Hierarchical Spatio–Temporal Graph Convolutional Networks and Transformer Network for Traffic Flow Forecasting

Guangyu Huo, Yong Zhang, *Member, IEEE*, Boyue Wang, Junbin Gao, Yongli Hu, *Member, IEEE*, and Baocai Yin, *Member, IEEE*

***Abstract*— Graph convolutional networks (GCN) have been applied in the traffic flow forecasting tasks with the graph capability in describing the irregular topology structures of road networks. However, GCN based traffic flow forecasting methods often fail to simultaneously capture the short-term and long-term temporal relations carried by the traffic flow data, and also suffer the over-smoothing problem. To overcome the problems, we propose a hierarchical traffic flow forecasting network by merging newly designed the long-term temporal Transformer network (LTT) and the spatio-temporal graph convolutional networks (STGC). Specifically, LTT aims to learn the long-term temporal relations among the traffic flow data, while the STGC module aims to capture the short-term temporal relations and spatial relations among the traffic flow data, respectively, via cascading between the one-dimensional convolution and the graph convolution. In addition, an attention fusion mechanism is proposed to combine the long-term with the short-term temporal relations as the input of the graph convolution layer in STGC, in order to mitigate the over-smoothing problem of GCN. Experimental results on three public traffic flow datasets prove the effectiveness and robustness of the proposed method.**

***Index Terms*— Graph convolutional networks, traffic data forecasting, transformer.**

I. Introduction

TRAFFIC flow forecasting is a fundamental topic in the intelligent transportation system applications. Recently, continued economic development has led to the rapid growth of car ownership. As a result, traffic congestion has become a normal state due to the slow urban infrastructure development. Therefore, an accurate traffic flow forecasting model is crucial to mitigate the traffic congestion problem. Specifically, for a traffic management department, better or improved traffic flow forecasting can help them to formulate the most responsible policies, reducing the traffic congestion and improving the traffic efficiency. For individual travelers, the accurate traffic flow forecasting can help them make reasonable travel plans, saving their travelling time.

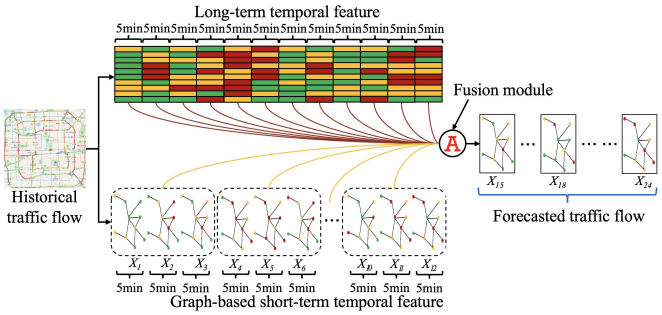


Fig. 1. A brief illustration of the proposed model: The upper layer extracts the long-term temporal relations among the traffic flow data. The lower layer captures the short-term temporal relation and spatial relations of the traffic flow data. The fusion module can produce more accurate traffic flow forecasting.

Early traffic flow forecasting methods mainly focus on analyzing time series, which learn the changing laws and trends hidden in the historical data. Representative methods include auto-regressive integrated moving average (ARIMA) [1], vector auto-regression (VAR) [2], and Kalman filter model [3]. Although these methods acquire good performance in the short-term traffic flow forecasting tasks, they neglect the important spatial information that existed in the traffic data, i.e., the topological graph of the road network.

To solve this problem, some researchers use the grid topology to introduce the spatial relations on the basis of the temporal relations between the traffic data [4], [5], [6], [7], which still fail to consider the irregular topology structure of natural road networks. The graph analytics is a natural choice for describing such irregular topology of the road networks. Therefore, we regard the traffic flow data as the graph signals that change over time.

Graph convolutional networks (GCN) [8], [9], [10] have attracted great attention due to its excellent feature extraction

Manuscript received 20 October 2021; revised 13 July 2022 and 24 October 2022; accepted 29 December 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0111902; in part by the National Natural Science Foundation of China under Grant 62272015, Grant 62072015, Grant U19B2039, and Grant U21B2038; and in part by the Beijing Natural Science Foundation under Grant 4222021.The Associate Editor for this article was A. Nunez. *(Corresponding author: Boyue Wang.)*

Guangyu Huo, Yong Zhang, Boyue Wang, Yongli Hu, and Baocai Yin are with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing Artificial Intelligence Institute, Beijing University of Technology, Beijing 100124, China (e-mail: gyhuo@emails.bjut.edu.cn; zhangyong2010@bjut.edu.cn; wby@bjut.edu.cn; huyongli@bjut.edu.cn; ybc@bjut.edu.cn).

Junbin Gao is with the Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Camperdown, NSW 2006, Australia (e-mail: junbin.gao@sydney.edu.au).

Digital Object Identifier 10.1109/TITS.2023.3234512

ability on the graph-structure data. GCN uses the spatial relations among the data with the help of the corresponding graph, which is greatly suitable for traffic flow forecasting tasks. In recent traffic flow forecasting studies, the graph convolutional networks and the temporal deep neural networks are usually integrated to simultaneously obtain the temporal and spatial relations among the traffic flow data. Spatio-temporal graph convolutional networks (STGCN) [11] and diffusion convolutional recurrent neural network (DCRNN) [12] are the first two GCN based traffic flow forecasting frameworks. Subsequent works including T-GCN [13], ASTGCN [14] and Graph WaveNet (GWN) [15] further mine the temporal and spatial relations from different aspects.

Except for the spatial relations, the above methods [11], [12], [13], [14], [15] extract the temporal relations among the traffic flow data through three types of temporal deep neural networks, namely RNN, one-dimensional convolution and causal diffusion convolution. However, the limited parallel computing capability of RNN restricts its ability to learn the long sequences [16]. One-dimensional convolution has a small receptive field and only learns the short-term temporal features [17]. Although the causal diffusion convolution can expand the receptive field, the time complexity increases by a square fold as the number of layers increases [15].

The traffic flow patterns in different time periods are discrepant. During the morning and evening rush hours, the road's vehicle carrying capacity often reach its limit and a slight congestion can result in a prominent impact for the traffic flow of nearby roads. In such condition, the short-term temporal relations are more important for the forecasting results. However, during the off-peak period, the road status is stable and a congestion has the relative small impact for the traffic of nearby roads. So, the traffic flow's long-term temporal relations can better reflect the changes of the entire transportation system. Base on the above-mentioned characteristics of traffic flow, how to comprehensively utilize the long-term and short-term temporal relations is the key to improving the traffic flow forecasting.

To better forecast the traffic flow, a novel hierarchical traffic flow forecasting model via spatio-temporal graph convolutional networks and Transformer network is proposed. Specifically, to capture the long-term temporal relations among the traffic flow data, we design a long-term temporal Transformer network. To capture the short-term temporal and spatial relations among the traffic flow data, the proposed STGC network consists of one-dimensional temporal convolution and GCN. In order to integrate long-term and short-term temporal relations, we propose a cross-attention mechanism based on the long-short-term temporal information fusion module, and transfer the fusion representation into STGC, which also alleviates the inherent over-smoothing problem [18] of GCN.

The main contributions of this paper are summarized as follows,

- We propose a novel hierarchical traffic flow forecasting model constructed by two paralleled networks, i.e., the spatio-temporal graph convolutional networks (STGC) and the long-term temporal Transformer network (LTT);
- **The long-term temporal Transformer network** encodes the temporal position information by the Transformer to extract the long-term temporal relations among the traffic flow data;
- **The spatio-temporal graph convolutional networks** captures the spatial dependencies among the traffic flow data at different short-term temporal granularity levels, where multiple one-dimensional convolutional kernels are exploited to mine the short-term temporal relations among data;
- **The long-short temporal information fusion module** designs an attention-based fusion module to adaptively integrate the above long-term and short-term temporal relations according to their importance, and transfers the fused representation to STGC, with the capability of mitigating the over-smoothing problem of GCN.

The rest of the paper is organized as follows. In Section II, we briefly review the traditional traffic flow forecasting methods, GCN-based traffic flow forecasting methods and the Transformer networks, respectively. In Section III, we present the hierarchical traffic flow forecasting model by using spatio-temporal graph convolutional networks and Transformer and its three main modules. In Section IV, the proposed method is evaluated on three public datasets for the traffic flow forecasting tasks. Finally, the conclusion is discussed in Section V.

II. Related Work

In this section, we review the necessary knowledge relating to *Traditional Traffic Flow Forecasting*, *Graph Convolutional Networks based Traffic Flow Forecasting* and *Transformer Network*.

A. Traditional Traffic Flow Forecasting

Traffic flow forecasting can be regarded as a mapping from the observed traffic data to the future traffic conditions. The average of historical traffic flow data is the easiest forecasting method [19]. However, the changes in the traffic flow data are complex and do not have the simple linear relations. To solve the problems, some machine learning methods are applied in the field of the traffic flow forecasting and receive good performance, such as the vector auto-regression (VAR) [2], the support vector regression (SVR) [20], the auto-regressive integrated moving average (ARIMA) [1], the ordered low-rank representation completion (OLRRC) [21], and the temporal and adaptive spatial constrained low rank (TAS-LR) [7].

Although some linear models have the better interpretability and simple operation, they assume the traffic data have the special structure or distribution (i.e. the traffic network is the structure of a graph, not the precise arrangement of pixels like an image.). Fortunately, deep learning can capture the non-linearity relations hidden in the traffic data, attracting much attention in the intelligent transportation field.

LSTM [6] and GRU [22] are initially used to forecast the traffic flow. But they only consider the temporal relations and ignore the spatial relations among the traffic flow data. Some researchers provide the improved versions. The fully

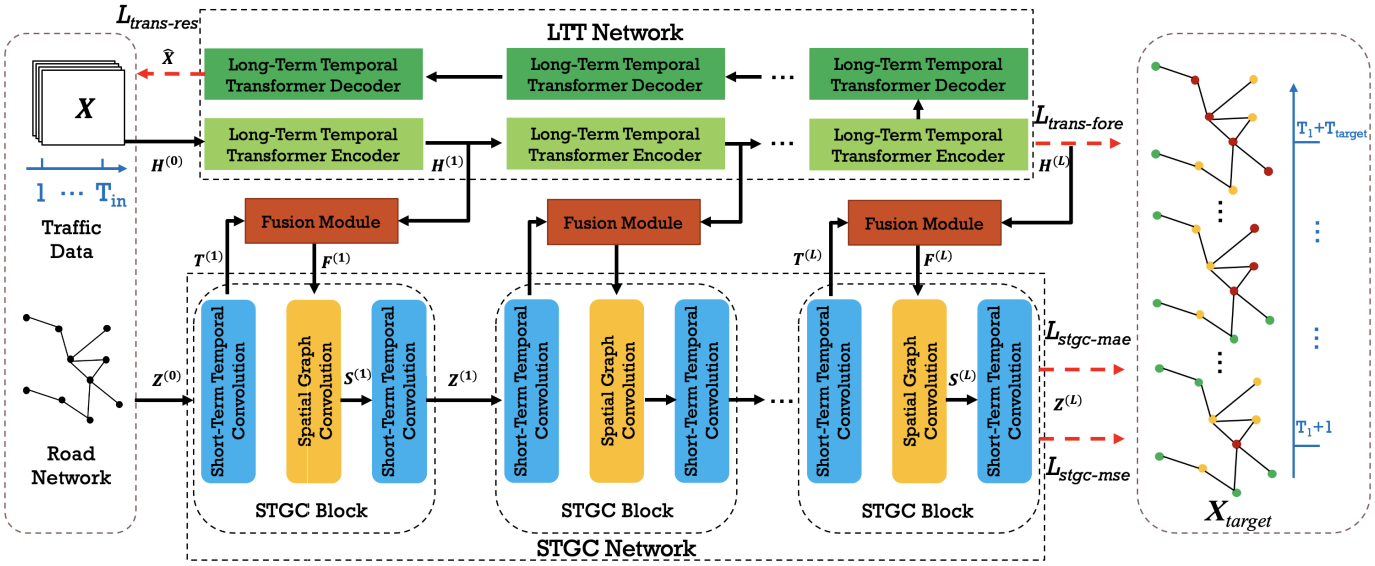


Fig. 2. The conceptual framework of the proposed hierarchical traffic flow forecasting model, which includes three modules: long-term temporal Transformer network, spatio-temporal graph convolutional networks, and long-short temporal information fusion module. $\mathbf{X}$ is the original traffic flow data, $\hat{\mathbf{X}}$ is the reconstructed traffic flow data product by LTT. $\mathbf{Z}^{(L)}$ is the forecasted traffic flow by STGC module, $\mathbf{H}^{(L)}$ is the output of the last layer of the LTT encoders. $\mathcal{L}_{trans-res}$ is the traffic flow reconstruction loss of LTT and $\mathcal{L}_{trans-fore}$ is the forecasting loss. $\mathcal{L}_{stgc-mae}$ and $\mathcal{L}_{stgc-mse}$ are the forecasting loss of STGC, respectively.

connected LSTM (FC-LSTM) [6] models both the spatial and temporal relations. Shi et al. [23] proposed Convolutional LSTM (ConvLSTM) that replaces the fully connected layer in FC-LSTM by a convolutional layer. Guo et al. [24] proposed spatial and temporal 3DNet (ST-3DNet) that introduces the 3D convolution to capture the spatial and temporal relations among the traffic data.

However, these models formulate the traffic data into the matrix form and cannot adopt the irregular graph structure of the road network.

B. Graph Convolutional Networks Based Traffic Flow Forecasting

The road network topology constructed in the form of an irregular graph can be expressed as $G = (V, E, A)$, where V is the set of road segments, E means the set of edges, and $A \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix, N refers to the number of road segments. Graph convolutional networks (GCN) [25] can simultaneously process the data and its graph. GCN converts the graph into a Laplacian matrix and performs the convolution operations as follows,

$$f(\mathbf{Z}^{(l)}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(l-1)}\mathbf{W}^{(l)}), \tag{1}$$

where $\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}$ is the symmetric normalized Laplacian matrix, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\hat{\mathbf{D}} = \sum_j \hat{\mathbf{A}}_{ij}$. $\mathbf{Z}^{(l)}$ and $\mathbf{W}^{(l)}$ represent the data matrix and the feature transformation matrix (i.e. one of network weights) in the l-th layer, respectively. $\sigma(\cdot)$ denotes an activation function.

According to the characteristics of traffic flow data, researchers made several heuristic improvements to GCN as follows,

(1) *Spatio-Temporal* GCN (STGCN) [11] is the first work that applies GCN on the traffic data forecasting tasks, which cascades the temporal CNN and spatial GCN modules to capture the spatio-temporal relations among the traffic data.

(2) *Diffusion Convolutional Recurrent Neural Network (DCRNN)* [12] executes the bidirectional random walk on the road network graph to capture the spatial relations and uses the gated recurrent unit to obtain the temporal relations as DCRNN in TABLE I,

where $\mathbf{P}_{in}$ and $\mathbf{P}_{out}$ indicate the transition matrices of the diffusion process and the opposite one, respectively. $\mathbf{W}_{in}$ and $\mathbf{W}_{out}$ are the weight matrices of the graph convolution.

(3) *Graph WaveNet (GWN)* [15] extracts the spatial relations by constructing a self-adaptive adjacency matrix and gets the temporal relations by the stacked dilated one-dimensional convolution. The structure of the self-adaptive adjacency matrix is,

$$\tilde{\mathbf{A}}_{adp} = \text{SoftMax}\left(\text{ReLU}\left(\mathbf{E}_s\mathbf{E}_t\right)\right), \tag{2}$$

where $\mathbf{E}_s$ and $\mathbf{E}_t$ denote the representations of source and target nodes, respectively. By multiplying $\mathbf{E}_s$ and $\mathbf{E}_t$, we can obtain the corresponding spatial dependence weight. GWN adopts the self-adaptive adjacency matrix and proposes a new graph convolutional layer as Graph WaveNet in TABLE I, where $\mathbf{W}_{adp}$ is the adaptive graph convolution kernel.

(4) *Spatial-Temporal Synchronous Graph Convolutional Networks (STSGCN)* [26] enhances the capture ability of spatial-temporal relations by constructing a local spatial-temporal graph.

C. Transformer Based Traffic Flow Forecasting

Vaswani et al. [27] proposed the Transformer network, which replaces the LSTM module in the seq2seq model [28] by a full self-attention structure. Due to the powerful representation ability, Transformer obtains the outstanding performance

TABLE I
SEVERAL IMPORTANT VARIABLES USED IN THIS PAPER

Method	Formula
STGCN [11]	$f(\mathbf{Z}^{(l)}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(l-1)}\mathbf{W}^{(l)})$.
DCRNN [12]	$f(\mathbf{Z}^{(l)}, \mathbf{A}) = \sigma(\sum_{k=0}^{K-1}\left(\mathbf{W}_{k,in}\mathbf{P}_{in}^{k} + \mathbf{W}_{k,out}\mathbf{P}_{out}^{k}\right)\mathbf{Z}^{(l-1)})$.
Graph Wavenet [15]	$f(\mathbf{Z}^{(l)}, \mathbf{A}) = \sigma(\sum_{k=0}^{K-1}\left(\mathbf{W}_{k,in}\mathbf{P}_{in}^{k} + \mathbf{W}_{k,out}\mathbf{P}_{out}^{k} + \mathbf{W}_{k,adp}\tilde{\mathbf{A}}_{adp}^{k}\right)\mathbf{Z}^{(l-1)})$,

in the machine translation tasks, traffic flow forecasting tasks and attracts much attention from other fields.

In traffic flow forecasting tasks, Cai et al. [16] directly applied Google's Transformer machine translation framework to forecast the traffic flow, which can capture the continuity and periodicity of time series. Park et al. [29] proposed the spatio-temporal graph attention, which obtains the graph structure information (e.g., distances between roads) through the spatial attention and dynamically adjusts the spatial correlations according to road states. Temporal attention is responsible for capturing the changes in traffic flow. Wang et al. [30] proposed the learnable location attention mechanism to efficiently aggregate the information from adjacent roads. Xue et al. [31] proposed the Transformer-based urban flow prediction architecture TERMCast, which simultaneously extracts the tightness, period and trend components from the traffic flow series. Li et al. [32] proposed the LogSparse Transformer with the low spatial complexity, which improves the fine-grained performance under constrained memory budgets and the forecast accuracy with strong long-term dependencies. Zhou et al. [33] improved the Transformer model to reduce its time complexity and memory usage, greatly improving the inference speed for long sequence predictions through the self-attention refinement and generative decoding. Padhi et al. [34] proposed the tabular time series based neural network, which consists of two modules: a BERT-based representation learning module, and a GPT-like module that can be used to generate realistic synthetic tabular sequences. Padhi et al. [35] proposed the GAN-based time series prediction model that employs a Sparse Transformer as a generator to learn a sparse attention map for time series prediction. Daiya et al. [36] proposed a multimodal time series forecasting model that uses dilated causal convolutions and Transformer blocks to extract multimodal features.

The Transformer has powerful capabilities in handling the sequence data. We attempt to leverage Transformers to focus on extracting long-term relationship of traffic flow, thereby improving the performance of long-term traffic flow forecasting.

III. HIERARCHICAL SPATIO-TEMPORAL GRAPH CONVOLUTIONAL NETWORKS AND TRANSFORMER NETWORK

A. Motivation

As we have known, the traffic flow data possess both the complex temporal and spatial relations. How to effectively capture and fuse these two kinds of relations is a critical problem for the traffic flow forecasting tasks. Therefore, a novel hierarchical deep neural network model for the traffic flow forecasting tasks is proposed with the following motivations.

- To learn the hidden long-term temporal relations, we use the Transformer network [27] to construct the long-term temporal Transformer network (LTT).
- To make full use of the spatial relations, we build the spatio-temporal graph convolutional networks (STGC), which also learns the short-term relations from the traffic flow data.
- To integrate the above long-term temporal relations, short-term temporal relations and the spatial relations, we design the attention based fusion module to connect LTT and STGC layer-by-layer, which can mitigate the over-smoothing issue of STGC.

B. Overall Network Architecture

The overall network architecture is shown in Fig. 2, consisting of the following three main modules:

- *Long-Term Temporal Transformer network (LTT)* captures the long-term temporal dependencies from the raw traffic flow data, i.e., $\mathbf{H}^{(l)} = \text{LTT}(\mathbf{H}^{(l-1)})$;
- Each *Spatio-Temporal Graph Convolutional module (STGC)* cascades a one-dimensional convolution neural network ($\mathbf{T}^{(l)} = \text{STC}(\mathbf{Z}^{(l-1)})$), a graph convolution network ($\mathbf{S}^{(l)} = \text{GCN}(\mathbf{F}^{(l)}, \mathbf{A})$) followed by another one-dimension convolutional neural network ($\mathbf{Z}^{(l)} = \text{STC}(\mathbf{S}^{(l)})$), which simultaneously captures the short-term temporal dependencies and spatial dependencies;
- *Long-Short Temporal Information Fusion module* fuses the above long-term and short-term temporal dependencies, i.e., $\mathbf{F}^{(l)} = \text{Fusion}(\mathbf{H}^{(l)}, \mathbf{T}^{(l)})$, then we send it to the graph convolutional networks in STGC layer-by-layer.

C. Long-Term Temporal Transformer Network

This network encodes the temporal position information to extract the long-term temporal relations from the raw traffic flow data. As shown in Fig. 3, the designed long-term temporal Transformer network has two main components: temporal position embedding and long-term temporal information extraction.

1) Temporal Position Embedding: Since the traffic flow data is the classic time-series data, analyzing the temporal position relations among the data can boost the traffic flow forecasting performance. For example, when we forecast the traffic flow on a certain stretch of road at 7 : 30, the traffic flow data at 7 : 00 is more important than the traffic flow data at 5 : 00. The traditional attention mechanism is used to capture the relations

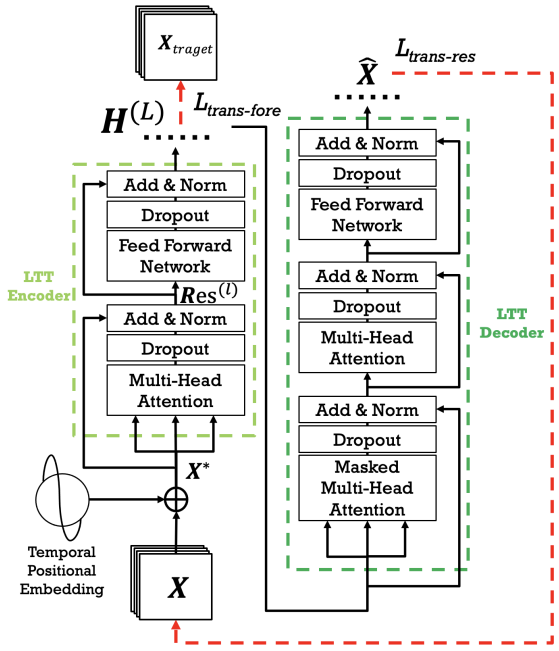


Fig. 3. The conceptual framework of the long-term temporal transformer.

among the data [37], while it neglects the order information in such time-series data.

To solve this drawback, Transformer introduces the sine and cosine position embedding to record the position information of the time-series data [27], and the traditional position embedding is defined as,

$$\begin{cases} \mathrm{PE}(\mathrm{pos}, 2\,i) = \sin\left(\dfrac{\mathrm{pos}}{10000^{2i/d}}\right) \\ \mathrm{PE}(\mathrm{pos}, 2\,i+1) = \cos\left(\dfrac{\mathrm{pos}}{10000^{2i/d}}\right), \end{cases} \tag{3}$$

where pos denotes the position in the sequence, i represents the length of the sequence, and d is the dimension of the sequence. For each element in the sequence data, we calculate different frequencies by the sin and cos functions to generate an embedding.

It is worth noting that the change of traffic data has obvious periodic laws. However, the above traditional position embedding method cannot express the periodic information hidden in the traffic data due to a long of period $10000^{2i/d}$. Therefore, it is necessary to set a proper period for the traffic data position embedding.

We propose an upgraded version of temporal position embedding with a given period to better describe the period characteristics of the traffic flow data as follows,

$$\mathrm{PE}(K) = \sin(\frac{2\pi K}{\mathrm{period}}), \tag{4}$$

where K represents the time step of one element in the sequence, $period$ represents the pre-set period.

We concatenate the above position embedding to the sequence data, so that the proposed model can obtain the relative position information of the data. Specifically, the position embedding $\mathbf{PE} \in \mathbb{R}^{N\times T_{in}\times D_p}$ and the corresponding traffic flow data $\mathbf{X} \in \mathbb{R}^{N\times T_{in}\times D_r}$ are concatenated as the input data $\mathbf{X}^* \in \mathbb{R}^{N\times T_{in}\times (D_p+D_r)}$,

$$\mathbf{X}^* = \mathrm{Concat}(\mathbf{X}, \mathbf{PE}), \tag{5}$$

where T_{in} represents the length of the input traffic flow data, D_p represents the dimension of $\mathbf{PE}$, and D_r represents the dimension of $\mathbf{X}$. The periodic attribute can better describe the periodic pattern of traffic flow data. Empirically, a smaller period can produce a position embedding with a higher degree of discrimination.

2) Long-Term Temporal Information Extraction: The long-term traffic forecasting for more than 30 minutes is still one fundamental and difficult problem in the current intelligent transportation field. Such a long-term temporal dependency brings in great difficulties for traditional CNN or RNN based temporal processing modules. To solve this problem, we propose a LTT network to extract the long-term temporal relations from the entire traffic flow sequence.

The LTT network consists of 3 Transformer encoders and the corresponding 3 Transformer decoders, and its architecture is shown in Fig. 2. Each encoder is constructed by four main components: *multi-head attention, fully-connected feed-forward network, dropout [38] and layer normalisation [39].* Here, We express the l-th Transformer encoder in an abstract manner as follows,

$$\begin{aligned} \mathbf{H}^{(l)} &= \mathrm{LTT}(\mathbf{H}^{(l-1)}) \\ &= \mathrm{LN}(\mathrm{dropout}(\mathrm{FeedForword}(\mathbf{Res}^{(l)})) + \mathbf{Res}^{(l)}), \end{aligned} \tag{6}$$

where $\mathbf{H}^{(l)}$ denotes the l-th LTT encoder's output. $\mathrm{LN}(\cdot)$ means the layer normalization operation, $\mathrm{dropout}(\cdot)$ represents the dropout operation. $\mathrm{FeedForword}(\cdot)$ is the feed-forward network and $\mathbf{Res}^{(l)}$ indicates the intermediate feature of the l-th LTT encoder, which can be expanded into,

$$\begin{aligned} \mathrm{FeedForword}(\mathbf{Res}^{(l)}) &= \max(0, \mathbf{Res}^{(l)}\mathbf{U}_1 + \mathbf{b}_1)\mathbf{U}_2 + \mathbf{b}_2 \\ \mathbf{Res}^{(l)} &= \mathrm{LN}(\mathrm{DP}(\mathrm{MultiHead}(\mathbf{H}^{(l-1)})) \\ &\quad + \mathbf{H}^{(l-1)}), \end{aligned} \tag{7}$$

where $\mathrm{FeedForword}(\cdot)$ contains two fully-connected layers and one ReLU activation, so $\mathbf{U}_1$, $\mathbf{b}_1$, $\mathbf{U}_2$ and $\mathbf{b}_2$ are the corresponding learnable parameters. This feed-forward network mainly provides the nonlinear transformation.

The multi-head attention $\mathrm{MultiHead}(\cdot)$ uses M different linear transformations to analyze the previous layer feature $\mathbf{H}^{(l-1)}$ from different aspects and $\mathbf{H}^{(0)} = \mathbf{X}^*$, which is the core part of LTT. This process can be written as,

$$\begin{aligned} &\mathrm{MultiHead}(\mathbf{H}^{(l-1)}) \\ &\quad = \mathrm{Concat}\left(\mathbf{head}_1, \ldots, \mathbf{head}_M\right)\mathbf{W}^O \\ &\mathbf{head}_m \\ &\quad = \mathrm{softmax}\left(\frac{\mathbf{H}^{(l-1)}\mathbf{W}_m^Q(\mathbf{H}^{(l-1)}\mathbf{W}_m^K)^T}{\sqrt{(D_p+D_r)}}\right)\mathbf{H}^{(l-1)}\mathbf{W}_m^V, \end{aligned} \tag{8}$$

where $\mathbf{W}^Q \in \mathbb{R}^{(D_p+D_r)\times q}$, $\mathbf{W}^K \in \mathbb{R}^{(D_p+D_r)\times q}$, $\mathbf{W}^V \in \mathbb{R}^{(D_p+D_r)\times q}$ and $\mathbf{W}^O \in \mathbb{R}^{(q*M)\times D_r}$ are the commonly-used linear mapping matrices.

The decoder has the similar structure with the encoder. We add a subsequent mask to the first attention block, and

only the traffic flow information in the first several time steps is used to forecast the traffic flow.

After 3 Transformer encoders, the forecasted traffic flow is obtained; therefore, we naturally minimize the difference between the forecasted traffic flow and its corresponding ground truth, i.e., the long-term temporal forecasting loss $\mathcal{L}_{\text{trans-fore}}$. To preserve more temporal information, we reconstruct the original traffic flow by 3 decoders and define its reconstructed loss $\mathcal{L}_{\text{trans-res}}$. So, we have,

$$\begin{aligned} \mathcal{L}_{\text{trans-fore}} &= \|\mathbf{H}^{(L)} - \mathbf{X}_{\text{target}}\|_1 \\ \mathcal{L}_{\text{trans-res}} &= \|\hat{\mathbf{X}} - \mathbf{X}\|_1, \end{aligned} \tag{9}$$

where $\mathbf{X}_{\text{target}} = \{\mathbf{X}_{(T_{in}+1)}, \cdots, \mathbf{X}_{(T_{in}+T_{target})}\} \in \mathbb{R}^{N \times T_{target} \times D_s}$ is the target traffic flow, T_{target} represents the time-series length of the target traffic flow data, D_s represents the dimension of $\mathbf{X}_{target}$. $\mathbf{H}^{(L)} \in \mathbb{R}^{N \times T_{target} \times D_s}$ is the forecasted traffic flow output from the last encoder. $\hat{\mathbf{X}}$ is the output of the last decoder.

Different from the RNN-based methods (the temporal relation is limited to the previous time steps) and CNN-based methods (only capturing the temporal relation within the receptive field), the receptive field of the self-attention mechanism can cover a long sequence of traffic data, and it is a computationally efficient model that supports the parallelism [27]; therefore, our LTT can learn the bidirectional and long-term temporal relations of the traffic flow sequences.

D. Spatio-Temporal Graph Convolutional Networks

The spatio-temporal convolutional network consists of three same blocks (shown in Fig. 2), and each block is constructed by a short-term temporal convolution and a spatial graph convolution. As the name suggests, the short-term temporal convolution aims to capture the short-term temporal dependence of the traffic flow data, and the spatial graph convolution is to capture the corresponding spatial dependence.

1) Short-Term Temporal Convolution: To extract the short-term temporal relation, we design the short-term temporal convolution STC(·) as follows,

$$\mathbf{T}^{(l)} = \text{STC}(\mathbf{Z}^{(l-1)}), \tag{10}$$

where $\mathbf{Z}^{(l-1)}$ is the input feature of l-th block and $\mathbf{Z}^{(0)} = \mathbf{X} \in \mathbb{R}^{N \times T_1 \times D_r}$.

Compared with the recurrent neural network, the convolutional neural network is less time-consuming because it can be trained in parallel. Moreover, CNN only focuses on the data within the receptive field, which captures the short-term drastic changes in the data. Therefore, we perform one-dimensional convolution along the temporal axis T_1 of traffic flow data to extract the short-term temporal relation. To further prevent from the vanishing gradient problem and ensure the sufficient information transmission, we use a gated linear unit [40] to activate the features extracted by one-dimensional convolution,

$$\begin{aligned} (\beta_1, \beta_2) &= \text{split}(\text{Conv}_{1\text{d}}(\mathbf{Z}^{(l-1)})), \\ \text{STC}(\mathbf{Z}^{(l-1)}) &= \beta_1 \odot \text{sigmoid}\left(\beta_2\right), \end{aligned} \tag{11}$$

where $\text{Conv}_{1\text{d}}(\cdot)$ means the one-dimensional convolution and the convolution kernel $\Psi \in \mathbb{R}^{t \times D_r \times D_1}$. split(·) equally divides the traffic features extracted by $\text{Conv}_{1\text{d}}(\cdot)$. The divided features are $\{\beta_1, \beta_2\} \in \mathbb{R}^{N \times T^{Conv} \times (D_1/2)}$ and $T^{Conv} = T_1 - (2 * t - 2)$. Operator sigmoid(·) is the activation function and $\odot$ stands for the Hadamard product operation.

In summary, formula (11) is a gated linear unit activation for the features extracted by one-dimensional convolution. Such gated linear unit activation removes the common-used tanh activation function. So, the gradient can easily pass through the activated units, and the gradient is not reduced during the back-propagation process, which overcomes the vanishing gradient problem.

2) Spatial Graph Convolution: Traffic flow data change in a complex manner and the road network has the non-Euclidean structure, so there are many implicit and complex relations among the road segments. For example, two road segments may not be directly connected in the road network, while they have many identical neighbors. It is a natural belief that these two road segments possess a high-order structural relation. As an important deep learning method, GCN [9], [10], [41] can mine such potential high-order relation in the road network.

We employ a spatial graph convolutional networks to extract the spatial relation from the traffic flow data below,

$$\begin{aligned} \mathbf{S}^{(l)} &= \text{GCN}(\mathbf{F}^{(l)}, \mathbf{A}) \\ &= \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{F}^{(l)}\mathbf{W}^{(l)}\right), \end{aligned} \tag{12}$$

where $\mathbf{S}^{(l)}$ is the output of the l-th spatial graph convolution layer. $\mathbf{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix constructed by the road network, which describes the interaction relations among road segments. $\mathbf{F}^{(l)}$ is the fused traffic flow features of the short-term temporal information and the long-term temporal information, which boost the forecasting performance. The specific fusion operation will be introduced in the next subsection. $\mathbf{W}^{(l)} \in \mathbb{R}^{D_2 \times D_3}$ is the weigh matrix of GCN.

It is well known that the stacking multiple layers of GCN often result in the over-smoothing problem. To alleviate this problem, the traffic information extracted by LTT is transferred to GCN layer-by-layer.

Finally, $\mathbf{S}^{(l)}$ goes through another short-term temporal convolution to adjust the time step of traffic features,

$$\mathbf{Z}^{(l)} = \text{STC}(\mathbf{S}^{(l)}). \tag{13}$$

Such alternating operation of spatial graph convolutions and short-term temporal convolutions is beneficial to obtain the spatial relations between the road segments in different time steps. Note that the final forecasted traffic flow of the entire model is $\mathbf{Z}^{(L)} = \text{STC}(\mathbf{S}^{(L)}) \in \mathbb{R}^{N \times T_2 \times D_s}$.

To make the forecasted traffic flow data close to the ground truth $\mathbf{X}_{\text{target}}$ from multiple aspects, we minimize the mean absolute error (MAE) and the mean square error (MSE), which can be defined as,

$$\begin{aligned} \mathcal{L}_{\text{stgc-mae}} &= ||\mathbf{Z}^{(L)} - \mathbf{X}_{\text{target}}||_1 \\ \mathcal{L}_{\text{stgc-mse}} &= ||\mathbf{Z}^{(L)} - \mathbf{X}_{\text{target}}||_2. \end{aligned} \tag{14}$$

The MSE loss is more sensitive to outliers, while the MAE loss mainly suits the traffic flow data in the stable periods.

E. Long-Short Temporal Information Fusion Module

In different periods, the importance of the long-term temporal information and the short-term one is different. For example, in the morning and evening peak periods, the traffic volume is large and the current traffic congestion has a significant impact on the traffic flow after 15 minutes. In this case, the short-term traffic flow information plays a crucial role in traffic flow forecasting tasks. In the traffic flat periods, the traffic volume is small and the traffic congestion dissipates quickly, so the long-term temporal information from 30 minutes to 1 hour is more instructive.

The LTT network captures the long-term temporal relations among the traffic flow data, and the STGC network learns the short-term temporal relations and the spatial relations, but how to properly integrate these heterogeneous data is a key issue.

We propose the long-short temporal information fusion module (LSTIF) to learn a better traffic flow representation layer-by-layer through an attention fusion mechanism. Then, the fused representation is sent to each STGC layer to solve the over-smoothing problem of GCN. The LSTIF module has the powerful global learning ability and good parallelism, which further highlights the critical information in the fusion representations according to the target traffic flow period and suppresses the useless noise. The specific form of LSTIF is expressed as,

$$\begin{aligned}\mathbf{F}^{(l)} &= \text{Fusion}(\mathbf{H}^{(l)}, \mathbf{T}^{(l)}) \\ &= \text{Attention}\left((\mathbf{T}^{(l)}, \mathbf{H}^{(l)}) \cdot \mathbf{Y}^{(l)}\right),\end{aligned} \tag{15}$$

where $\mathbf{F}^{(l)} \in \mathbb{R}^{N \times T^{\text{Conv}} \times D_2}$ is an attention fusion feature of $\mathbf{H}^{(l)}$ and $\mathbf{T}^{(l)}$. Attention$(\cdot)$ represents the attention fusion operation. $\mathbf{Y}^{(l)}$ is the concatenation of $\mathbf{H}^{(l)}$ and $\mathbf{T}^{(l)}$ as,

$$\mathbf{Y}^{(l)} = \text{Concat}(\mathbf{T}^{(l)}, \mathbf{H}^{(l)}), \tag{16}$$

where $\mathbf{Y}^{(l)} \in \mathbb{R}^{N \times T^{\text{Conv}} \times D_2}$ represents the concatenation feature and $D_2 = D_1/2 + D_r$.

Inspired by the heterogeneous graph Transformer [42], we also map the feature $\mathbf{H}^{(l)}$ to the Query vector and map the features $\mathbf{T}^{(l)}$ to the Key vector, and their dot product is calculated as the attention weight. The feature $\mathbf{Y}^{(l)}$ is used as the Value vector. The multi-head mechanism is also used here to enhance the algorithm robustness and the specific operations are given as follows,

$$\begin{aligned}\mathbf{F}^{(l)} &= \text{Concat}(\mathbf{head}_1, \cdots, \mathbf{head}_M)\mathbf{W}^C, \\ \mathbf{head}_m &= \text{SoftMax}(\left(\mathbf{K}^m(\mathbf{Q}^m)^T\right)) \cdot \frac{\mathbf{V}^i}{\sqrt{D_1/2}}, \\ \mathbf{K}^m &= \mathbf{W}_T^m \mathbf{T}^{(l)}, \\ \mathbf{Q}^m &= \mathbf{W}_H^m \mathbf{H}^{(l)}, \\ \mathbf{V}^m &= \mathbf{W}_Y^m \mathbf{Y}^{(l)},\end{aligned} \tag{17}$$

where $\mathbf{W}_T^m$, $\mathbf{W}_H^m$, $\mathbf{W}_Y^m$ and $\mathbf{W}^C \in \mathbb{R}^{D_2 \times D_2}$ are the corresponding linear transformation matrices.

The LSTIF module is an adaptive attention fusion module. For different target traffic flow periods, the LSTIF module can learn the changing trend from the historical traffic flow data, and assign proper weights to the long-term and short-term temporal representations, which dynamically analyzes the temporal relations among the traffic flow data.

F. The Objective Function

Each module boosts the traffic flow forecasting performance from different angles. We summarize them in an overall objective function as the following form,

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{stgc-mse}} + \lambda_1 \mathcal{L}_{\text{stgc-mae}} + \lambda_2 \mathcal{L}_{\text{trans-res}} \\ + \lambda_3 \mathcal{L}_{\text{trans-fore}},\end{aligned} \tag{18}$$

where λ_1, λ_2 and λ_3 are the hyper-parameters balancing the importance of difference modules. In order to better illustrate the operating procedure of the proposed model, we list the training process in APPENDIX A Algorithm 1.

IV. Experimental

A. Datesets

In order to prove the robustness and effectiveness of our method, we verify the proposed model on two types of traffic datasets, i.e., the road network traffic flow data and the metro passenger flow data.

- **PeMS-BAY**[1] [12] contains the six-month driving speed information (from Jan 1, 2017 to May 31, 2017) captured by 325 sensors located in the Bay Area. This dataset collects totally 52116 piece of traffic flow data.
- **PeMSD7(M)**[2] [43] collects the traffic status information captured by 228 sensors located in the California highway during the weekdays from May 2012 to June 2012. Since the 228 sensors are randomly selected, the relation between sensors is relatively weak.
- **Beijing Metro**[3] [44] gathers the passenger flow data from the Beijing metro system in August 2015, which covers the whole 325 stations and 22 lines in the Beijing subway system. Each piece of data includes three attribute values, i.e., the exit flow volume, the entrance flow volume and the flow volume. In this experiment, we use the entrance flow volume. Different from the other two road network traffic datasets, the stations on the same line have strong relations in the metro traffic dataset.

Each piece of data has a 5-minute interval, so there are 12 traffic flow data points in one hour. Then, we normalize these pieces of data by removing the mean and scaling to the unit variance,

$$\mathbf{X}' = \frac{\mathbf{X} - \text{mean}(\mathbf{X})}{\text{std}(\mathbf{X})}, \tag{19}$$

where mean$(\mathbf{X})$ and std$(\mathbf{X})$ denote the mean and the standard deviation operations of the observed traffic flow data $\mathbf{X}$, respectively.

[1] https://github.com/liyaguang/DCRNN
[2] https://github.com/VeritasYin/STGCN_IJCAI-18
[3] https://github.com/huogy/HSTGCNT

B. Compared Methods

The following representative and state-of-the-art traffic flow forecasting methods are chosen as the baselines,

- **History Average model (HA)** [45] uses the average traffic information in the historical periods as the prediction.
- **Linear Support Vector Regression (LSVR)** [20] utilizes a linear support vector machine to achieve the regression, which is a basic machine learning based traffic flow forecasting method relating to the time series.
- **Feed-forward Neural Network (FNN)** [46] is a standard full-connected neural network model.
- **Fully-Connected LSTM (FC-LSTM)** [6] is a standard LSTM network with the full-connected hidden units, while it only captures the temporal information from the traffic flow data.
- **Spatio-Temporal Graph Convolutional Networks (STGCN)** [11] combines the graph convolution with the one-dimensional convolution to capture the comprehensive spatio-temporal relation.
- **Diffusion Convolutional Recurrent Neural Network (DCRNN)** [12] exploits the bidirectional random walking graph convolution to model the spatial dependence, and uses the recurrent neural encoder-decoder network to model the temporal dependence.
- **Graph WaveNet (GWN)** [15] designs the graph convolution module by using the adaptive dependency matrix and the dilated one-dimensional convolution [47]. The receptive field of dilated one-dimensional convolution increases as the number of layers increases, improving the effect of processing long-term temporal data.
- **Spatial-Temporal Synchronous Graph Convolutional Networks (STSGCN)** [26] constructs a localized spatial-temporal graph, and utilizes the spatial-temporal synchronous graph convolutional module to capture the spatial and temporal relations at the same time.
- **Spatial-Temporal Fusion Graph Neural Network (STFGNN)** [48] utilizes the DTW algorithm to generate the temporal graph, and designs the spatial-temporal fusion graph neural module to synchronously capture the spatial-temporal relation.

In the field of traffic flow forecasting, DCRNN, STGCN, and GWN are the widely used baselines, especially GWN is currently the most popular traffic flow forecasting method due to its excellent experimental results.

To measure the performance of all methods, three common metrics are chosen, i.e., Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE),

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i|; \\ \text{MAPE} &= \frac{1}{N}\sum_{i=1}^{N}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \times 100\%; \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i|^2}, \end{aligned} \tag{20}$$

where $\hat{x}$ and x are the forecasted value and its ground truth.

However, Beijing Subway is idle between 23 : 00 and 5 : 00, and a large number of zeros exist in the Beijing Metro dataset. We cannot calculate its MAPE on account of the divisor is zero as shown in above function.

C. Hyperparameter Settings

Following the works in [12], we divide the PeMS-BAY and PeMSD7(M) datasets into the percentage ratio 7 : 1 : 2, then construct the corresponding training set, validating set and the testing set, respectively. Similarly, following the strategy in [44], the Beijing Metro dataset is segmented into a ratio of 8 : 1 : 1.

For the network parameters of compared methods, we pick up the recommended values and try our best to obtain the optimal experimental results. Specifically, the LSVR uses the linear kernel and sets the penalty term as 0.001. The FNN model uses a three-layer fully-connected network with the dimension $12-128-64$. The FC-LSTM model is a two-layer stacked LSTM network. For STGCN, the dimension of three hidden layers in each ST-Conv block is $64-16-64$. The number of the graph convolution kernels and the temporal convolution kernels are both set to 3. GWN consists of eight layers, which contains a series of expansion factors (1; 2; 1; 2; 1; 2; 1; 2). For both STSGCN and STFGNN, the size of the spatial-temporal fusion graph is $K = 4$, and the dimension of each latent layer is set as 64. Adam optimizer has a learning rate of 0.001.

Our HSTGCNT model contains two parallel deep neural networks, i.e., LTT and STGC. In the LTT network, it contains 3 encoders and 3 decoders. The dimensions Q, K, and V in the attention module are set as 8. And each attention module has 8 heads. The dimension of the fully-connected feed-forward block is 64, and the dropout rate is 0.3. The STGC network has 3 STGC modules. The dimension of the entire STGC network is $32-64-64-32-128-128-128-128$. The long-short temporal information fusion module is a bridge connecting above two deep networks, and each module also contains 8 attention heads. Due to the graphics memory limitations, we cannot add a fusion mechanism to the last STGC modules. We use the RMSProp optimizer to train our model, where the learning rate is set to 0.001. The hyper-parameters of the objective function are set as $\lambda_1 = 0.1$, $\lambda_2 = 0.01$, $\lambda_3 = 0.001$.

For all three datasets, no matter the forecast period is 15 min, 30 min, 45 min or 60 min, our input sequence is 60 min, i.e. 12 time steps.

For PeMS-BAY dataset, we set the batch size as 12. For PeMSD7(M) and Beijing Metro datasets, the batch size is set as 20. we repeat 200 epochs to train the proposed model, and select the model with the smallest loss on the validation set for testing.

The whole experiments are run on a workstation server with one Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz and one NVIDIA GeForce RTX 2080TI GPU.

D. Experiment Results Analysis

TABLE II and III illustrate the whole experimental results on three traffic datasets. Obviously, the proposed method is

TABLE II
PERFORMANCE COMPARISON ON THE TWO HIGHWAY TRAFFIC FLOW DATASETS. WE MARK THE BEST-PERFORMING RESULTS BY BOLDED FONT

Model	PeMS-BAY(15min/30min/60min)			PeMSD7(M)(15min/30min/60min)		
	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE
HA	2.94	6.61	6.69	4.82	11.78	9.17
LSVR	1.85/2.48/3.28	3.80/5.50/8.00	3.59/5.18/7.08	2.49/3.46/4.94	5.91/8.42/12.41	4.55/6.44/9.08
FNN	2.20/2.30/2.46	5.19/5.43/5.89	4.42/4.63/4.89	2.53/3.73/5.28	6.05/9.48/13.73	4.46/6.46/8.75
FC-LSTM	2.95/3.97/4.74	4.81/5.25/5.79	4.19/4.55/4.96	3.57/3.92/4.16	8.60/9.55/10.10	6.20/7.03/7.51
STGCN	1.39/1.84/2.42	3.00/4.22/5.58	2.92/4.12/5.33	2.25/3.05/4.04	5.26/7.33/9.77	4.04/5.70/7.55
DCRNN	1.38/1.74/2.07	2.90/3.90/4.92	2.95/3.97/4.74	2.37/3.31/4.01	5.54/8.06/9.99	4.21/5.96/7.19
GWN	1.36/1.85/1.98	2.84/3.79/4.59	2.93/3.86/4.63	2.17/**2.80**/3.44	5.13/6.89/8.68	4.01/5.48/6.71
STSGCN	1.57/1.98/2.53	4.34/4.64/6.13	4.42/4.51/5.97	2.59/3.34/4.62	6.19/8.18/11.71	4.91/6.59/8.75
STFGNN	1.47/1.91/2.44	3.14/4.32/6.07	3.04/4.28/5.54	2.47/3.23/4.21	5.86/8.10/10.35	4.54/6.27/8.07
HSTGCNT	**1.29/1.62/1.91**	**2.68/3.70/4.57**	**2.67/3.79/4.51**	**2.14/2.80/3.40**	**5.02/6.88/8.55**	**4.00/5.38/6.44**

TABLE III
PERFORMANCE COMPARISON ON THE METRO PASSENGER FLOW DATASETS. WE MARK THE BEST-PERFORMING RESULTS BY BOLDED FONT

Model	Beijing Metro(15min/30min/45min)	
	MAE	RMSE
HA	20.52	52.72
LSVR	14.71/16.55/17.75	25.12/31.33/32.93
FNN	11.01/14.46/18.78	23.61/31.22/40.75
FC-LSTM	10.76/12.27/12.86	21.22/22.33/23.74
STGCN	7.83/9.56/10.16	16.81/17.92/20.29
DCRNN	8.37/9.64/11.63	19.13/23.38/25.87
GWN	7.39/7.45/8.49	15.85/16.00/18.14
STSGCN	10.65/12.24/16.22	20.71/24.03/33.23
STFGNN	9.13/9.60/11.72	17.47/18.50/22.39
HSTGCNT	**6.72/7.15/7.69**	**14.91/15.65/17.06**

superior to other baseline methods in most cases. Especially compared to the most important baseline GWN, the proposed model improves 6% in MAE, 2.6% in MAPE, and 2.3% in RMSE on the PeMS-BAY dataset.

LSVR, FNN, and FC-LSTM only consider the temporal relations among the traffic flow data. However, both the road network and the rail station network exist obvious spatial relations among nodes (road segments or stations), which leads to the relatively lousy traffic forecasting performance for the above three methods. Furthermore, STGCN and DCRNN simultaneously consider the spatial and temporal relations, opening a new way for the traffic flow forecasting tasks and receiving the relatively satisfactory experimental results. Finally, GWN designs the dilated one-dimensional convolution and the adaptive adjacency matrix in the graph convolution, which greatly surpasses the previous convolution-based method STGCN and the RNN-based method DCRNN.

The structure and experimental performance of the proposed method are compared to STGCN and DCRNN below, respectively:

- STGCN uses the one-dimension convolution to extract the local temporal relations, while it lacks the ability to capture the long-term temporal relations among the traffic flow data. The fusion module in HSTGCNT fuses the short-term and long-term temporal relations, so it can capture the global temporal relation of the long sequences. On the three datasets, HSTGCNT achieves the better forecasting results than STGCN, especially for the long-term forecasting performance of more than 30 minutes.
- DCRNN uses the gated recurrent unit to remember the historical information in the sequence. However, the gradient vanishing problem makes GRU often fail to capture the long-term temporal relations in practical applications [49]. Since the GRU loop body cannot be performed parallelly, the algorithm efficiency of DCRNN is not high. Moreover, DCRNN shares the parameters of the GCN layer among all recurrent units, which makes it difficult to pay attention to the spatial relations among the traffic data on different time steps. Compared to DCRNN, HSTGCNT acquires small improvements for the short-time intervals (15-30 minutes); however, it achieves greater enhancements in the range from 45 minutes to 60 minutes.
- In most cases, GWN indeed achieves the best forecasting results among the comparison methods. On PeMSD7(M) and Beijing Metro datasets, our HSTGCNT's short-term forecasting effect is basically close to GWN. But for the long-term forecasting, HSTGCNT performs obviously better than GWN, which owes to the advantages of the LTT network in the proposed HSTGCNT. In addition, the fusion module adds the long-term temporal information to the graph convolution, which still solves the over-smoothing problem of GCN.
- STSGCN and STFGNN provide a new way to mine the temporal and spatial relations in the traffic flow. Different from other methods designing temporal and spatial networks separately, STSGCN and STFGNN capture the localized temporal and spatial relations simultaneously. STSGCN performs poorly on the three datasets due

TABLE IV
ABLATION EXPERIMENT ON PEMS-BAY AND PEMSD7(M) DATASET

Model	PeMS-BAY(15min/30min/60min)			PeMSD7(M)(15min/30min/60min)		
	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE
HSTGCNT-wLTT	1.35/1.68/2.06	2.85/3.79/5.19	2.78/**3.79**/5.31	2.16/2.95/3.40	5.20/7.59/8.98	4.11/5.54/6.65
HSTGCNT-Linear	1.34/1.72/2.11	2.89/3.84/5.07	2.71/3.97/5.15	2.22/2.89/3.57	5.45/7.42/8.91	4.29/5.58/6.82
HSTGCNT-wFUSE	1.31/1.66/2.02	2.69/3.75/4.86	2.73/3.92/4.95	2.17/2.82/3.50	5.33/7.20/8.72	4.11/5.51/6.76
HSTGCNT	**1.29/1.62/1.91**	**2.68/3.70/4.57**	**2.67/3.79/4.51**	**2.14/2.80/3.40**	**5.02/6.88/8.55**	**4.00/5.38/6.44**

TABLE V
ABLATION EXPERIMENT ON BEIJING METRO DATASET

Model	Beijing Metro(15min/30min/45min)	
	MAE	RMSE
HSTGCNT-wLTT	7.31/7.98/8.24	15.28/16.44/17.55
HSTGCNT-Linear	7.25/7.74/8.10	15.39/15.90/17.71
HSTGCNT-wFUSE	7.16/7.46/7.85	15.42/15.76/17.93
HSTGCNT	**6.72/7.15/7.69**	**14.91/15.65/17.06**

to the localized spatial-temporal graph only considering the adjacency relationship between two time steps. The experimental results of STFGNN are better than STS-GCN, which owes to STFGNN adding the temporal graph constructed by the DTW algorithm when it constructs the spatial-temporal fusion graph.

In the experiments, the proposed model performs slightly better on the Beijing Metro dataset than on other two datasets, and we owe to the degree of each node is relatively small in the metro network, i.e., the degree of non-transfer stations is 2. The spatial relationships in the Beijing Metro dataset are not very complex, so the temporal relationships are relatively important. Such simple graph structure of Beijing Metro limits the ability of GCN to extract the spatial relationships of data. Other baselines mainly mine the spatial and short-term temporal relationships, ignoring the long-term temporal relationships of traffic flow data. Our proposed model designs the LTT module to capture the long-term temporal relationships on basis of spatial and short-term temporal relationships, which improves the forecasting performance compared to other baselines. Therefore, the proposed model is more competitive on the Beijing metro dataset than on the PeMSD7(M) dataset.

E. Ablation Studies

In order to verify the effectiveness of the long-short temporal information fusion module and the long-term temporal Transformer network in the proposed HSTGCNT, we design three variants, including:

- **HSTGCNT-wLTT**: This variant deletes the LTT network, which studies the significance of the over-smoothing problem of GCN.
- **HSTGCNT-LINEAR**: This variant replaces the LTT network by a fully-connected network to prove the importance of the Transformer module.
- **HSTGCNT-wFUSE**: This variant replaces the fusion module by directly concatenating the long-term and short-term traffic information, which tests whether the fusion strategy is beneficial to the traffic flow forecasting tasks.

Although the above variants delete or replace some modules, the hyperparameter settings of the rest are the same as the proposed HSTGCNT. As shown in TABLE IV and TABLE V, HSTGCNT gets the optimal forecasting performance on all datasets, which independently demonstrates the effectiveness of the long-short temporal information fusion module and the long-term temporal Transformer network.

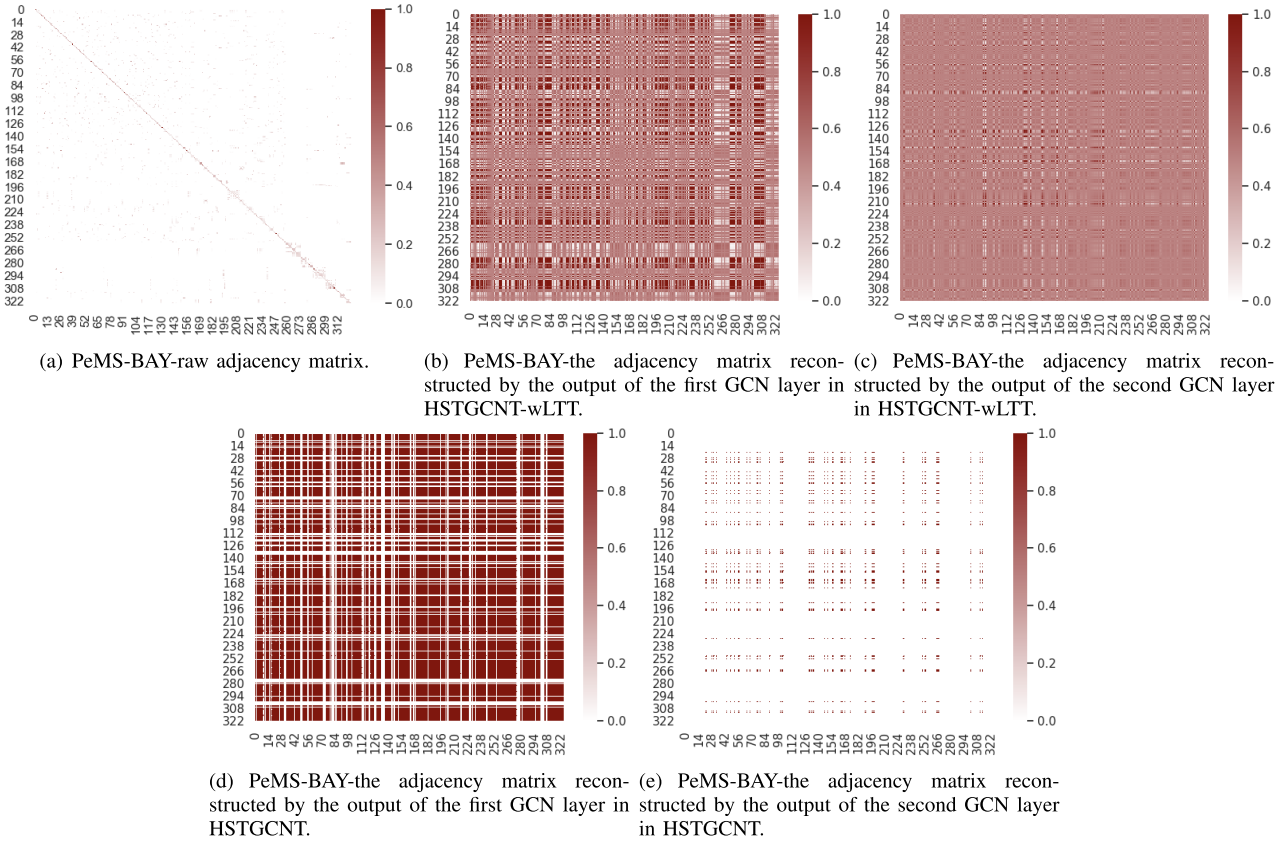
- Without the LTT module, there is no information supplemented into HSTGCNT-wLTT, which leads to the unsatisfied performance. We owe it to the over-smoothing of GCN. In other words, the learning ability of GCN is limited in the absence of additional information.
- To alleviate the over-smoothing problem of GCN, we introduce a fully-connected network to supplement the information to GCN, namely HSTGCNT-LINEAR, which is partly superior to HSTGCNT-wLTT and still inferior to our HSTGCNT. The reason may be that HSTGCNT-LINEAR lacks the ability to analyze the long-term traffic flow.
- We further introduce the long-term temporal information captured by the Transformer module into HSTGCNT-wFUSE, namely HSTGCNT-wFUSE, which acquires obvious improvements comparing to other variants in most cases. However, HSTGCNT-wFUSE also performs worse than HSTGCNT, which demonstrates the importance of the fusion module. This adaptive fusion mechanism of the long-term and short-term traffic flow information is significantly better than the simple concatenation.

F. Visualization Analysis

The over-smoothing problem has always been a major obstacle for the graph convolution network to obtain the more accurate forecasted results. In order to further prove that the proposed hierarchical network can alleviate the over-smoothing problem, we visualize the adjacency matrices constructed by the output of each GCN layer of HSTGCNT and HSTGCNT-wLTT, respectively. The adjacency matrix reflects the potential connection relation of the road network. The specific implementation is as follow,

$$\widetilde{\mathbf{A}} = \mathrm{ReLU}\left((\mathbf{S}^{(l)})^T \mathbf{S}^{(l)}\right), \tag{21}$$

where $\widetilde{\mathbf{A}}$ represents the reconstructed adjacency matrix.



(a) PeMS-BAY-raw adjacency matrix.

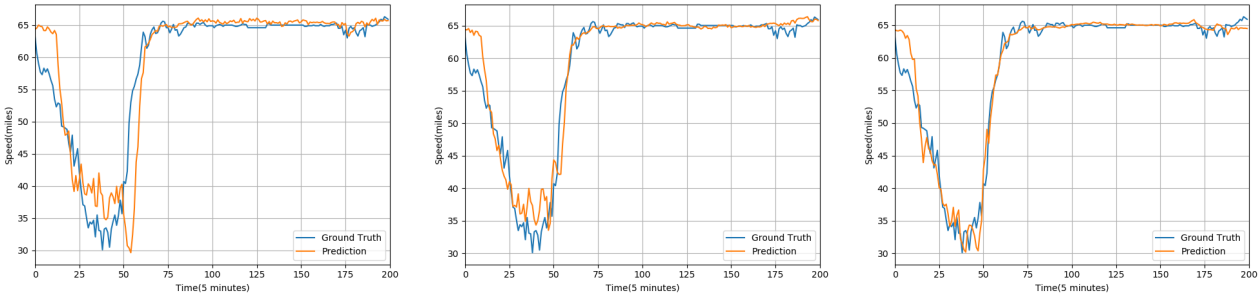
(b) PeMS-BAY-the adjacency matrix reconstructed by the output of the first GCN layer in HSTGCNT-wLTT.

(c) PeMS-BAY-the adjacency matrix reconstructed by the output of the second GCN layer in HSTGCNT-wLTT.

(d) PeMS-BAY-the adjacency matrix reconstructed by the output of the first GCN layer in HSTGCNT.

(e) PeMS-BAY-the adjacency matrix reconstructed by the output of the second GCN layer in HSTGCNT.

Fig. 4. Visualization of the raw adjacency matrix and the adjacency matrix reconstructed with the output of the GCN layer.



(a) Epoch=10.

(b) Epoch=30.

(c) Epoch=50.

Fig. 5. The visualization of HSTGCNT forecasted results after 10, 30 and 50 training epochs.

Fig. 4(a) firstly visualizes the adjacency matrix of the input data, which only includes the first-order adjacency relations. The adjacency matrices in Fig. 4(b) and 4(c) are constructed by the outputs of the first and the second GCN layers without the additional information transferred from the LTT network. Obviously, the adjacency matrix in Fig. 4(c) is smoother than that in Fig. 4(b), due to the stacking of GCN layers would cause the data overly smooth. In other words, it leads the homogeneity of road segments in the road network. Such oversmoothing problem also affects the traffic flow forecasting performance.

Similarly, Fig 4(d) and Fig 4(e) visualize two adjacency matrices relating to the first and the second GCN layers of the proposed model. Through the comparison, we can see that Fig. 4(b) and 4(c) are smoother than Fig. 4(d) and Fig. 4(e), which means the extra long-term temporal traffic flow information can effectively solve the over-smoothing problem of GCN.

G. Case Study

To illustrate that the proposed model can effectively handle complex traffic situations, we conduct a case study. We randomly select the 202-th sensor from the PeMSD7(M) dataset, then we visualize its traffic flow forecasting results of the proposed model and the corresponding Ground Truth in Fig. 5.

The forecasting results perform poorly after the 10 epochs, i.e. in the early training stage. After 30 epochs, the forecasting accuracy obtains the obvious improvement, but it still does

not adapt well to some sudden changes (about 20-th to 50-th time interval). As more training epochs are performed, the proposed model keeps improving its forecasting ability and fits the ground truth almost perfectly after 50 epochs.

In above, the proposed model can perform the reasonable forecasting in these complex situations quickly.

V. Conclusion

We proposed a hierarchical traffic flow forecasting model via spatio-temporal graph convolutional networks and Transformer, which connects the LTT network and the STGC network layer-by-layer through the long-short temporal information fusion module. The fusion representation is transferred to the GCN layer in the STGC network, which partially overcomes the over-smoothing problem of GCN. In the LTT network, we replaced CNN or RNN by the Transformer network to capture the long-term relations among the traffic flow data. The forecasting and reconstruction losses in the LTT module ensure that the captured long-term temporal information is suitable for the traffic forecasting tasks. The excellent experimental results on various datasets verify the superiority of the proposed method.

Appendix I
The Algorithm of HSTGCNT

Algorithm 1 The HSTGCNT Model for the Traffic Flow Data Forecasting

Require: The observed traffic flow data $\mathbf{X} \in \mathbb{R}^{N \times T_1 \times D_r}$ where T_1 denotes the length of traffic flow data, and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$.

1: Integrate the position embedding **PE** and the traffic flow data $\mathbf{X}$ to obtain $\mathbf{X}^* \in \mathbb{R}^{N \times T_1 \times (D_p + D_r)}$;
2: Pre-train the LTT network by minimizing the formula (9);
3: **for** $l \leqslant L$ **do**
4: Load the parameters of the pre-trained LTT network in step 2;
5: Update the data representation $\mathbf{H}^{(l)}$ through $\mathbf{H}^{(l-1)}$ and the formula (6);
6: Update the data representation $\mathbf{T}^{(l)}$ through $\mathbf{Z}^{(l-1)}$ and the formula (10);
7: Update the fusion data representation $\mathbf{F}^{(l)}$ by fusing $\mathbf{H}^{(l)}$ and $\mathbf{T}^{(l)}$ through the formula (15);
8: Transfer $\mathbf{F}^{(l)}$ into the STGC's spatial graph convolution to get $\mathbf{S}^{(l)}$ through the formula (12);
9: Transfer $\mathbf{S}^{(l)}$ to another one-dimensional convolution to obtain $\mathbf{Z}^{(l)}$ through the formula (13);
10: Optimize the entire network by minimizing the objective function (18);
11: **end for**
12: **return** The forecasted traffic flow $\mathbf{Z}^{(L)}$.

References

[1] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[2] Z. Lu, C. Zhou, J. Wu, H. Jiang, and S. Cui, "Integrating Granger causality and vector auto-regression for traffic prediction of large-scale WLANs," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 1, pp. 136–151, 2016.

[3] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1655–1661.

[5] 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.

[6] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[7] Y. Wang, Y. Zhang, X.-L. Piao, H. Liu, and K. Zhang, "Traffic data reconstruction via adaptive spatial–temporal correlations," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1531–1543, Aug. 2018.

[8] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[9] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[11] 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.*, Jul. 2018, pp. 3634–3640.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[13] L. Zhao, Y. Song, C. Zhang, and Y. Liu, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[14] 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.*, 2019, pp. 922–929.

[15] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.

[16] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Trans. GIS*, vol. 24, no. 3, pp. 736–755, Jun. 2020.

[17] C. Li, K. Yang, H. Tang, P. Wang, J. Li, and Q. He, "Fault diagnosis for rolling bearings of a freight train under limited fault data: Few-shot learning method," *J. Transp. Eng., A, Syst.*, vol. 147, no. 8, Aug. 2021, Art. no. 04021041.

[18] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.

[19] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, Sep. 2004.

[20] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[21] R. Du, Y. Zhang, B. Wang, H. Liu, G. Qi, and B. Yin, "Low-rank representation based traffic data completion method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 5127–5134.

[22] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Workshop Deep Learn. Represent. Learn.*, 2014, pp. 1–9.

[23] 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. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[24] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3913–3926, Oct. 2019.

[25] D. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[26] 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.

[27] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[29] C. Park et al., "ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1215–1224.

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

[31] H. Xue and F. D. Salim, "TERMCast: Temporal relation modeling for effective urban flow forecasting," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2021, pp. 741–753.

[32] S. Li et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[33] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021, pp. 1–9.

[34] I. Padhi et al., "Tabular transformers for modeling multivariate time series," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3565–3569.

[35] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, and J. Huang, "Adversarial sparse transformer for time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17105–17115.

[36] D. Daiya and C. Lin, "Stock movement prediction and portfolio management via multimodal learning with transformer," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3305–3309.

[37] K. Xu et al., "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[39] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[40] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.

[41] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[42] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, Apr. 2020, pp. 2704–2710.

[43] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: Mining loop detector data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1748, no. 1, pp. 96–102, Jan. 2001.

[44] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, "Metro passenger flow prediction via dynamic hypergraph convolution networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 1–13, Dec. 2021.

[45] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *J. Highway Transp. Res. Develop.*, vol. 21, no. 3, pp. 82–85, Mar. 2004.

[46] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

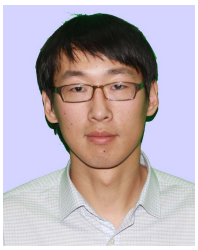
[47] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

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

[49] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, nos. 8–9, pp. 1421–1441, Sep. 2019.

Guangyu Huo received the B.Sc. degree in the IoT engineering and the M.S. degree in computer science from the Beijing University of Technology, China, in 2016 and 2019, respectively, where he is currently pursuing the PhD. degree in control science and engineering. His current research interests include intelligent transportation, computer vision, pattern recognition, and deep learning.

Yong Zhang (Member, IEEE) received the Ph.D. degree in computer science from the Beijing University of Technology (BJUT) in 2010. He is currently an Associate Professor of computer science with BJUT. His research interests include intelligent transportation systems, big data analysis and visualization, and computer graphics.



Boyue Wang received the B.Sc. degree in computer science from the Hebei University of Technology, China, in 2012, and the Ph.D. degree from the Beijing University of Technology, Beijing, China, in 2018. He is an Associate Professor with the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology. His current research interests include computer vision, pattern recognition, manifold learning, and kernel methods.

Junbin Gao received the B.Sc. degree in computational mathematics from the Huazhong University of Science and Technology (HUST), China, in 1982, and the Ph.D. degree from the Dalian University of Technology, China, in 1991. He was a Professor of computer science at the School of Computing and Mathematics. Charles Sturt University, Australia. He was a Senior Lecturer and a Lecturer of computer science at the University of New England, Australia, from 2001 to 2005. From 1982 to 2001, he was an Associate Lecturer, a Lecturer, an Associate Professor, and a Professor at the Department of Mathematics, HUST. He is a Professor of big data analytics with The University of Sydney Business School, The University of Sydney. His main research interests include machine learning, data analytics, Bayesian learning and inference, and image analysis.

Yongli Hu (Member, IEEE) received the Ph.D. degree from the Beijing University of Technology in 2005. He is a Professor with the Faculty of Information Technology, Beijing University of Technology. He is a Researcher with the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology. His research interests include computer graphics, pattern recognition, and multimedia technology.

Baocai Yin (Member, IEEE) received the Ph.D. degree from the Dalian University of Technology in 1993. He is a Professor with the Faculty of Information Technology, Beijing University of Technology. He is a Researcher with the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology. His research interests include multimedia, multifunctional perception, virtual reality, and computer graphics. He is a member of China Computer Federation.