Adaptive Traffic Forecasting on Daily Basis: A Spatio-Temporal Context Learning Approach

Xiaoyu Li, Yitian Zhang, Guodong Long, Yupeng Hu, *Member, IEEE*, Wenpeng Lu, *Member, IEEE*, Meng Chen, *Member, IEEE*, Chengqi Zhang, *Senior Member, IEEE*, and Yongshun Gong*, *Member, IEEE*

***Abstract*—Traffic forecasting plays a crucial role in establishing an Intelligent Transportation System (ITS) by providing essential insights. Existing traffic forecasting relies on the assumption that there is a hidden invariant spatial-temporal pattern in the large-scale dataset. However, the traffic patterns are easily influenced by many unpredictable external factors, such as policy interventions and climate changes. Due to the dynamic nature of these exogenous factors, the traffic network's spatial-temporal patterns are also changed, thus impacting the performance of traffic forecasting models. Thus, there is an urgent need to rethink the traffic forecasting model in a fast-adaptive manner. To solve this challenge, this paper proposes an Adaptive Spatio-Temporal Context Learning framework named ASTCL, which achieves desired forecasting accuracy using daily basis traffic data collected from dozens of sensors. ASTCL constructs adaptive spatio-temporal contexts for target locations in the traffic network and generates dynamic sequence graphs based on semantic similarities. The adaptive contexts aggregate valuable information from available data, while the graphs reveal dynamic trends in traffic properties. Further, ASTCL introduces a joint convolution and attention mechanism to model intricate spatio-temporal relationships from multiple perspectives. Extensive experiments conducted on four real-world datasets demonstrate that ASTCL achieves remarkable fast adaptability and outperforms other state-of-the-art methods by a significant margin.**

***Index Terms*—Spatio-temporal data mining, traffic forecasting, adaptive learning**

I. Introduction

With the rapid growth and urbanization of modern cities, traffic forecasting has become a critical element of intelligent transportation systems and smart city initiatives. Accurate traffic forecasting is essential for various applications, including congestion management, route planning, and emergency response. By learning from historical data, urban traffic systems aim to predict the future status of urban traffic networks, such as traffic flow, speed, and density [1]. Effective traffic forecasting can significantly enhance the efficiency of urban transportation systems, reduce travel times, and improve the overall quality of life for city inhabitants.

Recently, there has been a surge in research efforts focused on traffic forecasting, driven by the increasing complexity of urban transportation systems. These studies often recognize traffic networks as spatio-temporal graphs, where traffic sensors correspond to nodes, and the dependencies between these nodes correspond to edges [2]. Graph Neural Networks (GNNs) have become a popular choice for traffic prediction due to their ability to aggregate information from neighboring nodes through message-passing mechanisms [1], [3], [4]. However, GNN-based models may suffer from over-smoothing, where node representations may become indistinguishable [5], [6]. Additionally, GNNs can be notoriously inefficient due to the high computational and memory demands associated with processing complex graph structures [7]. Additionally, inspired by the success of attention mechanisms in spatio-temopral modeling, several attention-based models have been proposed [8], [9]. These methods leverage attention mechanisms to enhance model performance weighting the importance of spatial and temporal features.

Existing traffic forecasting methods are based on the assumption that there is a hidden invariant spatial-temporal pattern in the large-scale dataset. However, unforeseen distribution shifts can occur between historical training and future testing data due to the dynamic nature of spatio-temporal data [10]. The traffic patterns are very easily influenced by many external factors, such as social events which are unpredictable, daily weather impacted by climate changes, construction plans of the local area, and the economic development of the city. For example, during the COVID-19 pandemic, abrupt reductions in traffic volumes occurred due to widespread lockdowns and restrictions. These impacts may lead to sudden shifts in traffic patterns, thereby undermining the effectiveness of forecasting models. To address this challenge, we explore traffic forecasting from a novel perspective, emphasizing the importance of learning effective spatio-temporal representations in a fast-adaptive manner. We introduce the adaptive traffic forecasting on daily basis, using one day of data to train a model suitable for dynamic traffic situations. This daily adaptation enables the model to swiftly adjust to ever-changing

This work was supported in part by the National Natural Science Foundation of China (62476154, 62202270), in part by the Major Basic Research Project of Shandong Provincial Natural Science Foundation (ZR2024ZD03), in part by the Shandong Excellent Young Scientists Fund (Oversea)(2022HWYQ-044), in part by the Taishan Scholar Project of Shandong Province (tsqn202306066), and in part by the Qilu Young Scholar Project of Shandong University.

Xiaoyu Li, Yupeng Hu, Meng Chen and Yongshun Gong are with the School of Software, Shandong University, Jinan, China (e-mail: xyuli@mail.sdu.edu.cn; {huyupeng, mchen, ysgong}@sdu.edu.cn).

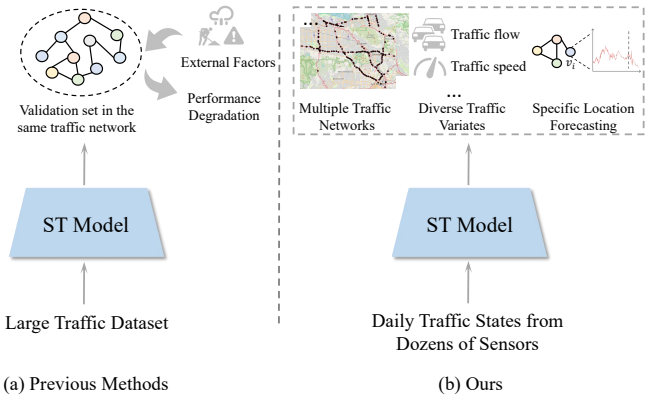
Yitian Zhang is with the Electrical and Computer Engineering, McGill University, Montreal, Canada (e-mail: yitian.zhang@mail.mcgill.ca).

Guodong Long, and Chengqi Zhang are with the School of Computer Science, University of Technology Sydney, Sydney, NSW, 2007, Australia (e-mail: {guodong.long, chengqi.zhang}@uts.edu.au).

Wenpeng Lu is with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences); Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan, China (e-mail: wenpeng.lu@qlu.edu.cn).

Our code is publicly available in https://github.com/Junzhidian/ASTCL.

*Corresponding authors.



(a) Previous Methods (b) Ours

Fig. 1: (a) The traffic forecasting framework in previous methods involves training the model on a large traffic dataset and evaluating it on the same traffic network; (b) Our proposed approach utilizes daily traffic data collected from dozens of sensors to train a model that is adaptive to various applications.

traffic patterns influenced by external factors. By training on daily data, the model captures the latest traffic trends and promptly responds to sudden changes, such as those caused by special events or weather anomalies. This real-time responsiveness is essential for maintaining accurate and reliable traffic forecasts in dynamic environments. The proposed approach not only achieves desired forecasting accuracy but is also adaptable to multiple application scenarios, such as cross-city and cross-domain prediction, as shown in Fig 1.

Differing from previous methods that exploit invariant spatio-temporal patterns from long historical data, our method highlights current traffic information and the application background. Specifically, we propose a novel adaptive spatio-temporal context learning method, named ASTCL, to achieve precise traffic forecasting on daily basis. We introduce a node-level forecasting task that takes the information aggregation of a target node (e.g. a traffic sensor) as the model input, rather than the entire traffic network. The motivation is to allow for targeted traffic predictions at specific locations of interest, which is more cost-effective than forecasting city-wide traffic. The trained model can also be applied to other locations within the city or to other cities, without being constrained by a specific traffic network structure. Additionally, node-level prediction can mitigate the issue of data scarcity in traffic forecasting on daily basis, as training samples increase proportionally with the number of traffic nodes in the graph. However, the information contained in an individual node is monotonous and sparse, leading to insufficient data diversity and temporal coverage. To overcome this limitation, we construct an adaptive spatio-temporal context for each target node to integrate information from a global perspective. The adaptive spatio-temporal context supplements the historical data from other available traffic sensors at a time fragment based on semantic similarities. This adaptive spatio-temporal context enriches the historical data by aggregating information from other traffic sensors that share semantic similarities within the same time fragment.

To model the relations between nodes within an adaptive context, we propose the dynamic sequence graph, where the edge weights are dynamic over the time series. Further, we introduce a novel spatio-temporal joint block, incorporating a joint attention mechanism and joint convolution, to capture intricate correlations from spatial, temporal, and spatio-temporal perspectives. It should be noted that ASTCL is a lightweight model with only thousands of parameters, implying low model complexity and computational demands. Additionally, extensive experiments are conducted on four real-world traffic datasets. The proposed ASTCL consistently outperforms other state-of-the-art baselines and achieves varying degrees of improvement across four datasets.

Our main contributions are summarised as follows:

- To the best of our knowledge, it is the first attempt to study the problem of traffic forecasting in a fast-adaptive manner. This novel perspective has the potential to advance future research.
- We tackle the challenging problem of traffic forecasting on daily basis and design a lightweight model with only thousands of parameters.
- We propose ASTCL, a novel spatio-temporal learning framework for traffic forecasting on daily basis. ASTCL leverages the adaptive spatio-temporal context through the joint attention mechanism and joint convolution to capture complex correlations and learns the evolving traffic patterns from dynamic sequence graphs.
- We perform extensive experiments on four real-world datasets. The experimental results show that our ASTCL achieves desired forecasting accuracy, consistently outperforming other state-of-the-art baselines.

The rest of this paper is organized as follows: Section II provides a literature review about traffic forecasting. Section III includes the related concepts. The proposed model is introduced in Section IV. The experimental results and analysis are shown in Section V. Finally, we summarize this work in Section VI.

II. Related Work

In this section, we discuss recent developments in the field of traffic forecasting in detail.

Numerous studies have been conducted in the field of traffic prediction due to its significance in transportation management. Traditional time series algorithms such as ARIMA [11] and its variants [12], [13] have long been used for traffic prediction. These methods model linear patterns based on historical data but often ignore spatial dependencies and complex traffic patterns. CNN-based models [14]–[16] apply convolutional networks to model the spatial dependencies among regions in grid-based traffic maps. This grid-based representation falls short in capturing the irregular and flexible node-wise connections inherent in spatio-temporal graphs [2].

Recent advancements in GCNs have significantly improved traffic forecasting by representing traffic sensor networks as graphs and employing graph convolutional networks to aggregate information from neighboring nodes. Various studies [17]–[21] have utilized GCNs to model spatial dependencies.

However, several of these models rely on pre-defined adjacency matrices based on road network distances, which can introduce biases and lack adaptability in domains without appropriate prior knowledge [1]. To overcome the limitations of predefined graphs, recent studies have focused on designing adaptive graphs that are learned during the training process. Methods such as those proposed by [1], [4], [22], [23] dynamically adjust the graph structure, enabling more flexible and accurate modeling of spatial dependencies. Given the computational complexity of GCNs, some researchers have explored the use of basic MLPs to alleviate computation bottlenecks. Studies like [7], [24], [25] have demonstrated the efficiency of MLPs in traffic prediction tasks, providing a simpler yet effective alternative to GCNs.

The attention mechanism has also been leveraged to enhance spatio-temporal representation learning. Methods such as [8], [9], [26]–[28] utilize attention mechanism to capture complex traffic patterns. However, manually designing models is time-consuming and often suboptimal, especially given the diverse nature of traffic scenarios across different regions and times. To tackle this problem, several studies [29]–[33]employ neural architecture search to discover neural network architectures for modeling spatio-temporal dependencies in traffic data. Besides that, some methods integrate self-supervised learning with spatio-temporal prediction to explore spatial and temporal heterogeneity [34]–[38].

There are some studies proposed to address the problem of data scarcity in traffic forecasting. Most approaches [2], [39]–[44] aim to train on extensive historical traffic data and transfer the knowledge to target networks. A common strategy involves pre-training a model on source domains with sufficient data and subsequently fine-tuning it on target domains with limited data. Additionally, some methods [45], [46] design algorithms to match regions from source cities to target cities based on feature similarity. The meta-learning paradigm [47] has also been explored to learn a well-generalized initialization of the network.

However, all these methods rely on the assumption that the learned spatio-temporal patterns are time-invariant. Our ASTCL is designed to address this limitation by focusing on daily traffic forecasting. It adapts to new and evolving conditions by leveraging one day of data to train a model that can be easily retrained or fine-tuned for dynamic traffic scenarios. This capability is particularly advantageous in practical scenarios where the data is dynamic, providing a more efficient and scalable solution for traffic forecasting.

III. Preliminaries

In this section, we provide the definitions of our prediction task, adaptive spatio-temporal context, and dynamic sequence graph.

A. Prediction Task

In this paper, we represent the historical traffic data collected from traffic sensors as $\mathbf{X} \in \mathbb{R}^{N \times T \times F}$, where N is the number of available sensors; T is the range of timestamps; and F denotes the features of interest (e.g. flow, speed, and occupancy). Given the overall traffic data, we divide the historical sequence into multiple intervals with a sliding window of size T_h and obtain $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{T_h}) \in \mathbb{R}^{N \times T_h \times F}$. Typically, $\mathcal{X}$ is directly fed into the model and undergoes several training iterations based on the size of T. However, with traffic forecasting on a daily basis, the constraint of having limited data for each day means that the value of T is typically limited to hundreds, generating hundreds of training samples. To complement this drawback, we formulate the training task of node-level forecasting. Specifically, for a target node (i.e. a traffic sensor) $\mathbf{v}_i \in \mathbb{R}^{T_h \times F}$ in the traffic network, the model takes its adaptive spatio-temporal context $\mathbf{C}$ (see in Section III-B) as input and predicts its traffic states of the next T_r steps, rather than predicting the entire traffic graph. Through this design, the training samples will increase multiple times based on the number of traffic sensors. The forecasting task is defined as:

$$(\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^{T_h}) \xrightarrow{g(\cdot)} (\hat{\mathbf{Y}}^{T_h+1}, \ldots, \hat{\mathbf{Y}}^{T_h+T_r}), \quad (1)$$

where $\hat{\mathbf{Y}}$ is the prediction results and $g(.)$ is the target forecasting function.

B. Adaptive ST Context

Although node-level forecasting expands the training samples, the traffic records of an individual node often lack diversity, thereby failing to provide valuable information for robust model training. Recent studies [48], [49] assume that the traffic states of a node are primarily influenced by the traffic states of its localized neighborhoods. These approaches construct the local or regional context of a target node by combining its adjacent nodes. However, they fail to account for global spatial influences.

To address this limitation, we introduce the concept of adaptive spatio-temporal (ST) context which aims to incorporate more diverse data into a target node from a global perspective. This approach allows us to consider the broader traffic environment and capture more comprehensive patterns that influence the target node. For each target node $\mathbf{v}_i$ over the time window T_h, the feature vector of the node is denoted as $\mathbf{V}_i = (\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots, \mathbf{v}_i^{T_h}) \in \mathbb{R}^{T_h \times F}$. Our objective is to adaptively identify a set of nodes that have a strong influence on the target node and integrate these nodes into the spatio-temporal context. To measure the correlation between the traffic states of different nodes, we employ fast Dynamic Time Warping (fastDTW) [50], an efficient algorithm based on the Dynamic Time Warping (DTW) [51] technique. This method evaluates the sequence similarity between two nodes over the same time interval. The value calculated by fastDTW serves as an indicator of the degree of traffic pattern similarity between nodes. The correlation weight between nodes $\mathbf{V}_i$ and $\mathbf{V}j$ is defined using the Gaussian kernel as follows:

$$S_{i,j} = exp(-\frac{\text{fDTW}(\mathbf{V}_i, \mathbf{V}_j)^2}{\sigma^2}) \in [0, 1], \quad (2)$$

where $S_{i,j}$ denotes the weight of the correlation between $\mathbf{V}_i$ and $\mathbf{V}_j$, with higher values indicating stronger relationships;

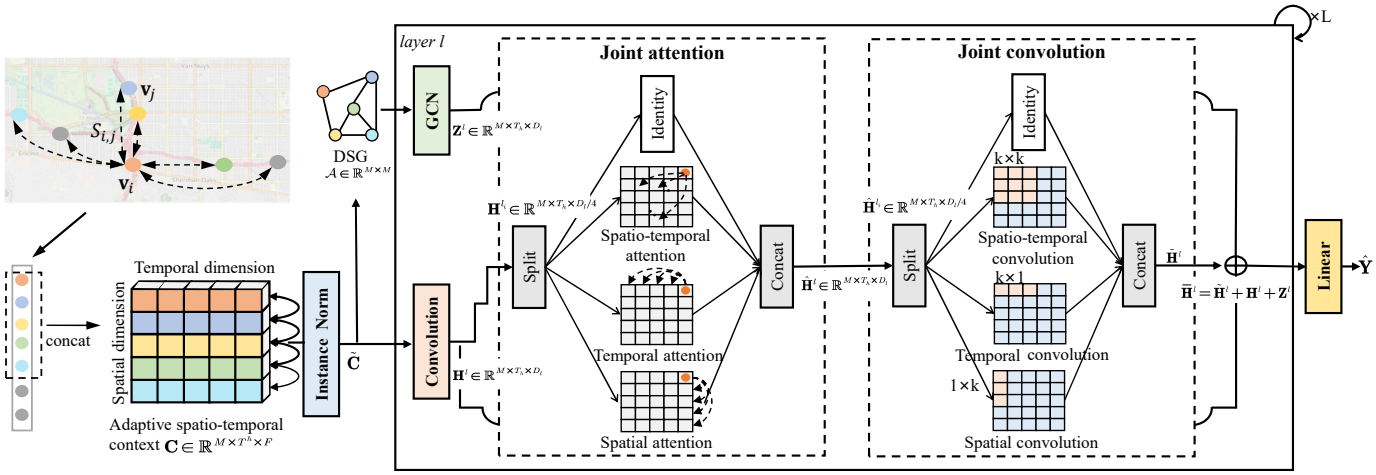


Fig. 2: The overall framework of ASTCL. DSG is the dynamic sequence graph and GCN is the graph convolution network.

fDTW$(\cdot)$ represents the fastDTW operation; and σ is the standard deviation among all fDTW$(\mathbf{V}_i, \mathbf{V}_j)$ values.

Based on the value of $S_{i,*}$, we select M nodes that exhibit the most similar traffic patterns to the target node. These nodes are concatenated to construct the adaptive spatio-temporal context $\mathbf{C} \in \mathbb{R}^{M \times T^h \times F}$. It is essential to include the target node itself in this adaptive context to ensure that the primary patterns of the target are retained. This method enables the model to dynamically and contextually select relevant information from the global traffic network, thereby enhancing the robustness of the predictions.

IV. METHOD

In this section, we introduce the details of ASTCL and showcase the overall framework of ASTCL in Fig 2.

A. Joint Spatial Temporal Learning

Our main task is to capture intricate spatio-temporal correlations from the adaptive spatio-temporal context and predict future trends of traffic states. In the time dimension, historical traffic data provides insights into recurring patterns and trends over time. In the spatial dimension, traffic conditions at a target node are often affected by other nodes in the adaptive context due to semantic similarities. Previous studies usually design separate spatial, temporal, and spatio-temporal encoders and serially stack them to capture features from each dimension. We notice that this stacking strategy not only increases the depth and complexity of the model but also leads to a loss of interactions across different dimensions. Inspired by work [52], we propose two novel joint operations: joint attention mechanism, and joint convolution, capable of learning spatial, temporal, and spatio-temporal representations in a single layer, while simultaneously maintaining comparable complexity. In the following subsections, we will introduce each component of ASTCL in detail.

1) Embedding initialization: Given the adaptive spatio-temporal context $\mathbf{C}$ as input, we first apply instance normalization [53] to stabilize the sequence. Specifically, we compute the mean and standard deviation for each spatio-temporal instance and normalize the adaptive spatio-temporal context as follows:

$$\mu = \frac{1}{T^h \times M} \sum_{t=1}^{T^h} \sum_{m=1}^{M} \mathbf{C}_{t,m} \in \mathbb{R}^F, \tag{3}$$

$$\sigma^2 = \frac{1}{T^h \times M} \sum_{t=1}^{T^h} \sum_{m=1}^{M} (\mathbf{C}_{t,m} - \mu)^2 \in \mathbb{R}^F, \tag{4}$$

$$\tilde{\mathbf{C}} = \frac{\mathbf{C} - \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{5}$$

where ϵ is a constant value of 1×10^{-5}. The normalized data will be de-normalized at the final stage of the model. Subsequently, we employ a convolutional layer to obtain the initial embedding of the normalized adaptive spatio-temporal context. It should be noted that we stack L layers of initial convolution, graph neural network, and joint operations to learn the high-level spatio-temporal representations as shown in Fig 2. In the first layer, we transform the dimension of the normalized adaptive spatio-temporal context $\tilde{\mathbf{C}}$ from F into D_1 and get the feature tensor $\mathbf{H}^1 \in \mathbb{R}^{M \times T_h \times D_1}$. For the outputs from $l-1$th layer, the computation is :

$$\mathbf{H}^l = f(\mathbf{W}^l * \overline{\mathbf{H}}^{l-1}), \tag{6}$$

where $\mathbf{H}^l \in \mathbb{R}^{M \times T_h \times D_l}$ is the projected representations; $\overline{\mathbf{H}}^{l-1}$ is the output from the $l-1$th layer; $*$ denotes the convolution operation; $f(\cdot)$ is an activation function; and $\mathbf{W}^l$ is the learnable parameters.

2) Joint attention mechanism: Drawing from the abilities of self-attention mechanism [54] in capturing long-range dependencies, we design a novel joint attention mechanism to effectively model the long-range spatial, temporal, and spatio-temporal correlations at one stage. Specifically, we split $\mathbf{H}^l$ into four parts on average according to the feature channels and get $\mathbf{H}^{l_1}$, $\mathbf{H}^{l_2}$, $\mathbf{H}^{l_3}$, and $\mathbf{H}^{l_4}$. Each of these partitions is a tensor with the same feature dimensions. These partitions are then processed using different attention mechanisms to capture various dependencies. For $\mathbf{H}^{l_1} \in \mathbb{R}^{M \times T_h \times D_l/4}$, we

employ the multi-head spatial self-attention to capture the spatial dependencies between nodes at the same time step. Formally, for time stamp t, we first derive the query, key, and value matrices as follows:

$$\mathbf{Q}_t = \mathbf{H}_t^{l_1}\mathbf{W}_Q, \mathbf{K}_t = \mathbf{H}_t^{l_1}\mathbf{W}_K, \mathbf{V}_t = \mathbf{H}_t^{l_1}\mathbf{W}_V, \quad (7)$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V \in \mathbb{R}^{D_l/4\times D'}$ are learnable parameters. Then we obtain the attention map of all nodes at time step t by self-attention operations. The spatial attention map demonstrates the time-invariant spatial properties by calculating the pairwise relations between different nodes at the same time step. Finally, we obtain the outputs from the spatial self-attention as:

$$\hat{\mathbf{H}}_t^{l_1} = \mathrm{softmax}(\frac{\mathbf{Q}_t\mathbf{K}_t^T}{\sqrt{D'}})\mathbf{V}_t. \quad (8)$$

Similarly, for $\mathbf{H}^{l_2}$, we apply a temporal self-attention mechanism to capture the unique traffic patterns of each node over the time dimension. For $\mathbf{H}^{l_3}$, we apply spatio-temporal attention to capture dynamic spatio-temporal correlations, providing a comprehensive understanding of how traffic patterns evolve over time and space. To maintain feature consistency and avoid excessive complexity, we apply an identity mapping to the last partition $\mathbf{H}^{l_4}$. After concatenating all the outputs, we append an extra channel shuffling to rearrange the learned features and obtain the output $\hat{\mathbf{H}}^l$.

3) Joint convolution: After the attention module, we design a joint convolution operation to capture the short-range spatio-temporal correlations. A similar strategy is applied to split the $\hat{\mathbf{H}}^l$ and different convolution modules are employed to these partitions. We use three different kernel sizes $1\times k$, $k\times 1$, and $k\times k$ corresponding to the spatial, temporal, and spatio-temporal perspectives, respectively. Finally, we add a residual connection to the outputs as:

$$\tilde{\mathbf{H}}^l = \mathcal{CS}([\mathbf{W}_1 * \hat{\mathbf{H}}^{l_1}, \mathbf{W}_2 * \hat{\mathbf{H}}^{l_2}, \mathbf{W}_3 * \hat{\mathbf{H}}^{l_3}, \hat{\mathbf{H}}^{l_4}]) + \mathbf{H}^l, \quad (9)$$

where $\mathbf{W}_1, \mathbf{W}_2$, and $\mathbf{W}_3$ are learnable parameters of different convolutions; $*$ is the convolution; $\mathcal{CS}$ is the channel shuffling; and $\tilde{\mathbf{H}}^l$ is the learned spatio-temporal representations.

B. Dynamic Sequence Graph Modeling

Graph neural networks are widely used to capture the spatial dependencies in traffic forecasting. The common method to build the traffic graph involves pre-defining an inter-connection graph through similarity or distance measures [1], [55]. The generated graph in this manner is normally time-invariant and applied to all timestamps as shown in Fig 3 (a). Similarly, the adaptive graphs [1] learning from node embedding also remain unchanged after the training phase. To better capture traffic dynamics, recent studies [22], [56] have proposed constructing dynamic graphs at each timestamp as shown in Fig3 (b). However, they often neglect the interactions between different time intervals, leading to a loss of valuable temporal dependency information.

We propose a novel approach, the Dynamic Sequence Graph (DSG), to address this challenge. The DSG aims to model complex relationships between nodes across the entire input sequence, capturing both spatial and temporal dependencies dynamically. We compute the pairwise edge weight between nodes in a fragment by equation 2 and use a threshold to reduce the graph density. In this way, the edge weights can reflect the dynamic relationships between nodes in both spatial and temporal dimensions. The construction of DSG is already completed during the phase of data preparation and does not incur additional overhead in training.

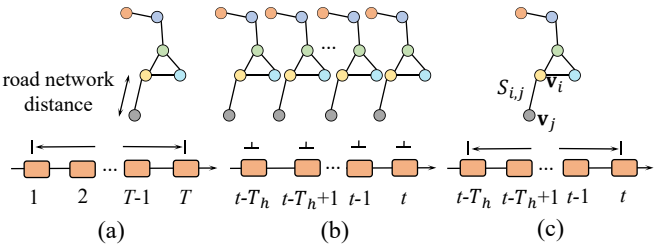


Fig. 3: (a) Pre-defined traffic graph based on the road network. (b) Dynamic graphs at each time stamp. (c) Dynamic sequence graph in a time fragment.

Therefore, we can get the dynamic sequence graph for the adaptive spatio-temporal context $\tilde{\mathbf{C}}$ and derive the corresponding adjacency matrix $\mathcal{A} \in \mathbb{R}^{M\times M}$. To extract features from this graph, we employ the graph convolution network proposed in study [4]. The GNNs aim to smooth the node' signal by aggregating and transforming its neighborhood information [4]. Given the hop K, the computation at layer l is as following:

$$\mathbf{Z}^l = \sum_{k=0}^{K} \mathbf{P}^k\mathbf{C}\mathbf{W}_\mathbf{k}, \quad (10)$$

where $\mathbf{W}_k$ is the learnable parameter and $\mathbf{P}^k$ denotes the power series of the transition matrix. We generate $\mathbf{P}$ by $\mathbf{P} = \mathcal{A}/rowsum(\mathcal{A})$. Finally, we integrate the features from the dynamic sequence graph modeling, joint operations and residual connections into a unified representation:

$$\overline{\mathbf{H}}^l = \mathbf{H}^l + \tilde{\mathbf{H}}^l + \mathbf{Z}^l, \quad (11)$$

where $\overline{\mathbf{H}}^l$ is the output at the lth layer.

C. Predictor

After processing the input through the stacked L joint blocks, we obtain the final high-level spatio-temporal features $\overline{\mathbf{H}}^L$. To generate multi-step traffic predictions, we apply an MLP to transform the hidden state as:

$$\hat{\mathbf{Y}} = \mathrm{MLP}((\overline{\mathbf{H}}^L)), \quad (12)$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{T_r\times F}$ denotes the prediction results of the next T_r steps. To ensure the predictions are on the same scale as the original data, we perform de-normalization by adding the mean and standard deviation back to the output.

The model is trained by minimizing the Mean Absolute Error (MAE) between the traffic predictions $\hat{\mathbf{Y}}$ and the actual traffic states $\mathbf{Y}$.

$$\mathcal{L}(\theta) = |\hat{\mathbf{Y}} - \mathbf{Y}|, \quad (13)$$

where θ are learnable parameters in the ASTCL.

V. EXPERIMENTS

In this section, we evaluate the performance of ASTCL from different perspectives, summarizing the results to address the following research questions:

•*RQ1:* How does ASTCL perform compared to other baselines for traffic forecasting on daily basis?

•*RQ2:* How effective is ASTCL when it adapts to various prediction tasks?

•*RQ3:* How do different components and parameter settings affect the overall performance of ASTCL?

•*RQ4:* How is the complexity of ASTCL compared against other deep-learning methods?

A. Experimental Setup

Datasets: We evaluate the performance of our model on two types of public traffic datasets summarized in TABLE I. **METR-LA** [57] is the traffic data gathered from loop detectors in the highway of Los Angeles County [58]. The time span of this dataset is from Mar 1st to Jun 30th, 2012. **PEMS-BAY** [57] refers to the traffic data collected by California Transportation Agencies (CalTrans) Performance Measurement System (PEMS) from the Bay Area. It contains 6 months traffic data ranging from Jan 1st 2017 to May 31st, 2017. **PEMS04** [26] comprises traffic information collected from traffic sensors in San Francisco Bay Area. It spans from Jan 1st to Feb 28th, 2018. **PEMS08** [26] includes the traffic data in San Bernardino from Jul 1st to Aug 31th in 2016. To ensure a fair comparison, we adopted the same data processing and selection strategies as those used in previous studies [26], [57]. In our evaluations, each detector in the traffic network is represented as a node in the graph. The raw data, collected by detectors at 30-second intervals, were aggregated into five-minute time windows. Following the methodology of [26], redundant traffic sensors in the PEMS04 and PEMS08 datasets were removed to ensure that the distance between any pair of adjacent detectors exceeds 3.5 miles. This preprocessing step resulted in a refined subset of traffic records from the PEMS datasets, on which we evaluated the performance of our model.

Metrics: We assess the effectiveness of the proposal by employing three commonly used metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{Y}_i - \hat{\mathbf{Y}}_i\right)^2} \tag{14}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{Y}_i - \hat{\mathbf{Y}}_i\right| \tag{15}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\mathbf{Y}_i - \hat{\mathbf{Y}}_i}{\mathbf{Y}_i}\right| \tag{16}$$

where $\hat{\mathbf{Y}}_i$ is the prediction result, $\mathbf{Y}_i$ is the ground truth; and N is the number of prediction values. We conducted the experiments three times and reported the average results.

TABLE I: Statistics of Datasets.

Data type	Traffic flow		Traffic speed	
Dataset	PEMS04	PEMS08	METR-LA	PEMS-BAY
Nodes	307	170	207	325
Edges	340	295	1515	2369
Time steps	16,992	17,856	34,272	52,116
Time interval	5 min	5 min	5 min	5 min

Implementation: We implement the model with the PyTorch on a machine with the NVIDIA GeForce 4090 GPUs with 24GB GPU memory. With traffic forecasting on daily basis, we take the traffic data from **the first day** in each dataset as the training set for all baselines. The validation and test sets are consistent with previous work settings [24], [59]. The last 40% data of PEMS04 and PEMS08 are divided into validation and test sets in a fraction of 1:1. The last 30% data of METR-LA and PEMS-BAY are divided in a fraction of 1:2. We access the performance across three horizons: 15 minutes ($T_r = 3$), 30 minutes($T_r = 6$), and 60 minutes($T_r = 12$). After conducting a grid search, the number of nodes used to construct the adaptive context is set to 15. We tune the hyperparameters of the following two types: the number of layers and the hidden size in the model, and the best performance is on settings 2 and 32, respectively. We adopt the Adam optimizer [60] with an initial learning rate of 0.01 to train our model. The batch size is 1024, and the training epoch is 120. The early-stop mechanism is applied when the validation error converges within 15 continuous steps.

Baselines: We compare our model with two types of baselines: traffic flow prediction and traffic speed prediction. For a fair comparison, some deep-learning based baselines are evaluated on their native prediction task without being applied to the other.

Traffic flow prediction:

• ***HA***: Historical Average (HA) method. We take the average results of the past 12 time steps to forecast the next value.

• ***VAR*** [61]: Vector Auto-Regression (VAR) is a statistical model used to capture the linear interdependencies among multiple all traffic series. It generalizes the univariate autoregressive model to multivariate time series.

• ***ASTGCN*** [26]: ASTGCN combines the spatial-temporal attention mechanism and the spatial-temporal convolution to capture the recent, daily-periodic and weekly-periodic dependencies.

• ***AGCRN*** [1]: AGCRN incorporates a node-specific parameter learning module to identify unique patterns for each node, along with a data-driven graph generation module that deduces the inter-dependencies among various traffic series.

• ***STSGCN*** [62]: STSGCN is designed to capture the complex localized spatial-temporal correlations and heterogeneities in traffic data through a spatial-temporal synchronous modeling mechanism.

• ***Bi-STAT*** [27]: BI-STAT employs a dynamic halting module for iterative computation and utilizes a dual-decoder

TABLE II: Overall prediction performance of different methods on the traffic flow datasets (PEMS04 and PEMS08). The best result in each column is highlighted in **bold** and the second best is underlined.

Datasets	PEMS04									PEMS08								
Models	15 minutes			30 minutes			60 minutes			15 minutes			30 minutes			60 minutes		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	121.31	154.05	102.74	121.31	154.05	102.74	121.31	154.05	102.74	119.73	149.72	155.53	119.73	149.72	155.53	119.73	149.72	155.53
VAR	46.72	62.01	23.18	80.62	108.92	44.47	51.42	65.35	36.71	44.63	62.02	14.99	71.83	91.71	26.77	108.34	144.77	38.83
ASTGCN	41.08	58.85	31.78	42.34	60.67	32.00	46.90	67.30	38.42	23.40	35.27	17.95	25.08	38.05	18.94	31.16	45.91	22.88
AGCRN	160.25	213.83	100.67	160.74	214.41	100.22	160.87	214.58	100.12	160.33	207.24	108.17	159.48	206.33	108.76	159.24	206.13	108.83
STSGCN	41.96	64.12	30.23	44.04	66.43	31.65	50.22	73.91	34.93	35.13	53.83	24.60	36.56	55.74	25.23	42.05	62.93	27.57
Bi-STAT	33.28	51.20	21.19	37.30	58.01	23.63	47.15	72.65	30.25	22.86	33.10	16.34	24.76	36.45	17.92	32.13	46.79	23.09
STID	83.29	103.01	62.34	97.49	123.25	67.01	122.46	156.90	76.15	61.91	80.01	39.41	64.04	84.99	38.04	79.10	106.81	47.71
FourierGNN	29.37	44.12	24.50	37.65	55.26	31.85	48.47	70.89	39.57	24.01	35.60	16.45	30.27	44.82	20.29	43.56	62.56	26.83
PDFormer	32.00	44.65	33.26	79.41	115.50	121.85	96.94	131.52	155.95	65.35	91.46	108.86	93.03	121.78	176.17	101.42	129.15	197.72
STAEformer	82.15	107.78	58.66	92.23	119.36	71.45	96.54	125.44	74.51	38.40	51.44	30.76	44.24	58.84	35.11	52.38	69.45	35.88
ASTCL$_{20}$	22.20	34.88	14.54	26.25	40.96	16.77	34.41	52.79	22.09	18.01	27.60	11.53	20.86	32.54	13.08	28.54	43.43	17.59
ASTCL$_{50}$	21.67	34.19	14.20	25.28	39.58	16.28	33.55	51.41	21.99	16.84	26.10	10.80	19.74	30.87	12.61	26.44	40.78	16.70
ASTCL	**21.36**	**33.78**	**14.05**	**24.90**	**39.01**	**16.17**	**32.83**	**50.01**	**21.36**	**16.77**	**25.95**	**11.15**	**19.55**	**30.43**	**12.76**	**25.77**	**39.46**	**16.86**

approach to enhance predictions by recollecting past traffic conditions

• ***STID*** [24]: STID addresses the indistinguishability in both spatial and temporal dimensions by attaching spatial and temporal identity information.

• ***FourierGNN*** [63]: FourierGNN introduces the hypervariate graph regarding each series value as a graph node for multivariate time series forecasting. It includes a Fourier graph operator to perform matrix multiplications in Fourier space.

• ***PDFormer*** [9]: PDFormer captures dynamic spatial dependencies through a spatial self-attention module and graph masking matrices. It employs a delay-aware feature transformation module to model long-range spatial information along with traffic condition propagation delays.

• ***STAEformer*** [59]: STAEformer is a transformer-based model that leverages a novel spatio-temporal adaptive embedding component to capture intricate spatio-temporal patterns and chronological information.

The following five methods: HA, VAR, STID, FourierGNN, and STAEformer can be utilized for traffic speed prediction as well.

Traffic speed prediction:

• ***STGCN*** [3]: STGCN incorporates graph convolution and gated temporal convolution to model spatio-temporal correlations.

• ***DCRNN*** [57]: DCRNN utilizes the diffusion convolution and the sequence to sequence learning framework together with scheduled sampling to capture the spatial and temporal dependencies in traffic data.

• ***GWNET*** [4]: Graph WaveNet (GWNET) introduces an adaptive dependency matrix that is learned through node embedding and captures the spatial dependencies by the graph neural network. To model temporal correlations, it employs stacked dilated 1D convolution layers.

• ***GMAN*** [8]: GMAN consists of multiple spatio-temporal attention blocks to capture the impact of spatio-temporal factors on traffic conditions.

• ***MTGNN*** [17]: MTGNN introduces a general graph neural network framework to extract the uni-directed relations among variables while integrating external knowledge for multivariate time series data.

B. Performance Comparison

In this section, we evaluate the ASTCL by extensive experiments from various perspectives.

1) Traffic Forecasting Performance on Daily Basis(RQ1): From TABLE II and III, valuable observations can be summarized as following:

ASTCL consistently outperforms state-of-the-art traffic forecasting baselines across four datasets in terms of all evaluation metrics. This demonstrates the robustness and effectiveness of the model in handling various traffic conditions with minimal information. Notably, on the large error datasets, PEMS04 and PEMS08, ASTCL achieves a performance improvement of over 30% in the majority of cases highlighting its superior capability in traffic flow prediction. For METR-LA and PEMS-BAY, the proposed model also performs the best, achieving an improvement of up to 11.7% compared with

TABLE III: Overall prediction performance of different methods on the traffic speed datasets (METR-LA and PEMS-BAY). The best result in each column is highlighted in bold and the second best is underlined.

Datasets	METR-LA									PEMS-BAY								
Methods	15 minutes			30 minutes			60 minutes			15 minutes			30 minutes			60 minutes		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	11.30	15.86	30.99	11.30	15.86	30.99	11.30	15.86	30.99	6.42	9.12	14.71	6.42	9.12	14.71	6.42	9.12	14.71
VAR	7.74	12.50	23.02	8.10	11.25	19.15	7.15	10.79	18.18	4.24	6.86	7.55	4.33	6.39	7.39	5.32	8.10	8.61
STGCN	8.75	16.13	8.84	10.24	18.42	9.89	12.21	21.08	13.78	4.64	9.67	6.29	4.84	9.63	6.52	5.21	10.38	6.95
DCRNN	3.71	7.92	9.18	4.78	10.17	11.28	6.48	13.01	17.38	2.22	5.08	6.57	2.64	5.81	7.44	3.29	7.01	8.97
GWNET	3.24	6.09	8.44	3.83	7.58	11.01	4.90	9.51	15.49	1.76	3.73	4.52	2.24	4.92	5.71	2.97	6.78	7.53
GMAN	10.57	16.79	21.06	11.01	17.52	21.91	11.73	18.61	23.44	3.82	6.48	8.98	4.05	6.92	9.50	4.49	7.77	10.54
MTGNN	11.54	14.61	32.38	10.96	13.80	30.74	11.22	13.98	30.51	5.17	10.14	14.32	5.26	10.34	14.59	5.53	10.78	31.09
STID	5.48	10.07	19.08	5.84	10.51	20.30	6.26	11.38	22.21	2.94	6.05	8.18	4.07	9.03	11.96	4.80	10.28	13.84
FourierGNN	4.17	7.11	10.57	4.81	8.92	12.77	5.35	10.01	14.13	2.46	5.42	5.47	3.55	6.50	7.61	6.45	9.40	13.20
STAEformer	5.77	9.68	14.55	6.16	10.59	16.26	6.89	12.13	19.20	3.68	6.61	8.71	3.93	7.12	9.35	4.39	8.06	10.32
ASTCL$_{20}$	2.90	5.84	7.67	3.52	7.26	9.84	4.65	9.36	13.60	1.62	3.57	3.42	2.16	4.98	4.85	2.99	6.93	7.03
ASTCL$_{50}$	2.89	5.74	7.72	3.50	7.18	9.76	4.61	9.21	13.69	1.57	3.44	3.37	2.10	4.87	4.65	2.94	6.84	6.77
ASTCL	**2.81**	**5.71**	**7.50**	**3.43**	**7.16**	**9.65**	**4.46**	**9.10**	**13.17**	**1.56**	**3.42**	**3.33**	**2.07**	**4.83**	**4.62**	**2.89**	**6.71**	**6.74**

the second-best results. This significant margin underscores ASTCL's advanced ability to capture and predict complex traffic patterns.

Forecasting traffic states on daily basis poses significant challenges for both traditional statistical methods and advanced deep-learning models. Statistical methods like HA and VAR can be easily fitted with traffic data for one day of data. However, these methods are highly susceptible to outliers and fail to adapt to the dynamic and complex nature of traffic patterns. On the other hand, deep-learning models may suffer performance degradation on daily basis due to the risk of overfitting. For instance, as shown in TABLE II, AGCRN attempts to generate an adaptive graph through learnable node embeddings, while it struggles to derive meaningful representations and maintain performance with one day of data. A similar issue of generating embeddings arises in GWNET and STAEformer, critically hampering their ability to learn effective features. ASTGCN adopts a different approach by incorporating traffic data from previous days, such as recent days or weeks, an advantage not feasible with only a single day of available data. For traffic speed prediction, some deep learning models perform even worse than the statistical method VAR, highlighting the difficulties deep learning models face in capturing traffic patterns on daily basis. The proposed ASTCL leverages the adaptive spatio-temporal context and two types of joint encoders to enhance its ability in learning traffic patterns. Consequently, ASTCL makes significant progress in extracting traffic representations and achieves better results.

To sum up, extensive experiments prove that the investigated problem is challenging and has not been properly resolved at present. Our ASTCL provides novel solutions, exhibiting excellent prediction accuracy and robustness.

2) Comprehensive Evaluation of Adaptive Traffic Forecasting (RQ2): We provide a comprehensive evaluation of our ASTCL across various scenarios to demonstrate its versatility and robustness.

Traffic Prediction in a Few-shot setting. In the early stages of smart city development, many urban areas face the challenge of data scarcity due to insufficient deployment of traffic sensors. This allows us to introduce a few-shot traffic forecasting, where the model must adapt to and predict traffic patterns with minimal data. To simulate traffic networks with sparse traffic sensors, we randomly remove the traffic records of most nodes within the dataset, retaining only a small fraction of the data for training purposes. In the experiments, the validation and test sets, as well as other settings, remain consistent with the baselines to ensure fairness. Consequently, the model trained on a subset of sensors undergoes validation and testing across the entire traffic network, which means it is evaluated not only on future time series but also on previously unseen sensors. This experimental design significantly increases the problem's difficulty and has not been thoroughly explored in the context of traffic forecasting. The variants denoted as ASTCL_{50} and ASTCL_{20} in TABLE II and TABLE III, respectively, represent models trained with only 50% and 20% of the available traffic sensors. Taking the ASTCL_{20} on PEMS08 for example, the training set comprises solely one day's traffic data from 34 (20% scale of PEMS08) traffic sen-

TABLE IV: Cross-city Prediction Results. The backslash symbol \ in the table indicates that the value is more than one hundred.

Source Data	Target Data	Methods	15 minutes			30 minutes			60 minutes		
			MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	PEMS-BAY	TransGTR	2.25	5.44	6.87	2.81	6.50	8.35	3.46	7.71	10.12
		pFedGTP	1.78	3.45	3.89	2.32	4.77	5.23	3.21	6.58	7.38
		ASTCL	**1.52**	**3.31**	**3.25**	**2.05**	**4.68**	**4.59**	**2.86**	**6.51**	**6.74**
PEMS-BAY	METR-LA	TransGTR	3.52	7.34	9.69	4.35	8.96	12.65	5.21	10.10	15.95
		pFedGTP	3.15	5.78	8.42	3.93	7.23	11.23	5.06	9.41	15.30
		ASTCL	**2.88**	**5.75**	**7.62**	**3.51**	**7.22**	**9.60**	**4.62**	**9.34**	**13.16**
PEMS04	PEMS08	TransGTR	18.03	26.83	\	20.77	33.96	\	27.68	41.93	\
		pFedGTP	18.48	27.81	12.02	21.19	33.40	14.29	27.59	42.29	22.46
		ASTCL	**17.15**	**26.66**	**11.97**	**20.11**	**33.28**	**14.10**	**27.50**	**41.74**	**22.30**
PEMS08	PEMS04	TransGTR	21.69	34.51	14.40	25.74	40.38	17.50	**32.94**	50.49	22.82
		pFedGTP	22.30	35.03	\	26.25	40.76	\	34.57	51.97	\
		ASTCL	**21.67**	**34.25**	**13.77**	**25.30**	**39.56**	**17.12**	32.98	**50.24**	**22.44**

sors. In this few-shot setting, our two model variants continue to demonstrate superior performance compared to baselines trained on the entire traffic network. Significantly, there exists a mere 2% and 5% average performance decline for ASTCL_{50} and ASTCL_{20}, respectively. These results further demonstrate ASTCL's adaptability in few-shot traffic forecasting.

Cross-city Performance Analysis. We evaluate the transferring ability of ASTCL for cross-city traffic prediction in this section and report the transferring results in TABLE IV. The evaluation involves pre-training models on a source dataset and transferring them to a target dataset. We compared two state-of-art transferring baselines TransGTR [2] and pFedCTP [64]. TransGTR learns graph structures from data-rich source cities and transfers them to the target city using knowledge distillation and temporal decoupled regularization techniques. pFedCTP is designed for traffic transfer learning with personalized federated learning method. Following their original training strategy, we first train TransGTR and pFedCTP on the complete source dataset and fine-tune them on the target dataset using one day's data. In contrast, ASTCL is pre-trained using only one day of data from the source dataset and directly evaluated on the target dataset without any fine-tuning or additional operations. This key difference highlights ASTCL's efficiency, as it does not require extensive pre-training or fine-tuning. As shown in TABLE IV, ASTCL consistently outperforms the TransGTR and pFedCTP and achieves an average improvement up to 10%. The results demonstrate the transferability and robustness of ASTCL in predicting both traffic flow and traffic speed. A particularly valuable observation is that ASTCL can be trained on any traffic dataset and directly applied to similar forecasting tasks in other cities, eliminating the need for additional training processes and reducing computational costs.

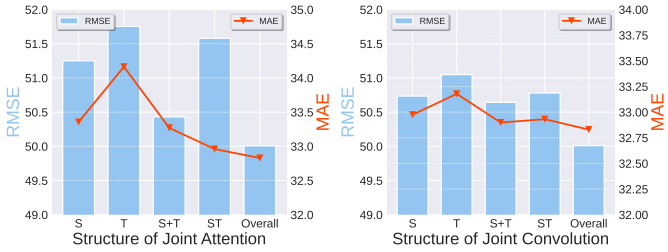
Cross-domain Performance Analysis. We aim to demonstrate ASTCL's adaptability to different prediction domains. Unlike the cross-city analysis, which examines the transferability of the model across different cities but within the same prediction variable, the cross-domain evaluation focuses on the model's capability to adapt to different types of prediction variables. For consistency, we use only the first two dimensions of traffic flow features across all four datasets in this section. We evaluate the performance when a model trained on one type of traffic prediction task (e.g., traffic speed prediction) is transferred to a different dataset for another task (e.g., traffic flow prediction). Due to the inherent heterogeneity between variables, directly transferring a pre-trained model to

TABLE V: Cross-domain Prediction Results

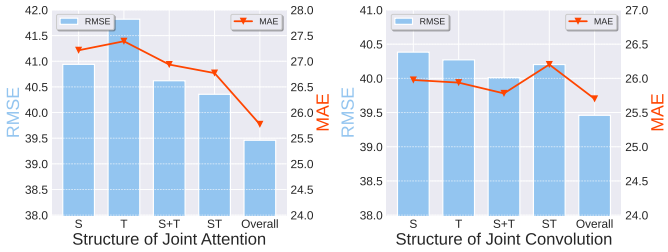
Source Data	Target Data	Methods	MAE	RMSE	MAPE
METR-LA	PEMS04	ASTCL	47.92	66.48	40.24
		pFedGTP	35.22	52.39	\
		FT-ASTCL	**33.05**	**51.69**	**21.17**
	PEMS08	ASTCL	41.34	57.51	32.03
		pFedGTP	28.00	41.45	23.28
		FT-ASTCL	**26.66**	**41.01**	**16.43**
PEMS-BAY	PEMS04	ASTCL	45.42	62.99	37.67
		pFedGTP	35.23	52.12	\
		FT-ASTCL	**32.87**	**51.92**	**21.24**
	PEMS08	ASTCL	38.31	53.17	30.93
		pFedGTP	27.68	41.52	23.58
		FT-ASTCL	**26.41**	**40.93**	**16.65**

TABLE VI: Ablation study on PEMS04 and PEMS08.

Models	PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o STA	35.54	53.57	23.50	32.88	47.45	24.11
w/o STC	33.94	51.53	22.15	28.60	42.40	21.77
w/o DSG	33.71	51.31	21.72	26.54	39.86	17.27
w/o IN	33.92	51.89	22.30	27.54	40.94	21.20
ASTCL	**32.83**	**50.01**	**21.36**	**25.77**	**39.46**	**16.86**



(a) Results on PEMS04



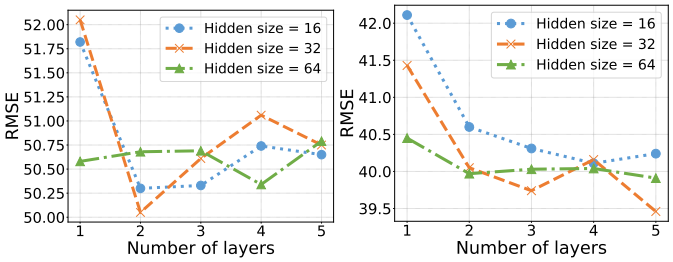
(b) Results on PEMS08

Fig. 4: Analysis of the joint attention and joint convolution on PEMS08 and PEMS04.

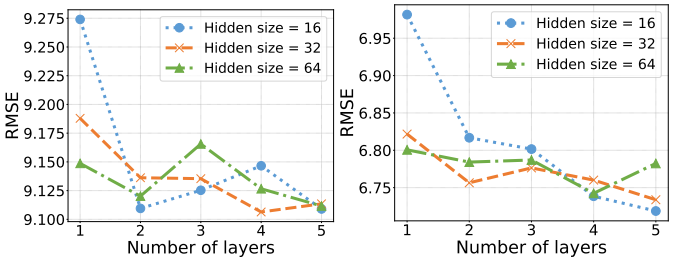
a new prediction task often leads to performance degradation. Here, we introduce the FT-ASTCL variant, which fine-tunes the pre-trained model on the target variable dataset for an additional 20 epochs. The experimental results compared with pFedCTP are shown in TABLE V. The results show that with simple fine-tuning, our ASTCL improves performance to levels comparable with models trained from scratch on the target dataset. ASTCL consistently outperforms pFedCTP in cross-domain scenarios, highlighting its superior adaptability and generalization capabilities. The results highlight ASTCL's ability to generalize and perform well across varied traffic forecasting tasks.

C. Model Analysis

In this section, we analyze the ASTCL from the following aspects: effects of individual components, parameter settings, and efficiency.



(b) RMSE on PEMS04 (b) RMSE on PEMS08



(c) RMSE on METR-LA (d) RMSE on PEMS-BAY

Fig. 5: Parameter analysis on four datasets.

1) Ablation Experiments (RQ3): We have designed two types of experiments to validate the effectiveness of individual components in ASTCL. First, we focus on analyzing the performance impact of individual design elements by creating the following model variants: (1) w/o STA: removes the joint spatio-temporal attention mechanism; (2) w/o STC: removes the joint spatio-temporal convolution; (3) w/o DSG: completely removes the modules related to the dynamic sequence graph; (4) w/o IN: removes the instance normalization. We conduct the ablation study on two datasets, PEMS04 and PEMS08, with a 60-minute forecasting horizon. The results are summarized in TABLE VI. From the results, it is evident that all variants perform worse than our complete ASTCL model, demonstrating the importance of each component. The absence of the joint spatio-temporal attention mechanism (w/o STA) leads to a significant increase in MAE, RMSE, and MAPE across both datasets. This highlights the critical role of attention in effectively capturing complex spatio-temporal dependencies. Additionally, the removal of the joint spatio-temporal convolution (w/o STC) also degrades performance notably, underscoring the importance of local spatio-temporal correlations. Similarly, the dynamic sequence graph and instance normalization also play crucial roles in accurate sequence modeling.

The second set of experiments is designed to analyze each operation in proposed joint encoders. We design more variants to investigate the influence of the structure of joint attention and joint convolution. As previously mentioned, our joint encoders utilize three types of operations to model the traffic network from different perspectives: spatial modeling (S), temporal modeling (T), and spatio-temporal modeling (ST). Therefore, we only keep parts of the set of operations and remove others to explore their individual roles as shown in Fig 4. The results indicate that the proposed joint attention

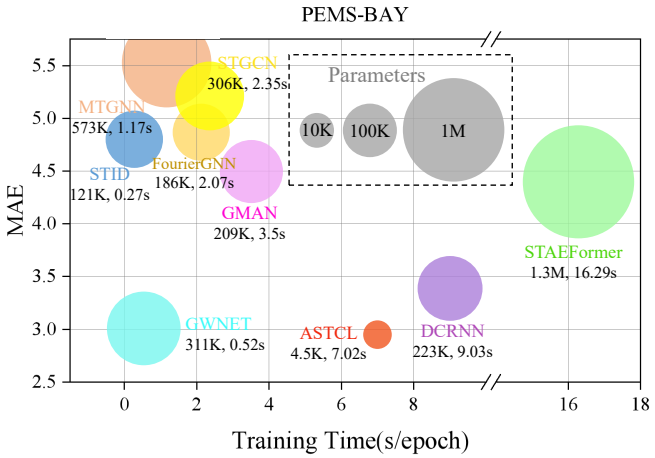


Fig. 6: Model efficiency comparison on PEMS-BAY.

and joint convolution modules achieve the best performance compared to other variants. The model can achieve better forecasting accuracy by combining the spatial and temporal modules than using either one alone. In summary, our proposed joint encoders demonstrate a significant advantage in capturing intricate spatio-temporal correlations.

2) Parameters Analysis: We applied a grid search on the number of layers and the hidden size in the model. The RMES values on four datasets are presented in Fig 5. In our experiments, we observed that 2-layer architecture is a balanced choice that maintains high performance while mitigating overfitting and reducing computational load on PEMS04 and METR-LA. For PEMS08 and PEMS-BAY datasets, a deeper model with 5 layers can reduce the RMSE. A larger hidden size generally enables the model to learn more complex representations but also increases the number of parameters, leading to higher computational costs and potential overfitting. Therefore, we chose 16 for METR-LA and 32 for the other three datasets based on the results.

3) Model Efficiency(RQ4): Fig 6 presents the MAE, parameter count, and the training time of different models on the PEMS-BAY dataset. Notably, our ASTCL model comprises only 4.5k parameters, which is less than one-tenth of the simplest baseline. This parameter efficiency is attributed to the innovative design where we feed only the adaptive context of the target node to the model rather than the entire traffic network. Such a design not only reduces the model's complexity but also enhances its practicality for real-world applications, significantly lowering deployment costs. The lower parameter count enables quick adaptation to new urban environments and different variables. In summary, ASTCL's node-based design enables accurate predictions for specific locations without requiring extensive network-wide data. The model's low parameter count and computational requirements make it suitable for deployment in most environments. By making reliable and efficient predictions, ASTCL shows great potential to contribute to smarter cities and more sustainable transportation networks.

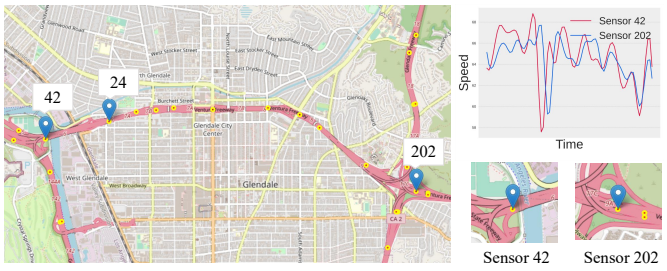


Fig. 7: Traffic network on METR-LA. The number identifies specific traffic nodes.

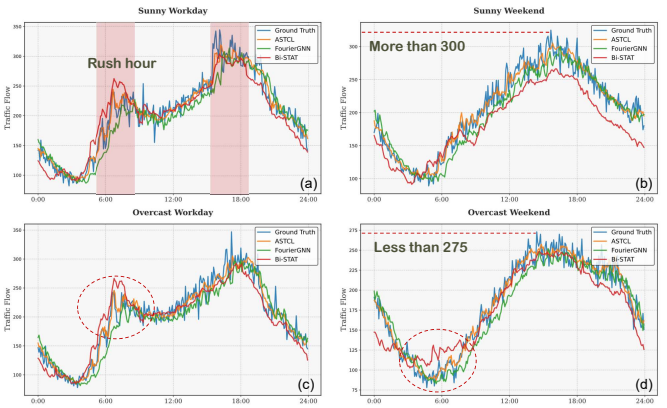


Fig. 8: The visualization results on PEMS08 across four traffic prediction scenarios: (a) Sunny Workday, (b) Sunny Weekend, (c) Overcast Workday, (d) Overcast Weekend.

D. Case Study

In this section, we present a comprehensive case study to intuitively demonstrate the advantages of ASTCL and validate its effectiveness across diverse scenarios.

First, we investigate the benefits of constructing the adaptive spatio-temporal context for learning valuable representations. We visualize a segment of a highway in METR-LA and mark the traffic sensors with yellow circles, as shown in Fig 7 (left). In this example, we set Node 42 as the target location and derive its adaptive context as described in Section III-B. Node 24 and 202 are two locations selected by the algorithm to form the adaptive context for Node 42. Intuitively, Node 24, being geographically adjacent to Node 42, naturally exerts a significant influence on it. However, Node 202 is located far from the target node, and such long-range dependencies are typically challenging to capture using standard GNN approaches. To illustrate the effectiveness of our method, we showcase the speed records of these two sensors within a specific time segment in the upper right corner of Fig 7. The visualization reveals a distinct correlation between the traffic speeds recorded at Node 42 and Node 202. We highlight the more fine-grained positions of Node 42 and 202. Both nodes are situated at the entrances of complex overpasses, which introduces significant semantic similarities despite their physical distance. This case study confirms that our ASTCL model effectively captures not only local spatio-temporal correlations but also global dependencies. By leveraging the adaptive

spatio-temporal context, ASTCL demonstrates its robustness in understanding complex traffic patterns, thereby enhancing its predictive performance.

Second, we evaluate ASTCL's performance on PEMS08 across a wide range of real-world scenarios, including weekdays versus weekends, rush hour versus off-peak hours, and sunny versus overcast days. As shown in Fig 8, ASTCL consistently outperforms the two best-performing baselines FourierGNN and Bi-STAT, across all conditions, highlighting its ability to handle diverse and complex traffic dynamics. In Fig 8(a) and (c), ASTCL accurately predicts the rapid changes in traffic flow during weekday rush hours. On weekends, ASTCL adapts to another traffic pattern, which exhibits a more continuous increase in traffic flow throughout the day. From dashed red circles in Fig 8(c) and (d), we can observe that our model achieves better predictions compared to other methods during periods of rapid traffic flow changes, both on weekends and weekdays. Additionally, ASTCL can extract the underlying influence of weather conditions. As illustrated in Fig 8(b) and (d), peak traffic flow under sunny conditions is notably higher than under overcast conditions. This variation poses a challenge for models such as Bi-STAT to distinguish underlying patterns in traffic flow. In conclusion, the visualization results demonstrate ASTCL's potential to effectively handle complex real-world traffic prediction tasks and adapt to diverse scenarios.

VI. Conclusion and Future Work

In this study, we address the challenge of accurate traffic forecasting on daily basis. We introduce the novel ASTCL model, designed to learn effective spatio-temporal representations with one day of data. Our approach involves constructing an adaptive spatio-temporal context for the target node and developing both a joint attention mechanism and joint convolution to capture intricate spatio-temporal correlations. Additionally, we introduce a dynamic sequence graph to model the fluctuating nature of traffic patterns. Comprehensive experiments on four real-world datasets demonstrate that ASTCL consistently achieves desirable performance. The adaptability of ASTCL allows it to be directly applicable to various traffic datasets and to predict different traffic attributes. Further efforts will be made towards extending the proposed framework to include domain adaptation techniques for diverse domains. Additionally, incorporating external factors such as weather conditions and calendar information to guide the forecasting in data scarcity scenarios may be an interesting research direction.

References

[1] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 804–17 815, 2020.

[2] Y. Jin, K. Chen, and Q. Yang, "Transferable graph structure learning for graph-based traffic forecasting across cities," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2023, pp. 1032–1043.

[3] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[4] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *International Joint Conference on Artificial Intelligence*, 2019, pp. 1907–1913.

[5] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," *In International Conference on Learning Representations*, 2019.

[6] C. Cai and Y. Wang, "A note on over-smoothing for graph neural networks," *arXiv preprint arXiv:2006.13318*, 2020.

[7] X. Liu, Y. Liang, C. Huang, H. Hu, Y. Cao, B. Hooi, and R. Zimmermann, "Do we really need graph neural networks for traffic forecasting?" *CoRR*, vol. abs/2301.12603, 2023.

[8] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 1234–1241.

[9] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 4365–4373.

[10] H. Liu, H. Kamarthi, L. Kong, Z. Zhao, C. Zhang, and B. A. Prakash, "Time-series forecasting for out-of-distribution generalization using invariant learning," *arXiv preprint arXiv:2406.09130*, 2024.

[11] C. Moorthy and B. Ratcliffe, "Short term traffic forecasting using time series methods," *Transportation Planning and Technology*, vol. 12, no. 1, pp. 45–56, 1988.

[12] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transportation Research Record*, vol. 2024, no. 1, pp. 116–125, 2007.

[13] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.

[14] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 1655–1661.

[15] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[16] X. Li, Y. Gong, W. Liu, Y. Yin, Y. Zheng, and L. Nie, "Dual-track spatio-temporal learning for urban flow prediction with adaptive normalization," *Artificial Intelligence*, p. 104065, 2024.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 753–763.

[18] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic graph convolution network for traffic forecasting based on latent network of laplace matrix estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1009–1018, 2020.

[19] 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 *International Conference on Machine Learning*. PMLR, 2022, pp. 11 906–11 917.

[20] Y. An, Z. Li, W. Liu, H. Sun, M. Chen, W. Lu, and Y. Gong, "Spatio-temporal graph normalizing flow for probabilistic traffic prediction," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 45–55.

[21] P. Yu, X. Zhang, Y. Gong, J. Zhang, H. Sun, J. Zhang, X. Zhang, and Y. Yin, "Enhancing origin–destination flow prediction via bi-directional spatio-temporal inference and interconnected feature evolution," *Expert Systems with Applications*, vol. 264, p. 125679, 2025.

[22] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2021.

[23] W. Duan, X. He, Z. Zhou, L. Thiele, and H. Rao, "Localised adaptive spatial-temporal graph neural network," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2023, pp. 448–458.

[24] 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 *International Conference on Information and Knowledge Management*, 2022, pp. 4454–4458.

[25] Z. Wang, Y. Nie, P. Sun, N. H. Nguyen, J. M. Mulvey, and H. V. Poor, "ST-MLP: A cascaded spatio-temporal linear framework with channel-independence strategy for traffic forecasting," *CoRR*, vol. abs/2308.07496, 2023.

[26] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 922–929.

[27] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting," *IEEE Transactions on Neural Networks and Learning System*, vol. 34, no. 10, pp. 6913–6925, 2023.

[28] R. Wang, Y. Liu, Y. Gong, W. Liu, M. Chen, Y. Yin, and Y. Zheng, "Fine-grained urban flow inference with unobservable data via space-time attraction learning," in *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2023, pp. 1367–1372.

[29] T. Li, J. Zhang, K. Bao, Y. Liang, Y. Li, and Y. Zheng, "Autost: Efficient neural architecture search for spatio-temporal prediction," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 794–802.

[30] Z. Pan, S. Ke, X. Yang, Y. Liang, Y. Yu, J. Zhang, and Y. Zheng, "Autostg: Neural architecture search for predictions of spatio-temporal graph," in *International Conference of World Wide Web*, 2021, pp. 1846–1855.

[31] S. Ke, Z. Pan, T. He, Y. Liang, J. Zhang, and Y. Zheng, "Autostg+: An automatic framework to discover the optimal network for spatio-temporal graph prediction," *Artificial Intelligence*, vol. 318, p. 103899, 2023.

[32] X. Wu, D. Zhang, C. Guo, C. He, B. Yang, and C. S. Jensen, "Autocts: Automated correlated time series forecasting," *Proceedings of the VLDB Endowment*, vol. 15, no. 4, pp. 971–983, 2021.

[33] X. Wu, D. Zhang, M. Zhang, C. Guo, B. Yang, and C. S. Jensen, "Autocts+: Joint neural architecture and hyperparameter search for correlated time series forecasting," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–26, 2023.

[34] X. Liu, Y. Liang, C. Huang, Y. Zheng, B. Hooi, and R. Zimmermann, "When do contrastive learning signals help spatio-temporal graph forecasting?" in *International Conference on Advances in Geographic Information Systems*, 2022, pp. 5:1–5:12.

[35] H. Qu, Y. Gong, M. Chen, J. Zhang, Y. Zheng, and Y. Yin, "Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8008–8023, 2023.

[36] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 4356–4364.

[37] X. Zhang, Y. Gong, X. Zhang, X. Wu, C. Zhang, and X. Dong, "Mask-and contrast-enhanced spatio-temporal learning for urban flow prediction," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3298–3307.

[38] Y. Gong, T. He, M. Chen, B. Wang, L. Nie, and Y. Yin, "Spatio-temporal enhanced contrastive and contextual learning for weather forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[39] B. Y. Lin, F. F. Xu, E. Q. Liao, and K. Q. Zhu, "Transfer learning for traffic speed prediction: A preliminary study," in *AAAI Conference on Artificial Intelligence Workshops*, vol. WS-18, 2018, pp. 174–177.

[40] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, 2019.

[41] Y. Ren, X. Chen, S. Wan, K. Xie, and K. Bian, "Passenger flow prediction in traffic system based on deep neural networks and transfer learning method," in *International Conference on Intelligent Transportation Engineering*, 2019, pp. 115–120.

[42] Z. Fang, D. Wu, L. Pan, L. Chen, and Y. Gao, "When transfer learning meets cross-city urban flow prediction: Spatio-temporal adaptation matters," in *International Joint Conference on Artificial Intelligence*, L. D. Raedt, Ed., 2022, pp. 2030–2036.

[43] B. Lu, X. Gan, W. Zhang, H. Yao, L. Fu, and X. Wang, "Spatio-temporal graph few-shot learning with cross-city knowledge transfer," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2022, pp. 1162–1172.

[44] Y. Tang, A. Qu, A. H. F. Chow, W. H. K. Lam, S. C. Wong, and W. Ma, "Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities," in *InternationalConference on Information and Knowledge Managemen*, 2022, pp. 1905–1915.

[45] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2016, pp. 1905–1914.

[46] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *International Joint Conference on Artificial Intelligence*, 2019, pp. 1893–1899.

[47] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *International Conference of World Wide Web*, 2019, pp. 2181–2191.

[48] S. Yang, J. Liu, and K. Zhao, "Space meets time: Local spacetime neural network for traffic flow forecasting," in *International Conference on Data Mining*. IEEE, pp. 817–826.

[49] M. Li, Y. Tang, and W. Ma, "Few-shot traffic prediction with graph networks using locale as relational inductive biases," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1894–1908, 2022.

[50] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *AAAI Conference on Artificial Intelligence*, 2021, pp. 4189–4196.

[51] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *AAAI Conference on Artificial Intelligence Workshop*, 1994, pp. 359–370.

[52] W. Yu, P. Zhou, S. Yan, and X. Wang, "Inceptionnext: When inception meets convnext," *CoRR*, vol. abs/2303.16900, 2023.

[53] T. Kim, J. Kim, Y. Tae, C. Park, J. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*. OpenReview.net, 2022.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[55] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, "Online spatio-temporal crowd flow distribution prediction for complex metro system," *IEEE Transactions on knowledge and data engineering*, vol. 34, no. 2, pp. 865–880, 2020.

[56] Y. Zhang, F. Regol, A. Valkanas, and M. Coates, "Contrastive learning for time series on dynamic graphs," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 742–746.

[57] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[58] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94.

[59] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *International Conference on Information and Knowledge Management*, 2023, pp. 4125–4129.

[60] D. Kingma, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations*, 2014.

[61] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling financial time series with S-PLUS®*, pp. 385–429, 2006.

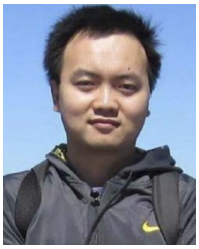
[62] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 914–921.

[63] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, "Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[64] Y. Zhang, H. Lu, N. Liu, Y. Xu, Q. Li, and L. Cui, "Personalized federated learning for cross-city traffic prediction," in *33rd International Joint Conference on Artificial Intelligence, IJCAI*, 2024, pp. 5526–5534.

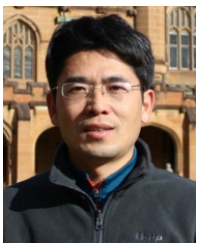
Xiaoyu Li is a graduate student in the School of Software, Shandong University, China. He received his B.S. degree in software engineering in 2022 from Shandong University, China. His research interests include spatio-temporal data mining and machine learning. He has published one journal paper in Artificial Intelligence.

Yitian Zhang is currently a Ph.D. candidate at McGill University, focusing on developing sequential models for time-series analysis. She received her BEng degree in Communication Engineering from Jilin University, and earned her Master's degree in Electrical and Computer Engineering at University of Toronto, specializing in machine learning and computer vision.



Guodong long received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia, in 2014. He is currently an Associate Professor and Core Member with the Center for Artificial Intelligence(CAI), Faculty of Engineering and Information Technology, University of Technology Sydney. His research focuses on machine learning, data mining, and cloud computing.

Yupeng Hu (Member, IEEE) obtained his Ph.D. degree in Software Engineering from Shandong University in 2018. He is currently an Associate Professor with the School of Software, Shandong University. His research interests include information retrieval, data mining, and explainable AI. Various parts of his work have been published in famous journals and forums, such as IEEE TIP, ACM TOIS, ACM MM, AAAI etc. He has served as a PC member for ACM MM, AAAI and a reviewer for IEEE TKDE and TMM.



Wenpeng Lu (Member, IEEE) received the B.S. and M.S. degrees in educational technology from Shandong Normal University in 2002 and 2005, respectively, and the Ph.D. degree in computer application technology from the Beijing Institute of Technology in 2014. He is currently a Professor with the Qilu University of Technology (Shandong Academy of Sciences). His research interests include artificial intelligence, natural language processing, and machine learning.

Meng Chen is currently an associate professor in the School of Software, Shandong University, China. He received his Ph.D. degree in computer science and technology in 2016 from Shandong University, China. He worked as a Postdoctoral fellow from 2016 to 2018 in the School of Information Technology, York University, Canada. His research interest is in the area of data mining. He has published over 40 papers in prestigious journals and conferences in data mining field such as TKDE, TOIS, TITS and CIKM.

Chengqi Zhang (Senior Member, IEEE) received the Ph.D. degree from The University of Queensland, Brisbane, QLD, Australia, in 1991, and the D.Sc. (Higher Doctorate) degree from Deakin University, Geelong, VIC, Australia, in 2002. Since 2001, he has been a Professor of information technology with the University of Technology Sydney (UTS), Sydney, NSW, Australia, where he has been the Executive Director of UTS DataScience since 2016. He has published more than 200 research papers, including several in first-class international journals, such as Artificial Intelligence and the IEEE and ACM TRANSACTIONS. He has published six monographs, edited 16 books, and attracted 11 Australian Research Council grants. His research interests mainly focus on data mining and its applications. Dr. Zhang has been the Chairman of the Australian Computer Society, National Committee for Artificial Intelligence, since 2005.

Yongshun Gong (S'19-M'21) is a Professor at the School of Software, Shandong University, China. He received his Ph.D. degree from University of Technology Sydney. His principal research interest covers the data science and machine learning, in particular, the following areas: spatio-temporal data mining and traffic prediction. He has published above 70 papers in top journals and refereed conference proceedings, including the IEEE TPAMI, Artificial Intelligence, IEEE T-KDE, IEEE T-NNLS, IEEE T-CYB, NeurIPS, KDD, CVPR, CIKM, AAAI and IJCAI.