A Lightweight and Accurate Spatial-Temporal Transformer for Traffic Forecasting

Guanyao Li, Shuhan Zhong, Xingdong Deng, Letian Xiang, S.-H. Gary Chan, Ruiyuan Li, Yang Liu, Ming Zhang, Chih-Chieh Hung, and Wen-Chih Peng, *Member, IEEE*

Abstract—We study the forecasting problem for traffic with dynamic, possibly periodical, and joint spatial-temporal dependency between regions. Given the aggregated inflow and outflow traffic of regions in a city from time slots 0 to $t-1$, we predict the traffic at time t for any region. Prior arts in the area often considered the spatial and temporal dependencies in a decoupled manner, or were rather computationally intensive in training with a large number of hyper-parameters which needed tuning. We propose ST-TIS, a novel, lightweight and accurate **S**patial-**T**emporal **T**ransformer with **i**nformation fusion and region **s**ampling for traffic forecasting. ST-TIS extends the canonical Transformer with information fusion and region sampling. The information fusion module captures the complex spatial-temporal dependency between regions. The region sampling module is to improve the efficiency and prediction accuracy, cutting the computation complexity for dependency learning from $O(n^2)$ to $O(n\sqrt{n})$, where n is the number of regions. With far fewer parameters than state-of-the-art deep learning models, ST-TIS's offline training is significantly faster in terms of tuning and computation (with a reduction of up to 90% on training time and network parameters). Notwithstanding such training efficiency, extensive experiments show that ST-TIS is substantially more accurate in online prediction than state-of-the-art approaches (with an average improvement of 9.5% on RMSE, and 12.4% on MAPE compared to STDN and DSAN).

Index Terms—Efficient Transformer, joint spatial-temporal dependency, region sampling, spatial-temporal forecasting, spatial-temporal data mining

1 Introduction

Traffic forecasting is to predict the inflow (i.e., the number of arriving objects per unit time) and outflow (i.e., the number of departing objects per unit time) of any region in a city at the next time slot. The objects can be people, vehicles, goods/items, etc. Traffic forecasting has important applications in transportation, retail, public safety, city planning, etc [1]. For example, with traffic forecasting, a taxi company may dispatch taxis in a timely manner to meet the supply and demand in different regions of a city. Yet another example is bike sharing, where the company may want to balance bike supply and demand at dock stations (regions) based on such forecasting.

Although there has been much effort focused on deep learning to improve the prediction accuracy of the state-of-the-art forecasting models, progressive improvements on benchmarks have been correlated with an increase in the number of parameters and the amount of training resources required to train the model, making it costly to train and deploy large deep learning models [2]. Therefore, a lightweight and training-efficient model is essential for fast delivery and deployment.

In this work, we study the following spatial-temporal traffic forecasting problem: Given the historical (aggregated) inflow and outflow data of different regions from time slots 0 to $t-1$ (with slot size of, say, 30 minutes), what is the design of a training-efficient model to accurately predict the inflow and outflow of any region at time t? (Note that even though we consider predicting for the next time slot, our work can be straightforwardly extended to any future time slot by successive application of the algorithm.) We seek a "small" training model with substantially fewer parameters, which naturally leads to efficiency in tuning, memory, and computation time. Despite its training efficiency, it should also achieve higher accuracy than the state-of-the-art approaches in its online predictions.

Intuitively, region traffic is spatially and temporally correlated. As an example, the traffic of a region could be

- *Guanyao Li was with the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. He is now with the Guangzhou Urban Planning and Design Survey Research Institute, Guangdong Enterprise Key Laboratory for Urban Sensing, Monitoring and Early Warning, Guangzhou, Guangdong 510060, China. E-mail: gyli@gzpi.com.cn.*
- *Shuhan Zhong, Letian Xiang, and S.-H. Gary Chan are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: {szhongaj, lxiangab, gchan}@cse.ust.hk.*
- *Xingdong Deng, Yang Liu, and Ming Zhang are with the Guangzhou Urban Planning and Design Survey Research Institute, Guangdong Enterprise Key Laboratory for Urban Sensing, Monitoring and Early Warning, Guangzhou, Guangdong 510060, China. E-mail: dengxd_updsrig@163.com, {liuyang, ming.zhang}@gzpi.com.cn.*
- *Ruiyuan Li is with the College of Computer Science, Chongqing University, Chongqing 400044, China. E-mail: liruiyuan@cqu.edu.cn.*
- *Chih-Chieh Hung is with the Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan. E-mail: smalloshin@email.nchu.edu.tw.*
- *Wen-Chih Peng is with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan. E-mail: wcpeng@g2.nctu.edu.tw.*

Manuscript received 10 January 2022; revised 17 November 2022; accepted 17 December 2022. Date of publication 30 December 2022; date of current version 6 October 2023.
This work was supported in part by Hong Kong General Research Fund under Grant 16200120, in part by the Guangdong Enterprise Key Laboratory for Urban Sensing, Monitoring and Early Warning under Grant 2020B121202019, in part by the National Natural Science Foundation of China under Grant 62202070, and in part by China Postdoctoral Science Foundation under Grant 2022M720567.
(Corresponding authors: S.-H Gary Chan and Xingdong Deng.)
Recommended for acceptance by V. Moreira.
Digital Object Identifier no. 10.1109/TKDE.2022.3233086

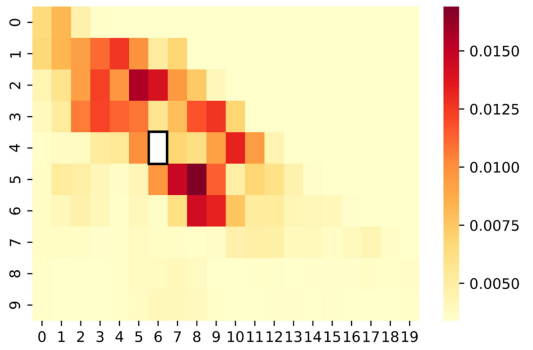


Fig. 1. Visualization of attention scores between a target region (6,4) and other regions. The color of a cell (x_i, y_i) indicates the dependency of (6,4) on (x_i, y_i), where a darker color indicates stronger dependency.

correlated with that of another with some temporal lag due to the travel time between them. Moreover, such dependency may be dynamic over different time slots, and it may have temporal periodic patterns. This is the case for the traffic of office regions, which exhibit high correlation with the residence regions on workday mornings but much less than at night or on weekend. To accurately predict the region traffic, it is hence significantly crucial to account for the dynamic, possibly periodical, and joint spatial-temporal (ST) dependency between regions, no matter how far apart the regions are.

Much effort has been devoted to capturing the dependency between regions for traffic forecasting. While commendable, most works consider spatial and temporal dependency separately with independent processing modules [3], [4], [5], [6], [7], [8], which can hardly capture their joint nature in our current setting. Some recent works applied canonical Transformer [9] to capture region dependency [10], [11], [12], [13]. While impressive, canonical Transformer limits the training efficiency because it learns a region's embedding as the weighted aggregation of all the other regions based on their computed attention scores. This results in $O(n^2)$ computation complexity per layer, where n is the number of regions. Moreover, it has been observed that the attention scores from the canonical Transformer have a long tail distribution [14]. We illustrate this in Fig. 1 using a taxi dataset collected in New York City. We split New York City into 10×20 regions and visualize the attention scores (after a SoftMax operation) between a target region (i.e., the region (6,4)) and other regions. Clearly, most regions have very small attention scores (i.e., the long-tail phenomenon), with the attention scores of more than 60% of regions being less than 0.004. Such a long-tail effect may introduce noise for the region embedding learning, and degrade the prediction performance.

We propose ST-TIS, a novel, small, efficient and accurate **S**patial-**T**emporal **T**ransformer with **i**nformation fusion and region **s**ampling for traffic forecasting. Given the historical (aggregated) inflow and outflow of regions from 0 to $t-1$, ST-TIS predicts the inflow and outflow of any region at t, without relying on the transition data between regions. With a small set of parameters, ST-TIS is efficient to train (offline phase). It extends the canonical Transformer with novel information fusion and region sampling strategies to learn the dynamic and possibly periodical spatial-temporal dependency between regions in a joint manner, hence achieving high prediction accuracy (online phase).

ST-TIS makes the following contributions:

- *A data-driven Transformer scheme for dynamic, possibly periodical, and joint spatial-temporal dependency learning.* ST-TIS jointly considers the spatial-temporal dependencies between regions, rather than considering the two dependencies sequentially in a decoupled manner. In particular, ST-TIS considers the dynamic spatial-temporal dependency for any individual time slot with an information fusion module (IFM), and also the possibly periodical characteristic of spatial-temporal dependency from multiple time slots using an attention mechanism. With IFM, spatial-temporal dependency is jointly considered for traffic forecasting. Moreover, the dependency learning is data-driven, without any assumption of spatial locality. Due to its design, ST-TIS is small (in parameter footprint), fast (in training time), and accurate (in prediction).
- *A novel region sampling strategy for computationally efficient dependency learning.*ST-TIS leverages the Transformer framework [9] to learn region dependency. To address the quadratic computation issue and mitigate the long-tail effect of the canonical Transformer, it employs a novel region sampling strategy to generate a connected region graph, and learns the dependency based on the graph. The dependencies between any pair of regions (both close and distant dependencies) are guaranteed to be considered in ST-TIS via information propagation, and the computational complexity is reduced from $O(n^2)$ to $O(n\sqrt{n})$, where n is the number of regions.
- *Extensive experimental validation:* We evaluate ST-TIS on three large-scale real datasets. Our results show that ST-TIS is substantially more accurate than the state-of-the-art approaches, with a significant improvement in RMSE and MAPE (an average improvement of 9.5% on RMSE, and 12.4% on MAPE compared to STDN and DSAN). Furthermore, it is much more lightweight than most state-of-the-art models, and is ultra fast for training (with a reduction of $46\% \sim 95\%$ on training time and $23\% \sim 98\%$ on network parameters).

The remainder of this paper is organized as follows. We first discuss related works in Section 2. After preliminaries in Section 3, we detail ST-TIS in Section 4. We present the experimental settings and results in Section 5, and conclude in Section 6.

2 Related Works

Traffic forecasting has raised much attention in both academia and industry due to its social and commercial values. Some early traffic forecasting works proposed using regression models, such as auto-regressive integrated moving average (ARIMA) models [15], [16] and non-parametric region models [17], [18]. All of these works consider temporal dependency, but they have not considered the spatial dependency between regions. Some other works extract features from heterogeneous data sources (e.g., POI, weather, etc.), and use machine learning models such as Support Vector Machine [19], Gradient Boosting Regression Tree [20], and linear regression model [21]. Despite the encouraging results,

they rely on manually defined features and have not considered the joint spatial-temporal dependency.

In recent years, deep learning techniques have been employed to study spatial and temporal correlations for traffic forecasting. Most existing works consider the spatial and temporal dependency in a decoupled manner [3], [4], [5], [6], [7], [8], which can hardly capture their joint effect. For spatial dependency, Convolution Neural Network (CNN), Graph Neural Network (GNN), and Transformer have been widely applied. Regarding temporal dependency, Recurrent Neural Network (RNN) and its variants such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been extensively studied. Compared with these works, ST-TIS considers the spatial and temporal dependency in a joint manner.

CNN has been applied in many works to capture dependencies between close regions [3], [4], [5], [6], [22]. In these works, a city is divided into some connected but non-overlapping grids, and the traffic in each grid is then predicted. However, these works cannot be used for fine-grained flow forecasting at an individual location [23], such as predicting flow for a docked bike-sharing station or a subway station. Moreover, CNN can hardly capture distant traffic dependency due to its relatively small receptive field [24], [25].

Some other works used GCN to capture spatial dependency [7], [8], [26], [27], [28], [29], [30], [31], [32], [33]. In these works, a city is represented as a graph structure, and convolution operations are applied to aggregate spatially distributed information in the graph. In each aggregation layer, a region would aggregate the embedding of its neighbouring regions in the graph. However, these works highly rely on the graph structure for dependency learning. Prior works usually construct graphs based on the distance between regions or road networks, based on the locality assumption (i.e., close regions have higher dependency). They have to stack more layers to learn dependency if the distance between two regions in the graph is long, and it ends up with an inefficient and over-smoothing model [28]. In recent years, Transformer [9] has been applied in traffic forecasting [10], [11], [12], [13]. The canonical Transformer can be seen as a special graph neural network with a complete graph, in which any pair of regions are connected. Consequently, the dependencies between both close and distant regions could be considered. Moreover, the self-attention mechanism and the Transformer network structure have been demonstrated to be powerful in many prior works. However, the computational complexity of each aggregation round is $O(n^2)$ for Transformer where n is the number of regions, while that for GNN is $O(E)$ where E is the number of edges in the graph ($E \leq n^2$). Furthermore, as a region may only have a strong dependency on a small portion of regions, aggregating the embedding of all regions would introduce noise and degrade its performance. In ST-TIS, we propose a region connected graph (i.e., there exists a path between any pair of regions in the graph), in which the degree of any node (i.e., region) is $O(\sqrt{n})$ and the distance between any two regions in the graph is no more than 2. We extend the canonical Transformer with the proposed region connected graph, so that it can inherit the advantage of efficiency and effectiveness from both GNN and Transformer.

RNN and its variants such as LSTM and GRU have been used to capture temporal dependency [5], [31], [32], [34]. However, the performance of RNN-based models deteriorates rapidly as the length of the input sequence increases [35]. Some works incorporate the attention mechanism to improve their capability of modeling long-term temporal correlations [6], [8], [26], [36]. Nevertheless, RNN-based networks are widely known to be difficult to train and are computationally intensive [7]. As recurrent networks generate the current hidden states as a function of the previous hidden state and the input for the position, they are in general more difficult to be trained in parallel. To address this issue, the self-attention mechanism is proposed as the replacement of RNN to model sequential data [9]. It has enjoyed success in capturing temporal correlations for traffic forecasting [10], [11], [12], [37]. Compared with RNN-based models, self-attention models can directly model long-term temporal interactions, but the computational complexity of using self-attention for temporal dependency learning in existing works is $O(q^2)$, where q is the number of historical time slots. Compared with them, ST-TIS is conditional on the spatial-temporal dependency at any individual slot to generate weights for different time slots, so its computational complexity is $O(q)$ in our work. In addition, the temporal dependency is jointly considered with the spatial dependency in ST-TIS, instead of in a decoupled manner.

Some variants of Transformer have been proposed to address the efficiency issues of the canonical Transformer, such as LogSparse [38], Reformer [39], Informer [14], etc. While impressive, these approaches cannot be used in the scenario of capturing spatial-temporal dependency between regions for traffic forecasting we are considering in this work.

3 Preliminaries

3.1 Problem Formulation

Definition 1. *(Region) The area (e.g., a city or subway route) is partitioned into n non-overlapping regions. We use $R = \{r_1, r_2, \ldots, r_n\}$ to denote the partitioned regions, in which r_i denotes the i-th region.*

The way to partition an area is *flexible* for ST-TIS, e.g., grid map, road network, clustering, or train/bus/bike stations, etc.

Definition 2. *(Traffic data) We use $\mathcal{I}$ and $\mathcal{O}$ to denote the inflow and outflow data of all regions over time, respectively. Specifically, $\mathcal{I}^t \in \mathbb{R}^{1\times n}$ and $\mathcal{O}^t \in \mathbb{R}^{1\times n}$ is the inflow and outflow of the n regions at time t. Moreover, $\mathcal{I}_i^t$ and $\mathcal{O}_i^t$ is the inflow and outflow of region r_i at time t (i.e., the number of objects arriving at/departing from the region r_i at time slot t). Furthermore, we use $\mathcal{I}_i^{t':t} = [\mathcal{I}_i^{t'}, \mathcal{I}_i^{t'+1}, \ldots, \mathcal{I}_i^t]$ to denote the inflow of r_i from t' to t. Similarly, $\mathcal{O}_i^{t':t} = [\mathcal{O}_i^{t'}, \mathcal{O}_i^{t'+1}, \ldots, \mathcal{O}_i^t]$ indicates the outflow of r_i from t' to t.*

The formal formulation of the traffic forecasting problem is as follows:

Definition 3. *(Traffic Forecasting) Given the traffic data of regions from 0 to $t-1$, namely $\mathcal{I}^{0:t-1}$ and $\mathcal{O}^{0:t-1}$, the traffic forecasting problem is to predict the inflow and outflow of any region at t, namely $\mathcal{I}^t$ and $\mathcal{O}^t$.*

3.2 ST-TIS Overview

We overview the proposed ST-TIS in Fig. 2. Given the traffic data of all regions from time slots 0 to $t-1$, ST-TIS is an end-to-end model to capture the spatial-temporal dependency

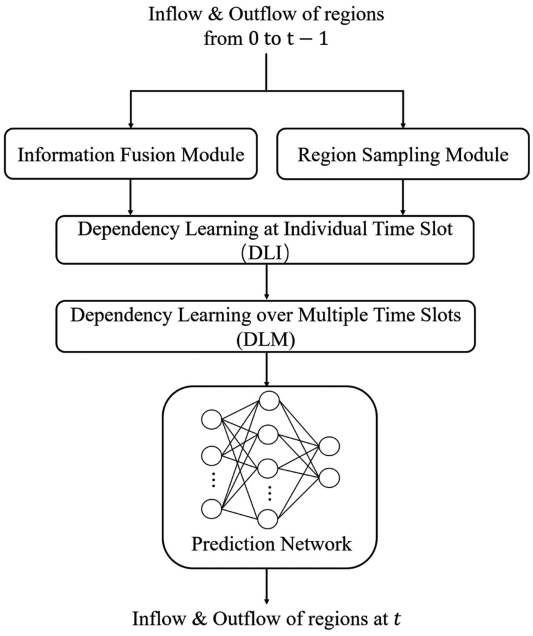


Fig. 2. ST-TIS overview.

and predict the inflow and outflow for any region at t. There are five modules in ST-TIS. We explain the design goals and the relationship among modules as follows:

- *Information Fusion Module:* A key to accurately predicting the traffic for regions is capturing the dynamic spatial-temporal (ST) dependency between regions in a joint manner. To this end, ST-TIS employs an information fusion module to learn the spatial-temporal-flow (STF) embedding by encoding its spatial, temporal, and flow information for any individual region at a time slot.
- *Region Sampling Module:* To address the issues of quadratic computational complexity and long-tail distribution of attention scores for the canonical Transformer, ST-TIS uses a novel region sampling strategy to generate a region connected graph. The dependency learning would be based on the generated graph.
- *Dependency Learning for Individual Time Slot (DLI):* Given the STF embedding at a time slot and the region connected graph, ST-TIS extends the canonical Transformer to jointly capture the dynamic spatial-temporal dependency between regions at the time slot. As both the spatial and temporal information has been encoded in the STF embedding, the joint effect of spatial-temporal dependencies between regions could be captured. Moreover, with the region connected graph, only the attention scores between neighbouring nodes (i.e., regions) are computed, and only the embedding of one's neighbouring regions is aggregated. Consequently, it cuts the computational complexity of a layer from $O(n^2)$ to $O(n \times \sqrt{n})$ where n is the number of regions, and addresses the issue of the long-tail effect for dependency learning.
- *Dependency Learning over Multiple Time Slots (DLM):* Given the region embedding from DLI at multiple time slots, DLM captures the periodic patterns of spatial-temporal dependency using the attention mechanism. The influence of spatial-temporal dependency at historical time slots on the predicted time slot is hence considered in ST-TIS. The learning process is data-driven, without any prior assumption of traffic periods.
- *Prediction Network (PN)*: Given the results from DLM, ST-TIS uses a fully connected neural network to predict the inflow and outflow of regions ($\mathcal{I}^t$ and $\mathcal{O}^t$) simultaneously.

4 ST-TIS Details

We present the details of ST-TIS in this section. We first elaborate the information fusion module in Section 4.1, followed by the region sampling module in Section 4.2. After that, we introduce the dependency learning for individual time slot (DLI) in Section 4.3, and the dependency learning over multiple time slots (DLM) in Section 4.4. Finally, we present the prediction network (PN) in Section 4.5.

4.1 Information Fusion Module

ST-TIS employs the information fusion module to learn one's spatial-temporal-flow (STF) embedding by fusing its position, time slot, and flow information at any individual time slot. Given n regions, we first use a one-hot vector $\mathcal{S}_i \in \mathbb{R}^{1\times n}$ to represent a region r_i, in which only the i-th element in $\mathcal{S}_i$ is 1 and otherwise 0. After that, we encode the position information for a region r_i as follows:

$$\hat{\mathcal{S}}_i = \mathcal{S}_i \cdot W^S + b^S, \tag{1}$$

where $\hat{\mathcal{S}}_i \in \mathbb{R}^{1\times d}$ is the spatial embedding of r_i, $W^S \in \mathbb{R}^{n\times d}$ and $b^S \in \mathbb{R}^{1\times d}$ are learnable parameters, and d is a hyperparameter.

In terms of temporal information, we first split a day into o time slots and represent the i-th time slot using a one-hot vector $\mathcal{T}_i \in \mathbb{R}^{1\times o}$. After that, we learn the temporal embedding by

$$\hat{\mathcal{T}}_i = \mathcal{T}_i \cdot W^T + b^T, \tag{2}$$

where $\hat{\mathcal{T}}_i \in \mathbb{R}^{1\times d}$ is the temporal embedding of the i-th time slot in a day, and $W_T \in \mathbb{R}^{o\times d}$ and $b^T \in \mathbb{R}^{1\times d}$ are learnable parameters.

Recall that the traffic of one region at t_i may depend on that of another region at previous time slots due to the travel time between two regions. Thus, we use one's surrounding observations instead of solely using a snapshot to learn the dependency [38]. We define the surrounding observations of a region at a time slot t_j as follows:

Definition 4. *(Surrounding observations) For a region r_i at a time slot t_j, its surrounding observations are defined as the flow in the w previous time slots:* $\{\mathcal{I}_i^{t_j-w}, \ldots, \mathcal{I}_i^{t_j-2}, \mathcal{I}_i^{t_j-1}, \mathcal{O}_i^{t_j-w}, \ldots, \mathcal{O}_i^{t_j-2}, \mathcal{O}_i^{t_j-1}\}$.

Note that by employing the surrounding observations, the temporal lag of the dependency between regions could be considered. Given the surrounding observations of r_i at t_j, we first apply the 1-D convolution of kernel size $1 \times p$ ($p < w$) with stride 1 and f output channels on its inflow and outflow surrounding observations to extract different patterns. The flow embedding $\mathcal{F}_i^{t_j} \in \mathbb{R}^{1\times d}$ of r_i at t_j is computed as

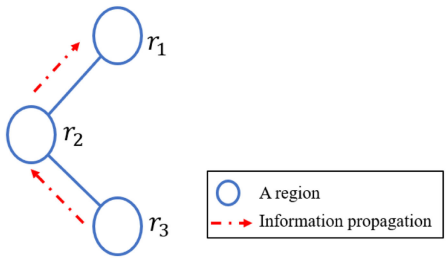


Fig. 3. Information propagation in a graph.

$$\mathcal{F}_i^{t_j} = (Conv^I(\mathcal{I}_i^{t_j-w:t_j-1}) || Conv^O(\mathcal{O}_i^{t_j-w:t_j-1})) \cdot W^F + b^F, \quad (3)$$

where $||$ is the concatenation operation, $Conv^I(\cdot)$ and $Conv^O(\cdot)$ are the convolutional operation for inflow and outflow data respectively, $W^F \in \mathbb{R}^{l\times d}$, $b^F \in \mathbb{R}^{1\times d}$ are learnable parameters, and $l = 2 \times f \times (w - k + 1)$.

Finally, we fuse the position, time slot and flow information to learn the STF embedding of a region r_i at time t_j. We define that t_j is the $M(t_j)$-th time slot in a day, where $M(\cdot)$ is a matching function. The fusion process is defined as

$$\mathcal{L}_i^{t_j} = (\hat{\mathcal{S}}_i + \hat{\mathcal{T}}_{M(t_j)} + \mathcal{F}_i^{t_j}) \cdot W^L + b_i^L, \quad (4)$$

where $\mathcal{L}_i^j \in \mathbb{R}^{1\times d}$ is the STF embedding, and $W_L \in \mathbb{R}^{d\times d}$ and $b_i^L \in \mathbb{R}^{1\times d}$ are learnable parameters.

4.2 Region Sampling

To capture a region's dependency on others (both nearby and distant), a canonical Transformer computes the attention scores between the target region and all other regions, and aggregates the embedding of all other regions based on the computed attention scores. However, this results in the issues of quadratic computation and the long-tail effect for dependency learning.

Fortunately, prior works have shown that information could be propogated between nodes in a graph via a multi-layer network structure [40]. Hence, with a proper graph structure and network structure, a region can aggregate the embedding of another even without directly evaluating their attention score. We present a toy example in Fig. 3, in which regions are represented as nodes in the graph and are connected with egdes. In the first aggregation layer, r_1 would capture the information from r_2, while r_2 would capture the information from r_3. Since the information of r_3 has been aggregated in r_2, r_1 could also capture it in the second aggregation layer without computing the attention score between r_1 and r_3. From this example, we conclude that a node's information can reach another with a β-layer aggregation operation if their distance is not more than β in the graph. However, a prior work has shown that the network achieves the best performance when $\beta = 2$ but degrades dramatically from 3 layers [41]. Therefore, we set β as 2 in our work.

To address the limitations of the canonical Transformer, we propose to generate a connected graph (i.e., there is at least one path between any two nodes in the graph), in which the distance between any pair of nodes is not more than 2 and the degree of any node (i.e., region) is not more than $c \times \sqrt{n}$. The reason is that, given n nodes, when $\beta = 2$, we should have $k + k^2 > n - 1$, and the minimum degree k of the constructed graph is hence $O(\sqrt{n})$. In this way, for a target region, we only have to aggregate the embedding of $O(\sqrt{n})$ regions in an aggregation layer, and the influence from other regions can be captured in a two-layer aggregation process by information propagation. With such design, we do not have to compute the attention scores between any pairs of regions and aggregate the embedding of all n regions for a target region, so that the computation complexity is reduced to $O(n\sqrt{n})$ for each layer, and the long-tail issue is also addressed.

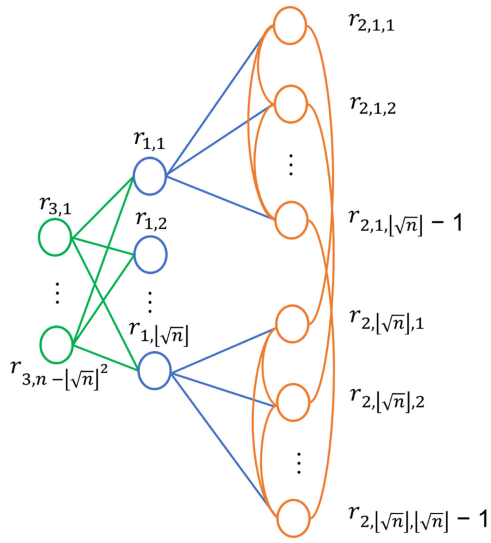


Fig. 4. The process of connected graph generation.

In this work, we present a heuristic approach for region connected graph generation. We first calculate the traffic similarity between any pair of regions. The calculation of the traffic similarity is flexible and it could be any similarity metric. Following a prior work [42], we use DTW [43] as an example in this work to measure the similarity in terms of the average traffic over time slots of a day. We use $\mathcal{M} \in \mathbb{R}^{n\times n}$ to represent the similarity matrix, in which $\mathcal{M}_{i,j}$ is the similarity between r_i and r_j. Based on $\mathcal{M}$, the process of the connected graph generation is illustrated in Fig. 4.

We first select top $\lfloor\sqrt{n}\rfloor$ regions without replacement, which have the largest sum of similarity with other regions, represented as $\{r_{1,1}, r_{1,2}, \ldots, r_{1,\lfloor\sqrt{n}\rfloor}\}$. For any region $r_{1,i}$, we then select $\lfloor\sqrt{n}\rfloor - 1$ regions without replacement, which have the largest similarity between them and $r_{1,i}$, represented as $\{r_{2,i,1}, r_{2,i,2}, \ldots, r_{2,i,\lfloor\sqrt{n}\rfloor-1}\}$, and we connect $r_{1,i}$ to $r_{2,i,j}$ $(j = 1, 2, \ldots, \lfloor\sqrt{n}\rfloor - 1)$. After that, we connect a region $r_{2,i,j}$ to regions $r_{2,i,k}$ and $r_{2,u,j}$, where $k \in \{\{1, 2, \ldots, \lfloor\sqrt{n}\rfloor - 1\}\backslash\{j\}\}$ and $u \in \{\{1, 2, \ldots, \lfloor\sqrt{n}\rfloor\}\backslash\{i\}\}$. Finally, if $\sqrt{n} \notin \mathbb{Z}$, the remaining regions would be connected to $r_{1,i}$ where $i \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor\}$, represented as $\{r_{3,1}, r_{3,2}, \ldots, r_{3,n-\lfloor\sqrt{n}\rfloor^2}\}$. If $\sqrt{n} \in \mathbb{Z}$, we randomly select a region from $r_{1,i}$, and connect it to $r_{1,j}$, where $j \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor\}\backslash\{i\}$, represented as r^*.

Theorem 1. *The degree of any node in the region connected graph is $O(\sqrt{n})$, and the distance between any two nodes in the graph is less than 2.*

Proof. The generation of the region connected graph ensures that a node is connected to at most $\max(2 \times \lfloor\sqrt{n}\rfloor - 2, n - \lfloor\sqrt{n}\rfloor^2 + \lfloor\sqrt{n}\rfloor - 1)$ other nodes, and hence the degree is $O(\sqrt{n})$.

The distance of $(r_{3,i}, r_{1,j})$ (or $(r^*, r_{1,j})$) and $(r_{1,j}, r_{2,j,k})$ is both 1, where $i \in \{1, 2, \ldots, n - \lfloor\sqrt{n}\rfloor^2\}$, $j \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor\}$, and $k \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor - 1\}$. Thus, the distance between $r_{3,i}$ (or r^*) and any other region is no more than 2 in the graph.

For region $r_{1,j}$, since the distance of $(r_{1,j}, r_{2,j,k})$ and $(r_{2,j,k}, r_{2,m,k})$ is both 1 where $k \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor - 1\}$ and $m \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor\}\backslash\{j\}$, the distance of $(r_{1,j}, r_{2,m,k})$ is hence 2. Thus, the distance between $r_{1,j}$ and any other region is also no more than 2.

In terms of $r_{2,j,k}$, as the distance of $(r_{2,j,k}, r_{2,m,k})$ and $(r_{2,j,k}, r_{2,j,v})$ is 1, where $m \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor\}\backslash\{j\}$ and $v \in \{1, 2, \ldots, \lfloor\sqrt{n}\rfloor - 1\}\backslash\{k\}$, the distance between $r_{2,j,k}$ and any other regions is hence no more than 2. Therefore, the distance between any two regions in the graph is less than 2. □

Because the distance between any two regions in the proposed graph is less than 2, the dependencies between any two regions could be considered if the layer number of the aggregation network is larger than 2.

4.3 Dependency Learning for Individual Time Slot (DLI)

Following the canonical Transformer, ST-TIS employs a multi-head attention mechanism, so that it could account for different dependencies between regions. For the m-th head, the attention score between r_i and r_v at t_j is defined as

$$\mathcal{A}_m(r_i, r_v, t_j) = \frac{(\mathcal{L}_i^{t_j} \cdot W_{Q_m}) \cdot (\mathcal{L}_v^{t_j} \cdot W_{K_m})^T}{\sqrt{d}}, \tag{5}$$

where $\mathcal{L}_i^{t_j} \in \mathbb{R}^{1\times d}$ and $\mathcal{L}_v^{t_j} \in \mathbb{R}^{1\times d}$ are the STF embedding of regions r_i and r_v at t_j (Equation 4), $W_{Q_m} \in \mathbb{R}^{d\times d}$ and $W_{K_m} \in \mathbb{R}^{d\times d}$ are learnable parameters, and d is a hyperparameter for the embedding size.

Unlike the canonical Transformer, we do not evaluate the attention scores between one region and all other regions. Instead, we only compute one's attention scores with its neighbouring regions in the region connected graph, and aggregate their embedding in terms of their attention score to update the region's embedding. For the m-th head, the embedding of a region r_i at time t_j is then updated as

$$\begin{aligned}\hat{\mathcal{L}}_{i,m}^{t_j} &= \sum_{r_v \in Neigh(r_i)} softmax(\mathcal{A}_m(r_i, r_v, t_j)) \cdot \mathcal{L}_v^{t_j} \\ &= \sum_{r_v \in Neigh(r_i)} \frac{\exp(\mathcal{A}_m(r_i, r_v, t_j))}{\sum_{r_u \in Neigh(r_i)} \exp(\mathcal{A}_m(r_i, r_u, t_j))} \cdot \mathcal{L}_v^{t_j},\end{aligned} \tag{6}$$

where $\hat{\mathcal{L}}_{i,m}^{t_j} \in \mathbb{R}^{1\times d}$ is the output of the m-th head, $Neigh(r_i)$ is the neighbouring regions of r_i in the graph, and $\mathcal{A}_m(r_i, r_v, t_j)$ is the attention score defined in Equation 5. As the degree of any node is $O(\sqrt{n})$, the computation complexity of attention score evaluation and embedding aggregation is hence $O(n\sqrt{n})$ for all regions in a layer.

Finally, we concatenate the results of multi-heads and the embedding of r_i is computed as

$$\hat{\mathcal{L}}_i^{t_j} = Concat(\hat{\mathcal{L}}_{i,1}^{t_j}, \ldots, \hat{\mathcal{L}}_{i,M}^{t_j}) \cdot W^O, \tag{7}$$

where $Concat(\cdot)$ is the concatenation operation, $\hat{\mathcal{L}}_i^t \in \mathbb{R}^{1\times d}$ is the embedding of r_i, $W_O \in \mathbb{R}^{(d\times M)\times 1}$ are learnable parameters, and M is the number of heads.

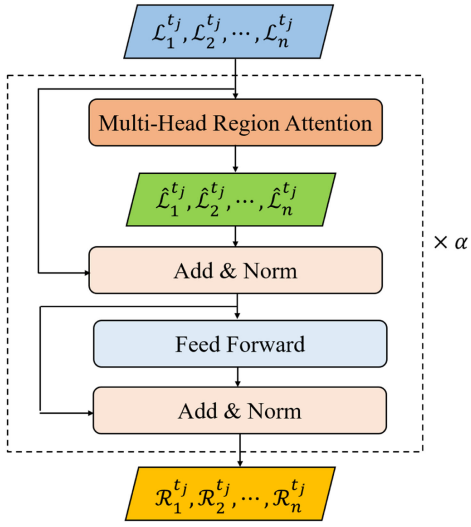


Fig. 5. Processing of dependency learning for individual time slot.

Following the structure of Transformer[9], the output of the multi-head region attention layers $\hat{\mathcal{L}}_i^{t_j}$ is then passed to a fully connected neural network (Fig. 5). We also employ a residual connection between each of the two layers. As we discussed in Section 4.2, the information propagation is achieved with a multi-layer network structure. Thus, we stack the layers α times (the effect of α will be discussed in Section 5.7). We denote the final output of the DLI as $\mathcal{R}^{t_j} = \{\mathcal{R}_1^{t_j}, \mathcal{R}_2^{t_j}, \ldots, \mathcal{R}_n^{t_j}\}$, where $\mathcal{R}_i^{t_j}$ is the embedding of region r_i at t_j.

Compared with the canonical Transformer, we only need to compute the attention scores and aggregate the embedding between adjacent nodes in the region connected graph. The computation complexity is hence reduced from $O(n^2)$ to $O(n\sqrt{n})$ for each layer.

4.4 Dependency Learning Over Multiple Time Slots (DLM)

Considering that the spatial-temporal dependency may have a periodical characteristic, DLM learns the periodic dependency by evaluating the correlation between $\mathcal{R}_n^t$ and the dependency at other historical time slots $\mathcal{R}_i^{\hat{t}}$ (where $\hat{t} < t$). After that, it generates a new embedding for r_i by aggregating the embedding at different time slots according to their correlations.

Specifically, we consider the short-term and long-term period for traffic data in this work. The spatial-temporal dependency at the following historical time slots is used as the model input to predict the inflow and outflow of regions at t: spatial-temporal dependency in the recent h time slots (i.e., short-term period); and the same time interval in the recent l days (i.e., long-term period).

We employ a point-wise aggregation with self-attention to evaluate their correlations and aggregate the embedding accordingly. To capture the multiple periodic dependency, we use a multi-head attention network in DLM.

Given a set of historical time slot Q, the z-th dependency on a historical time slot $\hat{t} \in Q$ is calculated as follows:

$$e_z(t, \hat{t}) = \frac{(\mathcal{R}_i^t \cdot W_{Q_z}^{\mathcal{T}}) \cdot (\mathcal{R}_i^{\hat{t}} \cdot W_{K_z}^{\mathcal{T}})^T}{\sqrt{d}}, \tag{8}$$

where $W^{\mathcal{T}}_{Q_z} \in \mathbb{R}^{d\times d}$ and $W^{\mathcal{T}}_{K_z} \in \mathbb{R}^{d\times d}$ are learnable parameters.

We use a softmax function to normalize the dependency and aggregate the context of each time slot by weight

$$\beta_z(t,\hat{t}) = softmax(e_z(t,\hat{t})) = \frac{\exp(e_z(t,\hat{t}))}{\sum_{t_p\in Q}\exp(e_z(t,t_p))}. \tag{9}$$

The aggregation of the z-th head is hence

$$\hat{\mathcal{R}}^t_{i,z} = \left(\sum_{\hat{t}\in Q}(\beta_z(t,\hat{t})\cdot\mathcal{R}^{\hat{t}}_i\right)\cdot W^{\mathcal{T}}_V, \tag{10}$$

where Q is the set of historical time slots, $\hat{\mathcal{R}}^t_{i,z}$ is the aggregation result of the z-th head, and $W^{\mathcal{T}}_V \in \mathbb{R}^{d\times d}$ are learnable parameters. Finally, we concatenate the results of different heads with

$$\hat{\mathcal{R}}^t_i = Concat(\hat{\mathcal{R}}^t_{i,1},\hat{\mathcal{R}}^t_{i,2},\ldots,\hat{\mathcal{R}}^t_{i,Z})\cdot W_{\mathcal{T}}, \tag{11}$$

where $Concat(\cdot)$ is the concatenation operation, and $W_{\mathcal{T}} \in \mathbb{R}^{(Z\times d)\times d}$ are learnable parameters. We then pass $\hat{\mathcal{R}}^t_i$ to a fully connected neural network to obtain the spatial-temporal embedding of r_i at t. Note that we also employ a residual connection between each of the two layers to avoid gradient exploding or vanishing. The output of the DLM is denoted as Ω^t_i for region r_i at t.

Different from prior works, the periodic dependency learning is conditional on the spatial-temporal dependency at each individual time slot. Consequently, the spatial and temporal dependencies are jointly considered during the periodic dependency learning. The computation complexity is $O(|Q|)$, where $|Q|$ is the number of historical time slots used for learning.

4.5 Prediction Network (PN)

Given the spatial-temporal embedding $\mathcal{T}^t_{r_i}$ of a region, the prediction network predicts the inflow and outflow using the fully connected network. The forecasting function is defined as

$$[\hat{\mathcal{I}}^t_i,\hat{\mathcal{O}}^t_i] = \sigma(\Omega^t_i\cdot W^P + b^P), \tag{12}$$

where $\hat{\mathcal{I}}^t_i$ and $\hat{\mathcal{O}}^t_i$ is the forecasting inflow and outflow respectively, $\sigma(\cdot)$ is the ReLU activation function, and $W^P \in \mathbb{R}^{d\times 2}$ and $b^P \in \mathbb{R}^{1\times 2}$ are learnable parameters.

We simultaneously forecast the inflow and outflow in our work, and define the loss function as follows:

$$LOSS = \sqrt{\frac{\sum_{i=1}^n(\mathcal{I}^t_i-\hat{\mathcal{I}}^t_i)^2+\sum_{i=1}^n(\mathcal{O}^t_i-\hat{\mathcal{O}}^t_i)^2}{2n}}, \tag{13}$$

where n is the number of regions.

5 Illustrative Experimental Results

In this section, we first introduce the datasets and the data processing approaches in Section 5.1, and the evaluation metrics and baseline approaches in Section 5.2. Then, we compare the accuracy of ST-TIS with the state-of-the-art methods and some variations in Sections 5.3 and 5.4, respectively. After that, we evaluate the training efficiency and the effect of surrounding observations in Sections 5.5 and 5.6 respectively, followed by a discussion of the hyperparameters of layer number in Section 5.7 and head number in Section 5.8.

5.1 Datasets

We conducted an extensive traffic study and model evaluations based on three real-world traffic flow datasets, the NYC-Taxi dataset, the NYC-Bike dataset, and the PeMS dataset (PeMSD4). The NYC-Taxi dataset and the NYC-Bike dataset contain trip records, each of which consists of origin, destination, departure time, and arrival time. The NYC-Taxi dataset contains 22,349,490 taxi trip records from NYC in 2015, from 01/01/2015 to 03/01/2015. The NYC-Bike dataset was collected from the NYC Citi Bike system from 07/01/2016 to 08/29/2016, and contains 2,605,648 trip records. The PeMSD4 refers to the total flow, average speed, and average occupancy data collected in the San Francisco Bay Area from January to February in 2018, containing 3,848 detectors on 29 roads. We consider the flow data in our experiments.

For the NYC-Taxi dataset and the NYC-Bike dataset, we split New York city into 10×20 regions with a size of 1km $\times 1$ km. The time interval was set as 30 minutes for both datasets. For the PeMSD4, following prior works [30], [44], we removed some redundant detectors to ensure the distance between any adjacent detectors was longer than 3.5 miles, resulting in 307 detectors in the PeMSD4. The time interval was set as 5 minutes.

In all datasets, we used data from the previous 40 days as the training data, and the remaining days as the testing data. In the training data, we selected 80% of the training data to train our model and the remaining 20% for validation. We used the Min-Max normalization to rescale the range of volume value in $[0,1]$, and recovered the result for evaluation after forecasting. In our experiments, we excluded the results of those regions for inflow or outflow which was less than 10 when evaluating the model. It is a common practice used in industry and many prior works [5], [6], [37].

5.2 Performance Metrics and Baseline Methods

We use Root Mean Squared Errors (RMSE), Mean Average Percentage Error (MAPE) and R^2 as the evaluation metrics, which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N(y_i-\hat{y}_i)^2}{N}}, \tag{14}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^N\frac{|y_i-\hat{y}_i|}{y_i}, \tag{15}$$

$$R^2 = 1-\frac{\sum_{i=1}^N(\hat{y}_i-y_i)}{\sum_{i=1}^N(\bar{y}_i-y_i)}, \tag{16}$$

where y_i and $\hat{y}_i$ are the ground-truth and forecasting result of the i-th sample, $\bar{y}$ is the mean of all y_i, and N is the total number of samples.

We compare our model with the following state-of-the-art approaches:

- *Historical average (HA):* It uses the average of traffic at the same time slots in historical data for prediction.
- *ARIMA:* It is a conventional approach for time series data forecasting.
- *Ridge Regression:* A regression approach for time series data forecasting.

TABLE 1
Comparison With the State-of-the-Art Methods on the Taxi and Bike Dataset

Dataset	Method	Inflow		Outflow	
		RMSE	MAPE	RMSE	MAPE
NYC-Taxi	HA	33.83	21.14%	43.82	23.18%
	ARIMA	27.25	20.91%	36.53	22.21%
	Ridge	24.38	20.07%	28.51	19.94%
	XGBoost	21.72	18.70%	26.07	19.35%
	MLP	22.08±0.50	18.31±0.83%	26.67±0.56	18.43±0.62%
	ConvLSTM	23.67±0.20	20.70±0.20%	28.13±0.25	20.50±0.10%
	ST-ResNet	21.63±0.25	21.09±0.51%	26.23±0.33	21.13±0.63%
	STDN	19.05±0.31	16.25±0.26%	24.10±0.25	16.30±0.23%
	ASTGCN	22.05±0.37	20.25±0.26%	26.10±0.25	20.30±0.31%
	STGODE	21.46±0.42	19.22±0.36%	27.24±0.46	19.30±0.34%
	STSAN	23.07±0.64	22.24±1.91%	27.83±0.30	25.90±1.67%
	DSAN	18.32±0.39	16.07±0.31%	24.27±0.30	17.70±0.35%
	ST-TIS	**17.73±0.23**	**14.65±0.32%**	**21.96±0.13**	**14.83±0.76%**
NYC-Bike	HA	11.93	27.06%	12.49	27.82%
	ARIMA	11.25	25.79%	11.53	26.35%
	Ridge	10.33	24.58%	10.92	25.29%
	XGBoost	8.94	22.54%	9.57	23.52%
	MLP	9.12±0.24	22.40±0.40%	9.83±0.19	23.12±0.24%
	ConvLSTM	9.22±0.19	23.20±0.47%	10.40±0.17	25.10±0.45%
	ST-ResNet	8.85±0.13	22.98±0.53%	9.80±0.12	25.06±0.36%
	STDN	8.15±0.15	20.87±0.39%	8.85±0.11	21.84±0.36%
	ASTGCN	9.05±0.31	22.25±0.36%	9.34±0.24	23.13±0.30%
	STGODE	8.58±0.38	23.33±0.26%	9.23±0.31	23.99±0.23%
	STSAN	8.20±0.45	20.42±1.33%	9.87±0.23	23.87±0.71%
	DSAN	7.97±0.25	20.23±0.18%	10.07±0.58	23.92±0.39%
	ST-TIS	**7.57±0.04**	**18.64±0.23%**	**7.73±0.10**	**18.58±0.19%**

- *XGBoost [45]:* A powerful approach for building supervised regression models.
- *Multi-Layer Perceptron (MLP):* A three-layer fully-connected neural network.
- *Convolutional LSTM (ConvLSTM) [46]:* It is a special recurrent neural network with a convolution structure for spatial-temporal prediction.
- *ST-ResNet [4]:* It uses multiple convolutional networks with residual structures to capture spatial correlations from different temporal periods for traffic forecasting. It also considers external data such as weather, holiday events, and metadata.
- *STDN [6]:* It considers the dynamic spatial correlation and temporal shifted problem using the combination of CNN and LSTM. External data such as weather and event are considered in the work.
- *ASTGCN [44]:* It is an attention-based spatial-temporal graph convolutional network (ASTGCN) model to solve the traffic flow forecasting problem.
- *STGODE [28]:* It uses a spatial-temporal graph ordinary differential equation network to predict traffic flow based on two predefined graphs, namely a spatial graph in terms of distance, and a semantic graph in terms of flow similarity.
- *STSAN [11]:* It uses CNN to capture spatial information and the canonical Transformer to consider the temporal dependencies over time. In particular, transition data between regions are used to indicate the correlation between regions.
- *DSAN [12]:* It uses the canonical Transformer to capture the spatial-temporal correlations for spatial-temporal prediction, in which transition data between regions are used for correlation modeling.

We used the identical datasets and data process approach as the work STDN [6], and used the results of the work [6] as the benchmark for discussion. The experiment results of (1) $\sim$ (8) in Table 1 are reported in the work [6]. The evaluation of ASTGCN, STGODE, STSAN, and DSAN is based on the code from their authors' GitHubs.

We implemented our model using PyTorch. Data and code can be found in https://github.com/GuanyaoLI/ST-TIS. We used the following data as the model input since they achieved the best performance on the validation datasets: data in the recent past 3 hours (i.e., $h = 6$ as the slot duration is 30 minutes); the same time slot in the recent past 10 days (i.e., $l = 10$). The other default hyperparameter settings are as follows. The default length w of the surrounding observations is 6 (i.e. 3 hours), the number of convolution kernels F is 4, and the dimension d is set as 8. The number of k for DLI in Fig. 5 is set as 3, and the number of heads is set as 6 for the two modules. Furthermore, the dropout rate is set as 0.1, the learning rate is set as 0.001, and the batch size is set to be 32. Adam optimizer was used for model training. We trained our model on a machine with a NVIDIA RTX2080 Ti GPU.

5.3 Prediction Accuracy

We first compared the accuracy of ST-TIS with the state-of-the-art methods using the metrics RMSE and MAPE. Then, we used R^2 to further evaluate our proposed model and the state-

TABLE 2
Comparison With the State-of-the-Art Methods on the PeMSD-4

Method	RMSE	MAPE
HA	57.83	26.44%
ARIMA	48.56	22.06%
Ridge	41.41	19.28%
XGBoost	33.26	15.31%
MLP	39.41 $\pm$ 0.84	18.01 $\pm$ 0.79%
ASTGCN	38.92 $\pm$ 0.52	17.71 $\pm$ 0.47%
STGODE	27.77 $\pm$ 0.76	12.38 $\pm$ 0.57%
ST-TIS	**26.88$\pm$ 0.45**	**11.74$\pm$ 0.21%**

TABLE 3
Comparison of R^2

Dataset	Method	Inflow	Outflow
NYC-Taxi	HA	0.6637	0.4122
	ARIMA	0.7818	0.5915
	Ridge	0.8253	0.7512
	XGBoost	0.8614	0.7919
	MLP	0.8567	0.7822
	ConvLSTM	0.8353	0.7577
	ST-ResNet	0.8625	0.7894
	STDN	0.8933	0.8222
	ASTGCN	0.8571	0.7914
	STGODE	0.8647	0.7728
	STSAN	0.8436	0.7629
	DSAN	0.9013	0.8197
	ST-TIS	**0.9076**	**0.85239**
NYC-Bike	HA	0.3157	0.2606
	ARIMA	0.3915	0.3699
	Ridge	0.4869	0.4348
	XGBoost	0.6157	0.5659
	MLP	0.6001	0.5420
	ConvLSTM	0.5913	0.4873
	ST-ResNet	0.6234	0.5448
	STDN	0.6806	0.6288
	ASTGCN	0.6062	0.5865
	STGODE	0.6460	0.5962
	STSAN	0.6767	0.5383
	DSAN	0.6946	0.5194
	ST-TIS	**0.7244**	**0.7168**

of-the-art methods. In our experiments, each approach was run 10 times, and the mean and standard deviation are reported.

The results of RMSE and MAPE on the taxi dataset and the bike dataset are presented in Table 1. ST-TIS significantly outperforms all other approaches on all metrics and datasets. Specifically, the performance of the conventional time series forecasting approaches (HA and ARIMA) is poor for both datasets because these approaches do not consider the spatial dependency. Conventional machine learning approaches (Ridge, XGBoost, and MLP), which consider spatial dependency as features, have better performance than HA and ARIMA. However, they fail to consider the joint spatial-temporal dependencies between regions. Most deep learning-based models have further improvements than conventional works, illustrating the ability of deep neural networks to capture the complicated spatial and temporal dependency. ST-TIS is substantially more accurate than state-of-the-art approaches (i.e., ConvLSTM, STResNet, STDN, ASTGCN and STGODE). For example, it has an average improvement of 9.5% on RMSE and 12.4% on MAPE compared to STDN and DSAN. The reasons for the improvements are that it can capture the correlations between both nearby and distant regions, and it considers the spatial and temporal dependency in a joint manner. We find that the improvement is more significant on the NYC-Taxi dataset than on the NYC-Bike dataset. The reason could be that people prefer using taxis instead of bikes for long-distance travel, and so the correlations with distant regions are more important for the prediction task on the taxi dataset. The significant improvement demonstrates that ST-TIS has a better ability to capture the correlations for distant regions than the other approaches which have the spatial locality assumption. ST-TIS also outperforms other Transformer-based approaches (such as STSAN and DSAN). The reason is that with the information fusion and region sampling strategies in ST-TIS, the long-tail issue of the canonical Transformer is addressed and the joint spatial-temporal correlations are considered. The significant improvements demonstrate the effectiveness of our proposed model.

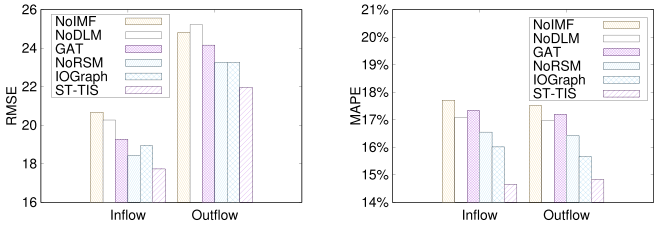
The results of RMSE and MAPE on the PeMSD4 dataset are shown in Table 2. Note that the inflow of a region is equal to its outflow in the PeMSD4. We do not include CNN-based methods (ConvLSTM, ST-ResNet and STDN) in the comparison because they are only appliable for grid-based prediction and cannot be applied for the scenario of PeMSD4 (graph-based prediction). We also exclude STSAN and DSAN because they rely on transition data between regions which are not available in the PeMSD4. As shown in Table 2, ST-TIS substantially outperfoms other baseline methods, which is consistent with the results of the other two datasets. The experimental results on the three datasets demonstrate the excellent performance for both grid-based and graph-based prediction.

To further evaluate our proposed model, we compare it with the state-of-the-art methods on the taxi and bike dataset using R^2. The results are presented in Table 3. ST-TIS achieves the highest R^2 value compared with any other state-of-the-art methods on both datasets, illustrating its outstanding prediction performance.

5.4 Design Variations of ST-TIS

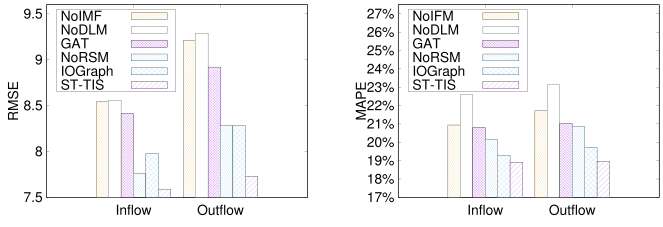
We compare ST-TIS with its variants to evaluate the effectiveness of the proposed modules. RMSE and MAPE are used as evaluation metrics. The following variants are discussed:

- *NoIFM*: We remove the information fusion module from ST-TIS. Only the surrounding observation of a time slot is used as the input of the model instead of the fusion result.
- *NoRSM*: We remove the region sampling module from ST-TIS. The canonical Transformer is used to capture the dependencies between regions without region sampling.
- *NoDLM*: We remove the module of dependency learning over multiple time slots, and only use $\mathcal{R}_i^t$ as the input of the prediction network for prediction.
- *GAT*: We use graph attention [47] to replace the Transformer in ST-TIS.
- *IOGraph*: We use two distinct directed graphs for outflow and inflow prediction. In particular, two DLI



(a) RMSE. (b) MAPE.

Fig. 6. Performance of variants on the NYC-Taxi dataset.



(a) RMSE. (b) MAPE.

Fig. 7. Performance of variants on the NYC-Bike dataset.

modules and two DLM modules are used to capture the correlation in the two graphs. Region embedding for inflow and outflow is concatenated as the input for the prediction network.

The RMSE and MAPE on the NYC-Taxi dataset are presented in Figs. 6a and 6b, respectively. After taking the information fusion module away, the performance degrades significantly. The reason is that the information fusion module plays a fundamental role in jointly considering the spatial-temporal dependency. Without such a module, our approach would degenerate to consider the spatial and temporal dependency in a decoupled manner. The experimental results demonstrate the importance of considering spatial-temporal dependency jointly and the effectiveness of the proposed information fusion module. Moreover, without the region sampling module, our approach still achieves good prediction performance because the canonical Transformer is good at capturing dependencies between regions. The performance is further improved with the region sampling since it could address the long-tail issue of the canonical Transformer for embedding aggregation. Furthermore, the RMSE and MAPE increase when the periodical characteristic of spatial-temporal dependency is not considered (i.e., NoDLM), which indicates the necessity of considering the period of spatial-temporal dependency and demonstrates the effectiveness and rationality of our model design. RMSE and MAPE increase when we replace Transformer with graph attention networks (GAT), illustrating the excellent performance of Transformer in capturing region correlation. The performance declines when we use two distinct directed graphs for outflow and inflow prediction (i.e., IOGraph). The reason could be that inflow and outflow are not independent in flow forecasting, and using two distinct graphs can hardly consider the joint effect of inflow and outflow. The comparison results illustrate the effectiveness of our proposed approach for capturing the joint effect of inflow and outflow. Similar and consistent findings can be observed on the NYC-Bike dataset in Figs. 7a and 7b.

5.5 Training Efficiency

We compare the training efficiency of ST-TIS with XGBoost, IOGraph and several state-of-the-art deep learning based approaches (i.e., ST-ResNet, STDN, ASTGCN, STGODE, STSAN, and DSAN) in terms of training time and number of learnable parameters. Other traditional baseline methods are not selected for comparison because their prediction accuracy is far inferior to that of XGBoost.

The results of the average training time per epoch and the total training time are presented in Table 4.

XGBoost achieves the least training time among all comparison approaches due to its lightweight design. However, it could hardly extend to consider the spatial-temporal correlation between regions, which limits its prediction accuracy. ST-ResNet achieves less training time than other deep learning models because it solely employs simple CNN and does not rely on RNN for temporal dependency learning. The average time per epoch of ST-TIS is close to ST-ResNet. In addition, ST-TIS is trained significantly faster than STDN, STSAN, and DSAN (with a reduction of $46\% \sim 95\%$). STDN uses LSTM to capture temporal correlation, which is in general more difficult to be trained in parallel. STSAN and DSAN also employ Transformer with self-attention for spatial and temporal correlation learning, but our proposed approach is significantly more efficient than them, illustrating the efficiency of the proposed region graph for model training. ST-TIS is also trained faster than IOGraph because IOGraph relies on two distinct graphs for inflow and outflow, leading to two distinct DLI modules and two DLM modules for training.

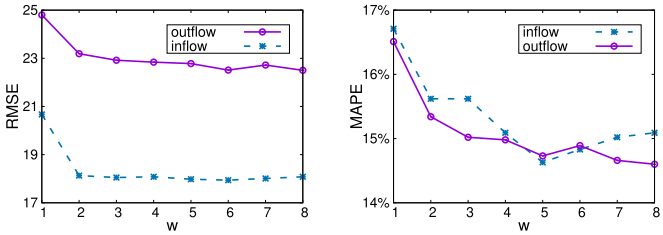
Furthermore, we also compare the number of learnable parameters of each model in Table 5. More parameters may lead to difficulties in model training, and would require more memory and training resources. Compared with other state-of-the-art approaches, ST-TIS is much more lightweight with fewer parameters for training (with a reduction of $35\% \sim 98\%$). The comparison results in Tables 1, 4 and 5

TABLE 4
Comparison of Training Time

Dataset	Method	Average time per epoch (s)	Total time (s)
NYC-Taxi	XGboost	–	313.67
	ST-ResNet	7.31	3077.51
	STDN	445.47	34746.66
	ASTGCN	25.31	6272.88
	STGODE	18.53	3423.48
	STSAN	426.75	33769.32
	DSAN	386.17	29390.75
	IOGraph	26.90	2636.21
	ST-TIS	10.21	1231.50
NYC-Bike	XGBoost	–	266.02
	ST-ResNet	7.25	2921.75
	STDN	480.21	23066.43
	ASTGCN	25.68	5084.64
	STGODE	18.76	3752.89
	STSAN	426.28	31216.28
	DSAN	434.03	26476.20
	IOGraph	27.03	3324.69
	ST-TIS	10.37	1556.80

TABLE 5
Comparison of the Number of Parameters

Method	Number of parameters
ST-ResNet	4,917,041
STDN	9,446,274
ASTGCN	450,031
STGODE	433,073
DSAN	1,621,634
STSAN	34,327,298
IOGraph	216,210
ST-TIS	**139,506**



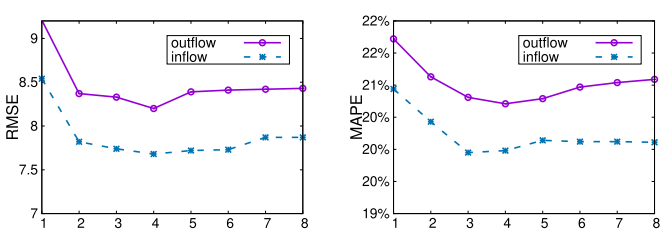
(a) RMSE. (b) MAPE.

Fig. 8. Impact of surrounding observations on the NYC-Taxi dataset.

demonstrate that our proposed ST-TIS is faster and more lightweight than other deep learning based baseline approaches, while achieving even better prediction accuracy.

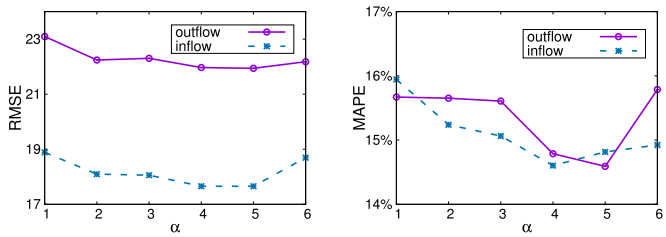
5.6 Surrounding Observations

We evaluated the impact of w on the performance of our model. Figs. 8a and 8b show the RMSE and MAPE versus different lengths on the NYC-Taxi dataset. A larger length w indicates that more information is encoded from the surrounding observations. When $w = 1$, only the observation at a time slot is used for dependency learning, and the model fails to capture the lagging characteristic of dependency. As the length increases, the RMSE and MAPE of both inflow and outflow forecasting decrease ($w \leq 5$). The performance improvement demonstrates the importance of using the surrounding observations to learn the dependencies between regions. RMSE and MAPE increase slightly but remain stable when the length is large ($w \geq 5$). The potential reason is that, when w is larger than the travel time between regions, increasing w would not introduce more information for dependency learning. On the other hand, a larger w may introduce some noise and more parameters for the model, leading to difficulties in model training [12]. The RMSE and MAPE versus different lengths of surrounding observations on the NYC-Bike dataset are shown in Figs. 9a and 9b, which are consistent with the results for the NYC-Taxi dataset. When the length is small ($w \leq 4$), RMSE and MAPE decrease as the length becomes larger, but they slightly increase when the length is large ($w \geq 4$).



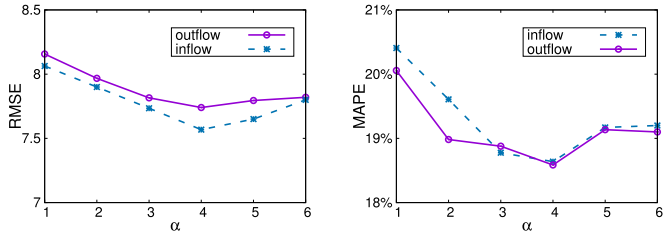
(a) RMSE. (b) MAPE.

Fig. 9. Impact of surrounding observations on the NYC-Bike dataset.



(a) RMSE. (b) MAPE.

Fig. 10. Impact of layer number on the NYC-Taxi dataset.



(a) RMSE. (b) MAPE.

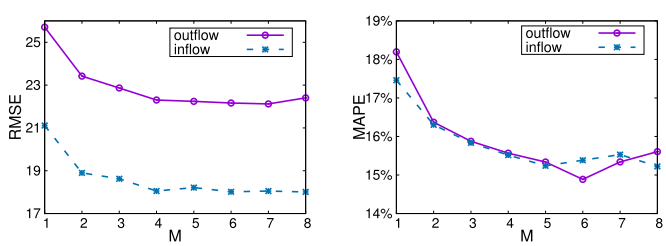
Fig. 11. Impact of layer number on the NYC-Bike dataset.

5.7 Layer Number

In ST-TIS, DLI is stacked α times to ensure the information propagation between regions and to improve the model robustness. We evaluate the effect of α on the prediction performance using RMSE and MAPE. The results for the taxi dataset are presented in Fig. 10. When $\alpha = 1$, only the dependencies on one's neighbouring regions in the graph are considered, resulting in the worst prediction accuracy. When $\alpha \geq 2$, the dependencies on all other regions could be captured. We found that with the layer number α increases, the RMSE and the MAPE declines for both inflow and outflow, indicating the performance improvement. However, we found that when the layer number is too large ($\alpha > 5$ in our experiments), the performance on RMSE and MAPE degrades, because too many layers may result in difficulties of model training. Similar results are observed on the bike dataset (Fig. 11).

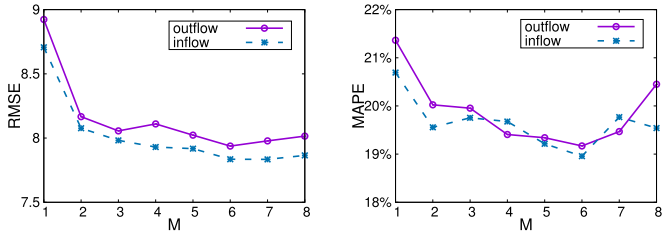
5.8 Head Number

We evaluate the impact of the head number M on the prediction performance. The results of RMSE and MAPE versus the head number M on the taxi and bike datasets are presented



(a) RMSE. (b) MAPE.

Fig. 12. Impact of head number on the NYC-Taxi dataset.



(a) RMSE. (b) MAPE.

Fig. 13. Impact of head number on the NYC-Bike dataset.

in Figs. 12 and 13, respectively. As shown in Figs. 12a and 13a, as the head number increases, the RMSE on the two datasets declines, demonstrating that the multi-head mechanism could benefit the dependency learning and improve the prediction accuracy. We also observe that when the head number becomes larger ($M > 4$ on the taxi dataset while $M > 6$ on the bike dataset), the improvements are not significant. The reason could be that some heads may focus on the same pattern when there are many heads. In terms of MAPE, similar findings could be observed in Figs. 12a and 13b. Moreover, the MAPE increases slightly when the head number becomes large ($M > 6$). It is because increasing the head number leads to more learnable parameters, which would result in difficulties for model training.

6 Conclusion

We propose ST-TIS, a novel, small (in parameters), computationally efficient and highly accurate model for traffic forecasting. ST-TIS employs a spatial-temporal Transformer with information fusion and region sampling to jointly consider the dynamic spatial and temporal dependencies between regions at any individual time slots, and also the possibly periodic spatial-temporal dependency from multiple time slots. In particular, ST-TIS boosts the efficiency and addresses the long-tail issue of the canonical Transformer using a novel region sampling strategy, which reduces the complexity from $O(n^2)$ to $O(n\sqrt{n})$, where n is the number of regions. We have conducted extensive experiments to evaluate ST-TIS, using three real-world datasets. Our experimental results show that ST-TIS significantly outperforms the state-of-the-art deep learning approaches in terms of training efficiency (with a reduction of $46\% \sim 95\%$ on training time and $35\% \sim 98\%$ on network parameters), and hence is efficient in tuning, training, and memory. Despite its small size and fast training, it achieves higher accuracy in its online predictions than other state-of-the-art works (with improvement of up to 9.5% on RMSE, and 12.4% on MAPE).

References

[1] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–55, 2014.

[2] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," 2021, *arXiv:2106.08962*.

[3] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, pp. 1–4.

[4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[5] H. Yao et al., "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.

[6] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5668–5675.

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

[8] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3656–3663.

[9] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[10] Y. Zhou, J. Li, H. Chen, Y. Wu, J. Wu, and L. Chen, "A spatiotemporal attention mechanism-based model for multi-step citywide passenger demand prediction," *Inf. Sci.*, vol. 513, pp. 372–385, 2020.

[11] H. Lin, W. Jia, Y. You, and Y. Sun, "Interpretable crowd flow prediction with spatial-temporal self-attention," 2020, *arXiv:2002.09693*.

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

[13] M. Xu et al., "Spatial-temporal transformer networks for traffic flow forecasting," 2020, *arXiv:2001.02908*.

[14] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, pp. 11106–11115, 2021.

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

[16] B. Pan, U. Demiryurek, and C. Shahabi, "Utilizing real-world transportation data for accurate traffic prediction," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 595–604.

[17] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. Part C: Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, 2002.

[18] R. Silva, S. M. Kang, and E. M. Airoldi, "Predicting traffic volumes and estimating the effects of shocks in massive transportation systems," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 18, pp. 5643–5648, 2015.

[19] Y. Zhang and Y. Xie, "Forecasting of short-term freeway volume with v-support vector machines," *Transp. Res. Rec.*, vol. 2024, no. 1, pp. 92–99, 2007.

[20] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2015, pp. 1–10.

[21] Y. Tong et al., "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 1653–1662.

[22] T. Li, J. Zhang, K. Bao, Y. Liang, Y. Li, and Y. Zheng, "AutoST: Efficient neural architecture search for spatio-temporal prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 794–802.

[23] S. He and K. G. Shin, "Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems," in *Proc. Web Conf.*, 2020, pp. 88–98.

[24] H. Yao et al., "Graph few-shot learning via knowledge transfer," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6656–6663.

[25] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNS?," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9267–9276.

[26] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1720–1730.

[27] X. Wang et al., "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, 2020, pp. 1082–1092.

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

[29] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, pp. 4189–4196, 2021.

[30] 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 *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 914–921.

[31] H. Shi et al., "Predicting origin-destination flow via multi-perspective graph convolutional network," in *Proc. IEEE 36th Int. Conf. Data Eng.*, 2020, pp. 1818–1821.
[32] H. Yuan, G. Li, Z. Bao, and L. Feng, "An effective joint prediction model for travel demands and traffic flows," in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 348–359.
[33] G. Li et al., "A data-driven spatial-temporal graph neural network for docked bike prediction," in *Proc. IEEE 38th Int. Conf. Data Eng.*, 2022, pp. 713–726.
[34] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–16.
[35] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. in Stat. Transl.*, 2014, pp. 103–111.
[36] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3428–3434.
[37] Y. Li and J. M. Moura, "Forecaster: A graph transformer for forecasting spatial and time-dependent data," in *Proc. Eur. Conf. Artif. Intell.*, 2020, Art. no. 274.
[38] S. Li et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 5243–5253, 2019.
[39] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *Proc. Int. Conf. Learn. Representations*, 2020.
[40] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[41] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 338–348.
[42] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3337–3348, Jun. 2020.
[43] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining Knowl. Discov.*, vol. 26, no. 2, pp. 275–309, 2013.
[44] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 922–929.
[45] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
[46] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. iNeural Inf. Process. Syst.*, 2015, pp. 802–810.
[47] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

Guanyao Li received the bachelor's of engineering degree from Zhejiang University, in 2015, the master's of engineering degree from National Chiao Tung University (NCTU), Taiwan, in 2017, and the PhD degree in computer science from The Hong Kong University of Science and Technology (HKUST), in 2022. Currently, he is a data scientist with Guangzhou Urban Planning and Design Survey Research Institute. His research interests include spatial-temporal data mining and urban computing.

Shuhan Zhong received the bachelor's and master's of engineering degrees from Wuhan University, and the master's of science degree from the Hong Kong University of Science and Technology (HKUST). He is currently working toward the PhD degree with the Department of Computer Science and Engineering, HKUST. His research interests include time series forecasting, deep learning, spatio-temporal Big Data modeling, and urban computing.

Xingdong Deng received the PhD degree in engineering with the South China University of Technology, Guangzhou, China, in 2011. He is currently the president of Guangzhou Urban Planning and Design Survey Research Institute and an adjunct professor of the South China University of Technology. He received the Special Government Allowances Expert of the State Council and was awarded "Guangzhou Excellent Expert," in 2012. He was included in the second echelon of the 121 Talent Echelon Project, in 2014. His research interests include urban planning, urban transportation, urban design, geospatial technology, and geotechnical engineering.

Letian Xiang received the bachelor's of engineering degree in computer science from the Hong Kong University of Science and Technology. He is currently working toward the MS degree in information networking with Carnegie Mellon University. Her interests include high-dimensional data management and software systems.

S.-H. Gary Chan received the BSE degree (highest honor) in electrical engineering from Princeton University (Princeton, NJ), with certificates in applied and computational mathematics, engineering physics, and engineering and management systems, and the MSE and PhD degrees in electrical engineering with a minor in business administration from stanford university (Stanford, CA). He is currently a professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), and affiliate professor in innovation, policy and entrepreneurship thrust with HKUST(GZ). His research interests include smart sensing and IoT, indoor localization and mobile computing, video/location/data analytics, cloud and fog/edge AI, and IT entrepreneurship. He has been an associate editor of *IEEE Transactions on Multimedia*, and guest editor of *ACM Transactions on Multimedia Computing, Communications and Applications*, *IEEE Signal Processing Magazine*, *IEEE Communication Magazine*, etc. He has been area chair of the multimedia symposium of IEEE Globecom and IEEE ICC for many years. His research technologies have received local and international awards due to their innovations and commercial or societal impacts. Notably, he received Hong Kong Chief Executive's Commendation for Community Service for "outstanding contribution to the fight against COVID-19" in 2020. He is a chartered fellow of The Chartered Institute of Logistics and Transport (FCILT).

Ruiyuan Li received the BE and MS degrees from Wuhan University, China, in 2013 and 2016, respectively, and the PhD degree from Xidian University, China, in 2020. He is an associate professor with Chongqing University, China. He was the head of Spatio-Temporal Data Group in JD Intelligent Cities Research, leading the research and development of JUST (JD Urban Spatio-Temporal data engine). Before joining JD, he interned in Microsoft Research Asia from 2014 to 2017. His research focuses on Spatio-temporal Data Management and Urban Computing.

Yang Liu received the PhD degree in cartography and geographic information system from Wuhan University, in 2008. He is currently the head of the Technology Department, Guangzhou Urban Planning and Design Survey Research Institute. He also has the title of registered surveyor, registered planner, and professional engineer. His research interests include spatial data mining and disaster prediction.

Ming Zhang receivied the PhD degree in nature resource from Beijing Normal University, in 2015. From 2015-2020, he worked as a senior research scientist with Aramco Beijing Research Center. In 2020, he jointed the Innovation Center, Guangzhou Urban Planning and Design Survey Research Institute, mainly focused on urban computing and digital twin city.

Chih-Chieh Hung received the MS and PhD degrees from the National Chiao Tung University, Taiwan, in 2005 and 2011, respectively. Currently, he is an assistant professor with the department of management information system, National Chung Hsing University, He published some papers in several prestigious conferences, such as IEEE International Conference on Data Engineering (ICDE), IEEE International Conference on Data Mining (ICDM) and ACM Conference on Information and Knowledge Management (ACM CIKM) and prestigious journals (e.g., *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics*, *The VLDB Journal*). He has the Best Paper Award in ACM Workshop on location-based social network 2009. His research interests include data mining, mobile and pervasive computing, Big Data analytics, and artificial intelligence.

Wen-Chih Peng (Member, IEEE) received the BS and MS degrees from the National Chiao Tung University, Taiwan, in 1995 and 1997, respectively, and the PhD degree in electrical engineering from the National Taiwan University, Taiwan, R.O.C, in 2001. Currently, he is a professor with the department of Computer Science, National Chiao Tung University, Taiwan. Prior to joining the department of Computer Science and Information Engineering, National Chiao Tung University, he was mainly involved in the projects related to mobile computing, data broadcasting and network data management. He published some papers in several prestigious conferences, such as IEEE International Conference on Data Engineering (ICDE), IEEE International Conference on Data Mining (ICDM) and ACM Conference on Information and Knowledge Management (ACM CIKM) and prestigious journals (e.g., *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Parallel and Distributed Systems*). He has the Best Paper Award in ACM Workshop on location-based social network 2009 and the Best Student Paper Award in IEEE International Conference on Mobile Data Management 2011. His research interests include mobile computing, network data management and data mining.