DSTAGNN: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting

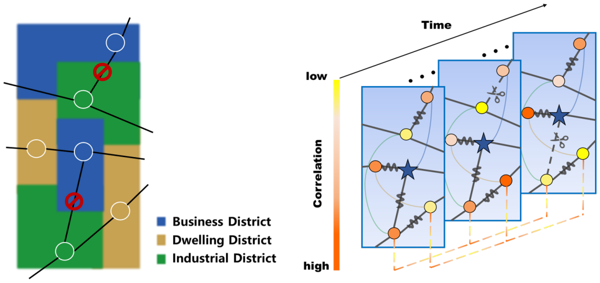
Shiyong Lan [*1] **Yitong Ma** [*1] **Weikang Huang** [2] **Wenwu Wang** [3] **Hongyu Yang** [1] **Piaoyang Li** [1]

Abstract

As a typical problem in time series analysis, traffic flow prediction is one of the most important application fields of machine learning. However, achieving highly accurate traffic flow prediction is a challenging task, due to the presence of complex dynamic spatial-temporal dependencies within a road network. This paper proposes a novel Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN) to model the complex spatial-temporal interaction in road network. First, considering the fact that historical data carries intrinsic dynamic information about the spatial structure of road networks, we propose a new dynamic spatial-temporal aware graph based on a data-driven strategy to replace the pre-defined static graph usually used in traditional graph convolution. Second, we design a novel graph neural network architecture, which can not only represent dynamic spatial relevance among nodes with an improved multi-head attention mechanism, but also acquire the wide range of dynamic temporal dependency from multi-receptive field features via multi-scale gated convolution. Extensive experiments on real-world data sets demonstrate that our proposed method significantly outperforms the state-of-the-art methods.

1. Introduction

With a growing number of vehicles in road networks, there is increasing pressure on traffic management systems. The development of Intelligent Transportation Systems (ITS) is urgently needed for efficient traffic management. Traffic flow prediction plays a key role in ITS, as it is a necessary prerequisite for the implementation of an intelligent traffic management system.



(a) Road network of a certain zone.
(b) Dynamic spatial-temporal relevance in traffic flow data.

Figure 1: Dynamic spatial-temporal correlations in real-world traffic data. In (a), the black line represents the actual road, and the nodes indicate recording points. In (b), elastic connection means that the spatial adjacency state between recording points is dynamically changing, while scissors cutting means that the road may be temporarily closed. The curve shows the spatial dependency of inter-regional nodes of similar urban functions, and the dashed line represents the temporal dependency among different time steps.

The traffic at each recording point (also called node in the road network) presents patterns of highly dynamic and complex temporal-spatial dependency. On the one hand, the agents in the road network are affected by various random factors (such as potentially random traffic accidents, and temporary road closure for maintenance), which influences the arrival time interval between adjacent nodes. On the other hand, similar urban functional areas may lead to correlations of the traffic data between the nodes in the road network, regardless of the distance between them. For example, during the daily rush hours, the office clustered areas are often congested at the same time. Fig. 1 illustrates the dynamic spatial-temporal correlation within the traffic flow. Moreover, the complex spatial-temporal interaction in road traffic network can substantially degrade the performance of traffic prediction algorithms (Yin et al., 2020).

In recent years, deep learning methods have become a popular choice for traffic flow prediction from high-dimensional

[*] Equal contribution , This work was funded in part by Key R&D Project of Sichuan Science and Technology Department, China (2021YFG0300), and in part by Key Project of Natural Science Foundation of China (U20A20161). [1] College of Computer Science, Sichuan University, Chengdu, China. [2] National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu, China. [3] Department of Electrical and Electronic Engineering, University of Surrey, Guildford, UK. Correspondence to: Shiyong Lan <lanshiyong@scu.edu.cn>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

spatial-temporal data. One type of classic methods is to cascade convolutional neural networks (CNN) and recurrent neural networks (RNN) to address the spatial-temporal dependencies in road network (Zhang et al., 2016; Li & Shahabi, 2018). Although the CNN method is suitable for capturing local spatial correlations in regular spatial grids, it is out at elbows for traffic prediction in road network which has a non-grid structure containing various long-distance spatial correlations.

Another method is to use graph convolutional network (GCN) for representing the spatial-temporal correlation of time-series data with non-Euclidean spatial structure that is suitable for road network (Li et al., 2017; Yu et al., 2017; Zhao et al., 2019). Nevertheless, most existing GCN methods leverage a static adjacency matrix to describe the spatial correlation of road network, which cannot reflect the actual dynamic changes of spatial dependency within the road network. Recently, SFTGNN (Li & Zhu, 2021) uses the Dynamic Time Warping (DTW) (Berndt & Clifford, 1994) technique to capture the similarity between traffic nodes through shape matching between data sequences. However, we argue that the spatial dependency between nodes is not only related to the shape similarity of data sequences, but also to the semantic relevance between them, just like in language processing where two semantically similar sentences may have different language structures (Kusner et al., 2015).

In addition, regardless of the short-term or long-term time scale, time series traffic data has an intricate interweaving between dynamic similar patterns and random irregular patterns. At the macro level, similar data patterns are presented, such as homomorphic wide-dynamic congestion during rushing hours, and stable traffic pattern deviations between weekdays and weekends. At the micro level, traffic data presents dynamic and complex fluctuations, in which the traffic states at any given time are significantly different due to the random and complex interactions among a large number of unique traffic components (such as different driving habits, vehicle mechanical characteristics, and adaptive traffic control strategies). However, most existing methods, such as (Oord et al., 2016; Bai et al., 2018; Zhou et al., 2021), lack attention to the simultaneous use of short-term and long-term temporal correlations, as a result, they have limitations in capturing the dynamic temporal dependency within the road network.

To address the above problems, we propose a novel neural network framework, Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN)[1], for traffic flow forecasting, which can capture both short-range and long-range spatial-temporal correlations of the road network. The main contributions of this paper can be summarized as follows:

- We construct a novel graph to capture dynamic attributes of spatial association among nodes by mining from their historic traffic flow data directly, without using a predefined static adjacency matrix. We call this dynamic association attribute as Spatial-Temporal Aware Distance (STAD).
- We design a new spatial-temporal attention module to exploit the dynamic spatial correlation within multi-scale neighborhoods based on multi-order Chebyshev polynomials in GCN. Specifically, the spatial weights of the input for each order of the Chebyshev polynomials are adjusted adaptively by an improved self-attention, meanwhile, the wide range of temporal dependency is exploited by the multi-head self-attention.
- An improved gated convolution module is designed, which can further enhance the awareness of the model to dynamic temporal dependency within the road network, via fusing temporal features of multi-receptive fields with multi-scale gated convolution.
- Extensive experiments on real road traffic data sets demonstrate the improved performance of our proposed algorithm, as compared with several baselines including the state of the art algorithms.

2. Related Works

2.1. Graph Convolution

Graph convolutional networks (GCN) are widely used in many tasks (Wu et al., 2020). They usually include two types of methods. One is spectral-type GCN. Bruna *et al.* (2013) extended convolution operations on the graph with the help of Laplacian spectrum in spectral domain. However, the calculation of the spectral domain convolution involves calculating all the eigenvalues of the Laplacian matrix, which is computationally expensive. ChebNet (Defferrard et al., 2016) uses the Chebyshev-polynomial expansion of the eigenvalue-based diagonal matrix to approximate the graph convolution in order to reduce its computational complexity. In the classic GCN (Kipf & Welling, 2016), graph convolution is used in a deep network architecture similar to CNN to achieve effective embedding of the graph structure and node attributes.

The other one is the spatial-type GCN. Micheli and Alessio (2009) performed graph convolution by directly summarizing the neighborhood information of nodes. Atwood *et al.* (2016) regarded graph convolution as a diffusion process, and introduced the probability of spreading information through different paths between any two nodes. Veličković *et al.* (2017) proposed Graph Attention Network (GAT) where an attention mechanism is adopted to adjust the weights between adjacent nodes.

[1] https://github.com/SYLan2019/DSTAGNN

2.2. Measuring Differences in Probability Distributions

In a road network, the traffic data collected from each node can be treated as discrete data in a multi-dimensional space. Therefore, the correlation between two nodes can be obtained by measuring the similarity between the data captured at these nodes, e.g. using the Minkowski distance (Singh et al., 2013). Furthermore, considering the relationship among the local and the global data, we can convert the discrete data into probability distributions of each node, and then compute the differences of probability distributions to obtain the spatial correlation between nodes. There are many methods for measuring the difference in probability distributions, such as Kullback-Leibler (KL) divergence (Goldberger et al., 2003), Hellinger Distance (Kailath, 1967), and Total Variation Distance (Devroye et al., 2018). These methods all compare the probability density functions of the corresponding points in the time series, however, they ignore the geometric characteristics within the data. To mitigate this problem, the Wasserstein distance (Panaretos & Zemel, 2019) has emerged as an effective method. Given two probability distributions u and v, the probability mass $u(a)$ at the current position a, and the probability mass $v(a)$ stored at the final position a, the Wasserstein Distance is defined as:

$$W[u, v] = \inf_{\gamma \in \Pi[u,v]} \int_x \int_y \gamma(x, y) d(x, y) dx dy \tag{1}$$

where γ is a joint probability distribution $\Pi[u, v]$, which requires its marginal distribution to be exactly u and v, that is $\int \gamma(x, y)dy = u(x)$ and $\int \gamma(x, y)dx = v(y)$, $d(x, y)$ is the cost of moving the unit mass from x to y, which is generally derived from the Minkowski distance. Here inf means the infimum, that is, the solution with the smallest cumulative moving distance from all the schemes to convert one probability distribution u to another v, and the cost of this scheme is $W[u, v]$.

2.3. Spatial-Temporal Forecasting

Recently, various deep learning methods have been proposed to capture the spatial-temporal correlation for traffic forecasting (Li et al., 2017; Yao et al., 2018). Despite their promising performance, the spatial dependency derived from these models cannot well reveal its dynamic nature due to the use of a pre-defined static adjacent graph. In ASTGCN (Guo et al., 2019) attention mechanisms are incorporated into standard convolution, where node information is updated by fusing information from adjacent time slices. However, the spatial dependencies only came from the static adjacency graph structure, which may miss potential dynamic dependency information. Graph WaveNet (Wu et al., 2019) and AGCRN (Bai et al., 2020) found hidden spatial dependency through learnable embedding from nodes, but with these models, the spatial-temporal layers cannot be stacked while expanding the receptive field. Transformer algorithms (Park et al., 2019; Wang et al., 2020) resorted to a self-attention mechanism to model the spatial-temporal correlations. However, due to the use of the auto-regressive mechanism, these algorithms are prone to error accumulation in the inference stage.

Different from the aforementioned methods, some works have focused on designing new graph structures. STSGCN (Song et al., 2020) concatenated the spatial graphs of multi-neighborhood time steps. However, this method only stitched the local spatial graphs within a fixed number of time step (e.g., 3), which may be corrupted by missing measurements in the collected data. On the basis of (Song et al., 2020), STFGCN (Li & Zhu, 2021) constructed a spatial-temporal fusion graph for traffic prediction, in which the static adjacent graph was supplemented by the information derived from historic sequences, similar to DTW (Berndt & Clifford, 1994). STGODE (Fang et al., 2021) incorporated Ordinary Differential Equations (ODE) into GCN based on the combination of semantic adjacency matrix and static spatial adjacency matrix, in which the semantic adjacency matrix is also calculated by using DTW. Nevertheless, these models do not explicitly consider the dynamic spatial-temporal dependency between the nodes of the road network.

3. Methodology

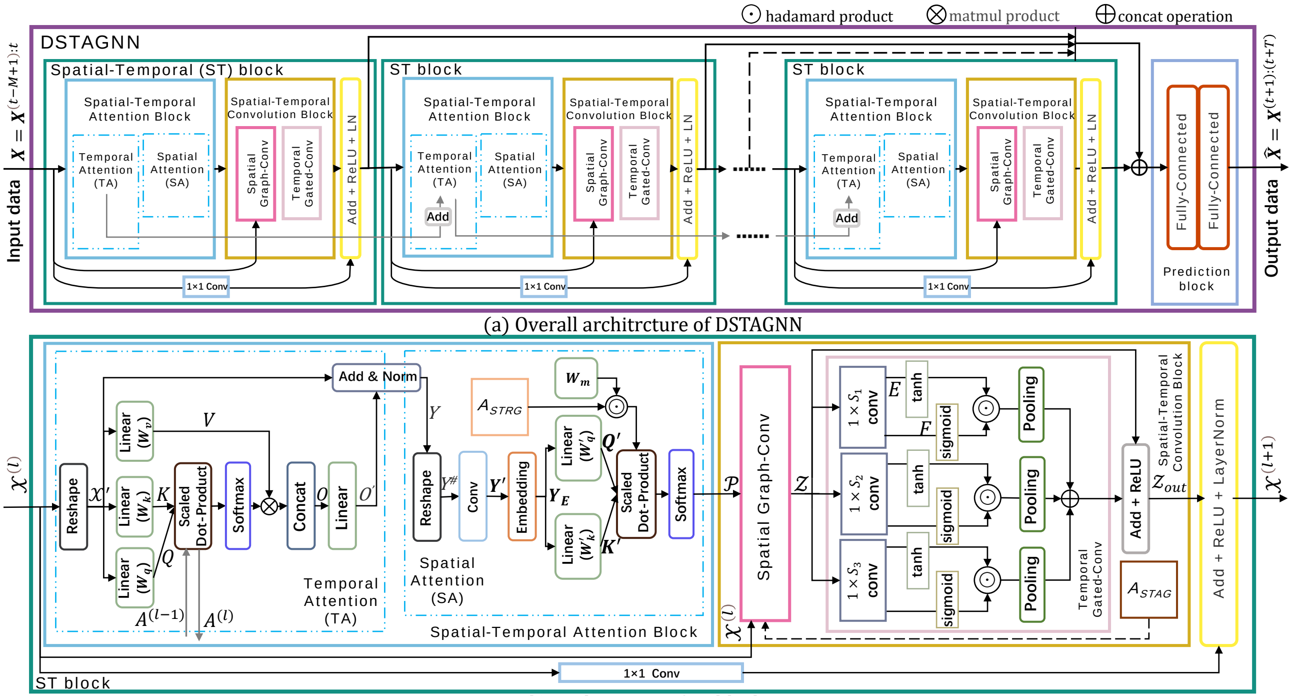
3.1. Preliminaries

We denote the road network as a graph $\mathcal{G} = (V, E)$, where V represents a set of N nodes (i.e. recording points) within the road network, and E is a set of edges that indicate the connectivity between nodes. The adjacency matrix of $\mathcal{G}$ is represented by $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, and $\boldsymbol{A}_{ij}$ is equal to 1 if $v_i, v_j \in V$ and $(v_i, v_j) \in E$. Hence, the traffic status at any time step t can be regarded as a graph signal $X^t \in \mathbb{R}^{N \times C_p}$, where C_p counts the types of traffic parameters (e.g., traffic volume, speed, etc.). In this work, we aim to predict only one type of parameter, i.e. the traffic volume (hence $C_p = 1$). Given the recorded data $X^{(t-M+1):t} \in \mathbb{R}^{N \times C_p \times M}$, a model $\mathcal{F}$ can be trained to predict the traffic volumes of the future T time steps $X^{(t+1):(t+T)} \in \mathbb{R}^{N \times C_p \times T}$ on the road network $\mathcal{G}$, as follows:

$$X^{(t+1):(t+T)} = \mathcal{F}\left[X^{(t-M+1):t}; \mathcal{G}\right] \tag{2}$$

3.2. Network architecture

The proposed DSTAGNN is shown in Fig. 2, which is composed of stacked Spatial-Temporal (ST) blocks and a prediction layer. The output of each ST block is concatenated and then sent to the prediction layer in a manner similar to residual connection. The specific details of the model are discussed in the following subsections.



(a) Overall archirtcture of DSTAGNN

(b) Archirtcture of ST block

Figure 2: Detailed framework of DSTAGNN. (a) is overall structure of DSTAGNN, consisting of several Spatial-Temporal (ST) blocks and a prediction block. (b) shows the detail of the ST block, which is combined with a Spatial-Temporal Attention (STA) block and a spatial-temporal convolution block. The STA block includes temporal attention module (TA) and spatial attention module (SA). The spatial-temporal convolution block includes a graph convolutional layer and a multi-scale GTU convolutional layer. The Spatial-Temporal Relevance Graph ($\boldsymbol{A}_{STRG}$) with correlation information between nodes is used in SA to further adjust spatial-temporal attention, and the pre-defined static adjacency graph used in conventional graph convolution is replaced by the Spatial-Temporal Aware Graph ($\boldsymbol{A}_{STAG}$) with dynamic spatial dependency information. $\boldsymbol{A}_{STRG}$ and $\boldsymbol{A}_{STAG}$ are derived from $\boldsymbol{A}_{STAD}$ (see Eq. 5 for details) based on historical traffic data.

3.2.1. Spatial-Temporal Aware Graph Construction

In the road network, the connectivity between nodes cannot fully reflect their spatial dependency. Inaccurate spatial dependency can degrade the performance of traffic flow prediction. This section discusses how to extract more accurate spatial dependencies. We argue that the spatial dependency between nodes in the road network mainly comes from two situations that cannot be characterized by a simple static adjacency matrix as used in conventional methods. One is due to the dynamic effect of traffic flow propagation between adjacent connected nodes. The other is caused by similar urban functions between nodes, even if these nodes are far apart.

With the traffic flow data at each day and each node of the road network, we can represent the dynamic spatial dependency among nodes by capturing the correlation attributes between the probability distributions of each node, using e.g. the Wasserstein Distance (Panaretos & Zemel, 2019) which measures the minimum effort taken to reconfigure a probability distribution to another.

Therefore, we propose a new data-driven strategy to formulate the degree of spatial association among nodes directly from historic traffic data. We name this degree of spatial association as Spatial-Temporal Aware Distance (STAD), and call the structure as Spatial-Temporal Aware Graph (STAG).

Take the traffic flow $X^f \in \mathbb{R}^{D \times d_t \times N}$ at the N recording points for D days as an example, where d_t is the number of recording times per day (if recordings are taken once every 5 minutes, then $d_t = 288$). For each recording point, the one-day traffic data is treated as a vector, then a set of multi-day traffic data is denoted as a vector sequence. For example, the vector sequence obtained at recording point n ($n \in \mathbb{N}$) is denoted as $\boldsymbol{X}_n^f = (\boldsymbol{w}_{n1}, \boldsymbol{w}_{n2}, ..., \boldsymbol{w}_{nD})$, $\boldsymbol{w}_{nd} \in \mathbb{R}^{d_t}$, where $d \in [1,\ D]$. We first extract the daily traffic volume information at each recording point through the modulus length of the traffic flow vector:

$$m_{nd} = \frac{\|\boldsymbol{w}_{nd}\|_2}{Z_n}, \qquad Z_n = \sum_{d=1}^{D} \|\boldsymbol{w}_{nd}\|_2 \tag{3}$$

where $\|\cdot\|_2$ represents Euclidean norm. In this way, the vector sequence of the recording point n is transformed into a probability distribution $P_n\{X_d = m_{nd}\}$, and each day has a probability mass $m_{nd} \in [0,\ 1]$ and $\sum_d m_{nd} = 1$, which denotes the proportion of traffic volume in a certain day over a period of time. Then we need to obtain the conversion cost of each probability mass. We use the cosine distance between traffic flow vectors as a cost function. For example, the conversion cost of the traffic flow vector $\boldsymbol{w}_{n_1 i}$ on the i-th day at n_1 point and the traffic flow vector $\boldsymbol{w}_{n_2 j}$ on the j-th day at n_2 recording point is:

$$cost\left(\boldsymbol{w}_{n_1 i}, \boldsymbol{w}_{n_2 j}\right) = 1 - \frac{\boldsymbol{w}_{n_1 i}^{\top} \cdot \boldsymbol{w}_{n_2 j}}{\left\|\boldsymbol{w}_{n_1 i}\right\|_2 \times \left\|\boldsymbol{w}_{n_2 j}\right\|_2} \tag{4}$$

where the superscript $\top$ is a transpose operator for a vector or matrix. Thus, the spatial-temporal aware distance is:

$$\begin{gathered} d_{STAD}(n_1, n_2) \triangleq STAD\left(\boldsymbol{X}_{n_1}, \boldsymbol{X}_{n_2}\right) = \\ \inf_{\gamma\in\Pi\left[P_{n_1}, P_{n_2}\right]} \int_x \int_y \gamma(x,y)\left(1 - \frac{\boldsymbol{w}_{n_1 x}^{\top} \cdot \boldsymbol{w}_{n_2 y}}{\sqrt{\boldsymbol{w}_{n_1 x}^{\top}\boldsymbol{w}_{n_1 x}} \times \sqrt{\boldsymbol{w}_{n_2 y}^{\top}\boldsymbol{w}_{n_2 y}}}\right) dxdy \\ \text{s.t.} \int \gamma(x,y)dy = \frac{\left\|\boldsymbol{w}_{n_1 x}\right\|_2}{\sum_{x=1}^{D}\left\|\boldsymbol{w}_{n_1 x}\right\|_2}, \int \gamma(x,y)dx = \frac{\left\|\boldsymbol{w}_{n_2 y}\right\|_2}{\sum_{y=1}^{D}\left\|\boldsymbol{w}_{n_2 y}\right\|_2} \end{gathered} \tag{5}$$

We get a matrix $\boldsymbol{A}_{STAD} \in \mathbb{R}^{N\times N}$ that represents the degree of relevance between the recording points, where $\boldsymbol{A}_{STAD}\left[i,\, j\right] = 1 - d_{STAD}(i,\, j) \in [0,\ 1]$. Under the premise of satisfying a certain sparsity level P_{sp} (as a hyper-parameter, e.g. 0.01), for each node i of the road network, we retain the $N_r = N \times P_{sp}$ elements that have the biggest value (and set the remaining elements to 0) in the i-th row of $\boldsymbol{A}_{STAD}$, hence we get the Spatial-Temporal Relevance Graph $\boldsymbol{A}_{STRG} \in \mathbb{R}^{N\times N}$. We use this $\boldsymbol{A}_{STRG}$ as prior knowledge to supplement the attention $\mathcal{P}$ learned from the spatial-temporal attention module. Furthermore, we use a learnable parameter $\boldsymbol{W}_m \in \mathbb{R}^{N\times N}$ (shown in Fig. 2 (b)) to adjust the influence of $\boldsymbol{A}_{STRG}$ on $\mathcal{P}$. In addition, we obtain $\boldsymbol{A}_{STAG} \in \mathbb{R}^{N\times N}$ as the graph structure by binarizing $\boldsymbol{A}_{STRG}$, i.e. setting those elements to 1 if their values are non-zero, which means that the most relevant N_r nodes for each given node are aggregated in graph convolution.

3.2.2. Spatial-Temporal Attention Block

The spatial-temporal aware distance (STAD) can provide more accurate estimates for the dependencies between nodes, but the dynamic characteristics of these dependencies need to be further refined to adapt to the changes in real-time data. Therefore, we design a new spatial-temporal attention module to further enhance the representation of dynamic spatial-temporal dependency, by combining sequentially the temporal attention with spatial attention.

Temporal attention The multi-head self-attention offers a mechanism for parallelism, which can effectively focus on the long-range correlation in time series data. We leverage this mechanism to capture the dynamic temporal dependency between nodes. For the multi-head attention of H heads, we define the variable:

$$\mathcal{X}'^{(l)}\boldsymbol{W}_q^{(l)} \triangleq Q^{(l)}, \quad \mathcal{X}'^{(l)}\boldsymbol{W}_k^{(l)} \triangleq K^{(l)}, \quad \mathcal{X}'^{(l)}\boldsymbol{W}_v^{(l)} \triangleq V^{(l)} \tag{6}$$

$$\operatorname{Att}\left(Q^{(l)}, K^{(l)}, V^{(l)}\right) = \operatorname{Softmax}\left(A^{(l)}\right)V^{(l)}, \quad A^{(l)} = \frac{Q^{(l)}K^{(l)\top}}{\sqrt{d_h}} + A^{(l-1)} \tag{7}$$

where $\mathcal{X}'^{(l)} \in \mathbb{R}^{c^{(l-1)}\times M\times N}$ is reshaped from the input of the lth ST block $\mathcal{X}^{(l)} \in \mathbb{R}^{N\times c^{(l-1)}\times M}$, which represents the $c^{(l-1)}$-dimensional feature extracted from the N recording points at time steps $t-M+1, t-M+2, ..., t$ output by the $(l-1)$th ST block. $\boldsymbol{W}_{q,k,v}^{(l)} \in \mathbb{R}^{N\times d}$ are the parameters learned to obtain $Q^{(l)}, K^{(l)}, V^{(l)} \in \mathbb{R}^{c^{(l-1)}\times M\times d}$. Using the residual attention idea (He et al., 2020), we directly connect the output from the temporal attention module in each ST block with that in the next ST block, which enhances the connection among the temporal attention in different layers of the ST block. This residual attention mechanism, i.e., $A^{(l)}$ in Fig. 2(b) derived by Eq. (7), allows the model to fuse both shallow and deep temporal dependency, which can not only reduce the risk of gradient disappearance, but also exploit effectively the dynamic temporal dependency within traffic data.

After that, $Q^{(l)}, K^{(l)}, V^{(l)}$ are projected H times with H different matrices, respectively, and then stitched together, as follows (where, to simplify the notation, the superscript l has been dropped, and the superscript h indicates the h-th attention head, $h = 1, 2, ..., H$),

$$O^{(h)} = \operatorname{Att}\left(Q\boldsymbol{W}_q^{(h)}, K\boldsymbol{W}_k^{(h)}, V\boldsymbol{W}_v^{(h)}\right) \tag{8}$$

$$O = \left[O^{(1)}, O^{(2)}, \ldots, O^{(H)}\right] \tag{9}$$

$$Y = \operatorname{LayerNorm}(\operatorname{Linear}(\operatorname{Reshape}(O) + \mathcal{X}')) \tag{10}$$

where $\boldsymbol{W}_{q,k,v}^{(h)} \in \mathbb{R}^{d\times d_h}(d_h = d/H)$, then $O \in \mathbb{R}^{c^{(l-1)}\times M\times H\times d_h}$ concatenates the multi-head outputs from the temporal attention, then it is input into a fully connected layer to get the output $O' \in \mathbb{R}^{c^{(l-1)}\times M\times N}$ of the Temporal Attention (TA) module. Finally, following the residual connection of O' and input $\mathcal{X}'$, the output $Y \in \mathbb{R}^{c^{(l-1)}\times M\times N}$ is obtained through a normalization layer, and then Y is input into the Spatial Attention (SA) module.

Spatial attention The TA module adaptively encodes the time series data, and obtains a feature representation with global dynamic temporal dependencies. Here, we design an improved self-attention mechanism to calculate the spatial dependency from the output of the TA module, in which the weight coefficients from the two branches of the input embedding vectors (i.e., Query (Q') and Key (K')) are calculated. However, unlike traditional transformers, the weight coefficients obtained are not used to weight the Value (V') branch of the input embedding vector Y_E, but used to adjust $\boldsymbol{A}_{STRG}$, as shown in Fig. 2(b).

We first transpose the output Y of the TA module to $Y^{\#} \in \mathbb{R}^{c^{(l-1)}\times N\times M}$, then map the time dimension M to the high-dimensional space with dimension d_E and aggregate feature dimension $c^{(l-1)}$ through one-dimensional convolution, which gives us a two-dimensional matrix $\boldsymbol{Y}' \in \mathbb{R}^{N\times d_E}$

denoting the set of the embedded vector representations for each recording point. Then, we add positional information to $\boldsymbol{Y}'$ through an embedding layer to get $\boldsymbol{Y}_E$. Instead of using the self-attention fully generated from $\boldsymbol{Y}_E$ as in conventional transformers, here we introduce the temporal-spatial relevance graph $\boldsymbol{A}_{STRG}$ with learned correlation between nodes to amend the attention in the SA module. Thus the improved spatial attention with H heads is denoted as:

$$\boldsymbol{P}^{(h)} = \text{Softmax}\left(\frac{\left(\boldsymbol{Y}_E\boldsymbol{W}'^{(h)}_k\right)^{\top}\left(\boldsymbol{Y}_E\boldsymbol{W}'^{(h)}_q\right)}{\sqrt{d_h}} + \boldsymbol{W}^{(h)}_m \odot \boldsymbol{A}_{STRG}\right) \tag{11}$$

$$\mathcal{P} = \left[\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ldots, \boldsymbol{P}^{(H)}\right] \tag{12}$$

where $\boldsymbol{W}'^{(h)}_k, \boldsymbol{W}'^{(h)}_q \in \mathbb{R}^{d_E \times d_h}, \boldsymbol{W}^{(h)}_m \in \mathbb{R}^{N\times N}$ are learnable parameters, $\odot$ is the element-wise Hadamard product, $\boldsymbol{W}^{(h)}_m$ is used to amend $\boldsymbol{A}_{STRG}$ for adjusting the attention of each head $\boldsymbol{P}^{(h)} \in \mathbb{R}^{N\times N}$, and the output $\mathcal{P} \in \mathbb{R}^{H\times N\times N}$ denotes the dynamic spatial-temporal attention tensor by combining the outputs from each head.

3.2.3. Spatial-Temporal Convolution Block

Spatial graph convolution For traffic road networks, many studies focus on the connectivity and globality of the road network, using pre-defined graph structure for graph convolution, and obtaining node features by aggregating information from its neighboring nodes (Yu et al., 2017; Guo et al., 2019; Song et al., 2020). In order to make full use of the topological characteristics of the traffic network, we retain the above idea and use graph convolution based on Chebyshev polynomial approximation (Simonovsky & Komodakis, 2017) to learn structure-aware node features. Unlike the existing methods, however, we use our spatial-temporal aware graph (STAG), instead of the pre-defined graph structure. In addition, each term of the Chebyshev polynomial is dynamically adjusted to extract more meaningful and wider-range of features on the traffic network in the spatial dimension.

In this paper, the scaled Laplacian matrix for Chebyshev polynomial is defined as $\tilde{\boldsymbol{L}} = \frac{2}{\lambda_{\max}}(\boldsymbol{D} - \boldsymbol{A}^*) - \boldsymbol{I}_N$, where $\boldsymbol{A}^* = \boldsymbol{A}_{STAG}$, $\boldsymbol{I}_N$ is the unit matrix, $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is the degree matrix, and the element $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}^*_{ij}$, $\lambda_{\max}$ is the maximum eigenvalue of the Laplacian matrix $\boldsymbol{L} = (\boldsymbol{D} - \boldsymbol{A}^*)$.

In graph convolution, the information at each node is derived from the nodes in its neighborhood. To incorporate the dynamical attributes of the nodes, we aggregate the information from the graph signal $x = \boldsymbol{x}_t \in \mathbb{R}^N$ at each time step by using the K-th order Chebyshev polynomial T_k, as follows:

$$g_\theta * Gx = g_\theta(\boldsymbol{L})x = \sum_{k=0}^{K-1} \boldsymbol{\theta}_k \left(T_k(\tilde{\boldsymbol{L}}) \odot \boldsymbol{P}^{(k)}\right) x \tag{13}$$

where g_θ denotes the approximate convolution kernel, $*G$ denotes the graph convolution operation, the learnable vector $\boldsymbol{\theta} \in \mathbb{R}^K$ contains the polynomial coefficients, updated iteratively in training, and $\boldsymbol{P}^{(k)} \in \mathbb{R}^{N\times N}$ is the spatial-temporal attention matrix of the k-th head. For the multi-channel input $\mathcal{X}^{(l)} \in \mathbb{R}^{N\times c^{(l-1)}\times M}$ of this module, the feature of each node has $c^{(l-1)}$ channels, and $g_\theta \in \mathbb{R}^{K\times c^{(l-1)}\times c^{(l)}}$ is the convolution kernel parameter (Kipf & Welling, 2016). As a result, each node can aggregate the information from $0 \sim (K-1)$-th order adjacent nodes.

Temporal gated convolution Different from TSSRGCN (Chen et al., 2020) where a cycle-based dilated deformable convolution was used to capture both long-term and short-term temporal dynamics in traffic data, we propose a new Multi-scale Gated Tanh Unit (M-GTU) convolution module to capture the temporal dynamic information of the traffic flow data. The specific structure of the module is shown in Fig. 2 (b), which is mainly composed of three Gated Tanh Unit (GTU) (Dauphin et al., 2017) modules with different receptive fields.

The input of the temporal gated convolution module is $\mathcal{Z}^{(l)} \in \mathbb{R}^{N\times M\times C^{(l)}}$. The traditional GTU doubles the number of channels by using the convolution kernel $\Gamma \in \mathbb{R}^{1\times S\times c^{(l)}\times 2c^{(l)}}$, where its kernel size is $1 \times S$, that is, $\mathcal{Z}'^{(l)} = \Gamma * \mathcal{Z}^{(l)} \in \mathbb{R}^{N\times(M-(S-1))\times 2C^{(l)}}$. Therefore, the GTU in time dimension can be defined as:

$$\Gamma *_\tau \mathcal{Z}^{(l)} = \phi(E) \odot \sigma(F) \in \mathbb{R}^{N\times(M-(S-1))\times 2C^{(l)}} \tag{14}$$

where $*_\tau$ is gated convolution operator, $\phi(\cdot)$ is tanh function, $\sigma(\cdot)$ is *sigmoid* function, and E and F are the first half and the second half of $\mathcal{Z}'^{(l)}$ relative to the channel dimension, respectively. The receptive field of the time dimension is expanded via stacking gated convolution, to improve its ability in extracting the long-range temporal dependencies in the data. Furthermore, we propose M-GTU by extending GTU as follows:

$$\begin{aligned} Z^{(l)}_{out} &= \text{M-GTU}(\mathcal{Z}^{(l)}) = \text{ReLU}(\text{Concat}(\text{Pooling}(\Gamma_1 *_\tau \mathcal{Z}^{(l)}), \\ &\quad \text{Pooling}(\Gamma_2 *_\tau \mathcal{Z}^{(l)}), \text{Pooling}(\Gamma_3 *_\tau \mathcal{Z}^{(l)})) + \mathcal{Z}^{(l)}) \end{aligned} \tag{15}$$

where Γ_1, Γ_2, Γ_3 are the convolution kernels with size $1 \times S_1$, $1 \times S_2$, $1 \times S_3$, respectively. The Concat$(\cdot)$ operation concatenates the features obtained from three GTUs that are of different scales, resulting in a feature of dimension $3M - (S_1 + S_2 + S_3 - 3)$. After that, the dimension is changed to $(3M - (S_1 + S_2 + S_3 - 3))/W$ through the pooling layer with a window of size W. In this part, we can adjust the hyper-parameters S_1, S_2, S_3, W to ensure the dimension of the output to be equal to that of the input, i.e., $(3M - (S_1 + S_2 + S_3 - 3))/W = M$, so that they can be connected via skip connection. Finally, the output $Z^{(l)}_{out} \in \mathbb{R}^{N\times M\times C^{(l)}}$ is obtained through the ReLU activation function. The M-GTU uses GTU and residual structure to effectively reduce the gradient dispersion and retains the non-linearity. In addition, our M-GTU has an advantage in extracting both long-term and short-term features of the traffic data by using multi-scale causal convolution.

4. Experiments

4.1. Datasets

In order to evaluate the performance of DSTAGNN, we conducted comparative experiments on four real road traffic data sets from California, PEMS03, PEMS04, PEMS07 and PEMS08 which were released by (Song et al., 2020). The original traffic data is aggregated into 5-minute intervals, and normalized to zero mean. Besides, the spatial adjacency graph of each dataset is constructed based on the actual road network. Table 1 shows more details about the datasets.

Table 1: Description and statistics of datasets

Datasets	Nodes	Edges	TimeSteps	MissingRatio
PEMS03	358	547	26208	0.672%
PEMS04	307	340	16992	3.182%
PEMS07	883	866	28224	0.452%
PEMS08	170	295	17856	0.696%

4.2. Baseline methods

We compare our DSTAGNN with the following baselines: (1) **FC-LSTM** (Sutskever et al., 2014), which is a special RNN model; (2) **TCN** (Bai et al., 2018) which claimed to be effective in learning local and global temporal relations; (3) **DCRNN** (Li et al., 2017) that integrated graph convolution into a gated recurrent unit; (4) **STGCN** (Yu et al., 2017) that integrated graph convolution into a 1D convolution unit; (5) **ASTGCN** (Guo et al., 2019) that introduced a spatial-temporal attention mechanism in the model. For fair comparison, only recent components of the modeling periodicity (**ASTGCN(r)**) is used; (6) **STSGCN** (Song et al., 2020) that included local spatial-temporal subgraph modules; (7) **STFGNN** (Li & Zhu, 2021) that used a spatial-temporal fusion graph to complement the spatial correlation; (8) **STGODE** (Fang et al., 2021) that applied continous graph neural network to traffic prediction in multivariate time series forecasting; (9) **Z-GCNETs** (Chen et al., 2021) that introduced the concept of zigzag persistence into time-aware graph convolutional network for time series prediction; (10) **AGCRN** (Bai et al., 2020) that exploited learnable embedding of nodes in graph convolution.

4.3. Experiment settings

To be fair, we divide the data into training set, validation set, and test set in the same way as the baselines, i.e. 6:2:2 on PEMS datasets. We use the historical data of one hour to predict the traffic flow in the next one hour. All experiments are trained and tested on a Linux server (CPU: Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz, GPU: NVIDIA GeForce GTX 3090). Through experiments (referring to Appendix A for details), we set the following hyper-parameters: the number of terms of the Chebyshev polynomial (equal to the number of spatial attention heads) $K = 3$. The size of the M-GTU convolution kernel along the time dimension $\{S_1, S_2, S_3\}$ ={3, 5, 7}, and the window size W of the pooling layer is 2. The number of attention heads in the temporal attention module is 3, and $d_h = 32$ in the spatial-temporal attention module. All graph convolutional layers and time convolution layers use 32 convolution kernels. All the experiments use a stack of 4 ST blocks. In this work, the loss function we use is Huber loss (Huber, 1992), and the threshold parameter of the loss function is set to 1. We adopt the Adam optimizer to train our model, in which the number of epochs is 100, the learning rate is 0.0001, and the batch size is 32. The hyper-parameter of sparsity $P_{sp} = 0.01$. The mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean squared error (RMSE) are used to measure the performance of models. In addition, some baseline results (marked with * in Table 2) were derived from running their respective open source code, while the other baseline results are taken directly from the corresponding papers.

4.4. Experiment results and analysis

4.4.1. Results on the PEMS dataset

Table 2 shows the results of DSTAGNN and ten baseline methods. It can be seen that our DSTAGNN has achieved the best results in all indicators on the four data sets. The graph structure composed of our proposed spatial-temporal aware distance can help the model to capture the spatial dependency among the nodes, which demonstrates that our model can be applied in the absence of spatial prior information.

In addition, our proposed spatial-temporal attention mechanism can better capture the dynamic changes of data, with significantly improved prediction performance. We plot 5, 60 minutes ahead prediction values, respectively, and ground truth on a snapshot of the test data, as shown in Fig. 3 (refer to Appendix C for more visualization results), to demonstrate the difference between the proposed method and STGODE. From the part marked in blue dot-dashed boxes, it can be seen that DSTAGNN responds more quickly and accurately to dynamic changes in peak traffic than the baseline method. In the case of missing data, the proposed DSTAGNN recovers faster and maintains a higher accuracy due to our more accurate modelling of spatial-temporal dependency, as shown the part of (b) marked with brown dashed boxes.

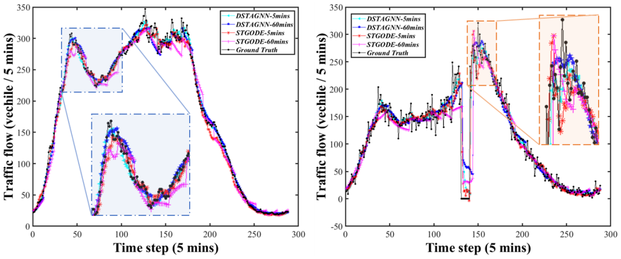
4.4.2. Ablation experiment

To verify the effectiveness of the individual components in DSTAGNN, we made the following variants of DSTAGNN: (1) RemSTA: completely removes the spatial-temporal attention mechanism; (2) RemM-A: removes the multi-head

Table 2: Performance comparison of our DSTAGNN and baseline models on PEMS datasets. Our DSTAGNN-G uses the pre-defined spatial adjacency graph in datasets as the graph structure of the model, and our DSTAGNN uses our new graph structure A_{STAG}, which is generated by binarizing the A_{STRG} derived from the traffic data in the training set.

Datasets	Metric	*FC-LSTM	*TCN	DCRNN	STGCN	ASTGCN(r)	STSGCN	AGCRN	STFGNN	STGODE	Z-GCNETs	DSTAGNN-G	**DSTAGNN**
PEMS03	MAE	21.33	19.31	18.18	17.49	17.69	17.48	*15.98	16.77	16.50	*16.64	15.61	**15.57**
	MAPE(%)	22.33	19.86	18.91	17.15	19.40	16.78	*15.23	16.30	16.69	*16.39	14.79	**14.68**
	RMSE	35.11	33.24	30.31	30.12	29.66	29.21	*28.25	28.34	27.84	*28.15	27.23	**27.21**
PEMS04	MAE	26.24	23.11	24.70	22.70	22.93	21.19	19.83	19.83	20.84	19.50	19.41	**19.30**
	MAPE(%)	19.30	15.48	17.12	14.59	16.56	13.90	12.97	13.02	13.77	12.78	12.84	**12.70**
	RMSE	40.49	37.25	38.12	35.55	35.22	33.65	32.26	31.88	32.82	31.61	31.63	**31.46**
PEMS07	MAE	29.96	32.68	25.30	25.38	28.05	24.26	*22.37	22.07	22.59	*21.77	21.67	**21.42**
	MAPE(%)	14.34	14.22	11.66	11.08	13.92	10.21	*9.12	9.21	10.14	*9.25	9.06	**9.01**
	RMSE	43.94	42.23	38.58	38.78	42.57	39.03	*36.55	35.80	37.54	*35.17	35.04	**34.51**
PEMS08	MAE	22.20	22.69	17.86	18.02	18.61	17.13	15.95	16.64	16.81	15.76	15.90	**15.67**
	MAPE(%)	15.02	14.04	11.45	11.40	13.08	10.96	10.09	10.60	10.62	10.01	9.97	**9.94**
	RMSE	33.06	35.79	27.83	27.83	28.16	26.80	25.22	26.22	25.97	25.11	25.24	**24.77**

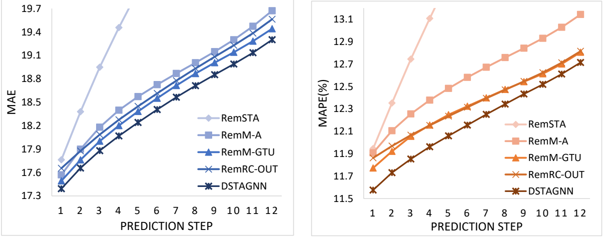
$*$ denotes re-implementation or re-training. ____ denotes the best indicator in baselines.



(a) Node #148 in PEMS04 (b) Node #22 in PEMS04

Figure 3: Comparison of prediction curves between STGODE and our DSTAGNN on a snapshot of the test data of PEMS04. Please zoom in the plots for a better view.

mechanism, and uses single-head attention to dynamically adjust the neighborhood of different scales of the graph convolution; (3) RemM-GTU: Remove multi-scale GTU and replace it with traditional convolution. (4) RemRC-OUT: The residual connection of the output of each ST block is removed. We performed ablation experiments on the above variants on the PEMS04 dataset. Fig. 4 shows the measurements of MAE and MAPE. It can be seen that the performance of our DSTAGNN is better than other variants, which confirms the effectiveness of each component in our model.

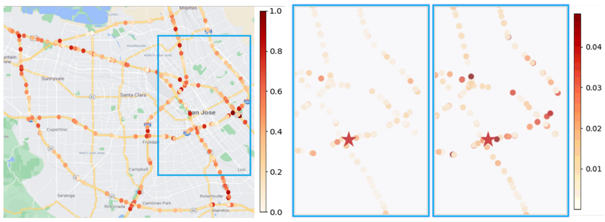


(a) MAE per prediction step (b) MAPE per prediction step

Figure 4: Ablation experiment of module effectiveness.

4.4.3. Visualization of spatial-temporal dependency

To enhance the interpretability of our proposed model and show the detail of our proposed attention module, we have visualized the spatial-temporal dependencies acquired by our model. It can be seen from Fig. 5 (a) that the proposed model has ability in identifying complex traffic conditions such as road network intersections. In addition, Fig. 5 (b) shows that, for a specific prediction point, the model obtains dynamic spatial dependency information from different scales. In summary, our model not only has achieved highly promising performance in traffic flow prediction, but also extracts complex information within the road network.



(a) (b)

Figure 5: Spatial-temporal dependency obtained by DSTAGNN on the PEMS-BAY dataset. (a) is the global self-attention from the 1st attention head. (b) are the dependencies between the target node (red star) and its surrounding nodes obtained by the 2nd and 3rd attention heads. The zone of (b) corresponds to the blue box area in (a).

5. Conclusion

We have presented a novel deep learning framework DSTAGNN for traffic flow prediction. Our DSTAGNN has utilized spatial-temporal aware distance (STAD) derived from historic traffic data without relying on a predefined static adjacency matrix. With this method, the representation of the internal dynamic association attributes between nodes of the road network can be enhanced effectively. In addition, graph convolution operated on the Spatial-Temporal Aware Graph (STAG) generated from STAD can reduce the dependency on prior information of the road network. Combined with our spatial-temporal attention module and multi-receptive field gated convolution, our DSTAGNN further boosts the awareness of dynamic spatial-temporal dependency in time series data. Therefore, our DSTAGNN achieves new state-of-the-art performance on the four public data sets for traffic flow prediction, compared to several recent baseline methods.

References

Atwood, J. and Towsley, D. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1993–2001, 2016.

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, 33, 2020.

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

Berndt, D. J. and Clifford, J. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pp. 359–370. Seattle, WA, USA:, 1994.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Chen, X., Zhang, Y., Du, L., Fang, Z., Ren, Y., Bian, K., and Xie, K. Tssrgcn: Temporal spectral spatial retrieval graph convolutional network for traffic flow forecasting. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 954–959. IEEE, 2020.

Chen, Y., Segovia, I., and Gel, Y. R. Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1684–1694. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/chen21o.html.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In *International Conference on Machine Learning*, pp. 933–941. PMLR, 2017.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 29:3844–3852, 2016.

Devroye, L., Mehrabian, A., and Reddad, T. The total variation distance between high-dimensional gaussians. *arXiv preprint arXiv:1810.08693*, 2018.

Fang, Z., Long, Q., Song, G., and Xie, K. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 364–373, 2021.

Goldberger, J., Gordon, S., Greenspan, H., et al. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *ICCV*, volume 3, pp. 487–493, 2003.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 922–929, 2019.

He, R., Ravula, A., Kanagal, B., and Ainslie, J. Realformer: Transformer likes residual attention. *arXiv preprint arXiv:2012.11747*, 2020.

Huber, P. J. Robust estimation of a location parameter. In *Breakthroughs in Statistics*, pp. 492–518. Springer, 1992.

Kailath, T. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. From word embeddings to document distances. In *International Conference on Machine Learning*, pp. 957–966. PMLR, 2015.

Li, M. and Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4189–4196, 2021.

Li, Y. and Shahabi, C. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *Sigspatial Special*, 10(1):3–9, 2018.

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

Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Panaretos, V. M. and Zemel, Y. Statistical aspects of wasserstein distances. *Annual Review of Statistics and Its Application*, 6:405–431, 2019.

Park, C., Lee, C., Bahng, H., won, T., Kim, K., Jin, S., Ko, S., and Choo, J. Stgrat: A spatio-temporal graph attention network for traffic forecasting. *arXiv preprint arXiv:1911.13181v1*, 2019.

Simonovsky, M. and Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3693–3702, 2017.

Singh, A., Yadav, A., and Rana, A. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10), 2013.

Song, C., Lin, Y., Guo, S., and Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 914–921, 2020.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., and Yu, J. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference*, pp. 1082–1092, 2020.

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

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., and Li, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., and Yin, B. A comprehensive survey on traffic prediction. *arXiv preprint arXiv:2004.08555*, 2020.

Yu, B., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

Zhang, J., Zheng, Y., Qi, D., Li, R., and Yi, X. Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4, 2016.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2019.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

A. Appendix: Best Hyper-parameters

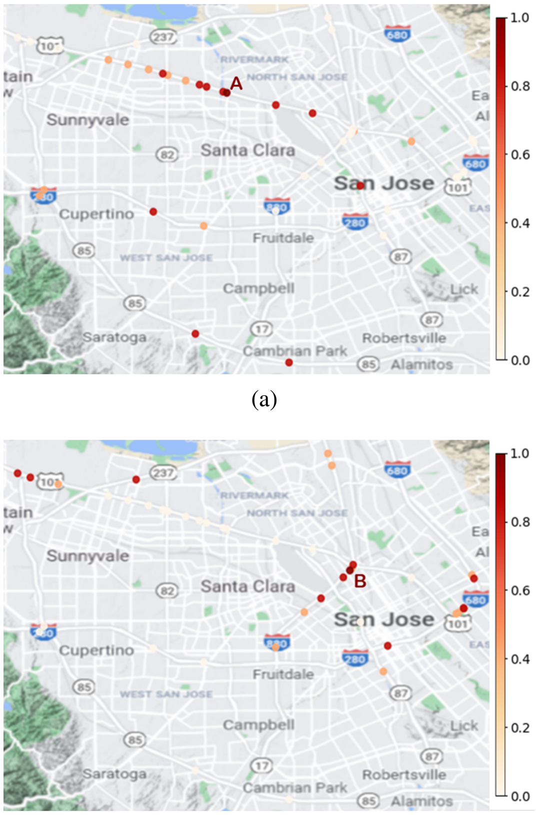
To compare the settings of various hyper-parameters (i.e., the number of ST blocks, the number of temporal attention heads, the number of spatial attention heads, the kernel size along the temporal dimension in the M-GTU model and the sparsity of $\boldsymbol{A}_{STAG}$) in our DSTAGNN. We conducted ablation experiments on PEMS04 and Table 3 shows results in terms of MAE, MAPE and RMSE. From Table 3, we can observe the following. The stacking of ST blocks can increase the performance of our model, but can degrade the performance for an increased number of layers. The setting of the number of attention heads shows a similar pattern. For the size of the convolution kernel used in the temporal convolution module, the combination of receptive fields allows the model to exploit both the short-term and the long-term temporal information, thereby improving the prediction performance.

Table 3: Ablation experiment for hyper-parameter setting on PEMS04. STn is the number of stacked layers of ST blocks. TA-h and SA-h correspond to the number of heads in temporal attention and spatial attention module, respectively. KS is the kernel size along the temporal dimension in the M-GTU model. For each given node P_{gn} of the road network, we set the $N_r = P_{sp} \times N$ nodes that are most relevant to P_{gn} as nonzero elements of this row (corresponding to P_{gn}) in $\boldsymbol{A}_{STAG}$, where N is the number of total nodes , P_{sp} is a hyper-parameter to control the ratio of N_r to N. The default configuration of DSTAGNN we use in this paper is [4, 3, 3, {3, 5, 7}, 1%].

[STn, TA-h, SA-h, KS, P_{sp}]	MAE	MAPE(%)	RMSE
[3, 3, 3, {3, 5, 7}, 1%]	19.33	12.82	31.54
[4, 3, 3, {3, 5, 7}, 1%]	19.30	**12.70**	31.46
[5, 3, 3, {3, 5, 7}, 1%]	19.34	12.88	31.43
[4, 4, 3, {3, 5, 7}, 1%]	19.28	12.74	31.42
[4, 5, 3, {3, 5, 7}, 1%]	19.42	12.89	31.73
[4, 3, 4, {3, 5, 7}, 1%]	19.47	12.85	31.72
[4, 3, 3, {1, 5, 9}, 1%]	19.32	12.81	31.35
[4, 3, 3, {2, 5, 8}, 1%]	**19.22**	12.79	**31.19**
[4, 3, 3, {4, 5, 6}, 1%]	19.32	12.72	31.35
[4, 3, 3, {3, 5, 7}, 5%]	19.42	12.90	31.72
[4, 3, 3, {3, 5, 7}, 0.5%]	19.37	12.85	31.37

B. Spatial-Temporal Aware Graph

This section will further analyze the Spatial-temporal Aware Graph (STAG) we proposed. We visualized the STAG information generated on the PEMS-BAY dataset (325 record points in total), and found the 20 recording points in the STAG that have the strongest correlation with a given recording point. From Fig. 6, the STAG of two recording points (i.e., A denotes node #247, B denotes node #69) in the PEMS-BAY dataset demonstrates that the spatial correlations based on the Spatial-Temporal Aware Distance (STAD) capture the node relation in a wide spatial range, not necessarily limited to its neighborhood. In Fig. 6(a), the strongest correlation comes from the nearest nodes (e.g.,the recording points that are on the same road and close to recording point A). However, the degree of the spatial correlation is not entirely proportional to their spatial connectivity distance, and there also are many recording points which have higher spatial correlation than those that have a smaller distance to A. In addition, our STAG can capture much more latent spatial correlations among the nodes. For example, the recording point B (in Fig. 6(b)) at an intersection shows a higher spatial correlation with the recording points around other intersections, due to their similar traffic patterns to the given recording point B.



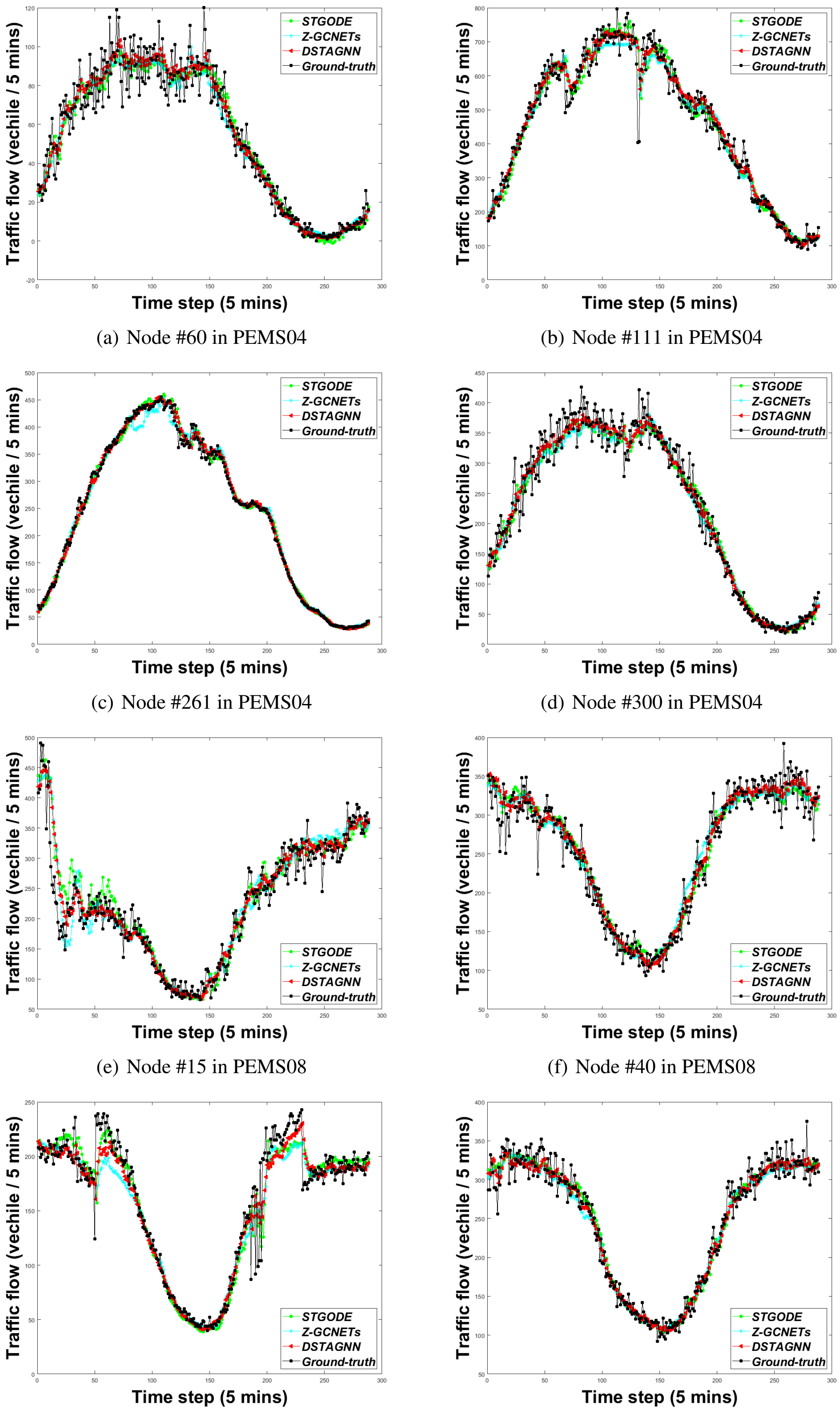
(a)

(b)

Figure 6: The spatial correlation included in the STAG on the PEMS-BAY dataset. (a) shows the recording points that have a strong correlation with the recording point A. (b) shows the recording points that have a strong correlation with recording point B.

C. Traffic Forecasting Visualization

In this section, we have supplemented the comparison of some prediction results by our method, STGODE (Fang et al., 2021) and Z-GCNETs (Chen et al., 2021) in Fig. 7.



(a) Node #60 in PEMS04

(b) Node #111 in PEMS04

(c) Node #261 in PEMS04

(d) Node #300 in PEMS04

(e) Node #15 in PEMS08

(f) Node #40 in PEMS08

(g) Node #85 in PEMS08

(h) Node #130 in PEMS08

Figure 7: Traffic forecasting visualization. Please zoom in the plots for a better view.