n$ and the input $\mathbf{H}_0$ through a residual connection to obtain the final output vector that incorporates spatial information. For both short-term and long-term components, $\overline{\mathbf{H}}_s^0 \in \mathbb{R}^{T\times N\times 3d_e}$ and $\widetilde{\mathbf{H}}_l^0 \in \mathbb{R}^{T\times N\times 3d_e}$, we first compress their time series and feature dimensions, and then input them into two independent HIM, generating input representations denoted as $\overline{\mathbf{H}}_s^s \in \mathbb{R}^{N\times T\times d_l}$ and $\widetilde{\mathbf{H}}_l^s \in \mathbb{R}^{N\times T\times d_l}$.

Computational complexity: Traditional mobile traffic prediction models utilize GCN for spatial feature extraction, with a time complexity of $\mathcal{O}(N^2)$, which has a quadratic relationship with the number of nodes [22], [43]. Some spatiotemporal learning models use Transformer, which can essentially be viewed as graph convolution operation on a fully graph. The computational complexity of the self-attention mechanism it uses is also $\mathcal{O}(N^2)$. These methods pose an intensive computational burden, especially when N is large. The complexity of calculating attention coefficients of our method in (5) and (7) are $\mathcal{O}(KN)$, typically, we set K to be much smaller than N. Therefore, the computational complexity is effectively approximated as $\mathcal{O}(N)$. This linear complexity significantly improves the efficiency of spatial feature extraction.

E. Multi-Scale Temporal Mixing

Time series data (such as mobile traffic sequences) inherently exhibit different characteristics at different scales [25], [49]. Fine scales excel at capturing intricate details, while coarse scales highlight broader macroscopic changes. This multi-scale perspective effectively reveals complex patterns in the data, thereby enhancing the modeling of temporal variations. Following the pioneer multi-scale temporal modeling works [39], [50], we introduce a multi-scale temporal mixing module for temporal features learning.

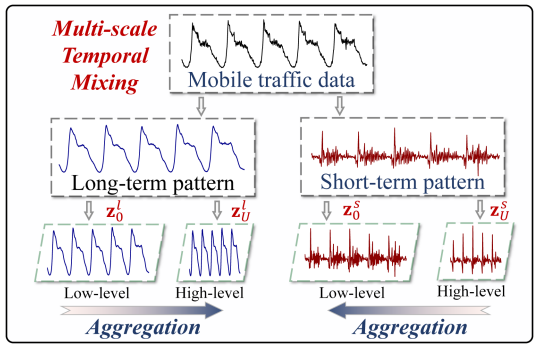

Fig. 8. Multi-scale temporal mixing.

Specifically, we first perform pooling operations along the temporal dimension to decompose long-term pattern sequences or short-term pattern sequences into time series at different scales. We use the long-term sequence $\widetilde{\mathbf{H}}_s$ as an example, and we first mix node and feature dimensions of the input, i.e., $\widetilde{\mathbf{H}}_s \in \mathbb{R}^{T\times C}$, where $C = N * d_l$. Then, we use the average pooling kernels with U scales to down-sample the $\widetilde{\mathbf{H}}_s$ and finally obtain a set of multi-scale time representation $\widetilde{\mathcal{Z}}_s = \{\widetilde{\mathbf{z}}_0, \ldots, \widetilde{\mathbf{z}}_U\}$, where $\mathbf{z}_u \in \mathbb{R}^{\lfloor \frac{T}{2^u}\rfloor\times C}, u \in \{0, \ldots, U\}$. The lowest-level sequence $\widetilde{\mathbf{z}}_0 = \widetilde{\mathbf{H}}_s$ represents the input sequence containing the most subtle time variations, while the highest-level sequence $\widetilde{\mathbf{z}}_U$ captures macro changes. Similarly, we can get short-term pattern series with multiple scales $\overline{\mathcal{Z}}_l = \{\overline{\mathbf{z}}_0, \ldots, \overline{\mathbf{z}}_U\}$, where $\overline{\mathbf{z}}_U = \overline{\mathbf{H}}_s$. These two sequences, $\overline{\mathcal{Z}} = \{\overline{\mathbf{z}}_0, \ldots, \overline{\mathbf{z}}_U\}$ and $\widetilde{\mathcal{Z}} = \{\widetilde{\mathbf{z}}_0, \ldots, \widetilde{\mathbf{z}}_U\}$, are fed into J temporal modules, each of which contains $2U$ MLP layers to independently model U scales for both long-term and short-term components. The use of multiple modules enables the model to capture more precise representations at different scales, thereby facilitating an accurate analysis of the complex temporal dynamics inherent in mobile traffic. We take ith layer as an example, and the input is denoted as $\widetilde{\mathcal{Z}}_s^i = \{\widetilde{\mathbf{z}}_0^i, \ldots, \widetilde{\mathbf{z}}_U^i\}$ and $\overline{\mathcal{Z}}_l^i = \{\overline{\mathbf{z}}_0^i, \ldots, \overline{\mathbf{z}}_U^i\}$.

1) Long-Term Temporal Modeling: Long-term patterns play a crucial role in spatiotemporal learning, showcasing periodic characteristics that reflect human mobility patterns. For instance, the weekly cycle of mobile traffic formed by daily variations. Accurately identifying these long-term patterns can assist in more precise predictions of future traffic conditions.

After decomposing the time series into information of different scales, such as $\widetilde{\mathcal{Z}}_l^i = \{\widetilde{\mathbf{z}}_0^l i, \ldots, \widetilde{\mathbf{z}}_U^i\}$, we first use two layers of MLP to map the multi-scale information, and then we fuse the information of different scales. Specifically, for the uth temporal scale of long-term representation $\widetilde{\mathbf{z}}_u^i \in \mathbb{R}^{\lfloor \frac{T}{2^{u-1}}\rfloor\times C}$ can be formalized as,

$$\widetilde{\mathbf{z}}_u^i = \widetilde{\mathbf{z}}_u^i + \mathrm{MLP}\left(\widetilde{\mathbf{z}}_{u-1}^i\right) \in \mathbb{R}^{\lfloor \frac{T}{2^u}\rfloor\times C} \tag{9}$$

As shown in Fig. 8, following the work [39], we aggregate information scale by scale from low level to high level. The output after aggregating information from different scales is denoted as $\widetilde{\mathcal{Z}}_l^{i+1} = \{\widetilde{\mathbf{z}}_0^i, \ldots, \widetilde{\mathbf{z}}_U^i\}$, which will be input to the next multi-scale temporal mixing module.

2) Short-Term Temporal Modeling: To model short-term patterns at varying scales, we progressively aggregate the multiple time series with various scales from high level to low level. Specifically, for the uth scale of short-term representation $\overline{\mathbf{z}}_u^i \in \mathbb{R}^{\lfloor \frac{T}{2^u} \rfloor \times C}$, we aggregate features scale by scale as follows,

$$\overline{\mathbf{z}}_u^i = \overline{\mathbf{z}}_u^i + \text{MLP}\left(\overline{\mathbf{z}}_{u+1}^i\right) \in \mathbb{R}^{\lfloor \frac{T}{2^u} \rfloor \times C} \tag{10}$$

The output after aggregating information from different scales is denoted as $\overline{\mathcal{Z}}_s^{i+1} = \{\overline{\mathbf{z}}_0^i, \ldots, \overline{\mathbf{z}}_U^i\}$.

3) Short-Term and Long-Term Temporal Fusion: After J layers of neural network modeling, we add the long-term component $\widetilde{\mathcal{Z}}_l^J = \{\widetilde{\mathbf{z}}_0^J, \ldots, \widetilde{\mathbf{z}}_U^J\}$ and the short-term component $\overline{\mathcal{Z}}_s^J = \{\overline{\mathbf{z}}_0^J, \ldots, \overline{\mathbf{z}}_U^J\}$. For uth temporal scale, we aggregate as follows:

$$\mathbf{z}_u = \text{FeedForward}\left(\widetilde{\mathbf{z}}_u^J + \overline{\mathbf{z}}_u^J\right) \tag{11}$$

where FeedForward$(\cdot)$ contains two MLP layers with GELU activation. The output with U scales is denoted as $\mathcal{Z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_U\}$.

Computational complexity: Recurrent Neural Networks such as RNN and LSTM are notorious for their inefficiency in modeling temporal dependencies due to their sequential nature, and their performance is often suboptimal [24], [39]. The Transformer architecture, which has gained popularity for time series modeling, overcomes this inefficiency by utilizing self-attention mechanisms to model dependencies at arbitrary time steps, resulting in excellent long-term modeling capabilities. However, the time complexity of Transformers grows quadratically with the length of the input sequence, $\mathcal{O}(T^2)$. In this paper, we use MLP for time series modeling, which is a lightweight architecture with a time complexity of $\mathcal{O}(T)$.

F. Future Prediction

Finally, we have temporal representations at different scales $\mathcal{Z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_U\}$, then, we integrate these time series information at different scales to predict future mobile traffic,

$$\widehat{\mathbf{Y}} = \sum_{u=0}^{U} \text{MLP}_u(\mathbf{z}_u), u \in \{0, \ldots, U\} \in \mathbb{R}^{P \times N \times d} \tag{12}$$

where $\text{MLP}_u(\cdot)$ denotes the predictor for the uth scale sequence. This predictor initially employs a single linear layer to directly extract historical information from $\mathbf{z}_u$ with a length of $\lfloor \frac{T}{2^u} \rfloor$ time steps. Then it forecasts this information over the subsequent P time steps and maps the resulting deep representation to N nodes. Because distinct predictors are configured for different scale time information because each scale encapsulates unique historical data with varying significance in generating the prediction $\widehat{\mathbf{Y}}$.

V. Experiment

In this section, we assess the efficacy of our model on two mobile traffic datasets from two cities. We answer the following potential concerns:

- **Q.1**. How effective is MobiMixer compared to advanced models? Refer to Section V-C.
- **Q.2**. Does each proposed component of MobiMixer contribute to the performance? Refer to Section V-D.
- **Q.3**. What is the computational complexity of the model? Refer to Section V-E.
- **Q.4**. How do hyperparameters affect the model's performance? Refer to Section V-F.
- **Q.5**. How scalability is the model in large-scale mobile networks? Refer to Section V-G.
- **Q.6**. Is MobiMixer effective for predicting peak mobile traffic and weekends? Refer to Section V-H.
- **Q.7**. How robust is MobileMixer to the time interval of mobile traffic data? Refer to Section V-I.

A. Mobile Traffic Datasets

We utilize mobile traffic datasets from four cities: Milan and Trentino in Italy, and Shanghai and Beijing, two modern metropolises in China. The Milan and Trentino datasets were provided by Telecom Italia, a well-known European telecommunications service provider [20].

❶ **Milan** dataset includes multiple mobile traffic features: outgoing calls (CALLOut), incoming calls (CALLIn), sent text messages (SMSOut), and received text messages (SMSIn). These features encompass mobility records collected over two months, from November 1, 2013, to January 1, 2014, across 400 regions. The data time interval is 1 h.

❷ **Trentino** dataset contains three mobile traffic features: SMS, calls (CALL), and Internet usage (Internet). This dataset covers data from 11466 regions spanning 62 days, from November 1, 2013, to January 1, 2014, at 23:00. The data time interval is also set to 1 h.

❸ **Beijing** dataset contains blog check-in data received from 528 regions in Beijing through the Weibo application from January to December 2023. The Weibo application is a mainstream social media platform in China, with 590 million monthly active users as of 2024, offering extensive coverage. The data points are aggregated at 5-minute intervals.

❹ **Shanghai** dataset [51], [52], [53] comprises over 7.2 million call records generated by 9,481 mobile phones accessing the Internet via 3,233 base stations from June 2014 to November 2014. The data time interval is 10 minutes.

1) Data Process: Each region or base station is treated as a node. Missing values are filled with 0. We primarily report the performance of various models in the Milan dataset because of its moderate size, which allows the majority of models to run freely. Subsequently, the Trentino dataset, which contains 10,000 nodes, is used to evaluate the scalability of the models in large-scale mobile networks; some complex models encounter memory complexity challenges and could not run. The Beijing and Shanghai datasets, representing two cities with completely different administrative functions in China, are used to assess the robustness and generalization of the models under fine-grained sampling rates. All datasets are split along the time axis into training, validation, and test sets in a 6:2:2 ratio, with missing values imputed as zero. The summary of four datasets is shown in Table II.

TABLE II
THE SUMMARY OF FOUR DATASETS

City	Milan	Trentino	Beijing	Shanghai
Node	400	11466	528	3233
Time Interval	1 Hour	1 Hour	5 mins	10 mins
Time Span	62 days	62 days	12 months	6 months

B. Experimental Setups

1) Settings and Hyperparameters: All datasets are split into training, validation, and testing sets in a 6:2:2 ratio along the time axis. Experiments are conducted three times, implemented in PyTorch with Python 3.11.5, and run on an NVIDIA H100 80 GB GPU. We employ 24 time steps observation window to forecast the mobile traffic for the subsequent 24 time steps, i.e., $T = P = 24$. We make predictions for each mobile traffic feature separately. For fair comparison, time step of day and day of week information are also integrated into baselines as input features.

We use the Adam optimizer [54] with initial learning rate 10^{-4}. The learning rate is halved after each gradient descent. We use the L_2 loss optimization function with early-stopping technique with a patience of 10 to prevent models from overfitting. All the hidden dimensions including d_e, d_q, d_l in the Mobimixer are 16. In the hierarchical interaction module, the number of super nodes K is equal to 8. In the temporal learning module, we set up two different timescales (i.e., $U = 2$). The number of layers of multi-scale temporal mixing J is equal to 5.

2) Metrics: We use three metrics to assess the gap between predicted $\widehat{\mathbf{Y}} \in \mathbb{R}^{P\times N\times d}$ and ground-truth values $\mathbf{Y} \in \mathbb{R}^{P\times N\times d}$, including Mean Square Error (MAE), Root Mean Square Error (RMSE), and Bit Error Rate (BER) which are defined as,

$$\text{MAE} = \frac{1}{P \times N \times d} \sum_{i=1}^{P\times N\times d} |\mathbf{Y}_i - \widehat{\mathbf{Y}}_i| \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{1}{P \times N \times d} \sum_{i=1}^{P\times N\times d} (\mathbf{Y}_i - \widehat{\mathbf{Y}}_i)^2} \tag{14}$$

$$\text{BER} = \frac{\#\left\{\hat{\mathbf{Y}}_t = \mathbf{Y}_t\right\}}{P} \times 100\% \tag{15}$$

3) Baseline: To compare the performance of MobiMixer with SOTA models, we select **open-source** models specifically designed for mobile traffic prediction and general spatiotemporal prediction models, including: **ConvLSTM**, **MVSTGN** [13], **STDenseNet** [55], **ST-Tran** [56], **AHSTGNN** [57], **AGCRN** [58], **ASTGCN** [59], **BigST** [60], **D ^{2} STGNN** [46], **DCRNN** [45], **D ^{2} STGNN** [46], **DGCRN** [61], **DLiner** [62], **AMD** [50], **TimeMixer** [39], **DSTAGNN** [63], **GWNet** [64], **STAEformer** [65], **STGCN** [66], **STGODE** [67], **STID** [68], **STNorm** [69], **STNN** [70].

C. Prediction Performance Comparison (Q.1)

Tables III and IV show the predictive performance of different models at time granularity on Milan dataset.

❶ *MobileMixer vs Time Series Model:* ConvLSTM exhibits the poorest performance as it relies exclusively on LSTM for modeling temporal dependencies. In contrast, AMD achieves improved predictive performance through its dual-dependency interaction module and adaptive multi-predictor synthesis, which effectively capture long-range sequence dependencies. DLinear performs relatively well among time series models due to its simpler architecture, which reduces the risk of overfitting the training data. However, these models struggle to adequately model spatial dependencies, resulting in significantly lower performance compared to MobileMixer. Our model addresses this limitation by introducing a hierarchical spatial interaction network that facilitates efficient spatial dependency learning.

❷ *MobileMixer vs Spatiotemporal Graph Learning Model:* Compared to STGCN and DGCRN, which integrate traditional graph convolutional networks, AHSTGNN and GWNet utilize static adaptive graph learning and dynamic graph learning modules to capture complex spatiotemporal dependencies, demonstrating superior performance. BigST and STID are spatiotemporal graph prediction models based on MLP architecture, and despite their simple structure, they surprisingly achieve results close to those of GCN-based models. STAEformer, STNN, and D ^{2} STGNN leverage Transformers to capture mobile traffic patterns, but their performance is unsatisfactory, possibly because Transformers use a uniform scale for temporal modeling, which may not effectively address the non-stationary characteristics of mobile traffic data. DCRNN achieves good predictive performance due to the representation power of its diffusion graph convolutional network for spatiotemporal graph data. In contrast to these models, the significant advantage of our model lies in its ability capture different granularities of temporal and spatial features. Spatial patterns are decomposed into fine-scale nodes and coarse-scale regions, and learning is performed separately for different granularities. This method mitigates the complexity of spatial features across different nodes, thereby reducing the complexity.

MobiMixer achieves the best performance across all metrics on all datasets. Impressively, in the SMS dataset, MobiMixer demonstrates a relative performance improvement of 11.98% to 48.49% compared to advanced models. In the CALL dataset, MobiMixer achieves a performance boost ranging from 13.33% to 43.30%.

D. Ablation Experiment (Q.2)

We evaluate the effectiveness of each component on the SMS-IN dataset. And we create the following variants by removing the individual components:

- **w/o pro** means that we remove the temporal prompt module of MobiMixer.
- **w/o HIM** means that we remove the hierarchical interaction module of MobiMixer.
- **w/o nemb** means that we remove the node embedding of MobiMixer.

TABLE III
SHORT-TERM PREDICTION PERFORMANCE FOR SMS-IN AND SMS-OUT

Model	SMS-IN							SMS-Out						
	6 Horizons		18 Horizons		24 Horizons		BER	6 Horizons		18 Horizons		24 Horizons		BER
	MAE	RMSE	MAE	RMSE	MAE	RMSE		MAE	RMSE	MAE	RMSE	MAE	RMSE	
ConvLSTM	63.77	134.67	70.73	146.14	72.27	144.07	0.5520	39.08	84.40	40.81	88.40	42.06	88.82	0.4951
MVSTGN	50.29	103.98	56.47	127.44	62.34	124.96	0.6849	32.75	68.67	33.00	74.92	35.27	74.66	0.7161
STDenseNet	57.11	115.34	66.48	133.48	62.31	128.21	0.7068	34.94	72.79	33.88	76.26	37.67	76.96	0.7239
AHSTGNN	45.00	91.48	48.75	98.10	49.63	104.15	0.6886	26.82	59.32	28.70	60.04	29.94	62.68	0.6894
AGCRN	45.48	96.91	51.34	106.15	50.17	105.11	0.6863	31.71	65.01	31.40	62.46	32.10	62.33	0.7012
ASTGCN	42.99	89.95	46.11	94.92	46.91	96.48	0.6879	26.22	51.34	27.80	53.97	28.36	53.97	0.7515
BigST	36.40	74.93	40.34	81.69	40.06	81.77	0.7058	27.46	52.21	27.68	52.69	30.10	55.65	0.6862
D^{2}STGNN	38.24	85.50	42.73	93.15	43.38	92.77	0.6562	23.87	50.84	25.13	52.76	27.72	57.86	0.7005
DCRNN	28.07	66.12	32.22	72.78	32.30	75.40	0.5178	19.05	41.40	20.75	43.26	21.03	45.05	0.5247
DGCRN	70.21	115.20	69.99	111.77	65.70	108.84	0.5557	34.94	54.46	34.52	57.51	34.30	57.61	0.6126
DLinear	35.64	76.66	38.14	81.03	38.55	82.35	0.5469	21.07	45.83	21.57	46.42	21.95	47.02	0.5036
AMD	30.18	71.65	35.69	77.68	36.10	81.05	0.5374	20.98	49.35	21.12	50.96	21.76	51.45	0.5763
TimeMixer	31.94	72.65	35.28	76.13	36.35	82.51	0.5413	20.16	48.86	20.90	49.63	21.54	51.28	0.5687
DSTAGNN	51.15	101.92	51.31	101.06	51.94	106.63	0.6498	26.51	55.74	30.24	61.42	29.62	58.17	0.6984
GWNet	40.30	82.92	44.67	90.32	44.48	92.55	0.7594	24.44	53.81	25.61	55.35	26.95	56.93	0.6432
STAEformer	42.40	81.45	46.57	90.48	51.09	95.06	0.8170	27.69	54.26	29.17	56.57	31.35	59.88	0.7752
STGCN	49.06	105.08	51.21	109.00	58.78	120.14	0.6894	27.75	59.04	29.28	61.62	32.87	65.92	0.6817
STGODE	39.32	84.50	42.23	88.49	43.46	90.77	0.6569	25.41	54.85	27.28	57.96	27.35	57.86	0.6142
STID	40.72	83.80	41.08	84.58	43.33	88.35	0.6900	23.91	49.31	26.65	53.66	27.83	54.45	0.6606
STNorm	56.19	115.29	49.55	101.51	46.22	91.90	0.7200	31.35	64.35	30.28	61.34	27.39	54.46	0.7185
STTN	46.77	94.76	57.03	109.37	48.09	98.22	0.7309	27.18	57.56	36.18	71.27	32.13	64.53	0.6931
MobiMixer	**22.52**	**34.06**	**24.12**	**40.68**	**28.08**	**52.00**	**0.3884**	**14.49**	**28.37**	**16.22**	**31.37**	**18.51**	**30.42**	**0.3265**
Improvement	+19.77%	+48.49%	+25.14%	+44.11%	+13.07%	+31.03%	+24.99%	+23.94%	+31.47%	+21.83%	+27.48%	+11.98%	+32.48%	+34.05%

Value indicates the best performance, while value indicates the second-best performance. 'Improvement' represents the percentage improvement of the best performance relative to the second-best performance.

TABLE IV
SHORT-TERM PREDICTION PERFORMANCE FOR CALL-IN AND CALL-OUT

Model	CALL-IN							CALL-Out						
	6 Horizons		18 Horizons		24 Horizons		BER	6 Horizons		18 Horizons		24 Horizons		BER
	MAE	RMSE	MAE	RMSE	MAE	RMSE		MAE	RMSE	MAE	RMSE	MAE	RMSE	
ConvLSTM	46.56	92.78	46.89	99.34	45.92	89.21	0.5688	46.40	95.99	52.30	103.07	51.51	98.79	0.5092
MVSTGN	37.01	73.34	34.55	78.59	33.78	72.97	0.7130	35.99	70.79	38.29	79.57	37.69	75.89	0.6891
STDenseNet	42.70	90.74	45.56	86.91	42.24	87.84	0.7015	41.89	94.54	49.6	98.38	49.31	95.29	0.6792
AHSTGNN	34.99	71.32	37.56	72.60	35.23	69.07	0.7165	33.59	75.57	38.43	77.23	39.30	77.96	0.7254
AGCRN	36.33	72.94	36.72	75.63	35.61	70.91	0.7035	35.60	75.37	39.76	82.08	40.29	79.31	0.6766
ASTGCN	31.34	66.33	35.05	70.78	34.48	70.12	0.7562	36.39	73.60	37.23	75.26	39.43	76.85	0.7106
BigST	26.07	49.53	31.16	60.01	30.01	59.36	0.7333	27.40	52.80	33.19	63.19	32.27	63.02	0.7269
D^{2}STGNN	28.07	62.39	33.65	74.59	33.29	73.76	0.7165	30.35	68.95	36.34	78.88	33.85	74.12	0.6269
DCRNN	18.22	41.82	21.95	48.30	22.20	50.72	0.5019	21.65	48.68	25.15	54.68	28.29	57.56	0.5041
DGCRN	54.55	81.64	43.09	73.76	39.09	71.08	0.6128	58.40	92.79	52.68	87.24	47.85	83.43	0.5746
DLinear	25.44	53.80	27.78	59.21	27.79	59.81	0.5412	28.58	58.70	31.34	64.82	31.27	65.01	0.5701
AMD	22.45	50.56	26.18	60.79	26.44	60.88	0.7080	27.89	51.89	32.47	58.23	35.41	65.79	0.6389
TimeMixer	21.13	47.89	24.46	58.09	24.56	58.14	0.6689	25.46	50.10	29.68	56.78	30.48	59.16	0.5598
DSTAGNN	33.07	67.75	37.17	74.45	37.77	74.33	0.6742	33.41	70.10	41.07	80.33	41.94	82.49	0.6523
GWNet	31.41	62.07	33.00	67.82	31.17	66.57	0.7182	33.00	6.65	37.68	78.27	35.22	74.14	0.7172
STAEformer	29.48	56.77	33.76	65.81	37.60	72.30	0.6956	34.80	66.52	38.61	75.47	42.86	81.71	0.6962
STGCN	35.47	71.56	38.79	79.01	44.52	86.85	0.6984	37.85	79.09	40.97	85.95	46.49	92.98	0.7159
STGODE	27.26	57.09	28.60	60.21	28.57	60.38	0.6485	31.02	62.08	34.38	69.20	36.38	74.98	0.6990
STID	26.74	52.23	29.02	58.38	29.43	59.83	0.6874	33.78	66.83	34.12	69.14	37.68	76.55	0.6797
STNorm	35.04	71.61	37.12	73.66	35.76	68.78	0.7223	37.17	75.03	39.96	78.92	33.24	67.93	0.6998
STTN	29.14	59.23	43.16	81.68	34.05	66.69	0.7431	30.98	63.51	38.77	79.23	36.74	73.53	0.6899
MobiMixer	**14.86**	**23.71**	**17.36**	**29.81**	**19.24**	**39.01**	**0.3691**	**15.66**	**27.30**	**19.94**	**35.06**	**23.93**	**49.48**	**0.3555**
Improvement	+18.44%	+43.30%	+20.91%	+38.28%	+13.33%	+23.09%	+26.45%	+27.67%	+43.92%	+20.72%	+35.88	+15.41%	+14.04%	+29.47%

- **w/o TD** means that we remove the temporal decomposition mechanism of MobiMixer.
- **w/o MT** means that we remove the multi-scale timing modeling mechanism of MobiMixer.

The experimental results are shown in Figs. 9 and 10, and we find that every component of MobiMixer is effective. "w/o pro" achieves higher prediction errors, indicating that encoding various temporal priors is beneficial for the model to accurately capture mobile traffic patterns. For example, integrating day-of-the-week information into the model helps capture periodic dynamics. On the other hand, "w/o nemb" achieves higher prediction errors because node embeddings can adaptively capture the fine-scale spatial features of each node. "w/o HIM" shows poorer predictive performance, highlighting the necessity of extracting spatial features. This module enhances predictions by extracting high-level features shared

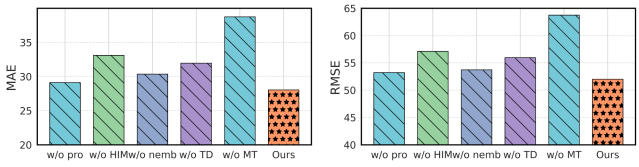

Fig. 9. Ablation experiments on the Milan SMS-IN dataset.

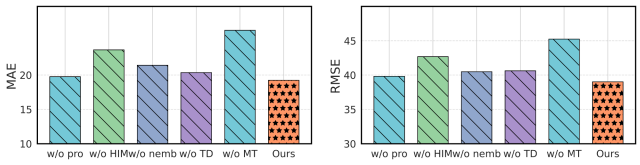

Fig. 10. Ablation experiments on the Milan CALL-IN dataset.

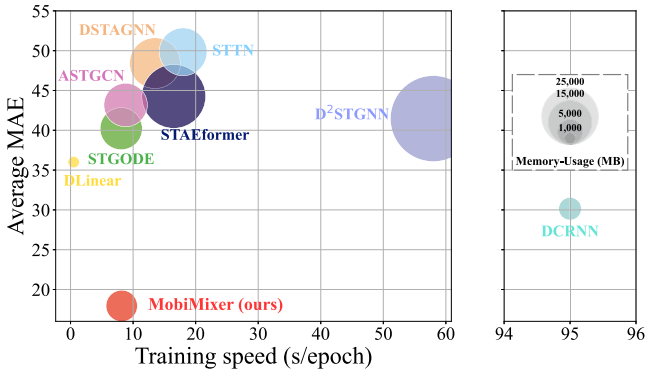

Fig. 11. Efficiency comparison on the Milan SMS-IN dataset.

TABLE V
EFFICIENCY COMPARISON ON CALL-IN DATASET

Model	Performance RMSE	Training speed (s/epoch)	Memory footprint (MB)
AGCRN	105.11	12.15	4,686
ASTGCN	96.48	8.81	5,906
D²STGNN	92.77	58.12	23,078
DSTAGNN	106.63	13.54	8,066
STAEformer	95.06	16.51	12,738
STGODE	60.38	90/77	5,560
STTN	98.22	18.03	7,140
DCRNN	75.40	95.63	4,604
MobiMixer	**52.00**	8.18	3,084

We report the RMSE of 24 time step.

among nodes. Additionally, "w/o TD" results in larger prediction errors, as decomposing traffic sequences into long-term and short-term patterns and modeling them separately helps the model capture comprehensive temporal dynamic. The mixture of multi-granularity temporal features further enhances modeling accuracy.

E. Efficiency Comparison (Q.3)

We compare the computational complexity of the models and report the training time per epoch and memory usage of several advanced models on the SMS-IN dataset. We run each model on the same device and environment. The results are shown in Fig. 11 and Table V. We find that Transformer-based models require more memory due to their complex model parameters, which also introduce a quadratic computational burden. The top-performing baseline DCRNN uses a recurrent structure, sequentially predicting future values, consuming the most computation time. Our model eschews graph convolution operators and primarily utilizes MLP as the underlying architecture. Therefore, while maintaining high performance, MobiMixer achieves the lowest computational burden. Compared to DCRNN, the training speed also improves by 10.69X, and memory usage decreased by 33.01%.

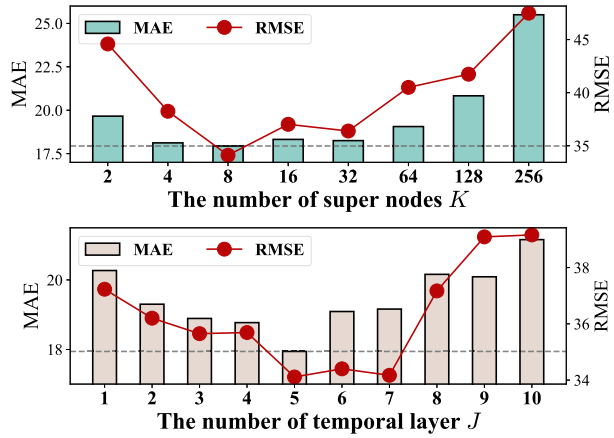

Fig. 12. Parameter sensitivity experiment on the Milan CALL-IN dataset.

F. Hyperparameter Experiment (Q.4)

Using CALL-IN dataset as example, we evaluate the impact of two important parameters on the model performance: the number of super nodes K and the number of layers of the multi-scale temporal mixing module J. As shown in Fig. 12, we find that the best predictive performance was achieved when K is equal to 8. When K exceeds this value, an excessive number of super nodes hinder the model's ability to effectively extract shared spatial features. When the number of temporal layers J is set to 5, the model exhibits good predictive performance. A smaller value may result in the model being unable to effectively fit complex mobile traffic patterns, leading to underfitting. Conversely, a larger value may cause the parameter scale to become excessively large, resulting in overfitting.

G. Scalability in Large-Scale Mobile Networks (Q.5)

We evaluate the scalability performance of the model on the Trentino dataset, which consists of 11,466 nodes in a large-scale mobile network. Notably, some models, such as D ^{2} STGNN, are absent from the results due to exceeding memory limits. The results are presented in Table VI. STID employs several embedding techniques to capture mobile traffic patterns, resulting in excellent predictive performance. GWNet achieves high accuracy by utilizing graph convolutional networks to learn spatiotemporal correlations, but introduces significant computational complexity. In contrast, our model demonstrates outstanding predictive performance on large-scale networks, with a maximum performance improvement of 54.5%.

TABLE VI
PERFORMANCE EVALUATION ON TRENTINO DATASET WITH TEN THOUSAND NODES

Model	SMS						CALL						Internet					
	6 Horizons		18 Horizons		24 Horizons		6 Horizons		18 Horizons		24 Horizons		6 Horizons		18 Horizons		24 Horizons	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
BigST	6.74	21.67	7.12	22.20	7.56	24.17	3.00	13.20	3.30	15.66	3.62	16.93	20.47	60.38	23.39	69.77	24.69	74.03
DLinear	7.19	24.23	7.38	24.76	7.25	24.34	3.53	16.57	3.71	18.07	3.76	18.01	17.42	59.13	18.30	62.55	19.68	66.03
GWNet	6.45	20.22	6.69	20.63	6.97	22.59	3.03	12.36	3.39	15.61	3.43	15.84	18.54	60.66	20.15	66.69	23.09	72.36
STGCN	6.99	28.28	7.16	27.85	7.48	28.57	4.08	20.25	4.21	21.40	4.62	22.46	30.70	113.32	32.26	116.43	34.79	124.06
STGODE	8.08	25.41	8.67	26.44	8.86	26.58	4.51	32.21	5.20	35.08	5.19	32.77	23.98	80.34	28.07	90.91	28.92	95.28
STID	6.28	20.09	6.84	21.58	7.17	22.74	2.92	12.47	3.12	15.12	3.39	16.21	16.97	54.17	18.51	62.06	19.99	65.25
STNorm	8.29	28.76	8.25	26.67	8.14	26.28	3.78	15.70	3.81	14.87	4.09	16.48	24.57	96.42	24.81	93.89	23.17	73.96
MobiMixer	**4.83**	**19.99**	**4.89**	**19.51**	**3.17**	**13.26**	**1.91**	**12.08**	**1.93**	**11.87**	**1.61**	**9.68**	**11.67**	**47.83**	**12.76**	**51.85**	**9.56**	**44.46**

Some models do not appear because they run out of memory.

TABLE VII
PERFORMANCE COMPARISON OF PEAK TRAFFIC PREDICTION

Model	SMS-IN		CALL-IN	
	MAE	RMSE	MAE	RMSE
DCRNN	32.71	76.59	33.62	48.84
MobiMixer	**25.49**	**38.10**	**19.24**	**27.33**

TABLE VIII
PERFORMANCE COMPARISON IN WEEKEND

Model	SMS-IN		CALL-IN	
	MAE	RMSE	MAE	RMSE
DCRNN	56.71	104.34	36.21	72.41
MobiMixer	**41.73**	**73.61**	**25.98**	**49.96**

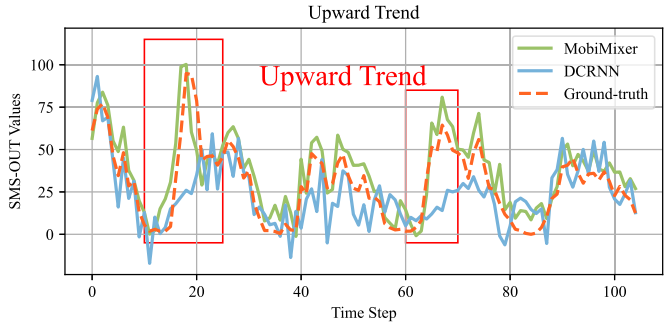

Fig. 13. Visualization of peak prediction results.

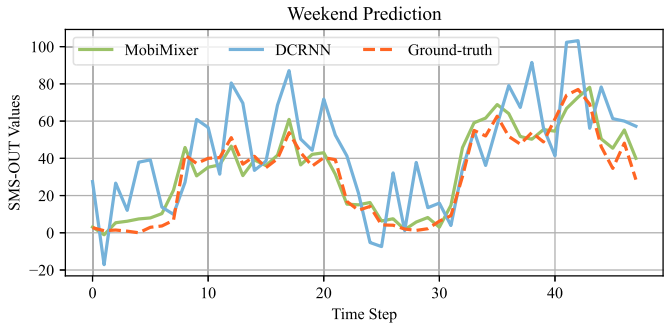

Fig. 14. Visualization of weekend prediction results.

H. Case Study (Q.6)

1) Peak Traffic Prediction: Peak traffic situations may be of greater concern to mobile management personnel in order to timely formulate network prevention policies. We compare the prediction performance of MobiMixer and DCRNN for peak traffic. Taking the Milan SMS-IN dataset as an example, we select the top 90% of traffic values for each node in the test dataset, and the predictive performance is shown in Table VII.

We further visualize the prediction results of MobiMixer and the top-performing baseline DCRNN in Fig. 13. We can find that MobiMixer can effectively predict future mobile traffic peaks. This is attributed to the multi-scale temporal modeling capability of MobiMixer, which can efficiently capture periodic peaks. Accurately predicting peaks can better assist network managers in formulating traffic management policies.

2) Weekend Traffic Prediction: We further compare the prediction performance on weekends. During weekends, due to travel or tourism activities, the mobile traffic patterns may exhibit non-stationary distributions, posing challenges for accurate prediction. The results, as shown in Table VIII and Fig. 14, indicate that MobiMixer still achieves outstanding predictive performance. This can be attributed to the decoupled temporal modeling, which enables a better fit for non-stationary mobile patterns. The multi-scale temporal modeling can sensitively perceive trend changes, while the periodic long-term patterns assist in making accurate predictions during weekend scenarios.

I. Robustness of Data Time Interval (Q.7)

We evaluate the robustness of the models under various time interval. In datasets with high sampling frequencies, such as the Beijing dataset (5 minutes) and the Shanghai dataset (10 minutes), mobile traffic data often contain a significant number of zero values, which complicates the models' ability to learn accurate spatiotemporal correlations from sparse data. The results, presented in Table IX, indicate that STGODE achieves relatively good predictive performance, likely due to its incorporation of partial differential equation techniques, which are effective in capturing high-level spatiotemporal correlations. However, MobileMixer demonstrates competitive predictive performance as well, thanks to its multi-scale spatiotemporal modeling capability, which helps mitigate the adverse effects of data sparsity.

TABLE IX
PREDICTION PERFORMANCE FOR SHANGHAI AND BEIJING DATASETS

Model	Shanghai (10 mins)							Beijing (5 mins)						
	6 Horizons		18 Horizons		24 Horizons			6 Horizons		18 Horizons		24 Horizons		
	MAE	RMSE	MAE	RMSE	MAE	RMSE	BER	MAE	RMSE	MAE	RMSE	MAE	RMSE	BER
ConvLSTM	0.2145	0.5798	0.2151	0.6011	0.2454	0.5923	0.0057	0.1889	0.5801	0.1899	0.5888	0.1918	0.5879	0.0051
AGCRN	-	-	-	-	-	-	-	0.1907	0.5737	0.1936	0.5819	0.1925	0.5815	0.0068
ASTGCN	-	-	-	-	-	-	-	0.1987	0.5914	0.2021	0.6024	0.2031	0.6052	0.0071
BigST	0.2153	0.5723	0.2160	0.5910	0.2143	0.5836	0.0068	0.1879	0.5664	0.1900	0.5753	0.1910	0.5795	0.0058
D^2STGNN	-	-	-	-	-	-	-	0.1991	0.5792	0.2050	0.5922	0.2036	0.5878	0.0062
DCRNN	-	-	-	-	-	-	-	0.2616	0.6900	0.2837	0.7003	0.2328	0.6331	0.0059
DGCRN	-	-	-	-	-	-	-	0.1930	0.5960	0.2001	0.5952	0.2136	0.6043	0.0056
Dlinear	0.2220	0.5769	0.2222	0.5946	0.2204	0.5888	0.0059	0.2151	0.6373	0.2140	0.6388	0.2139	0.6367	0.0068
AMD	0.2196	0.5931	0.2204	0.5916	0.2108	0.5928	0.0071	0.1935	0.5813	0.1943	0.5831	0.1950	0.5846	0.0068
TimeMixer	0.2188	0.5864	0.2189	0.5946	0.2193	0.5922	0.0066	0.1915	0.5731	0.1921	0.5763	0.1938	0.5756	0.0062
GWNET	0.2147	0.5812	0.2142	0.5994	0.2141	0.5927	0.0059	0.1904	0.5690	0.1975	0.5715	0.1983	0.5781	0.0061
STAEformer	-	-	-	-	-	-	-	0.1880	0.5679	0.1891	0.5733	0.1908	0.5786	0.0063
STGCN	0.2175	0.5958	0.2165	0.6155	0.2174	0.5847	0.0063	0.1907	0.5666	0.1941	0.5765	0.1948	0.5798	0.0073
STGODE	0.2150	0.5722	0.2165	0.5896	0.2146	0.5853	0.0064	0.1908	0.5661	0.1921	0.5703	0.2031	0.5700	0.0068
STID	0.2159	0.5807	0.2188	0.5976	0.2164	0.5918	0.0063	0.1892	0.5669	0.1898	0.5680	0.1903	0.5799	0.0064
STNorm	0.2198	0.5851	0.2202	0.6013	0.2198	0.5970	0.0072	0.2013	0.5913	0.1956	0.5828	0.1897	0.5895	0.0070
STTN	-	-	-	-	-	-	-	0.1894	0.5671	0.1911	0.5710	0.1888	0.5754	0.0064
MobiMixer	**0.1581**	**0.4493**	**0.1587**	**0.4754**	**0.1556**	**0.4761**	**0.0038**	**0.1355**	**0.4678**	**0.1408**	**0.4818**	**0.1455**	**0.4761**	**0.0047**

Some models marked with "-" are missing due to out-of-memory issue.

VI. DISCUSSION AND FUTURE WORK

In this section, we outline potential future research directions. Our experiments utilized mobile network datasets from four major cities. Moving forward, our goal is to collaborate with mobile network operators to release a comprehensive long-term data set that includes data from additional cities. Furthermore, as the number of mobile base stations continues to grow, mobile networks will exhibit increasingly dynamic characteristics. Investigating the challenges associated with making accurate predictions in these dynamic environments presents a valuable avenue for future research.

VII. CONCLUSION

In this paper, we propose a lightweight and efficient mobile traffic prediction model, MobiMixer. The model captures multi-scale information from spatial and temporal dimensions separately. Specifically, we design a hierarchical interaction component to capture macro-level shared features and integrate node embeddings to capture fine-grained spatial features. Subsequently, we introduce a multi-scale temporal modeling module that utilizes decoupled multi-scale information to enhance the model's handling of non-stationary in mobile traffic data. We evaluate the effectiveness on four mobile traffic datasets and MobiMixer achieves competitive performance while maintaining extremely low computational complexity.

REFERENCES

[1] R. Mavi, R. Singh, and R. Grover, "On time demand traffic estimation based on DBN with horse herd optimization for next generation wireless network," *Expert Syst. Appl.*, vol. 246, 2024, Art. no. 123189.

[2] Z. Feng, L. Ji, Q. Zhang, and W. Li, "A supply-demand approach for traffic-oriented wireless resource virtualization with testbed analysis," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 6077–6090, Sep. 2017.

[3] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Aug. 2018.

[4] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[5] H. Hu, J. Zhang, Y. Jiang, Z. Li, Q. Chen, and J. Zhang, "Computation offloading analysis in clustered fog radio access networks with repulsion," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10804–10819, Oct. 2021.

[6] J. Gong, Y. Liu, T. Li, J. Ding, Z. Wang, and D. Jin, "STTF: A spatiotemporal transformer framework for multi-task mobile network prediction," *IEEE Trans. Mobile Comput.*, vol. 24, no. 5, pp. 4072–4085, May 2025.

[7] L. Wang et al., "Hyper-parameter optimization for wireless network traffic prediction models with a novel meta-learning framework," 2024, *arXiv:2409.14535*.

[8] J. Guo, C. Tang, J. Lu, A. Zou, and W. Yang, "Wvett-net: A novel hybrid prediction model for wireless network traffic based on variational mode decomposition," *Electronics*, vol. 13, no. 16, 2024, Art. no. 3109.

[9] A. Roy, K. K. Roy, A. Ahsan Ali, M. A. Amin, and A. M. Rahman, "Sst-gnn: Simplified spatio-temporal traffic forecasting model using graph neural network," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2021, pp. 90–102.

[10] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, 2022, Art. no. 117921.

[11] H. Nan, X. Zhu, and J. Ma, "MSTL-GLTP: A global–local decomposition and prediction framework for wireless traffic," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5024–5034, Mar. 2022.

[12] J. Gong et al., "Empowering spatial knowledge graph for mobile traffic prediction," in *Proc. 31st ACM Int. Conf. Adv. Geographic Inf. Syst.*, 2023, pp. 1–11.

[13] Y. Yao, B. Gu, Z. Su, and M. Guizani, "MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2837–2849, May 2023.

[14] K. He, X. Chen, Q. Wu, S. Yu, and Z. Zhou, "Graph attention spatial-temporal network with collaborative global-local learning for citywide mobile traffic prediction," *IEEE Trans. Mobile Comput.*, vol. 21, no. 4, pp. 1244–1256, Apr. 2022.

[15] Y. Fang, S. Ergüt, and P. Patras, "SDGNet: A handover-aware spatiotemporal graph neural network for mobile traffic forecasting," *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 582–586, Mar. 2022.

[16] F. Sun et al., "Mobile data traffic prediction by exploiting time-evolving user mobility patterns," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4456–4470, Dec. 2021.

[17] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in *Proc. IEEE IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–5.

[18] N. Keriven, "Not too little, not too much: A theoretical analysis of graph (over) smoothing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 2268–2281.

[19] A. Bassolas et al., "Hierarchical organization of urban mobility and its connection with city livability," *Nature Commun.*, vol. 10, no. 1, 2019, Art. no. 4817.

[20] G. Barlacchi et al., "A multi-source dataset of urban life in the city of Milan and the Province of Trentino," *Sci. Data*, vol. 2, no. 1, pp. 1–15, 2015.

[21] T. Rakthanmanon et al., "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, doi: 10.1145/2339530.2339576.

[22] X. Chen, J. Wang, and K. Xie, "TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning," 2021, *arXiv:2106.06273*.

[23] Z. Shao, Z. Zhang, F. Wang, and Y. Xu, "Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 1567–1577.

[24] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.

[25] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 22419–22430, 2021, pp. 22419–22430.

[26] S. Gowrishankar, "A time series modeling and prediction of wireless network traffic," *Comput. Sci. Telecommun.*, vol. 2, pp. 40–52, 2008.

[27] G. Lin, A. Lin, and D. Gu, "Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient," *Inf. Sci.*, vol. 608, pp. 517–531, 2022.

[28] X. Chen, Y. Liu, and J. Zhang, "Traffic prediction for Internet of Things through support vector regression model," *Internet Technol. Lett.*, vol. 5, no. 3, 2022, Art. no. e336.

[29] M. Marvi, A. Aijaz, and M. Khurram, "On the use of ON/OFF traffic models for spatio-temporal analysis of wireless networks," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1219–1222, Jul. 2019.

[30] Y. Cai, P. Cheng, M. Ding, Y. Chen, Y. Li, and B. Vucetic, "Spatiotemporal gaussian process Kalman filter for mobile traffic prediction," in *Proc. IEEE 31st Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2020, pp. 1–6.

[31] Q. T. Tran, L. Hao, and Q. K. Trinh, "Cellular network traffic prediction using exponential smoothing methods," *J. Inf. Commun. Technol.*, vol. 18, no. 1, pp. 1–18, 2019.

[32] Z. Zhang, F. Li, X. Chu, Y. Fang, and J. Zhang, "dmTP: A deep meta-learning based framework for mobile traffic prediction," *IEEE Wireless Commun.*, vol. 28, no. 5, pp. 110–117, Oct. 2021.

[33] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, "DeepTP: An end-to-end neural network for mobile cellular traffic prediction," *IEEE Netw.*, vol. 32, no. 6, pp. 108–115, Nov./Dec. 2018.

[34] L. Zhang et al., "LNTP: An end-to-end online prediction model for network traffic," *IEEE Netw.*, vol. 35, no. 1, pp. 226–233, Jan./Feb. 2021.

[35] X. Wang et al., "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, Sep. 2018.

[36] B. Gu, J. Zhan, S. Gong, W. Liu, Z. Su, and M. Guizani, "A spatial-temporal transformer network for city-level cellular traffic analysis and prediction," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9412–9423, Dec. 2023.

[37] J. Gong et al., "KGDA: A knowledge graph-based decomposition approach for cellular traffic prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 15, pp. 1–22, 2024.

[38] Y. Hu, P. Liu, P. Zhu, D. Cheng, and T. Dai, "Adaptive multi-scale decomposition framework for time series forecasting," 2024, *arXiv:2406.03751*.

[39] S. Wang et al., "Timemixer: Decomposable multiscale mixing for time series forecasting," 2024, *arXiv:2405.14616*.

[40] G. Jin, Y. Liang, Y. Fang, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," 2023, *arXiv:2303.14483*.

[41] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[42] B. Wang et al., "Knowledge expansion and consolidation for continual traffic prediction with expanding graphs," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7190–7201, Jul. 2023.

[43] B. Wang et al., "Pattern expansion and consolidation on evolving graphs for continual traffic prediction," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2023, pp. 2223–2232.

[44] B. Wang et al., "Towards dynamic spatial-temporal graph learning: A decoupled perspective," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 9089–9097.

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

[46] Z. Shao et al., "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," 2022, *arXiv:2206.09112*.

[47] Y. Fang et al., "When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks," in *Proc. IEEE 39th Int. Conf. Data Eng.*, 2023, pp. 517–529.

[48] W. Zhang and K. Zheng, "Disentangled traffic forecasting via efficient graph neural network," *J. Phys.: Conf. Ser.*, IOP Publishing, 2024, Art. no. 012036.

[49] M. C. Mozer, "Induction of multiscale temporal structure," in *Proc. 5th Int. Conf. Neural Inf. Process. Syst.*, 1991, pp. 275–282.

[50] Y. Hu, P. Liu, P. Zhu, D. Cheng, and T. Dai, "Adaptive multi-scale decomposition framework for time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 17359–17367.

[51] Y. Li, A. Zhou, X. Ma, and S. Wang, "Profit-aware edge server placement," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 55–67, Jan. 2022.

[52] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Softw.: Pract. Experience*, vol. 50, no. 5, pp. 489–502, 2020.

[53] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 939–951, Mar. 2021.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[55] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, Aug. 2018.

[56] Q. Liu, J. Li, and Z. Lu, "ST-tran: Spatial-temporal transformer for cellular traffic prediction," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3325–3329, Oct. 2021.

[57] X. Wang et al., "Adaptive hybrid spatial-temporal graph neural network for cellular traffic prediction," 2023. [Online]. Available: https://arxiv.org/abs/2303.00498

[58] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17804–17815.

[59] 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, no. 01, 2019, pp. 922–929.

[60] J. Han, W. Zhang, H. Liu, T. Tao, N. Tan, and H. Xiong, "BigST: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks," in *Proc. VLDB Endowment*, 2024, vol. 17, no. 5, pp. 1081–1090.

[61] F. Li et al., "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discov. from Data*, vol. 17, no. 1, pp. 1–21, 2023.

[62] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting," 2022. [Online]. Available: https://arxiv.org/abs/2205.13504

[63] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 11906–11917.

[64] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," 2019, *arXiv: 1906.00121*.

[65] H. Liu et al., "Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 4125–4129.

[66] 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.*, 2018, pp. 3634–3640.

[67] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ode networks for traffic flow forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2021, pp. 364–373.

[68] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 4454–4458.

[69] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "ST-norm: Spatial and temporal normalization for multi-variate time series forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 269–278.

[70] Z. He, C.-Y. Chow, and J.-D. Zhang, "STNN: A spatio-temporal neural network for traffic predictions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7642–7651, Dec. 2021.

Jiaming Ma is currently working toward the master's degree in the School of Artificial Intelligence and Data Science, University of Science and Technology of China (USTC). His current research focuses on mobile traffic data mining, human mobility modeling, and spatio-temporal prediction. He has published papers in top conferences such as ACM KDD, VLDB, and IJCAI.

Binwu Wang received the PhD degree from USTC, in 2024. He is now an associate researcher with the University of Science and Technology of China (USTC). He has published over twenty research papers in top journals and conferences such as *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Transactions on Vehicular Technology*, ICLR, SIGKDD, and AAAI. His research interests mainly include data mining, machine learning, and continuous learning.

Pengkun Wang (Member, IEEE) received the PhD degree from USTC, in 2023. He is now an associate researcher with the University of Science and Technology of China (USTC), under the supervision of Professor Qi Liu and Yang Wang. His research interest mainly includes open environment machine learning, spatio-temporal data mining, and generalized AI for Science.

Zhengyang Zhou (Member, IEEE) received the PhD degree from USTC, in 2023. He is now an associate researcher with Suzhou Institute for Advanced Research, University of Science and Technology of China (USTC). He has published over twenty papers in top conferences and journals such as *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Knowledge and Data Engineering*, KDD, ICLR, WWW, AAAI, NeurIPS, and ICDE. His main research interests include human-centered urban computing and mobile data mining.

Yudong Zhang (Graduate Student Member, IEEE) is currently working toward the PhD degree in the School of Artificial Intelligence and Data Science, University of Science and Technology of China (USTC). He has published over twenty research papers in top journals and conferences such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Vehicular Technology*, ICLR, SIGKDD, AAAI, WSDM, and ICDM. His current research interests include mobile data mining and urban computing.

Xu Wang received the bachelor's degree in automation from North Eastern University, in 2017 and the PhD degree from USTC, in 2023, under the supervision of Professor Zheng-Jun Zha and Yang Wang. He is now an associate researcher with the University of Science and Technology of China (USTC). His research interests mainly encompass spatio-temporal data mining, time series analysis, and the application of AI in scientific research.

Yang Wang (Senior Member, IEEE) received the PhD degree from USTC, in 2007. He is now an associate professor with the School of Computer Science and Technology, School of Software Engineering, and School of Artificial Intelligence and Data Science at the University of Science and Technology of China (USTC). Since then, he keeps working at USTC till now as a postdoc and an Associate Professor successively. Meanwhile, he also serves as the vice dean of the School of Software Engineering of USTC. His research interests mainly include mobile (sensor) networks, distributed systems, data mining, and machine learning, and he is also interested in all kinds of applications of AI and data mining technologies, especially in urban computing and AI4Science. His work has been published in top-tier conferences and journals like ICLR, NeurIPS, ICML, KDD, AAAI, WWW, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Mobile Computing*, and *IEEE Transactions on Pattern Analysis and Machine Intelligence*, with over fifty papers as the first author or corresponding author.