Rethinking Urban Mobility Prediction: A Super-Multivariate Time Series Forecasting Approach

Jinguo Cheng, Ke Li, Yuxuan Liang, *Member, IEEE,* Lijun Sun, *Senior Member, IEEE,* Junchi Yan, *Senior Member, IEEE,* and Yuankai Wu✉, *Senior Member, IEEE*

***Abstract*—Long-term urban mobility predictions play a crucial role in the effective management of urban facilities and services. Conventionally, urban mobility data has been structured as spatiotemporal videos, treating longitude and latitude grids as fundamental pixels. Consequently, video prediction methods, relying on Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), have been instrumental in this domain. In our research, we introduce a fresh perspective on urban mobility prediction. Instead of oversimplifying urban mobility data as traditional video data, we regard it as a complex multivariate time series. This perspective involves treating the time-varying values of each grid in each channel as individual time series, necessitating a thorough examination of temporal dynamics, cross-variable correlations, and frequency-domain insights for precise and reliable predictions. To address this challenge, we present the Super-Multivariate Urban Mobility Transformer (SUMformer), which utilizes a specially designed attention mechanism to calculate temporal and cross-variable correlations and reduce computational costs stemming from a large number of time series. SUMformer also employs low-frequency filters to extract essential information for long-term predictions. Furthermore, SUMformer is structured with a temporal patch merge mechanism, forming a hierarchical framework that enables the capture of multi-scale correlations. Consequently, it excels in urban mobility pattern modeling and long-term prediction, outperforming current state-of-the-art methods across five real-world datasets. The code is available at: https://github.com/Chengyui/SUMformer.**

***Index Terms*—Urban mobility prediction, Multivariate time series forecasting, Efficient attention mechanism.**

I. Introduction

In the realm of urban mobility computing, a diverse array of spatiotemporal data exists, encompassing different organizational structures, scales, and modes. These data are characterized by their dynamic nature, evolving continuously across both time and space. Among the prominent forms of urban dynamic spatiotemporal data are point-based [1], graph-based [2], and grid-based data [3]. Grid-based data, in particular, involve the division of urban areas into grids based on latitude and longitude coordinates. Each grid contains a wealth of attributes for the current spatiotemporal slot, including latitude and longitude coordinate ranges, points of interest, cumulative in/out vehicle counts (in/out traffic flow) [4], and various other relevant information [5], [6]. Forecasting grid-based data is crucial as it serves as a foundational framework for spatial analysis and modeling, enabling the assessment, prediction, and management of various urban phenomena, spanning from congestion hotspots to land use dynamics.

Traditionally, the practice of structuring grid-based mobility data in a video format (T, C, H, W) has naturally emerged due to its alignment with the inherent characteristics of the data. Here, T corresponds to the number of time points, C represents the number of attributes, and H and W indicate the latitude and longitude dimensions of the urban area. In recent years, there have been remarkable advancements in deep video prediction techniques, with Convolutional Neural Networks (CNNs) [7] and more recent Vision Transformers (ViTs) [8], [9] serving as their core components. This has resulted in the extensive utilization of deep video prediction methods for the prediction of grid-based mobility data. Moreover, it is important to note that grid-based mobility data such as TaxiBJ also serves as a common dataset for evaluating video prediction models [10]–[13].

Both CNN and ViT-style video prediction frameworks aim to capture cross-channel and spatial correlations by treating small patches of the image as a unified entity. In CNNs, this is achieved through 2D/3D convolution filters [10], [14]–[17], while in ViTs, the input data is divided into non-overlapping 2D/3D patches [11], [18], [19], which are then used as input tokens for the Transformer model. This approach is particularly apt when dealing with images and video data. In the realm of visual data, adjacent pixels' RGB values merge into a cohesive whole within specific regions. This holistic perspective facilitates the extraction of diverse features, objects, and semantic insights. Merging RGB channels with image regions enriches the analysis, empowering deep learning models to capture meaningful features from the visual data [20], [21]. However, this approach may not be well-suited for urban mobility data where each channel (analogous to RGB channels) stores a specific attribute for a particular region, and the attributes of each region carry unique semantic meanings. Moreover, these attributes may exhibit complex spatiotemporal correlations that should not be hastily dismissed. Adopting conventional CNN and ViT methods to them without thoughtful consideration could potentially disrupt and neglect

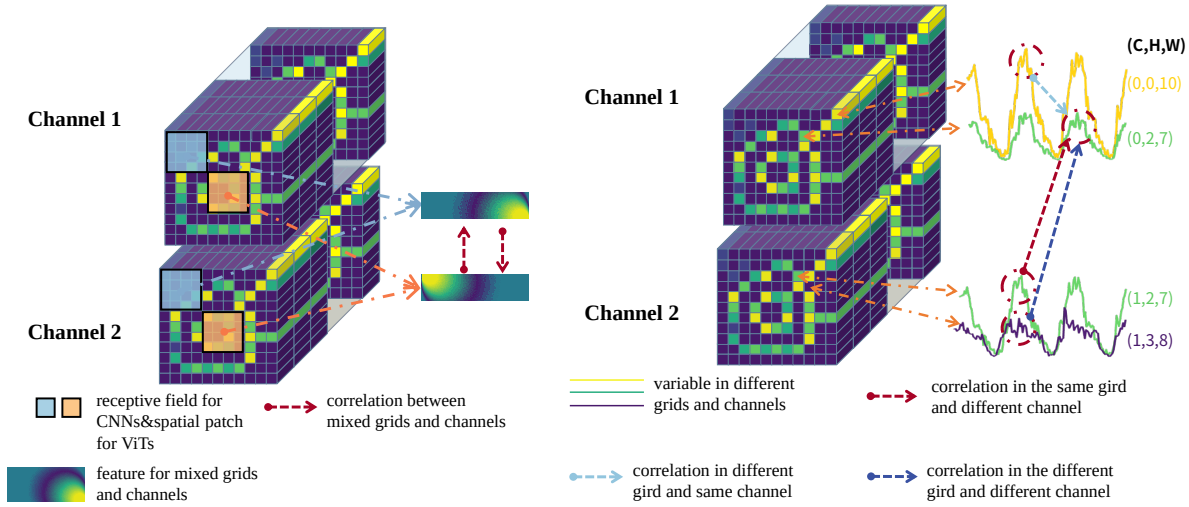
Jinguo Cheng, Ke Li and Yuankai Wu are all with the Department of Computer Science, Sichuan University, Chengdu 610065, China.

Yuxuan Liang is with INTR Thrust & DSA Thrust, The Hong Kong University of Science and Technology (GuangZhou).

Lijun Sun is with the Department of Civil Engineering, McGill University, Montreal, Quebec H3A 0C3, Canada.

Junchi Yan is with e Department of Computer Science and Engineering, and the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China.

✉Yuankai Wu is the corresponding author. Email: wuyk0@scu.edu.cn.



(a) Conventional Embedding Approaches in CNNs and ViTs. (b) A Super-Multivariate Perspective on Grid-Based Urban Mobility Data.

Fig. 1. Illustration depicting distinct perspectives on grid-based urban mobility data: (a) The image patch and/or convolution reception field mix both channels and variables. To simplify the illustration, we only presented 2D convolution and 2D patch partition; (b) Each variable within each spatial grid and across all channels is treated as an independent entity. We explore three distinct types of correlations to generate the embeddings.

significant cross-channel and spatial correlations [22].

In this paper, ***we contend that grid-based urban mobility data should be treated as a super-multivariate time series rather than as video data***. "Super-multivariate time series" refers to a time series characterized by a substantial number of variables and long-term temporal observations. We compare the video and super-multivariate time series perspectives of grid-based urban mobility data in Fig.1. In Fig.1(a), both CNNs and ViTs employ a shared approach, wherein they tend to mix the adjacent channels and grid variable. However, this perspective might overlook some crucial correlations. For example, consider a grid cell with many working offices surrounded by grids with numerous entertainment venues. The outflow from this working grid during the evening commute might be highly correlated with the inflow to a distant residential grid. Furthermore, the grids surrounding the corresponding residential grid may also exhibit significantly different characteristics compared to the residential grid. If we simply combine the inflow and outflow of these two grids and their surrounding grids into patches, it would likely lead to the oversight of the correlation related to the evening commute between the distant working and residential grids. In Fig.1(b), we present our perspective on urban mobility data. Instead of mixing channels and grid data into compact patches, we treat each channel's time series beneath every pixel as an independent entity. This approach allows us to directly utilize correlations within individual time series, both within the same grid and across different channels, as well as across different grids and channels, to generate embeddings.

Additionally, our focus is on addressing the long-term prediction of urban mobility data, in contrast to the prevalent practice found in standard video prediction paradigms, where typically only a few steps ahead are predicted (e.g., 4 steps, as seen in most literature [10]–[13]). Our emphasis on long-term prediction stems from its essential role in many urban management scenarios [23]. Long-term predictions offer management personnel the lead time needed for effective preparation and planning. This extended forecasting horizon is critical for enabling proactive decision-making, ensuring that administrators are well-prepared to address future challenges and opportunities [24], [25].

To this end, we introduce a novel **Super-Mutlviarate Urban Mobility Transformer (SUMformer)** for this task. We begin by converting the video data with dimensions (T, C, H, W) into a super-multivariate time series with $C \times H \times W$ variables. Next, we aggregate the time steps along the temporal dimension, organizing them into subseries-level patches for subsequent temporal, variable, and frequency (TVF) blocks of SUMformer. In the temporal dimension, SUMformer offers two options for capturing the temporal relationships between patches: one relies on a pure multilayer perceptron (MLP), while the other is based on self-attention. In the variable dimensions, given a multitude of time series in grid-based mobility data, our specific design incorporates an efficient self-attention mechanism inspired by [26]–[28] to compute cross-variable attention. Notably, this design achieves a computational time complexity that scales linearly with respect to the number of variables involved. Lastly, in the frequency domain, recognizing that low-frequency periodic information holds more long-term predictive information, SUMformer employs Fourier low-frequency filters to process the features. We organize the hierarchical TVF blocks together using the patch merge approach of the Swin Transformer [29], enabling SUMformer to capture multi-scale spatiotemporal and cross-variable correlations. By integrating these carefully-designed components, SUMformer achieves state-of-the-art (SOTA) performance in long-term urban mobility prediction.

Our contributions to the long-term urban mobility prediction challenge using SUMformer are as follows:

- We present a novel super-multivariate perspective on grid-based urban mobility data. Through this approach, we are able to utilize general multivariate time series forecasting

models to achieve long-term urban mobility predictions.
- We present the SUMformer: a Transformer model designed to leverage temporal, frequency, and cross-variable correlations for urban mobility forecasting. Notably, it stands out as one of the few Transformer models that explicitly taps into and harnesses cross-variable correlations across every channel and grid for urban mobility prediction.
- Experiments (detailed in Section IV) demonstrate that SUMformer surpasses state-of-the-art methods across five real-world datasets. We emphasize the significance of the super-multivariate perspective, explicit cross-variable correlation modeling, and frequency information for achieving optimal performance.

The remainder of this paper is organized as follows: Section II offers an overview of relevant works. Section III details our proposed architecture and its variants. Subsequently, Section IV presents a thorough comparison with existing methods, including detailed ablation studies. Finally, Section V concludes the paper.

II. Related Work

A. Urban Mobility Prediction as Video Prediction

Urban mobility prediction has garnered attention in machine learning recently. Initial research largely focused on CNN-based methodologies [30]–[35], stemming from their success in image processing, as demonstrated by [4]. While models like SimVP [10] predominantly utilize CNNs, newer works argue against their efficiency in capturing global spatial dependencies [22]. Such concerns led researchers to introduce enhancements like the ConvPlus structure in DeepSTN+ to address CNN's limitations in handling long-range spatial dependencies [30]. Meanwhile, the emergence of the Transformer [36] in NLP has influenced computer vision studies. ViT [37], for instance, adapted Transformer techniques for visual tasks. Its patch-based processing approach inspired other models, such as the MLP-Mixer [38], which segments images into patches and processes them using a standalone MLP architecture. These patch-based strategies have also been adopted in spatial-temporal forecasting [11], [39], including the application of large models like Pangu-Weather [40] in weather forecasting. Urban mobility datasets, notably TaxiBJ [4], serve as common benchmarks for video prediction algorithms. Many frameworks are evaluated using these urban mobility datasets, including [10], [11], [14], [39]. However, these studies typically prioritize general video prediction, often focusing on short-term forecasts. This contrasts with the urban management requirement for predicting mobility trends days ahead.

B. Multivariate time series forecasting Framework

Deep neural networks (DNNs), particularly Transformer models, have significantly advanced time series forecasting, emphasizing long-term predictions since pioneering works like Informer [23]. Multivariate time series forecasting's success hinges on modeling cross-variable correlations. Broadly, methods are classified into variable-dependent strategies [23], [41]–[43] and variable-independent strategies [44]–[46]. To clarify terminology, we use "variable" instead of "channels" as in [46]. Variable-dependent methods treat time series comprehensively, with the majority rudimentarily mapping the cross-variable dimension at the same time step to a latent space for **implicit** modeling. Yet, they have been critiqued for inconsistency during distribution shifts among variables [47]. On the other hand, variable-independent methods [45], [46] apply univariate models across multiple correlated variables. Despite neglecting correlations, they have shown enhanced performance [47]. However, this strategy can yield suboptimal forecasts due to limited capacity [47]. A special method is Crossformer [48], which leverages self-attention mechanism to **explicitly** explore cross-variable correlations, achieving good performance in general time series forecasting task. In urban mobility data, high-resolution grids produce a multitude of time series. Areas with similar semantic and geographical features tend to have strong correlations. Moreover, the high granularity of grids can result in a super-multivariate time series. This complexity makes capturing correlations with computational-heavy variable-dependent strategies a challenge. Efficient attention mechanisms, such as those found in [26]–[28], are crucial to address this issue.

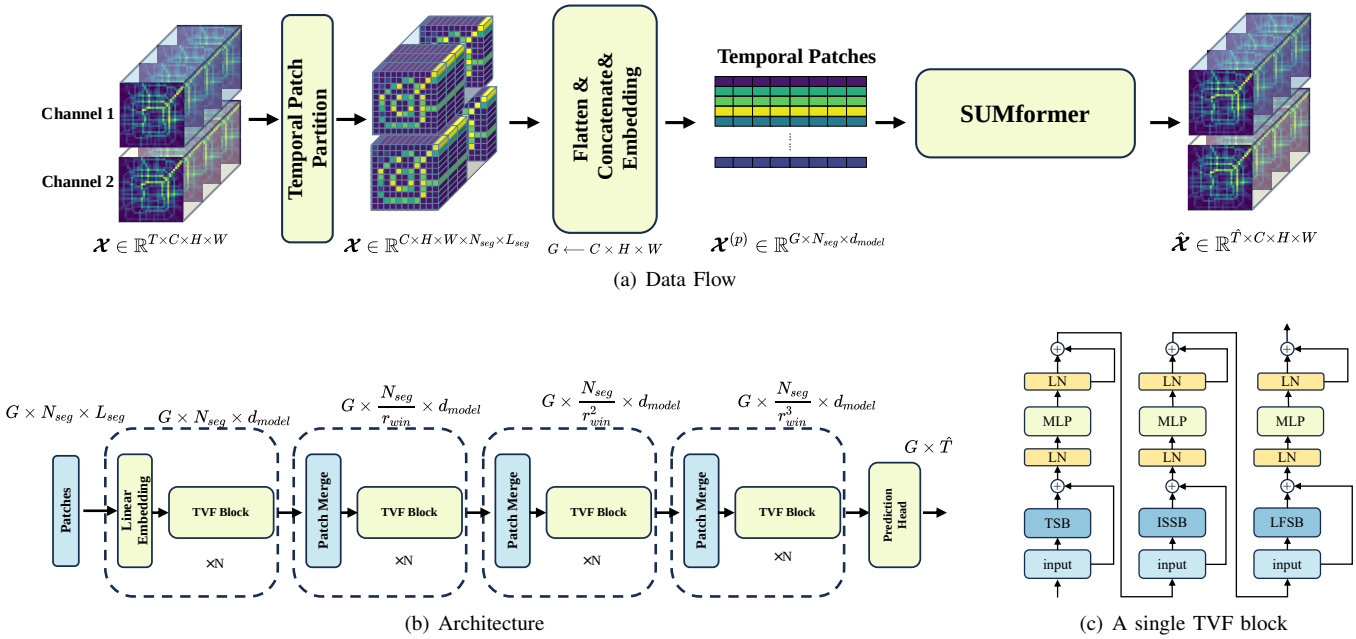
C. Deep learning model leveraging frequency-domain information

The frequency domain analysis algorithm like Fast Fourier Transform (FFT) converts data from the time domain to the frequency domain and serves as a frequency-domain feature extraction module in constructing neural network architectures [42], [49], [50]. Initially proposed as a data-driven method for solving partial differential equations (PDEs), the Fourier Neural Operator (FNO) [51] has subsequently proven effective in image classification [52] and time series forcasting [41], [42], [49], [53]. FourcastNet [54] accurately captures the formation and movement of weather patterns through the utilization of the adaptive Fourier neural operator (AFNO). CoST [55] leverage contrastive learning methods, transform the input series into frequency domain to learn discriminative seasonal representations. TimesNet [49] transforms the time series into 2D space based on multiple periods and applied a 2D kernel for features extraction. Even though urban mobility data clearly shows strong periodic patterns—a significant frequency domain feature—few studies have tapped into this frequency domain information for urban mobility prediction.

III. Methodology

A. Overall Architecture

Fig. 2 presents an overview of the SUMformer architecture. The core elements of SUMformer encompass a temporal patch mechanism for generating super-multivariate temporal patches from input videos, the TVF block, which fully exploits temporal, cross-variable (inter-series), and frequency-domain information, and the temporal patch merging mechanism for capturing multi-scale correlations. In the following sections, we will provide a detailed introduction to all the components of SUMformer.



(a) Data Flow

(b) Architecture

(c) A single TVF block

Fig. 2. (a) Data flow in a SUMformer, where grid-based mobility data is flattened into super-multivariate patches before processing; (b) Architecture inspired by the Swin Transformer [29]; (c) A single TVF (Temporal-Variate-Frequency) block, consisting of a temporal sub-block, an Inter-Series Sub-Block, and a Low-frequency Filter Sub-Block.

*1) **Temporal Patch Partition**:* In the temporal dimension, we begin by segmenting the input into patches at the sub-series level, with each patch functioning as a token input for the SUMformer. The input is denoted as $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T\times C\times H\times W}$, where T is the number of time steps, C is the number of variables per frame, and H, W are the height and width of the frame. First, we flatten it into a super-multivariate time series denoted as a $\boldsymbol{X} \in \mathbb{R}^{G\times T}$, where $G = C \times H \times W$. Subsequently, it is sliced into non-overlapping sub-series $\boldsymbol{\mathcal{X}}' \in \mathbb{R}^{G\times N_{seg}\times L_{seg}}$, where N_{seg} is the number of sub-series $N_{seg} = \frac{T}{L_{seg}}$. Through a linear embedding layer shared by G variables and N_{seg} segments, the length of each patch is projected to a fixed dimension of d_{model}:

$$\begin{aligned} \boldsymbol{x}_{i,j}^{(p)} &= \boldsymbol{x}'_{i,j}\boldsymbol{W}_{patch}, 1 \le i \le G, 1 \le j \le N_{seg}, \\ \boldsymbol{x}^{(p)} &= \boldsymbol{x}^{(p)} + \boldsymbol{W}_{pos}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_{i,j}^{(p)} \in \mathbb{R}^{d_{model}}$ denotes the embedding transferred from the original j-th patch of the i-th sub-series; $\boldsymbol{W}_{patch}$ denotes a linear projection layer; $\boldsymbol{W}_{pos}$ denotes the learnable positional encoding. We employ learnable positional encodings that are separately applied in the directions of longitude, latitude, and time:

$$\begin{aligned} \boldsymbol{W}'_{pos} &= \boldsymbol{W}_{long} + \boldsymbol{W}_{lat} + \boldsymbol{W}_{time}, \\ \boldsymbol{W}_{pos} &= \text{ReShape}\left(\boldsymbol{W}'_{pos}\right), \end{aligned} \tag{2}$$

where $\boldsymbol{W}_{long} \in \mathbb{R}^{C\times 1\times W\times 1\times d_{model}}$, $\boldsymbol{W}_{lat} \in \mathbb{R}^{C\times H\times 1\times 1\times d_{model}}$, $\boldsymbol{W}_{time} \in \mathbb{R}^{C\times 1\times 1\times N_{seg}\times d_{model}}$, $\boldsymbol{W}'_{pos} \in \mathbb{R}^{C\times H\times W\times N_{seg}\times d_{model}}$ and $\boldsymbol{W}_{pos} \in \mathbb{R}^{G\times N_{seg}\times d_{model}}$. Following the linear embedding layer, we obtain a tensor $\boldsymbol{\mathcal{X}}^{(p)} \in \mathbb{R}^{G\times N_{seg}\times d_{model}}$. In contrast to CNN or ViT-based methods, we do not downsample or divide it into spatial patches. Instead, we treat it as a super-multivariate time series to maintain the independence of information between grids and channels. We verify the advantages of preserving this grid independence in our subsequent experiments.

*2) **TVF Block**:* The input tensor $\boldsymbol{\mathcal{X}}^{(p)}$ then passes through a series of stacked Temporal-Variable-Frequency (TVF) blocks, as depicted in Fig. 2(c). Each of these blocks comprises three sub-blocks designed for feature processing from the temporal, inter-series, and frequency domains, respectively. Each sub-block comprises a corresponding processing module, followed by a LayerNorm (LN) layer and a 2-layer MLP using GELU activation. Residual connections are applied for each layer:

$$\begin{aligned} \hat{\boldsymbol{\mathcal{X}}}^{(p)l} &= \text{LayerNorm}\left(\boldsymbol{\mathcal{X}}^{(p)l} + sub\text{-}block\left(\boldsymbol{\mathcal{X}}^{(p)l}\right)\right), \\ \boldsymbol{\mathcal{X}}^{(p)l+1} &= \text{LayerNorm}\left(\hat{\boldsymbol{\mathcal{X}}}^{(p)l} + MLP\left(\hat{\boldsymbol{\mathcal{X}}}^{(p)l}\right)\right), \end{aligned} \tag{3}$$

where $\boldsymbol{\mathcal{X}}^{(p)l+1}$ and $\boldsymbol{\mathcal{X}}^{(p)l}$ represent the input and output of the l-th layer, respectively.

*3) **Patch Merging**:* To capture long-term temporal correlations within the super-multivariate time series, the number of patches is reduced through the patch merging [29] mechanism, as depicted in Fig. 2(b). The patch merging layer concatenates the features of each group of adjacent fixed-size windows and applies a linear layer to the N_{seg}-dimensional concatenated features. With each merging operation, The patch size of the input tensor is reduced by a factor of r_{win}, which denotes the number of merged sub-series. With several layer of patch merging, the model's temporal receptive field grows exponentially which is advantageous for effectively capturing correlations at different scales. Eventually, all sub-series in the same variable are fused into a single token. After passing through a linear prediction layer, we yield the prediction results.

B. Temporal Sub-Block

The main objective of this module is to capture temporal correlations within each individual univariate time series, with all time series sharing the same set of parameters. To achieve this, we have introduced two options, one based on Multi-Head Self Attention (MHSA) [36] and the other on MLP-Mixer [38].

*1) **TSB-MHSA**:* In this version, we utilize MHSA to extract the correlation among sub-series within the same variable. The patches from all time series, denoted as $\boldsymbol{\mathcal{X}}_{i,:}^{(p)} \in \mathbb{R}^{N_{seg} \times d_{model}}$, are initially mapped by $\boldsymbol{W}_Q^{(h)}$, $\boldsymbol{W}_K^{(h)}$, and $\boldsymbol{W}_V^{(h)}$ into $\boldsymbol{Q}_{i,:}^{(h)}$, $\boldsymbol{K}_{i,:}^{(h)}$, and $\boldsymbol{V}_{i,:}^{(h)} \in \mathbb{R}^{N_{seg} \times d_{qkv}}$ within the latent space, respectively. Here, $1 \leq i \leq G$, and h represents the current head number of the MHSA. Then, we have

$$\boldsymbol{Z}_{i,:}^{(h)} = \text{Softmax}\left(\frac{\boldsymbol{Q}_{i,:}^{(h)} \boldsymbol{K}_{i,:}^{(h)^T}}{\sqrt{d_{qkv}}}\right) \boldsymbol{V}_{i,:}^{(h)}, \tag{4}$$

where d_{qkv} is a constant, the h heads, denoted as $\boldsymbol{Z}_{i,:}^{(h)}$, together form a tensor $\boldsymbol{\mathcal{Z}}_{i,:} \in \mathbb{R}^{N_{seg} \times h \times d_{qkv}}$. Subsequently, the output, $\boldsymbol{\mathcal{Z}}_{i,:} \in \mathbb{R}^{N_{seg} \times h \times d_{qkv}}$, is flattened along the dimension of the head number and then mapped through linear output projection to produce the final output $\boldsymbol{O}_{i,:}^{(time)} \in \mathbb{R}^{N_{seg} \times d_{model}}$.

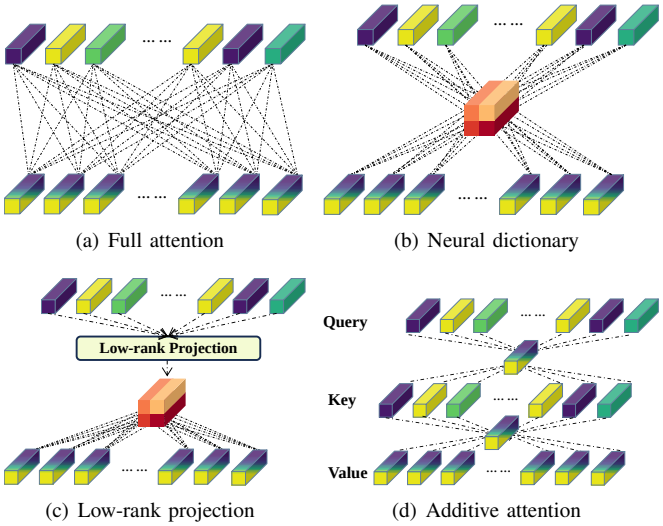
*2) **TSB-MLP**:* We also offer a pure-MLP architecture as an alternative to MHSA. We utilize two two-layer MLP networks, each incorporating the GELU activation function, Dropout, and residual connections to capture the internal features of the sub-series and the features spanning across sub-series. When provided with the input $\boldsymbol{\mathcal{X}}_{i,:}^{(p)}$, the process is as follows:

$$\begin{aligned} \hat{\boldsymbol{\mathcal{Z}}}_{i,j,:} &= \boldsymbol{\mathcal{X}}_{i,j,:}^{(p)} + \boldsymbol{W}_2 \sigma\left(\boldsymbol{W}_1 \text{LayerNorm}\left(\boldsymbol{\mathcal{X}}_{i,j,:}^{(p)l}\right)\right), \\ \boldsymbol{O}_{i,:,k}^{(time)} &= \hat{\boldsymbol{Z}}_{i,:,k} + \boldsymbol{W}_4 \sigma\left(\boldsymbol{W}_3 \text{LayerNorm}\left(\hat{\boldsymbol{\mathcal{Z}}}_{i,:,k}\right)\right), \end{aligned} \tag{5}$$

where $1 \leq i \leq G$, $1 \leq j \leq N_{seg}$ and $1 \leq k \leq d_{model}$; $W_n (n = 1, 2, 3, 4)$ denote the learnable weight matrices, $\boldsymbol{O}^{(time)} \in \mathbb{R}^{G \times N_{seg} \times d_{model}}$ denotes the output of TSB-MLP sub-block.

C. Inter-Series Sub-Block

The purpose of the Inter-Series Sub-Block (ISSB) is to capture correlations between different variables. In the case of urban mobility data, a significant challenge arises due to the potentially large number of time series present in an urban mobility video. For instance, the popular benchmark dataset TaxiBJ comprises a total of 2048 time series. It's important to note that this number can be even larger in real-world scenarios. For example, we may use finer latitude and longitude resolutions to define more detailed fine-grained videos [56] (larger H and W), or incorporate additional mobility attributes for each grid, such as the inflow and outflow of various travel modes (larger C). We can certainly use the raw transformer proposed in [36] to capture correlations between variables (Fig. 3(a)), but its computational cost is $O(G^2)$, which would make the model computationally burdensome for "hyperspectral fine-grained" urban mobility video data. Instead, SUMformer offers several alternative options with $O(G)$ complexity, as illustrated in Fig. 3.



(a) Full attention (b) Neural dictionary (c) Low-rank projection (d) Additive attention

Fig. 3. Alternative Attention Mechanism Choices for the Inter-Series Sub-Block.

*1) **Neural Dictionary**:* The neural dictionary model, proposed in [26], utilizes a learnable neural dictionary $\boldsymbol{Dic} \in \mathbb{R}^{g \times d_{model}}$ with a fixed number g (where $g \ll G$) of vectors to reduce computational complexity. For an input $\boldsymbol{\mathcal{X}}_{:,i}^{(p)l} \in \mathbb{R}^{G \times d_{model}}$ from the i-th patch, we have

$$\begin{aligned} \boldsymbol{M} &= \text{MHSA}\left(\boldsymbol{Dic}, \boldsymbol{\mathcal{X}}_{:,i}^{(p)l}, \boldsymbol{\mathcal{X}}_{:,i}^{(p)l}\right), \\ \boldsymbol{O}_{:,i}^{(spatial)} &= \text{MHSA}\left(\boldsymbol{\mathcal{X}}_{:,i}^{(p)l}, \boldsymbol{M}, \boldsymbol{M}\right), \end{aligned} \tag{6}$$

where we employ the neural dictionary $\boldsymbol{Dic}$ as the query, and the input $\boldsymbol{X}_{:,i}^{(p)l}$ as the key-value pair in the first MHSA layer, creating an aggregated message denoted as $\boldsymbol{M} \in \mathbb{R}^{g \times d_{model}}$ with the same shape as $\boldsymbol{Dic}$. Subsequently, this $\boldsymbol{M} \in \mathbb{R}^{g \times d_{model}}$ is utilized as the key-value pair in the second MHSA layer to interact with the query $\boldsymbol{X}_{:,i}^{(p)l}$, resulting in the final output denoted as $\boldsymbol{O}_{:,i}^{(spatial)} \in \mathbb{R}^{G \times d_{model}}$.

*2) **Low-rank Projection**:* Unlike aggregating information using a neural dictionary and computing the correlations with the input $\boldsymbol{\mathcal{X}}_{:,i}^{(p)l}$ through MHSA, in the low-rank projection version [27], we use a $(g \times G)$-dimensional projection matrix $\boldsymbol{W}_{lin}$ to map the key and value from G-dimension to g-dimension where g is a fixed number and $g \ll G$ (see Fig. 3(c)).

$$\begin{aligned} \hat{\boldsymbol{\mathcal{X}}}_{:,i}^{(p)l} &= \boldsymbol{W}_{lin} \boldsymbol{\mathcal{X}}_{:,i}^{(p)l}, \\ \boldsymbol{O}_{:,i}^{(spatial)} &= \text{MHSA}\left(\boldsymbol{\mathcal{X}}_{:,i}^{(p)l}, \hat{\boldsymbol{\mathcal{X}}}_{:,i}^{(p)l}, \hat{\boldsymbol{\mathcal{X}}}_{:,i}^{(p)l}\right), \end{aligned} \tag{7}$$

where all variable information is aggregated to a smaller size output denoted as $\hat{\boldsymbol{\mathcal{X}}}_{:,i}^{(p)l} \in \mathbb{R}^{g \times d_{model}}$ through $\boldsymbol{W}_{lin}$. Subsequently, we employ MHSA to calculate the correlation among the G variables.

*3) **Additive Attention**:* The patches from all variables at the same time steps denoted as $\boldsymbol{\mathcal{X}}_{:,i}^{(p)} \in \mathbb{R}^{N_{seg} \times d_{model}}$, are initially mapped by $\boldsymbol{W}_Q^{(h)}$, $\boldsymbol{W}_K^{(h)}$, and $\boldsymbol{W}_V^{(h)}$ into $\boldsymbol{Q}_{:,i}^{(h)}$, $\boldsymbol{K}_{:,i}^{(h)}$, and $\boldsymbol{V}_{:,i}^{(h)} \in \mathbb{R}^{G \times d_{qkv}}$ within the latent space, respectively. As shown in Fig. 3(d), the Additive Attention mechanism [28] first

summarizes the query $\boldsymbol{Q}_{:,i}^{(h)} \in \mathbb{R}^{G\times d_{qkv}}$ with a 1-D mapping vector $\boldsymbol{w}_q \in \mathbb{R}^{d_{qkv}}$ using Softmax activation:

$$\boldsymbol{\alpha} = \text{Softmax}\left(\frac{\boldsymbol{Q}_{:,i}^{(h)}\boldsymbol{w}_q}{\sqrt{d_{qkv}}}\right), \tag{8}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^G$ denotes the output query attention score. Then we get the global query vector $\boldsymbol{q} \in \mathbb{R}^{d_{qkv}}$ via $\boldsymbol{q} = \boldsymbol{\alpha}\boldsymbol{Q}_{:,i}^{(h)}$. Followed by an element-wise product between global query vector and key $\boldsymbol{K}_{:,i}^{(h)} \in \mathbb{R}^{G\times d_{qkv}}$, we model the correlation between query and key via $\boldsymbol{P}_{:,i}^{(h)} = \boldsymbol{q} * \boldsymbol{K}_{:,i}^{(h)}$, where $\boldsymbol{K}_{:,i}^{(h)} \in \mathbb{R}^{G\times d_{qkv}}$ and $*$ is the element-wise production. Subsequently, a similar procedure is employed to obtain a global key vector $\boldsymbol{k} \in \mathbb{R}^{d_{qkv}}$:

$$\begin{aligned}\boldsymbol{\beta} &= \text{Softmax}\left(\frac{\boldsymbol{P}_{:,i}^{(h)}\boldsymbol{w}_k}{\sqrt{d_{qkv}}}\right),\\ \boldsymbol{k} &= \boldsymbol{\beta}\boldsymbol{P}_{:,i}^{(h)}.\end{aligned} \tag{9}$$

Then, an element-wise product is applied between value $\boldsymbol{V}_{:,i}^{(h)} \in \mathbb{R}^{G\times d_{qkv}}$ and global key vector to compute the key-value interaction $\boldsymbol{U}_{:,i}^{(h)} \in \mathbb{R}^{G\times d_{qkv}}$. Then we applied linear matrices and residual connection to obtain the final output of the inter-series correlation output $\boldsymbol{O}^{(spatial)}$:

$$\begin{aligned}\boldsymbol{U}_{:,i}^{(h)} &= \boldsymbol{k} * \boldsymbol{V}_{:,i}^{(h)},\\ \boldsymbol{O}_{:,i}^{(spatial)} &= \boldsymbol{W}_1\left(\boldsymbol{Q}_{:,i}^{(h)} + \boldsymbol{W}_2\boldsymbol{U}_{:,i}^{(h)}\right),\end{aligned} \tag{10}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are learnable weight matrices.

D. Low-frequency Filter Sub-Block

We consider periodicity within urban mobility data to play a substantial role in long-term prediction task. In frequency-domain analysis, it's well-established that periodicity primarily resides within the low-frequency components, while the high-frequency components introduce some level of noise. To effectively address this and bolster the model's resilience, we have introduced a Low-Frequency Filter Sub-Block (LFSB). This module serves the purpose of filtering out superfluous high-frequency elements, enhancing the model's robustness and preserving the essential periodic features.

LFSB is composed of four translators, as illustrated in Fig. 4. Following TSB and ISSB, LFSB conducts low-frequency filtering within the frequency domain of the features generated by ISSB. The details are as follows:

$$\begin{aligned}\boldsymbol{X}_1 &= \boldsymbol{W}_1\text{Reshape}\left(\boldsymbol{\mathcal{X}}^{(p)l}\right),\\ \boldsymbol{\mathcal{F}} &= \text{DFT}\left(\boldsymbol{X_1}\right),\\ \boldsymbol{X}_2 &= \text{iDFT}\left(\hat{\boldsymbol{\mathcal{F}}}\right),\\ \boldsymbol{\mathcal{X}}^{(p)l+1} &= \text{Reshape}_{\text{i}}\left(\boldsymbol{W}_2\boldsymbol{X}_2\right),\end{aligned} \tag{11}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{T\times(N_{seg}\cdot d_{model})}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{(N_{seg}\cdot d_{model})\times T}$ are learnable parameters for the full series translators and patch translators, a Discrete Fourier Transform (DFT) translator is employed to transform the entire series into the frequency domain, denoted as $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{G\times f}$. Following this, we selectively

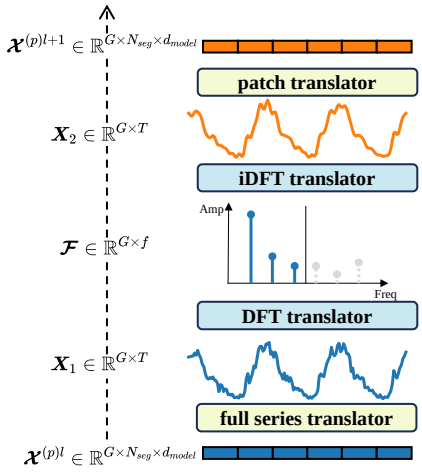


Fig. 4. Illustration of Low-frequency Filter Sub-Block

sample the first half of the spectrum, retaining the portions with the lowest frequency while zeroing out the remainder which is denoted as $\hat{\boldsymbol{\mathcal{F}}}$. Then, we turn to the Inverse Discrete Fourier Transform (iDFT) translator, which transforms the filtered signal back into the temporal domain, denoted as $\boldsymbol{X}_2 \in \mathbb{R}^{G\times T}$. Lastly, a patch translator is deployed to reshape the denoised full series back into patches $\boldsymbol{\mathcal{X}}^{(p)l+1} \in \mathbb{R}^{G\times N_{seg}\times d_{model}}$ using a linear layer.

E. Architecture Variants

Based on the aforementioned description, SUMformer is more akin to a framework than a specific method. We offer multiple choices in both time and space, as shown in Tab. I. Note that the main complexity of the attention algorithm is concentrated on the variable dimension since variable number is larger than the input length. The complexity mainly presented in the table is about the variable G. We have not listed the computation complexity associated with the input length T in Tab. I.

TABLE I
SEVEN VARIANTS OF SUMFORMER

Variant	TSB	ISSB	Complexity
SUMformer-AD	MHSA	NeuralDictionary	$O(G)$
SUMformer-MD	MLP	NeuralDictionary	$O(G)$
SUMformer-AL	MHSA	Low-rank Projection	$O(G)$
SUMformer-AA	MHSA	Additive Attention	$O(G)$
SUMformer-AF	MHSA	Full Attention	$O(G^2)$
SUMformer-TS	NeuralDictionary		$O(N_{seg}G)$[1]
SUMformer-ViT	MHSA	NeuralDictionary	$O\left(\frac{G}{L_{spatial}^2}\right)$

[1] both space and time.

From Tab. I, we can find that a model employing full attention is also included. Comparing this with other variants allows us to assess whether the expedited attention mechanism compromises the performance of SUMformer. Of particular note are two distinct variants: SUMformer-TS and SUMformer-ViT. These are instrumental in examining the necessity of temporal attention within SUMformer and evaluating the appropriateness of the super-multivariate perspective for urban mobility data.

*1) **SUMformer-TS**:* In the SUMformer-TS variant, we employ the NeuralDictionary mechanism to compute the attention across all variables and patch tokens. This design was intended to investigate the potential of Neural Dictionary, to concurrently capture both temporal and spatial correlations. The fixed value of g and other configuration settings are consistent with those used in SUMformer-AD, resulting in a computational complexity of $O(N_{seg}G)$ for both space and time.

*2) **SUMformer-ViT**:* In this variant, we conceptualize urban mobility data akin to a video format by mixing adjacent grids and channels into patch structures. Following standard ViTs, we first reshape the input data $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T\times C\times H\times W}$ at each time point into a sequence of flattened non-overlapping 2D spatial patches $\boldsymbol{X} \in \mathbb{R}^{T\times G_{spatial}\times\left(L^2_{spatial}\cdot C\right)}$, where $(L_{spatial}, L_{spatial})$ is the resolution of each spatial patch, and $G_{spatial} = \frac{HW}{L^2_{spatial}}$ is the resulting number of spatial patches. Subsequently, $\boldsymbol{X}$ is sliced into non-overlapping temporal subseries $\boldsymbol{\mathcal{X}'} \in \mathbb{R}^{G_{spatial}\times N_{seg}\times\left(L^2_{spatial}\cdot C\cdot L_{seg}\right)}$. Finally, we project the spatio-temporal "tubes" [18], [19] denoted as $\boldsymbol{\mathcal{X}}'_{i,j} \in \mathbb{R}^{L^2_{spatial}\times C\times L_{seg}}$ into $\boldsymbol{\mathcal{X}}^{(p)}_{i,j} \in \mathbb{R}^{d_{model}}$ through a linear embedding layer. The remaining settings align with those of SUMformer-AD.

IV. Experiment

We conducted experimental evaluations on five real urban mobility datasets spanning three cities: Beijing, Chengdu, and New York. We subsequently delineate various categories of prediction models for comparative analysis, including: the **Variable-Dependent** Time Series Forecasting Method, the **Variable-Independent** Time Series Forecasting Method, the **Video Prediction** Method, and the **Frequency-based** Method. Details pertaining to our training procedure are then elaborated upon. Following this, we juxtapose our novel SUMformer architecture with emblematic models from alternative variants, offering an analysis of the relative merits and demerits of each approach. Finally, we present an in-depth analysis of our proposed model.

A. Datasets

We utilized urban inbound and outbound traffic flow datasets from Beijing, Chengdu, and New York, designated as TaxiBJ, CDtaxi, CDbike, NYCtaxi, NYCbike, for our urban mobility prediction experiments. TaxiBJ, CDtaxi, NYCtaxi represent taxi flow datasets, while CDbike and NYCbike pertain to bike flow. The primary aim is to forecast the inbound and outbound traffic flow for each grid when urban traffic is spatially divided into grids based on latitude and longitude. Detailed attributes for each of the five datasets are presented in Tab. II. The TaxiBJ, NYCtaxi and NYCbike datasets are frequently used in existing literature[1]. In contrast, the CDtaxi and CDbike dataset are proprietary, and we do not have authorization to release it publicly. We primarily use a historical input of 128 frames to predict the traffic flow in the next 128, 64, and 32 frames. The dataset is split into training, validation, and test sets with a ratio of 7:1:2, respectively.

[1]https://github.com/LibCity/Bigscity-LibCity-Datasets

TABLE II
Statistics of traffic flow datasets

Datasets	Spatial size	Variables	Time range	Timesteps
TaxiBJ	32×32	2048	2013/7/1–2013/10/29 2014/3/1–2014/6/27 2015/3/1–2015/6/30 2015/11/1–2016/4/10	22484
CDtaxi	32×32	2048	2023/2/26–2023/7/22	7056
CDbike	32×32	2048	2023/3/13–2023/11/3	11280
NYCtaxi	10×20	400	2015/1/1–2015/3/1	2880
NYCbike	10×20	400	2016/7/1–2016/8/30	2880

B. Baselines

We utilized various baseline categories for comparative analysis:

*1) **Heuristic Method**:*

- **HA**: The **History Average (HA)** method predicts future values by averaging the input. It represents a widely accepted minimal baseline in existing time series forecasting research.

Furthermore, we introduce two enhanced simple baselines to our studies. Urban mobility is renowned for its pronounced daily and weekly periodicities. A high-quality long-term prediction method should ideally surpass both those two methodologies:

- **DH**: The **Daily History (DH)** method employs historical data from the corresponding time of the previous day for its predictions.
- **WH**: The **Weekly History (WH)** method employs historical data from the corresponding time of the previous week for its predictions.

*2) **Video Prediction**:* We compared three video prediction models treating the gird-based data as a video for prediction.

- **SimVP** [10] encodes the input frames into a latent space and uses a spatial-temporal translator to learn spatiotemporal variations.
- **TAU** [14] introduces an attention-based temporal module, extracting inter-frame dynamical attention and intra-frame statistical attention separately.
- **Earthformer** [57] represents a ViT-based prediction framework, distinct from the CNN-based SimVP and TAU. This method decomposes the video into spatiotemporal 3D cuboids and employs cuboid-level self-attention to discern the spatiotemporal correlations.

*3) **Variable-Independent Methods**:* We evaluated two time series forecasting techniques grounded in the variable-independent strategy. Such methods approach multivariate time series as a univariate issue, wherein all dimensions utilize shared parameters, overlooking the interrelated correlations between variables.

- **Nlinear** [45] is a one-layer linear model that directly maps the input sequence into the output sequence. To counteract distribution shift, the input is subtracted by the last value of the sequence, passed through a linear layer, and then the subtracted value is added back as a simple form of normalization.
- **PatchTST** [46] slices the time series into sub-series level patches, treating them as tokens input to a Transformer.

*4) **Frequency-based Methods**:* We also tested several methods that utilize frequency-domain information. Notably, we believe that the spread of human flow in cities may also partially conform to a certain partial differential equations (PDEs) [58]. Precisely because of this reason, we chose the Fourier neural operator (FNO) method, which is highly effective in modeling PDEs, for comparison.

- **FNO1D** [51] flattens the video into a super-multivariate time series. The historical input data serves as the initial condition for the PDE, and the FNO is employed to solve the PDE, which corresponds to the projected future data.
- **FNO3D** [51] treats the input video as a whole, and utilize a 3D Fourier transform layer to solve the PDE.

The comparative performance between FNO1D and FNO3D offers an opportunity to evaluate whether the super-multivariate view on urban mobility data is more appropriate than the video-based view.

- **Fedformer** [42] integrates a frequency-enhanced block (same as FNO1D), leveraging the seasonal-trend decomposition method to capture the overarching profile of the time series. Concurrently, the Transformer is deployed to discern finer structures. For forecasting, Fedformer employs an implicit variable-dependent strategy.

*5) **Variable-Dependent Methods**:*

- **TCN** [44] utilizes a 1-D causal convolution to ensure the model relies only on past information during predictions, while simultaneously implicitly modeling correlations across variable dimensions.
- **Crossformer** [48] devised a Two-Stage-Attention (TSA) layer to capture the cross-time and cross-variable dependency. A hierarchical encoder-decoder architecture is employed for multi-scale information utilization.

For the time series forecasting benchmarks, the variables across all channels and grids are evaluated as distinct time series.

C. Experiment Settings

Our experiments were conducted on a server equipped with a 20-core Intel Silver 4316 CPU and an NVIDIA GeForce RTX 4090 GPU. The SUMformer model was implemented using PyTorch, and the implementation code has been released on GitHub[2]. The dataset was divided into training, validation, and testing sets with a ratio of 7:1:2. We trained eight baselines and SUMformer for 80 epochs with a batch size of 16. The Adam optimizer was used, and the learning rate was scheduled using the CosineLRScheduler from the *timm* library. The warm-up phase consisted of 5 epochs, with the learning rate set to 1e-5, and in the training phase, the learning rate was set to 5e-4.

The primary hyperparameter settings for SUMformer are as follows: we utilize SUMformer-AD in Tab. III and Tab. IV, which employs MHSA in the temporal domain and a neural dictionary for the spatial domain. The patch merge ratio, r_{win}, is set to 2. The initial length of the sub-series, L_{seg}, is 16, while the fixed dictionary size for spatial linear attention is 256. The embedding size, d_{model}, is 128. Additionally, the number of TVF blocks is 4, indicating that the patch will undergo merging four times. We employed a grid search approach based on the default settings of other baselines reported in the literature, ensuring that these approaches achieved optimal performance. We evaluated each model by root-mean square error (RMSE) and mean absolute error (MAE).

[2]https://github.com/Chengyui/SUMformer

D. Main results

The comprehensive results are presented in Tab. III and Tab. IV in which lower MAE and RMSE values indicate more accurate predictions, with the optimal results highlighted in **bold** and the second-best results underlined. Our proposed SUMformer model consistently outperforms across all scenarios and metrics. Compared to video prediction methods and heuristic methods in Tab. III, SUMformer exhibits a significant performance advantage by a large margin. Besides demonstrating the superiority of SUMformer, the results in Tab. IV also provide a comprehensive comparison of different types of multivariate time series forecasting methods, including frequency-based, variable independent, variable dependent, as well as a comparison between explicit and implicit modeling methods related to cross-variable correlation. We can derive three key observations from these results:

1). Super-multivariate Modeling Outperforms Video Modeling in Long-term Urban Mobility Forecasting: For CDtaxi, CDbike and TaxiBJ datasets, which presumably have larger training data sizes, the strong time series forecasting baselines - PatchTST, Crossformer, and our SUMformer - consistently outshine the video-based prediction models such as SimVP, TAU, and Earthformer. Notably, even for the NYCtaxi and NYCbike datasets, both Crossformer and SUMformer continue to surpass the video-based forecasting models. This assertion is further bolstered when comparing the performance of FNO1d and FNO3d, where the former consistently delivers better results.

2). Explicit Variable Correlation Modeling is Crucial: We included the Fedformer baseline, which translates the variable dimension into a latent space for implicitly modeling cross-variable relationships. Despite being equipped with a frequency-enhancement block, this method doesn't yield impressive results. PatchTST, which doesn't even focus on cross-variable correlation, still surpasses Fedformer. In the majority of scenarios and across various metrics, both PatchTST and Fedformer lag behind Crossformer and our proposed SUMformer. This underscores the importance of explicitly modeling variable correlations when dealing with urban mobility forecasting. Our SUMformer, armed with an attention mechanism that boasts linear computational complexity, excels in both efficiency and effectiveness when explicit modeling cross-variable correlations.

3). Emphasizing the Periodicity of Urban Mobility is Crucial for Long-term Forecasting: We presented two robust historical benchmarks, WH and DH, tailored for extended-horizon forecasting. Intriguingly, though these methods rely primarily on data from the previous day and week for their predictions, they manage to outshine several advanced deep

TABLE III
MAE/RMSE COMPARISION FOR VIDEO PREDICTION, HEURISTIC METHOD AND SUMFORMER ON FIVE DATASETS

Model		SUMformer		Video Prediction SimVP		TAU		Earthformer		Heuristic Method HA		WH		DH	
Metric		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TaxiBJ	128-32	**15.268**	**29.815**	16.375	30.676	17.422	32.921	17.106	32.655	43.168	70.541				
	128-64	**16.681**	**32.619**	19.246	36.311	19.105	36.644	20.637	38.386	43.133	70.488	36.402	71.085	23.425	47.472
	128-128	**19.347**	**38.102**	22.154	40.764	23.082	42.177	25.816	48.435	43.023	70.321				
CDtaxi	128-32	**3.678**	**7.997**	4.108	8.904	4.118	8.856	4.142	9.744	11.384	26.602				
	128-64	**3.728**	**8.377**	4.437	9.600	4.286	9.373	4.187	10.150	11.425	26.684	4.543	10.512	4.549	10.762
	128-128	**3.789**	**8.830**	4.559	10.160	4.599	9.632	4.309	10.302	11.449	26.737				
CDbike	128-32	**11.941**	**41.082**	15.306	45.309	14.340	46.520	15.245	51.350	30.268	86.452				
	128-64	**12.442**	**43.359**	14.938	46.766	15.445	46.545	15.457	51.717	30.148	86.090	16.530	61.412	12.240	49.025
	128-128	**14.187**	**46.359**	15.798	51.893	15.540	51.310	15.582	53.168	29.990	85.622				
NYCtaxi	128-32	**5.121**	**14.879**	5.859	16.096	5.899	16.328	5.882	17.014	20.520	58.743				
	128-64	**5.389**	**15.607**	6.141	16.666	6.430	17.437	5.846	17.455	20.370	58.382	7.111	22.298	10.101	35.694
	128-128	**5.595**	**16.734**	6.978	19.014	7.298	18.993	6.465	17.848	20.395	58.328				
NYCbike	128-32	**1.170**	**3.934**	1.873	5.930	1.915	5.852	1.997	6.649	3.317	10.270				
	128-64	**1.201**	**4.243**	1.908	6.013	1.811	5.708	1.804	5.942	3.315	10.239	2.430	9.821	2.051	8.011
	128-128	**1.280**	**5.486**	1.922	6.070	2.005	6.247	1.941	6.235	3.361	10.338				

TABLE IV
MAE/RMSE COMPARISION FOR DIFFERENT MULTIVARIATE TIME SERIES FORECASTING APPROACHES AND SUMFORMER ON FIVE DATASETS

Model		Variable-Dependent SUMformer		TCN		Crossformer		Variable-Independent Nlinear		PatchTST		Frequency-based FNO1D		FNO3D		Fedformer	
Metric		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TaxiBJ	128-32	**15.268**	**29.815**	18.912	35.076	19.765	33.553	19.306	36.899	16.166	31.330	19.101	34.746	19.833	35.317	19.288	35.546
	128-64	**16.681**	**32.619**	21.503	38.693	18.784	34.626	21.393	40.411	18.291	34.933	21.057	38.481	22.083	39.911	21.244	38.820
	128-128	**19.347**	**38.102**	25.004	46.787	21.342	40.732	25.778	47.801	20.674	40.597	25.422	47.595	26.019	47.151	24.480	46.255
CDtaxi	128-32	**3.678**	**7.997**	4.082	9.669	3.683	8.082	4.065	9.796	3.710	8.629	3.789	8.839	4.388	9.607	4.406	9.287
	128-64	**3.728**	**8.377**	4.213	9.900	3.816	8.578	4.184	10.197	3.824	8.988	3.841	8.915	4.553	10.020	4.170	9.313
	128-128	**3.789**	**8.830**	4.369	10.691	3.997	9.203	4.383	10.800	4.015	9.424	4.011	9.551	4.555	10.426	4.370	9.885
CDbike	128-32	11.941	**41.082**	14.656	46.863	13.699	42.709	11.974	44.010	**11.750**	41.749	14.636	47.231	19.948	57.599	14.778	47.514
	128-64	12.442	43.359	16.384	50.474	15.693	45.959	12.718	46.182	**12.408**	**43.326**	16.133	52.086	21.514	61.044	15.057	49.497
	128-128	14.187	**46.359**	15.967	50.147	17.074	50.909	**13.816**	49.455	13.997	47.083	16.782	51.870	21.107	60.381	14.850	50.416
NYCtaxi	128-32	**5.121**	**14.879**	8.576	19.484	5.534	15.537	9.653	31.511	6.581	18.708	5.816	15.964	7.716	19.343	6.570	18.681
	128-64	**5.389**	**15.607**	7.599	19.662	5.696	16.346	10.593	34.118	6.987	20.706	6.027	16.886	8.040	20.552	7.420	20.987
	128-128	**5.595**	**16.734**	8.766	22.622	6.138	17.592	12.072	38.261	7.488	22.656	6.176	17.517	9.247	23.926	7.644	22.379
NYCbike	128-32	**1.170**	**3.934**	3.133	7.386	1.373	4.578	1.869	6.829	1.399	4.792	2.479	7.673	2.131	6.695	2.580	7.604
	128-64	**1.201**	**4.243**	2.824	7.144	1.433	5.170	2.040	7.457	1.496	5.295	2.730	8.406	2.465	7.447	2.409	7.867
	128-128	**1.280**	**5.486**	2.711	6.920	2.010	6.313	2.288	8.082	1.637	6.058	2.734	8.516	2.268	7.342	2.428	7.743

learning baselines, most notably in the longest forecasting window (128-128). This phenomenon can be attributed to the innate statistical properties of urban mobility. Research indicates that as the time lag extends, the autocorrelation coefficient of urban mobility wanes [59]. It underscores the importance of prediction frameworks that harness the intrinsic periodic, seasonal, and recurrent patterns in the data for long-term accuracy. Our SUMformer model, with its low-frequency filtering mechanism, adeptly captures and emphasizes these periodic patterns.

We also visualize predicted errors of SUMformer, Crossformer and SimVP on TaxiBJ, CDtaxi and NYCtaxi in Fig. 5. At various prediction points, labeled as $t = 32$, $t = 64$, $t = 96$, and $t = 128$, the target displays intricate patterns. In addition, we provide the outputs of the predicted time series from three methods for a specific grid within each dataset. We have observed that human mobility data exhibit a strong periodic pattern, and all three methods can offer predictions that are fundamentally accurate on a macro trend level. SUMformer's errors show minimal deviations from the target, suggesting a high level of accuracy. In contrast, both Crossformer and SimVP appear to have more pronounced error patterns, indicating larger discrepancies from the target. This visualization underscores the superior accuracy and precision of SUMformer in comparison to the other two models. SimVP performs the poorest, particularly in the NYC example, suggesting that CNN-based methods may not be suitable for long-term prediction tasks. SUMformer's advantage over Crossformer can be attributed to its use of low-frequency filters and a more sophisticated Swin Transformer-style architecture. It tracks larger values more accurately than Crossformer in those three examples. In Fig. 5, we selected one variable from each of the three datasets for temporal visualization. We observed that SUMformer can effectively capture the temporal patterns of traffic flow, particularly in predicting the peak traffic flow, which is a crucial metric for traffic agencies.

E. Performance Analysis for Seven Variants

Tab. V shows the results of seven variants on the three datasets, where SUMformer-ViT uses a 2×2 patch. We provide

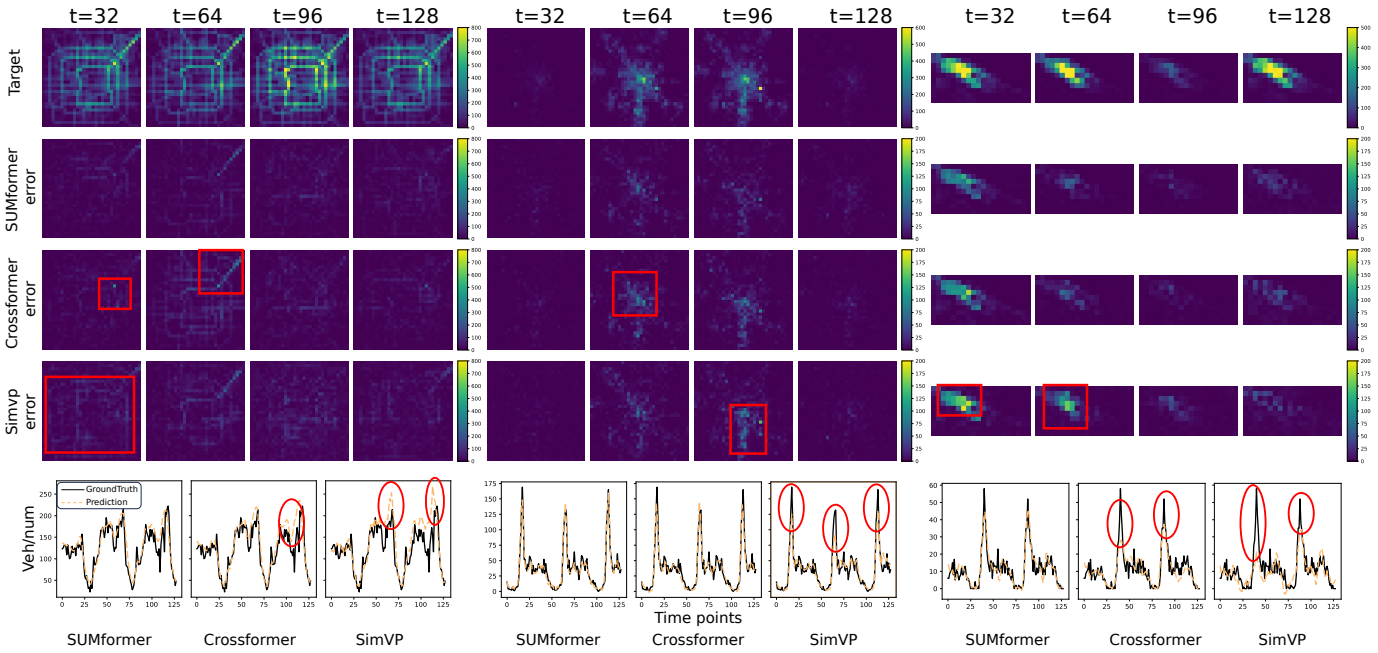


Fig. 5. Spatial and temporal visualization for SUMformer, Crossformer, and SimVP on TaxiBJ, CDtaxi, NYCtaxi respectively while the two channels are averaged for convenience. Significant errors are highlighted with red boxes.

TABLE V
MAE/RMSE COMPARISION ON SEVEN VARIANTS

Dataset	Scenario	128-128		128-64		128-32	
	Variant	MAE	RMSE	MAE	RMSE	MAE	RMSE
TaxiBJ	AD	19.347	**38.102**	**16.681**	32.619	15.268	29.815
	MD	20.091	39.364	17.403	33.542	15.733	30.503
	AL	**19.138**	38.361	17.169	**32.498**	**14.915**	28.883
	AA	20.442	39.384	17.628	34.009	15.774	30.133
	AF	19.890	38.801	16.765	32.664	15.109	**28.878**
	TS	21.416	41.222	17.797	34.033	16.242	31.432
	ViT	20.526	40.619	17.586	33.126	15.504	29.085
CDtaxi	AD	**3.789**	8.830	3.728	**8.377**	**3.678**	**7.997**
	MD	3.862	8.826	3.840	8.773	3.778	8.281
	AL	4.040	9.471	3.887	9.036	3.914	8.832
	AA	3.966	9.120	3.857	9.113	3.930	8.562
	AF	3.958	8.866	3.806	8.848	3.710	8.204
	TS	3.851	**8.765**	**3.724**	8.380	3.790	8.287
	ViT	4.132	9.326	3.988	9.036	3.845	8.576
NYCtaxi	AD	5.595	16.734	5.389	**15.607**	**5.121**	**14.879**
	MD	5.768	17.465	5.405	16.158	5.178	15.127
	AL	**5.561**	**16.631**	5.338	15.805	5.256	15.414
	AA	5.817	17.834	5.559	16.172	5.640	15.425
	AF	5.795	17.276	**5.318**	15.668	5.231	15.161
	TS	6.497	19.179	5.684	16.560	5.350	15.053
	ViT	6.235	17.909	5.715	16.646	5.439	15.532

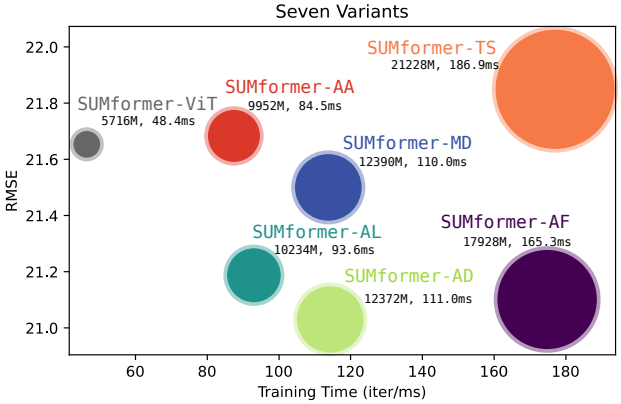


Fig. 6. Average memory usage, training speed, and performance for seven variants. Their sizes are proportional to their areas.

a comprehensive comparison of training speed, model size, and performance on TaxiBJ and CDtaxi in Fig. 6. Our key observations (Obs) are as follows:

Obs 1): SUMformer-AD and SUMformer-AL consistently achieve the best performance across most scenarios. This underscores that linear attention techniques—whether leveraging a neural dictionary or linear layers to produce low-rank key-value pairs—maintain high accuracy while reducing computational demands. In many instances, their results even outshine those of SUMformer-AF, which incorporates full attention. A possible explanation is that **the correlation between the urban mobility data is inherently low-rank**, and full attention models might face challenges in grasping this structure. The suboptimal results of SUMformer-AA, which uses additive attention, suggest that utilizing a global vector via additive attention might not be the optimal approach for capturing the low-rank correlations among variables. ***Obs 2):*** SUMformer-MD, which leverages MLP within the Temporal Sub-Block to extract temporal correlations, underperforms compared to SUMformer-AD. This implies that MHSA is more effective than MLP at extracting temporal correlations. ***Obs 3):*** SUMformer-TS, which omits the Temporal Sub-Block and relies solely on the Neural Dictionary to extract both temporal and cross-variable correlations, is the slowest method and yields the poorest prediction results. While SUMformer-ViT is the fastest method, thanks to its avoidance of computing attention scores for all variables, its prediction accuracy lags behind most other variants, with the exception of SUMformer-

TS and SUMformer-AA. This suggests that a two-stage approach to modeling temporal and cross-variable correlations step-by-step is well-suited for urban mobility data.

We conducted ablation testing to assess the effectiveness of the SUMformer architecture. We explored four ablation strategies on **TSB**, **ISSB**, **LFSB** as well as **Patch Merge mechanism** and assessed their impact on 128-step-ahead predictions using the TaxiBJ and CDtaxi datasets.

F. Ablation Study

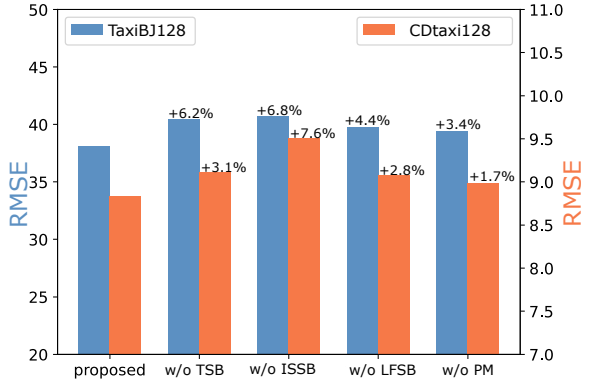
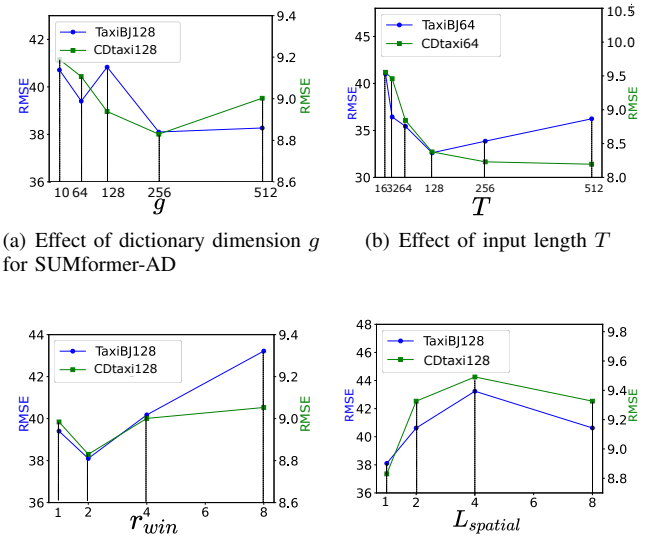


Fig. 7. Main component ablation for SUMformer

Fig. 7 presents the results of the ablation study. All experiments were conducted using the SUMformer-AD as the baseline for evaluation. In summary, each component of the SUMformer—TSB, ISSB, LFSB and the Patch Merge mechanism, holds significant importance in ensuring accurate predictions. The ablation study highlights the critical importance of each component. Omitting any of them consistently results in increased prediction errors, with the ISSB standing out as the most crucial. We observed a 6.8% increase in RMSE for TaxiBJ and a 7.6% increase for CDtaxi. This underscores the significance of modeling inter-series correlations within urban mobility data. The ISSB allows the model to explicitly capture correlations among variables. This ensures that predictions are not just based on individual variables but also encompass a comprehensive understanding of global patterns. Furthermore, our observations indicate that removing the TSB results in a less significant performance reduction compared to eliminating the ISSB. This indicates that the model is capable of conducting implicit temporal modeling through the synergistic effects of ISSB, LFSB, and particularly the patch merge mechanism. It's plausible that the main contribution to this capability arises from the patch merge mechanism, which merges temporally neighboring patches using linear layers, thereby implicitly modeling temporal correlations.

G. Effect of hyper-parameter

*1) **Selection of dictionary dimension** g **for SUMformer-AD**:* Fig. 8(a) shows the forecasting accuracy, measured in terms of the Root Mean Square Error (RMSE), for a 128



(a) Effect of dictionary dimension g for SUMformer-AD

(b) Effect of input length T

(c) Effect of the length of the patch merged window r_{win}

(d) Effect of spatial patch size $L_{spatial}$ for SUMformer-ViT

Fig. 8. Evaluation of hyper-parameter impact on prediction accuracy.

step-ahead prediction on both the TaxiBJ and CDtaxi datasets using the SUMformer-AD model with varying dictionary dimensions. Two datasets, TaxiBJ and CDtaxi, are plotted using blue and green lines, respectively. For both datasets, a dictionary dimension of $g = 256$ appears to yield the best forecasting accuracy. Lower dimensions, particularly $g = 10$, lead to inferior performance, indicating a requisite threshold for the dictionary size to capture cross-variable correlation effectively. Notably, while the performance for CDtaxi diminishes considerably for larger g values such as 256 and 512, the TaxiBJ dataset maintains a relatively consistent performance, highlighting inherent differences in the data characteristics of the two datasets. Note that both datasets contain over 2,000 variables. The fact that the model with $g = 256$ achieves the best performance suggests that both datasets exhibit low-rank cross-variable correlations.

*2) **Performances under Longer Input Sequences**:* A proficient time series forecasting model should accurately capture dependencies over extended review windows, thereby enhancing its results. A previous study [45] showed that Transformer-based models can exhibit significant fluctuations in performance, resulting in either a decline in overall efficiency or decreased stability with longer review windows. We conducted a similar analysis on both the TaxiBJ and CDtaxi datasets using the SUMformer-AD model. Various input lengths, specifically $\{16, 32, 64, 128, 256, 512\}$, were employed to forecast the values for the subsequent 64 time steps. Detailed results are presented in Fig. 8(b). SUMformer-AD also integrates a MHSA to extract temporal dependencies. However, unlike some prior models that may overfit to temporal noise, SUMformer-AD adeptly captures temporal information. While our model's performance is marginally suboptimal on TaxiBJ for input sequence lengths exceeding 64, there is a general trend of diminishing error. We believe this can be attributed to the SUMformer's temporal transformer

operating within the temporal patch itself. By condensing long sequences into shorter ones via the temporal patch partition, it effectively counters the transformer's inherent limitations in grasping prolonged sequence correlations within time series.

*3) **Performance Across Different Patch Merge Windows**:* Fig. 8(c) illustrates the relationship between the rise in r_{win} and the number of patch merges, given that N_{seg} remains constant at 16 and the network depth is fixed at 4. For uniformity in network depth, when r_{win} is set to 2, patch merges are executed at every layer. With r_{win} set to 4, patch merges occur at the first and third layers, while for r_{win} at 8, they are only conducted at the initial layer. The findings indicate that an r_{win} value of 2 yields the most favorable outcomes. The model without patch merge at $r_{win} = 1$ underperforms, emphasizing the critical role of patch merges in grasping multi-scale correlations. Conversely, a rise in r_{win} causes a decline in performance when $r_{win} > 2$, suggesting the model's essentiality to discern small-scale nuances.

*4) **Performance with Different Spatial Patch Size for SUMformer-ViT**:* To further validate that the multivariate time series perspective is better suited for urban mobility data, Fig. 8(d) displays the RMSE values in relation to various spatial patch sizes. For simplicity, we present the results of SUMformer-AD in Fig. 8(d) when the spatial patch size equals 1. For the TaxiBJ dataset, as the $L_{spatial}$ value increases from 1 to 2, the RMSE significantly climbs. However, as the $L_{spatial}$ continues to grow, the RMSE mildly declines, peaking at $L_{spatial} = 4$ and then slightly dropping at $L_{spatial} = 8$. A similar trend is observed for the CDtaxi dataset. This supports our hypothesis. Although setting the patch size to 8 introduces more larger range correlations during token embedding computation, its performance is still not as good as when the patch size is 1. This suggests treating the data of each grid as an independent variable and then extracting downstream spatial correlations is a better approach for the grid-based urban mobility data.

H. Effect of LFSB

We observed an interesting phenomenon after incorporating the LFSB (Low-frequency Filter Sub-Block), which is that the model's prediction accuracy for peak traffic flow has significantly improved. Peak traffic times are crucial since they are when congestion is most likely to occur, leading to longer travel times, increased fuel consumption, and higher emissions. Being able to predict these peak periods accurately is vital for implementing effective traffic management strategies to mitigate congestion [60].

We visualize the peak prediction bias in Fig. 9 where it is evident that the SUMformer without LFSB has a tendency to underestimate peak traffic volumes. Such underestimation could potentially introduce risks to the decision-making processes of intelligent transportation systems. The LFSB aids the model in better learning the temporal changes of traffic flow from low-frequency information by enforcing the loss of superfluous high-frequency details. Consequently, it facilitates more accurate estimation of traffic peaks.

Additionally, we offer a quantitative comparison of forecasting errors during peak hours, which are defined as follows:

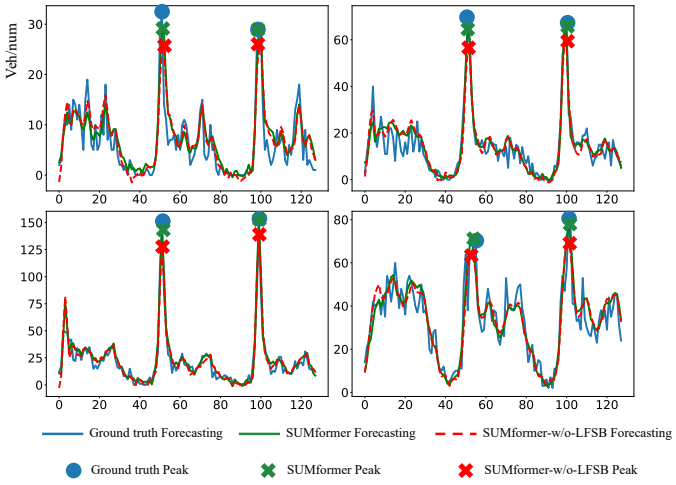


Fig. 9. When the LFSB added, the prediction accuracy on the two highest peak of traffic flow is well estimated.

We only consider the scenario when forecasting the next 128 time steps. The time interval for all five datasets used is 30 minutes. Therefore, a sequence of 48 steps corresponds to a daily traffic pattern. Consequently, within the subsequent 128 steps, there are at least two complete days. Inspired by the Seq2Peak approach [61], we utilize a 1D max pooling layer with a window size and stride of 64 to divide the input into two windows and select the maximum traffic value from each, which serves as the daily peak value:

$$\boldsymbol{X}_{\boldsymbol{peak}} = \text{MaxPooling1D}\left(\boldsymbol{X}\right), \tag{12}$$

where $\boldsymbol{X} \in \mathbb{R}^{G \times T}$ and $\boldsymbol{X}_{\boldsymbol{peak}} \in \mathbb{R}^{G \times 2}$. We apply the same procedure to the output of different forecasting methods and obtain $\boldsymbol{X}'_{\boldsymbol{peak}}$. Then we evaluated each forecasting methods on daily peak value by MAE and RMSE as shown in Tab. VI.

We observe that in all five datasets, LFSB aids SUMformer in improving the accuracy of predicting the daily peak. Additionally, compared to the other two deep models, PatchTST and SimVP, our model demonstrates superior capability in capturing the daily peak. Furthermore, we discovered that WH (Weekly History) outperforms both SimVP and PatchTST on the CDtaxi and NYCtaxi datasets. The possible reason is that the actual time ranges of these two datasets are relatively short, covering fewer peak hours. As a result, the model may overfit to non-peak hours, and when it comes to the more critical prediction of peak flows, it struggles to provide accurate predictions due to the scarcity of samples. Overall, SUMformer has a significant advantage over the vision-based SimVP and the variable-independent PatchTST in terms of peak hour prediction. Considering that peak flow prediction is more important, this further demonstrates the advantage of the super-multivariate time series view proposed in this paper.

I. Attention Score Visualization

To elucidate the SUMformer's proficiency in unraveling the inter-series correlations, we turn our focus to the attention scores that pertain to the inflow of the Zhongguancun area—a

TABLE VI
MAE/RMSE COMPARISION ON DAILY PEAK FLOW PREDICTION

Methods	SUMformer		SUMformer-w/o-LFSB		PatchTST		SimVP		WH		DH	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TaxiBJ	27.019	45.542	28.078	47.013	32.014	50.397	29.047	49.276	34.455	66.842	39.296	69.551
CDtaxi	8.509	17.886	8.790	18.238	10.663	20.757	10.830	22.044	7.611	18.327	9.991	25.505
CDbike	43.102	114.900	44.677	120.938	47.964	128.930	50.524	135.339	57.847	177.906	47.169	149.036
NYCtaxi	12.034	30.495	12.711	32.696	15.748	39.341	15.118	36.522	11.736	34.403	20.569	63.278
NYCbike	3.041	8.245	3.121	8.571	4.332	11.712	5.399	13.604	3.699	12.078	6.687	20.968

hub of academic institutions and central business districts. Fig. 10 offers a raw glimpse into these scores. For the sake of clarity in our visual exposition, we have employed SUMformer-AF for visualization.

A cursory glance at the attention maps reveals a nuanced picture. The SUMformer, with its super-multivariate lens, brings to light the point-to-point relationships that remain elusive to other architectures like ViTs and CNNs. The principal revelation from the figure is the attention map's innate capacity to autonomously spotlight key areas, the principal ring roads enveloping Beijing. These roads are not just mere strips of asphalt; they are the lifeblood of Zhongguancun's inflow, crucial arteries that dictate the pulse of traffic within the city. The attention map's ability to isolate these areas for scrutiny, absent any manual guidance, is indicative of the SUMformer's deep learning prowess. It identifies and accentuates areas where traffic congregates, providing invaluable insights into urban mobility patterns.

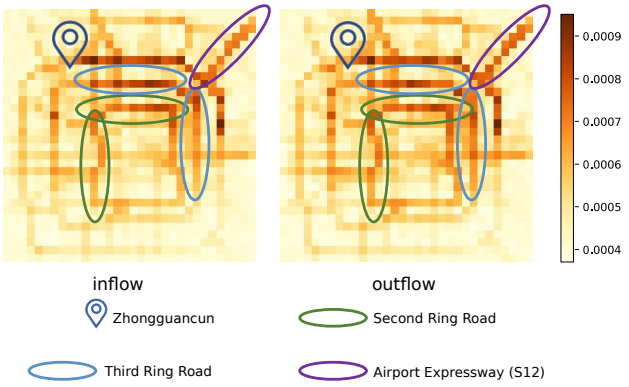


Fig. 10. The attention score reflects the relationship between the inflow to Zhongguancun in the Haidian District and the in/out flow to various other regions throughout Beijing.

V. CONCLUSION

In this study, we note that widely used CNNs and ViTs in video prediction architectures fall short in capturing crucial representations, as well as spatial and cross-channel correlations, essential for long-term grid-based urban mobility forecasting. To address this, we introduce SUMformer, a tailored backbone architecture specifically designed for grid-based urban mobility data, comprising three key components that focus on temporal dynamics, inter-series correlations, and frequency information. Our experimental results show that SUMformer delivers outstanding performance across five different datasets, demonstrating a remarkable versatility of the framework, as evidenced by our thorough analysis. Furthermore, our study reveals that grid-based urban mobility data represents a unique dataset within the domain of time series forecasting research, offering a more extensive array of data series than those typically encountered in existing datasets. In the future, we aim to extend the application of SUMformer to a broader range of video prediction tasks, including those involving AI4science datasets, which cover phenomena like weather patterns and fluid dynamics.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Sichuan Province (No. 2023NSFSC1423, No. 24ZHSF0023), the Fundamental Research Funds for the Central Universities and the Natural Science Foundation of China (No. 62371324). We would like to thank Wang Kun from the University of Science and Technology of China for his helpful suggestions. We also acknowledge the generous contributions of dataset donors.

REFERENCES

[1] S. Wang, J. Cao, and S. Y. Philip, "Deep learning for spatio-temporal data mining: A survey," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 8, pp. 3681–3700, 2020.

[2] 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," *arXiv preprint arXiv:2303.14483*, 2023.

[3] J. Wang, J. Jiang, W. Jiang, C. Han, and W. X. Zhao, "Towards efficient and comprehensive urban spatial-temporal prediction: A unified library and performance benchmark," *arXiv preprint arXiv:2304.14343*, 2023.

[4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[5] L. Liu, R. Zhang, J. Peng, G. Li, B. Du, and L. Lin, "Attentive crowd flow machines," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1553–1561.

[6] H. Lin, R. Bai, W. Jia, X. Yang, and Y. You, "Preserving dynamic attention for long-term spatial-temporal prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 36–46.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[9] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[10] Z. Gao, C. Tan, L. Wu, and S. Z. Li, "Simvp: Simpler yet better video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3170–3180.

[11] S. Tang, C. Li, P. Zhang, and R. Tang, "Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 470–13 479.

[12] V. L. Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 474–11 484.

[13] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, "Efficient and information-preserving future frame prediction and beyond," in *International Conference on Learning Representations*, 2019.

[14] C. Tan, Z. Gao, L. Wu, Y. Xu, J. Xia, S. Li, and S. Z. Li, "Temporal attention unit: Towards efficient spatiotemporal predictive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 770–18 782.

[15] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention convlstm for spatiotemporal prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 531–11 538.

[16] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d lstm: A model for video prediction and beyond," in *International conference on learning representations*, 2018.

[17] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, "Crevnet: Conditionally reversible video prediction," *arXiv preprint arXiv:1910.11577*, 2019.

[18] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[19] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3202–3211.

[20] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.

[21] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2286–2296.

[22] Y. Liang, K. Ouyang, Y. Wang, Y. Liu, J. Zhang, Y. Zheng, and D. S. Rosenblum, "Revisiting convolutional neural networks for citywide crowd flow analytics," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*. Springer, 2021, pp. 578–594.

[23] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[24] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, pp. 1–55, 2014.

[25] D. E. Boyce and H. C. Williams, *Forecasting urban travel: Past, present and future*. Edward Elgar Publishing, 2015.

[26] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 3744–3753.

[27] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[28] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," *arXiv preprint arXiv:2108.09084*, 2021.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[30] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1020–1027.

[31] Z. Xu, Y. Wang, M. Long, J. Wang, and M. KLiss, "Predcnn: Predictive learning with cascade convolutions." in *IJCAI*, 2018, pp. 2940–2947.

[32] H. Zhang, Y. Wu, H. Tan, H. Dong, F. Ding, and B. Ran, "Understanding and modeling urban mobility dynamics via disentangled representation learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2010–2020, 2020.

[33] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," *Advances in neural information processing systems*, vol. 30, 2017.

[34] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip, "Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5123–5132.

[35] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9154–9162.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale. arxiv 2020," *arXiv preprint arXiv:2010.11929*, 2010.

[38] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[39] Z. Zhang, Z. Huang, Z. Hu, X. Zhao, W. Wang, Z. Liu, J. Zhang, S. J. Qin, and H. Zhao, "Mlpst: Mlp is all you need for spatio-temporal prediction," *arXiv preprint arXiv:2309.13363*, 2023.

[40] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3d neural networks," *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.

[41] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 419–22 430, 2021.

[42] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 27 268–27 286.

[43] Z. Zhang, X. Wang, and Y. Gu, "Sageformer: Series-aware graph-enhanced transformers for multivariate time series forecasting," *arXiv preprint arXiv:2307.01616*, 2023.

[44] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[45] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[46] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[47] L. Han, H.-J. Ye, and D.-C. Zhan, "The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting," *arXiv preprint arXiv:2304.05206*, 2023.

[48] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The Eleventh International Conference on Learning Representations*, 2022.

[49] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[50] Y. Liu, C. Li, J. Wang, and M. Long, "Koopa: Learning non-stationary time series dynamics with koopman predictors," *arXiv preprint arXiv:2305.18803*, 2023.

[51] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.

[52] W. Johnny, H. Brigido, M. Ladeira, and J. C. F. Souza, "Fourier neural operator for image classification," in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2022, pp. 1–6.

[53] C. Yang, X. Chen, L. Sun, H. Yang, and Y. Wu, "Enhancing representation learning for periodic time series with floss: A frequency domain regularization approach," *arXiv preprint arXiv:2308.01011*, 2023.

[54] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli *et al.*, "Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators," *arXiv preprint arXiv:2202.11214*, 2022.

[55] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," *arXiv preprint arXiv:2202.01575*, 2022.

[56] Y. Liang, K. Ouyang, J. Sun, Y. Wang, J. Zhang, Y. Zheng, D. Rosenblum, and R. Zimmermann, "Fine-grained urban flow prediction," in *Proceedings of the Web Conference 2021*, 2021, pp. 1833–1845.

[57] Z. Gao, X. Shi, H. Wang, Y. Zhu, Y. B. Wang, M. Li, and D.-Y. Yeung, "Earthformer: Exploring space-time transformers for earth system fore-

casting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 390–25 403, 2022.
[58] M. Schläpfer, L. Dong, K. O'Keeffe, P. Santi, M. Szell, H. Salat, S. Anklesaria, M. Vazifeh, C. Ratti, and G. B. West, "The universal visitation law of human mobility," *Nature*, vol. 593, no. 7860, pp. 522–527, 2021.
[59] X. Wang and L. Sun, "Anti-circulant dynamic mode decomposition with sparsity-promoting for highway traffic dynamics analysis," *Transportation Research Part C: Emerging Technologies*, vol. 153, p. 104178, 2023.
[60] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.
[61] Z. Zhang, X. Wang, J. Xie, H. Zhang, and Y. Gu, "Unlocking the potential of deep learning in peak-hour series forecasting," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 4415–4419.

Jinguo Cheng received a bachelor's degree from Sichuan University in 2023, and he is currently continuing his studies in Computer Science and Technology at National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University. His main research interests include spatial-temporal data mining and ML4Science.

Ke Li is a full-time research staff member at the National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University. He is mainly engaged in research and development in fields such as image and video processing, pattern recognition, and computer vision. His specific research directions include video analysis and processing, deep learning, and face recognition. As the principal investigator or main researcher, he has participated in more than ten national research projects, and published multiple papers in international and domestic academic journals and conferences.

Yuxuan Liang is currently a tenure-track Assistant Professor at Hong Kong University of Science and Technology (Guangzhou), working on the research, development, and innovation of spatio-temporal data mining and AI, with a broad range of applications in urban computing. Prior to that, he obtained his PhD degree at the School of Computing, National University of Singapore. He has published over 60 papers in refereed journals (e.g., TPAMI, AI, TKDE) and conferences (e.g., KDD, NeurIPS, ICLR, WWW). These publications have attracted 3,600 citations with h-index of 29. He has served as a program committee member for prestigious conferences, such as KDD, ICML, ICLR, NeurIPS, WWW, CVPR, and ICCV. He has served as organizer or co-chair of Workshop on Urban Computing (Urbcomp@KDD-23) and Workshop on AI for Time Series (AI4TS@IJCAI-24). He has received The 23rd China Patent Excellence Award and SDSC Dissertation Research Fellowship 2020.

Lijun Sun received B.S. degree in Civil Engineering from Tsinghua University, Beijing, China, in 2011 and Ph.D. degree in Civil Engineering (Transportation) from the National University of Singapore in 2015. He is currently an Associate Professor and William Dawson Scholar with the Department of Civil Engineering at McGill University, Montreal, QC, Canada. His research centers on intelligent transportation systems, machine learning, spatiotemporal modeling, travel behavior, and agent-based simulation. He is an Associate Editor of Transportation Research Part C: Emerging Technologies.

Junchi Yan (S'10-M'11-SM'21) is a Tenured Professor with Department of Computer Science and Engineering, and AI College of Shanghai Jiao Tong University, Shanghai, China. Before that, he was a Senior Research Staff Member with IBM Research where he started his career since April 2011. His research interests include machine learning and its applications. He regularly serves as Senior PC/Area Chair for NeurIPS, ICML, ICLR, CVPR, AAAI, IJCAI, and Associate Editor for the Pattern Recognition.

Yuankai Wu (Senior Member, IEEE) received the Ph.D. degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2019. He is a tenure-track professor at the College of Computer Science, Sichuan University, China. Prior to joining Sichuan University in March 2022, he was an IVADO postdoc researcher with the Department of Civil Engineering, McGill University. His research interests include spatio-temporal data mining, intelligent transportation systems, and intelligent decision-making.