This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038380, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Dynamic Global-Local Spatial-Temporal Network for Traffic Speed Prediction

DONG FENG[1,2,3], ZHONGCHENG WU[1,2], JUN ZHANG[1,2,3] and ZIHENG WU [4]
[1]High Magnetic Field Laboratory, HFIPS, Chinese Academy of Sciences, Hefei 230031, China
[2]University of Science and Technology of China, Hefei 230026, China
[3]High Magnetic Field Laboratory of Anhui Province, Hefei 230031, China
[4]School of Electrical and Information Engineering, Anhui University of Technology, Maanshan, 243000, China.

Corresponding author: Jun Zhang (e-mail: zhang_jun@hmfl.ac.cn).

This work was financially supported by Anhui Province Internet of Vehicles Data Sharing Center and Operation Management Cloud Service Platform, the High Magnetic Field Laboratory of Anhui Province and Anhui Province Natural Science Foundation of China under Grant No.1908085QF253

ABSTRACT Predicting traffic speed accurately is a very challenging task of the intelligent traffic system (ITS), due to the complex and dynamic spatial-temporal dependencies from both temporal and spatial aspects. There not only exits short-term local neighboring fluctuation and long-term global trend in temporal aspect, but also local and global correlations in spatial aspect. Most existing work focus on the local spatial-temporal dependencies, ignoring the global dynamic spatial-temporal corrections, which is comparably critical for traffic speed prediction. To address this problem, we propose a novel **D**ynamic **G**lobal-**L**ocal **S**patial-**T**emporal **N**etwork(DGLSTNet) for traffic speed prediction, which consists of multiple spatial-temporal module considering the local and global information simultaneously from both temporal and spatial perspective. Each temporal module applies stacked dilated convolution block to exploit multi-scale local temporal information. Moreover, we empoly a global temporal attention block to capture global dependencies of temporal domain in an attention mechanism. In each spatial module, we not only learn the local but also focus on dynamic global spatial information learned by dymamic graph learning block. Combining the feature results from local and global perspective, the capability and expressiveness of traffic predicting model is improved. Experiment results on two real-world traffic datasets have demonstrated that our proposed model can effectively capture the comprehensive spatial-temporal dependencies and can achieve state-of-the-art prediction performance compared with the existing works.

INDEX TERMS traffic speed prediction, spatial-temporal network, graph convolutional network, dynamic graph learning.

I. INTRODUCTION

Predicting large-scale network-wide traffic becomes increasingly popular in the intelligent transportation systems due to its application and research significance. The development of an accurate and robust forecasting of multi-scale traffic conditions is a key consideration as it leads to many useful applications, such as designing and upgrading highway networks, improving traffic safety, reducing traffic congestion. Long-term traffic prediction is highly challenging due to the constantly changing nature of many impacting factors from both temporal and spatial aspects. There are two mainly issues should be considered in traffic prediction: (1) there exits short-term local neighboring fluctuation and long-term global trend in temporal aspect and traffic conditions are different at various times (i.e. morning peak, noon). (2) the saptial dependence is often found to exist in a wider range of the traffic networks, for instance, congestion can not only effect the neighboring regions but also reachable far distant regions. In other words, the spatial correlations over different regions are also both local and global. Moreover, since traffic is constantly evolving, the spatial corrections are not static but change over time. In summary, the local and global correlations exits from both temporal and spatial aspects in traffic systems. And therefore the approprite explicit local and global spatial-temporal modeling is great necessary and is the key to better prediction accuracy.

In the modern traffic forecasting systems, deep learning based works have received attention because they can model complex non-linear spatio-temporal information and achieve best results at present. A seemingly natural way is assuming city-wide traffic as a image [1] or grid [2], where each unit states the traffic condition of the corresponding region. Efforts [3] have been conducted to apply convolution neural network (CNN) to extract spatial correlation on traffic network since CNN exhibits superior capability for processing Euclidean-structured data. To make full use of the spatial and temporal dependence, the method of combining CNN and recurrent neural network (RNN) is proposed as the basic frame. Although the spatial-temporal features of the traffic data can be extracted by these models [4]–[6], their limitations are that the input must be standard 2D or 3D grid data, whereas the real-world traffic networks are irregular and complex topological structures. Current trends in modeling spatial dependency highlight the need for taking advantage of topological structure. Fortunately, in recent years, graph convolution network (GCN) [7], [8], which is accomplished in capturing structural feature of irregular graph, provides a good solution for traffic forecasting tasks. Substantial researches adopt GCN to model spatial dependencies, while the temporal dependencies among historical states are preserved by 1D convolution [9], [10] or RNN [11]–[14]. Although, introducing GCN have alleviated difficulties in traffic forecasting to some extent, from a careful review, there still remain two important problems neglected in current GCN-based approaches: (1) many existing methods only consider localized spatial dependencies but igore the global ones, which leads to inadequacy in capturing relevant information from distant links. Even though the global spatial dependencies are considered [15], the global adjacency matrix only be calculated once without considering the changing spatial correlations over time. (2) RNNs or 1D convolution based methods can not capture global temporal dependencies, since the receptive field of RNNs or 1D convolution is limited.

In light of preceding analysis, we argue that considering dynamic global and local spatial-temporal correlations simultaneously could enhance the capacity and expressiveness of traffic modeling. Consequently, in this paper we propose a novel dynamic global-local spatial-temporal network called DGLSTNet to predict traffic speed, which consists several spatial-temporal module (STM) considering the local and global information simultaneously from both temporal and spatial perspective. The main contributions of this paper are as follows:

- We develop a dynamic temporal module which considers the short-term local neighboring and the long-term global trend dependencies. It consists of a global temporal attention block(GTAB) and a stacked dilated convolution block (SDCB), where the former is used to extract whole-range global temproal features in an attention mechanism and the latter is used to capture the multi-scale local temporal features. In this way, dynamic temporal dependencies can be captured effectively.
- We design a dynamic spatial module which considers local spatial dependencies and global ones simultaneously. Especially, a dynamic graph learning block(DGLB) is introduced for learning the dynamic topological correction in a global way. By integrating the pre-defined and the adaptively learned global adjacency matrices into graph convolution operation to capture both local and global spatial dependencies simultaneously, the capacity and expressiveness of capturing saptial dependencies are enhanced.
- We conduct extensive experiments on two real-world traffic datasets, METR-LA and PEMS-BAY, and the proposed model achieves the state-of-the-art results.

II. RELATED WORKS

Traffic Prediction has been extensively studied in past few decades. Early statistical methods [16], [17] and traditional machine learning methods [18]–[21], mainly employ shallow machine learning for a single observation node or few nodes, which can not model the non-linear temporal correlations of traffic data effectively and neglect the spatial dependency. Recent advances in deep learning [22], [23] make it possible to model the complicated spatial-temporal dependency in traffic forecasting. To model spatial dependency, some attempts [1], [2] are assuming city-wide traffic as a regular grid structures(e.g.,images and videos), where each pixel states the traffic condition of the corresponding region. Then, Convolutional Neural Network (CNN) [3] or Recurrent Neural Network (RNN) [24]–[26] and the combination of CNN and RNN [4]–[6] are utilized for traffic forecasting. However, the main limitation of the above models is that CNN can only capture the spatial dependency of regular grid structures but do not work for data points with irregular topologies. Therefore, they fail to make an effective use of the topological structure of the traffic network to capture complex spatial correlations.

To bridge the above gap, a series of studies has generalized traditional convolution to model arbitrary graphs on spectral or spatial domain [27]. The introduction of graph convolution network (GCN) boosts the latest rapid development of graph-structured data learning. In traffic forecasting, many researchers [9]–[14] have applied GCN to capture more complex saptial dependencies. Li at al. [11] proposed Diffusion Convolutional Recurrent Neural Network (DCRNN), which replaces the fully-connected layers in Gated Recurrent Units (GRU) by the diffusion graph convolution operator. The diffusion convolution performs graph convolution on the given graph and its inverse to consider both inflow and outflow relationships. Yu et al. [9] proposed an Spatial-Temporal GCN (ST-GCN) , which applied GCN to capture the spatial dependency and employed CNN on time axis to capture the temporal dependency leading to much computationally efficient than RNN. Guo at al. [28] proposed a novel attention based spatial-temporal graph convolutional network (ASTGCN) to capture the dynamic local spatial correlations, which adjust the adjacency matrices by the attention score. But, all these approaches assuming that spatial correlations only existing connected or very close nodes are essentially local method, which do not consider the dynamic non-local spatial correlations between nodes on traffic networks. Afterward, GMAN [29] proposes an multi-head attention-based encoder-decoder architecture to capture dynamically non-local spatial correlations. Owing to calculate spatial attention score from all nodes and temporal attention score from all time steps , the time and memory consumption of GMAN is inevitably heavy. Although the schemes mentioned above have improved the accuracy of traffic prediction, they still fail to capture the global and local spatial-temporal dependencies simultaneously in the traffic network.

To overcome the shortcomings mentioned above, Graph Wavenet [10] developed a novel adaptive dependency matrix by computing the similarity of node embeddings to capture non-local the hidden spatial dependency in the data. STSeq2Seq [15] utilizes seq2seq architecture which couples a convolutional encoder and a recurrent decoder with attention mechanism for traffic predictions . In STSeq2Seq encoder, patten-aware adjacency matrices is construcetd and applied to GCN to model the non-local spatial correlations dynamically. The two models have proved the necessary to consider the global and local spatial-temporal dependencies simultaneously. However, both non-local adjacency matrices of the two model only be calculated once at the beginning of each model by softmax function, which results in dense fully connected adjacency matrices and introduce lots of noise into spatial correlations. Such dynamic global spatial information derived from GraphWavenet and STSeq2Seq is relatively weak and less effective. In this paper, we propose dymamic graph learning concept which is quite efficient to address these shortcomings.

III. MATERIALS AND METHODS

A. PROBLEM DEFINITION

Given the historical traffic data from N correlated traffic sensors located on a road network, the task of traffic predicting is to forecast the future traffic of the road network. Traffic predicting can be formulated as a graph modeling problem since the traffic flows are restricted on road networks, which is abstracted as graphs. Following previous studies, we define the N correlated traffic sensors as a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is the set of N nodes(roads or sensors) on the road network, $\mathcal{E}$ is a set of edges representing the connectivity among nodes, and $A \in \mathbb{R}^{N\times N}$ is a predefined weighted adjacency matrix representing the nodes' proximities (usually measured by road network distance between any pair of nodes or topological adjacency). Denote the traffic flow observed on $\mathcal{G}$ as a graph signal $X \in \mathbb{R}^{P\times N\times D}$, where P represents the number of historical time steps, D is the feature dimension of each node (e.g.,speed,volume), in which X_t represents the features of nodes of $\mathcal{G}$ at time step t. Given a graph $\mathcal{G}$, then the traffic forecasting problem is formed as learning a function $f(\cdot)$ that maps P historical graph signals to future Q graph signals:

$$\left[X_{(t-P+1):t}, \mathcal{G}\right] \xrightarrow{f} \left[X_{(t+1):(t+Q)}\right] \tag{1}$$

where $X_{(t-P+1):t} \in \mathbb{R}^{P\times N\times D}$ and $X_{(t+1):(t+Q)} \in \mathbb{R}^{Q\times N\times D}$

B. MODEL OVERVIEW

The overall framework of proposed model is illustrated in Figure 1. Our proposed DGLSTNet consists of a temporal embedding module to encoder the temporal information of input data, L-stacked saptial-temporal module(STM) and an output module. A STM is composed of a temporal sub-module and a spatial sub-module, which is constructed to capture dynamic temporal and spatial dependencies respectively. Each temporal sub-module contains a global temporal attention block and stacked dilated convolution block, where the former is used to extract the while-range global temporal features in an attention mechanism and the latter is used to extract multi-scale local temporal features by dilated convolution, shown in Figure 1(bottom left). Each spatial sub-module firstly constructe a dynamic graph learning block (DGLB) for global spatial correlations learning , and then integrate the static graph with the dynamic graph learned by DGLB into the hybird dynamic-static GCN block to capture both local and global spatial features, shown in Figure 1(bottom right). In addition, residual connections [30] is introduced in each sub-module to stabilize the learning process when it goes deep. To faciliate the residual connection, all modules produce the same dimensions F of outputs. Skip connections are added after each temporal module, which utilizes standard convolutions to standardize information that jumps to the output module to have the same sequence length. Finally, the output module consists of two standard convolution layers, which project the summary of each skip connection to the desired predicted data $Y \in \mathbb{R}^{Q\times N\times D}$. In more detail, the core components of our model are illustrated in the following.

C. TEMPORAL EMBEDDING

Following [29], we also adopt a temporal embedding module to encoder every time step into a vector. Specifically, in this paper, the sample rate of data is 5-minute interval, and 288 snapshots per day. We encoder the day-of-week and time-of-day of each time step into $e_{dayweek} \in \mathbb{R}^{7}$ and $e_{timeday} \in \mathbb{R}^{228}$ using one-hot coding, and concatenate them into a vector $e_{temp} \in \mathbb{R}^{228+7}$. Next, we apply a two-layer fully-connected neural network M to transform the time features of P historical time steps to a embedding matrices $X_{emb} \in \mathbb{R}^{P\times F}$, which contains the necessary temporal information to help the model capturing the spatial-temporal information. We add the temporal embedding matrix X_{emb} to the tensor $X_{conv} \in \mathbb{R}^{P\times N\times F}$ encodered by a 1×1 convolution with broadcast operation to obtain the new representations $X^{'}$ as initial input of the first STM :

$$\begin{aligned} X_{emb} &= M(e_{dayweek}||e_{timeday}) \\ X^{'} &= X_{conv} + X_{emb} \end{aligned} \tag{2}$$

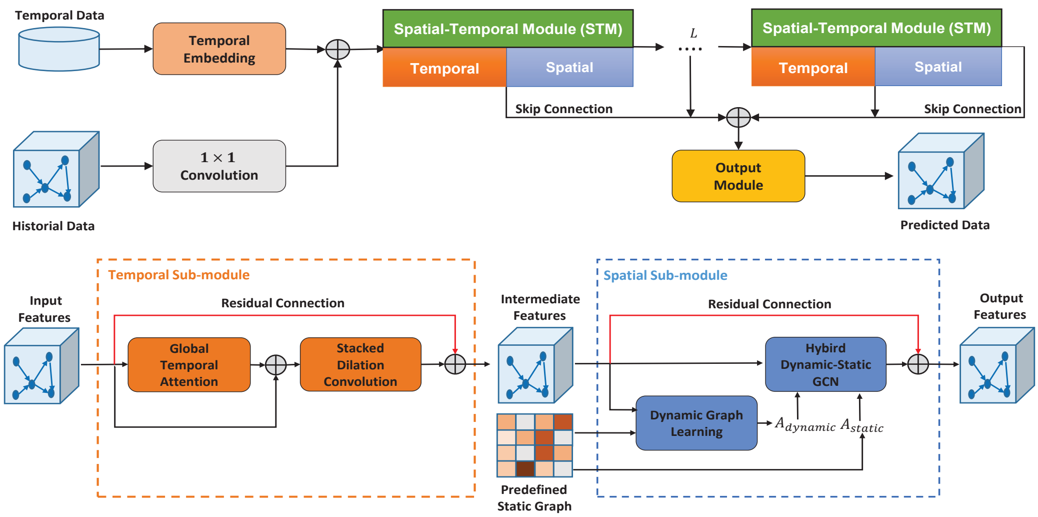


FIGURE 1: The architecture of DGLSTNet.

D. DYNAMIC TEMPORAL SUB-MODULE

The traffic conditions have complex non-linear relationship between different time steps. To capture the short-term local neighboring and the long-term global trend dependencies, we propose a dynamic temporal module which is composed of a global temporal attention block (GTAB) and a stacked dilated convolution block (SDCB), as shown in the left part of Figure 2. Given the feature maps $\hat{Z} \in \mathbb{R}^{T\times N\times F}$ as input of temporal module , where T is the length of temporal dimension of the current STM. In GTAB, we utilize a multi-head temporal attention mechanism to get a global representation $Z_g \in \mathbb{R}^{T\times N\times F}$, and then add Z_g to Z to get more effective representations $Z^{'}$ of the temporal domain. Then, SDCB utilizes dilated convolutions which have exponentially growing receptive field with few layers to extract multi-scale local temporal correlations $Z \in \mathbb{R}^{T^{'}\times N\times F}$, where $T^{'} = T - R + 1$ and R is the receptive field of SDCB.

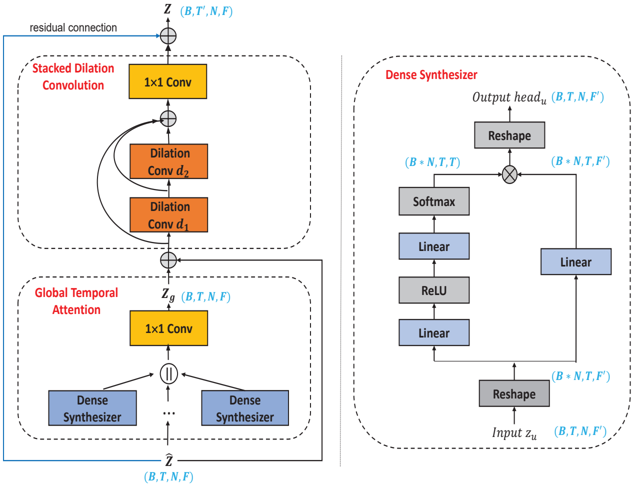


FIGURE 2: Pipeline of the dynamic temporal module.

1) Global Temporal Attention Block (GTAB)

Traditional temporal convolution layer can only capture temporal features between local time steps, being this limited by the fixed kernel size, so the global temporal correlations are not captured effectively. In this block, we design a global temporal attention block, that can directly attend to features across time steps without any restriction to extract global temporal features.

Dense Synthesizer [31] is used as the attention kernel in each attention head, which learns the attention weights directly and accelerates both the training and inference speed drastically. Following a self-attention strategy, we use multi-head mechanism to capture several independent diverse representation. As shown in right flow of Figure 2, each Dense Synthesizer operates on an input $z_u \in \mathbb{R}^{T\times N\times F^{'}}$, where u is the number of heads and F' is the dimensions of each head. The attention weights of the u-th head $e_u \in \mathbb{R}^{N\times T\times T}$ are generated by feeding z_u to a function $f(\cdot)$ with two hidden layers :

$$e_u = f(z_u) = W_2(\sigma(W_1 z_u + b_1)) + b_2 \tag{3}$$

where $W_1 \in \mathbb{R}^{F^{'}\times T}$, $W_2 \in \mathbb{R}^{T\times T}$, $b_1 \in \mathbb{R}^{F^{'}}$, $b_2 \in \mathbb{R}^{T}$ are learnable parameters, and $\sigma(\cdot)$ is the $ReLU$ activation function. The element $e_u^v(i,j)$ in the tensor e_u represent the correlation between time step i and time step j of the v-th node. After the normalizing attention scores via softmax function is obtained, the output features of each head $head_u$ can be computed as follow :

$$head_u = Softmax(e_u) z_u W_u \tag{4}$$

where $W_u \in \mathbb{R}^{F^{'}\times F^{'}}$. Finally, we concatenate the output of each head and project them by a learnable linear transformation $W_o \in \mathbb{R}^{uF^{'}\times F}$:

$$Z_g = Concat(head_1, ..., head_u) * W_o \tag{5}$$

In this work, we employ $u = 2$ paraller attention heads to extract global temporal features of the entrie sequence. Finally, combining Z_g and $\hat{Z}$, the outputs $Z^{'}$ of GTAB will be obtained by element-wise summation, which can be formulated as :

$$Z^{'} = \hat{Z} + Z_g \tag{6}$$

2) Stacked Dilated Convolution Block (SDCB)

Dilated convolution network(DCN) [32] allows an exponentially large receptive field by increasing the layer depth so as to capture both short-term neighboring and long-term periodic temporal dependencies with high effectiveness. Therefore, we adopt DCN to extract features at different temporal scales explicitly. Note that dilated convolution operation is based on 1D convolution, injecting holes into the convolution kernel, sliding over inputs by skipping values with a certain step. Mathematically, given a 1D sequence input $x \in \mathbb{R}^T$ and a convolution kernel $f \in \mathbb{R}^k$, x_t denotes the t-th value in the 1D sequence x, and a d-dilated convolution operation of x at step t is represented as

$$(f \star_d x)_t = \sum_{i=0}^{k-1} f(i) \cdot x_{t-d\times i} \tag{7}$$

where d is the dilated factor which controls the skipping distance. Suppose the progressively increasing dilated factor is $d_m = 2^{m-1}$, the receptive field size R of a m layer dilated convolution network with kernel size k is

$$R = 1 + (k-1)(2^m - 1) \tag{8}$$

In order to extract informative features on multi temporal scales , we stack two-layer dilated convolution following by rectified linear unit ($ReLU$), and then apply a 1×1 convolutional layer to fusion the summary of features extracted by every dilated convolution layer, as shown in Figure 2. Given an initial input $Z_0 = Z^{'} \in \mathbb{R}^{T\times N\times F}$, the output $Z \in \mathbb{R}^{(T-R+1)\times N\times F}$ after passing the SDCB is formulated as:

$$\begin{aligned} Z_{m+1} &= \sigma(\Phi_m \star_{d_m} Z_m) \\ Z &= \sigma(\Phi_{1\star1}(\sum_{i=0}^{m} Z_m)) + \hat{Z}W_r \end{aligned} \tag{9}$$

where Φ_m is the convolution kernel for the m-th dilated causal convolution layer and $W_r \in \mathbb{R}^{F\times F}$ is the learnable parameter of a residual connection. The adding features are truncated to the same length according to the latest filter and summary across the channel dimension. $\sigma(\cdot)$ is the $ReLU$ activation function, Φ_{1*1} represent the 1×1 convolution. After the dynamic temporal features Z are aggregated, a skip connection utilizing standard convolution to standardizes the Z is appended for jumping to the output module to have the same sequence length.

E. DYNAMIC SPATIAL MODULE

We argue that there are two conditions should be noticed in spatial feature representation: (1) The directly connected nodes; (2) The dis-connected but reachable nodes. For some extent, these two kinds of nodes all effect each other. The dependencies of the former are local while the latter are global. The key is how to model both local and global effection simultaneously. In this module, we model spatial dependencies using both static and dynamic strategies, as shown in Figure 3. From the static perspective, we use the pre-defined adjacency matrix $A_{static} = \{A, A^T\}$ as local spatial correlation and perform bidirectional diffusion convolution to extract local spatial information. From the dynamic perspective, we applied dynamic graph learning (DGL) concept to constructe adaptive adjacency matrix $A_{dynamic}$ as global spatial correlation for the sake of modeling the global spatial dependencies. By using the above two strategies and fusing the features between graph nodes in both static and dynamic way, the capacity and expressiveness of capturing saptial dependencies are enhanced.

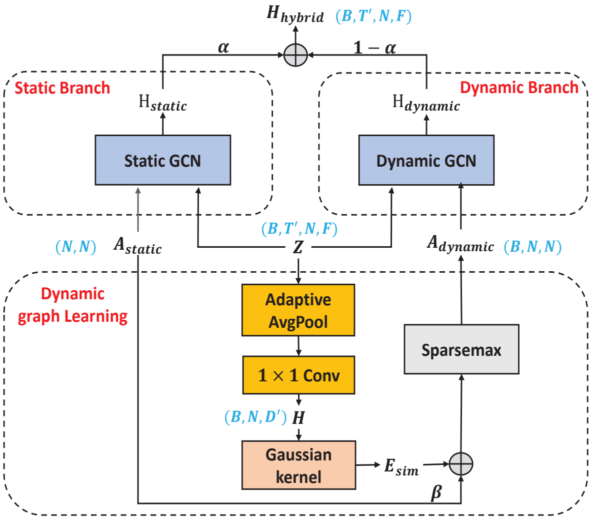


FIGURE 3: Pipeline of the dynamic spatial module.

1) Dynamic Graph Learning

Given the adjacency matrix A_{static} ,which is pre-defined with prior knowledge, we design an extremely lightweight dynamic graph learning layer to predict a dynamic matrix $A_{dynamic}$ directly, as shown in Figure 3. The DGL takes the outputs of the dynamic temporal module $Z \in \mathbb{R}^{T^{'}\times N\times F}$ as inputs, the feature and temporal dimensions of each node are firstly squeezed and compressed by an adaptive average pooling (AdaptiveAvgPool) layer and two 1×1 convolutional layers sequentially, then we can get the hidden representations $H \in \mathbb{R}^{N\times D^{'}}$, where $D^{'}$ denote the dimensions of the hidden layer. After that, we use the Gaussian kernel [33] to measure the feature similarity score $E_{sim}(p,q)$ between each pair of nodes (v_p, v_q) in the road network :

$$\begin{aligned} &E_{sim}(p,q) = exp(-\mathbb{D}_{p,q}/m) \\ &where \quad \mathbb{D}_{p,q} = \left\|(H_p - H_q)^T W_\phi (H_p - H_q)\right\|_2 \end{aligned} \tag{10}$$

where $H_p \in \mathbb{R}^{1\times D'}$ and $H_q \in \mathbb{R}^{1\times D'}$ denote the representations of node v_p and v_q, $\|\cdot\|_2$ perform ℓ_2 norm. $W_\phi \in \mathbb{R}^{D' \times D'}$ is one of the trainable weights, which is a shared transform basis to the Euclidean distance between node v_p and v_q. m is a positive hyperparameter used to adjust the scale of the distance $\mathbb{D}$ between nodes. Then, we incorporate A_{static} into a structure learning layer to get the final similarity score :

$$S(p,q) = E_{sim}(p,q) + \beta \cdot \mathrm{A}_{static}(p,q) \tag{11}$$

where $\beta \geq 0$ is a trade-off parameter. If the $\beta = 0$, the similarity score $S(p,q)$ is learned in a data-driven way without any prior assumption. And if the $\beta > 0$, $S(p,q)$ will be given a relatively larger similarity score between directly connected nodes, and at the same time try to learn the underlying pairwise relationships and directions between disconnected nodes [34] .

Moreover, to improve training efficiency, reduce the effect of noise, amplify the effective relations and make the model more robust, we use sparemax function [34] which can retain most important properties of softmax function and has the ability of producing sparse distributions. The sparsemax$(\cdot)$ function can be formulated as follows :

$$\begin{aligned} A_{dynamic} &= \mathrm{sparsemax}(S) \\ &= [S(p,q) - \tau(S(p,:))]_+ \end{aligned} \tag{12}$$

where $[x]_+ = \max\{0, x\}$ and $\tau(\cdot)$ is the threshold function that returns a threshold. Sparsemax$(\cdot)$ preserves the values above the threshold and the other values will be truncated to zeros, which make the adjacency matrix sparse. It is worth noting that $A_{dymamic} = \{A_d^f, A_d^b\}$, A_d^f, A_d^b are the dynamic graphs predicted by DGL according to the forward adjacency matrix A and and the backward adjacency matrix A^T, respectively. Additionally, the dynamic adjacency matrix $A_{dynamic}$ learned by the DGL layer differ among different samples as well as each STM. Then, the $A_{dynamic}$ is fed into the nexting graph convolution block to capture global spatial representations.

2) Hybird Dynamic-Static GCN

The block contains a static branch and a dynamic branch which aim to capture the local and global spatial dependencies respectively. The static branch takes the outputs of the dynamic temporal module $Z \in \mathbb{R}^{T' \times N \times F}$ and the pre-defined $A_{static} = \{A, A^T\}$ as inputs. Yet, the dynamic branch takes the same features Z and the dynamic $A_{dymamic} = \{A_d^f, A_d^b\}$ predicted by DGL as inputs. Finally, the outputs of the two branchs further apaptively get fused to model more effective spatial dependencies.

In the static branch, borrowing from Wu et al. [10], we use pre-defined static adjacency matrix A_{static} to perform two step diffusion convolution in both forword an backward directions to capture localized spatial dependencies, which corresponds to Eq.(13)

$$H_{static} = \sum_{k=0}^{K-1} (A^f)^k Z W_{s1} + \sum_{k=0}^{K-1} (A^b)^k Z W_{s2} \tag{13}$$

where $(\cdot)^k$ represents the power series of the transition matrix, K is the number of diffusion steps , W_{s1} and W_{s2} denote the learnable kernel . Here, the forward transition matrix $A^f = A/\mathrm{rowsum}(A)$ and the backward transition matrix $A^b = A^T/\mathrm{rowsum}(A^T)$. H_{static} is the output features of the static branch, which only capture the local spatial dependencies. However, the static branch has proved the ability to capture the influence from both the upstream and the downstream direction in traffic prediction model [10], [11].

More importantly, the dynamic branch can be formulated as :

$$\mathrm{H}_{dynamic} = A_d^f Z W_{d1} + A_d^b Z W_{d2} \tag{14}$$

where W_{d1} and W_{d2} are the learnable parameters of dynamic branch. $H_{dynamic}$ is the out of the dynamic branch , which extract the global spatial dependency of the traffic.

After extracting the dynamic local and global spatial dependencies, a weighted summation operation is applied for fusion. The combination is expressed as follows:

$$\mathrm{H}_{hybrid} = \alpha \odot \mathrm{H}_{static} + (1-\alpha) \odot \mathrm{H}_{dynamic} \tag{15}$$

where $\alpha \in R^F$ is learned parameters, differing in channels dimension, which is used to balance H_{static} and $\mathrm{H}_{dynamic}$. The $\odot$ is broadcasted hadamard product. Finally, a residual mechanism and LayerNorm [35] are applied to improve generalization performance.

F. OPTIMIZATION STRATEGY

Our model predicts the future Q timestamps speeds of all sensors based on the historical P timestamps traffic records. In the training phase, we choose Mean Absolute Error(MAE) as the loss function. Furthermore, to avoid overfitting, we adopt L2 regularization, which is defined by:

$$L(\Theta) = \frac{1}{Q} \sum_{i=t+1}^{t+Q} |Y^i - \hat{Y}^i| + \lambda \|\Theta\|_2 \tag{16}$$

where Y^i and $\hat{Y}^i$ denote the prediction value and ground truth at i-th step, respectively. λ is the coefficient of the $L2$ regularization and Θ denotes all trainable parameters in our model.

IV. EXPERIMENTS

A. DATASETS

We evaluate our proposed method on two large-scale real world datasets: METR-LA and PEMS-BAY [11]. The two datasets are similar and both the nodes represent sensors measuring traffic speed at 5-minutes intervals. METR-LA dataset contains four months(rangeing from Mar 1st 2012 to Jun 30th 2012) of traffic information on 207 sensors located on the highways of Los Angeles County. PEMS-BAY dataset contains six months(ranging from Jan 1st 2017 to May 31th 2017) of statistics on traffic speed on 325 sensors in the Bay area. We adopt the same data processing procedures as in the original papers [11]. A sequence of length 12 is used as the input to predict the future traffic speed in one hour(12 steps). Z-sore normalization is applied to all the input speed data. Detailed statistics of the datasets are shown in Table 1.

TABLE 1: Statistics of METR-LA and PEMS-BAY datasets

Dataset	#Sensors	#Edges	#Time Steps	# Time range
METR-LA	207	1515	34272	3/1/2012 - 6/30/2017
PEMS-BAY	325	2369	52116	1/1/2017 - 5/31/2017

We build weighted adjacency matrix by road network distances between sensors with a thresholded Gaussian kernel. The adjacency matrix is defined as :

$$A_{ij} = \begin{cases} \exp(-\frac{dist(\nu_i,\nu_j)^2}{\sigma^2}), & \text{if } dist(\nu_i,\nu_j)^2 \leq \delta \\ 0, & \text{if } dist(\nu_i,\nu_j)^2 > \delta \end{cases} \tag{17}$$

where A_{ij} represents the edge between sensor ν_i and ν_j, $dist\,(\nu_i,\nu_j)$ denotes the road network distance from sensor ν_i to ν_j. σ is the standard deviation of distances and δ is the threshold to control the sparsity of the adjacency matrix A.

B. BASELINES

We compare our model with the following models:

- **FNN**: Feed Forward Neural Network with two hidden layers.each contains 256 units.
- **FC-LSTM**: A Recurrent neural network with fully connected LSTM hidden units.
- **STGCN** [9]: A complete convolutional structure combining K-order chebyshev graph convolution with geted linear unit(GLU) convolution layers for traffic prediction.
- **DCRNN** [11]: Diffusion convolution recurrent neural network , which captures the spatial dependency using bidirectional random walks on the graph, and the temporal dependency using the encoder-decoder architecture with scheduled sampling.
- **GMAN** [29]: adapts an encoder-decoder architecture (similar with transformer), where both the encoder and the decoder consist of multiple spatio-temporal attention blocks to model the impact of the spatio-temporal factors on traffic conditions.
- **Graph Wavenet** [10]: A spatial-temporal graph convolutional network, which employs GCNs with a self-adaptive matrix and a stacked dilated 1D convolution to model the spatial-temporal dependencies.
- **STSeq2Seq** [15]: adapts an encoder-decoder architecture (similar with transformer), where both the encoder and the decoder consist of multiple spatio-temporal attention blocks to model the impact of the spatio-temporal factors on traffic conditions.

C. EXPERIMENT SETTINGS AND EVALUATION METRIC

We implemented our model based on the open source machine learning framework PyTorch [36] and conducted experiments on Inter(R) Xeon(R) E5-2609 CPU @1.70GHZ and one NVIDIA GeForce GTX 1080Ti GPU. We adopt the same proportion with Graph WaveNet [10] which generate training set (70%), validation set (10%) and test set (20%). For our model, the number of STM L were set to 4 ,the outputs dimensions of all modules F were set to 40, and the parameter β was set as 1. The batch size was set to 64; the maxinum diffusion step K was set as 2; the dropout rate of temporal module was 0.3. DGLSTNet was trained based on the Adam optimizer [37] for 100 epochs with early stopping to prevent model from overfitting. The initial learning rate was 0.001 with with a decay rate of 0.5 per 10 epochs after twentieth epoch. In addition, the Layernorm and L2 normalization with a weight decay of 2e-4 were applied for better generalization.

Three widely-used metrics are empolyed to evaluate predicting performance : Mean Absolute Error(MAE), Root Mean Squared Error(RMSE), and Mean Absolute Percentage Error(MAPE), defined as follows:

$$\begin{aligned} MAE &= \frac{1}{N \times Q} \sum_{i=1}^{N} \sum_{j=1}^{Q} |y_{i,j} - \hat{y}_{i,j}| \\ RMSE &= \sqrt{\frac{1}{N \times Q} \sum_{i=1}^{N} \sum_{j=1}^{Q} (y_{i,j} - \hat{y}_{i,j})^2} \\ MAPE &= \frac{1}{N \times Q} \sum_{i=1}^{N} \sum_{j=1}^{Q} \frac{|y_{i,j} - \hat{y}_{i,j}|}{y_{i,j}} \end{aligned} \tag{18}$$

where $\hat{y}_{i,j}$ and $y_{i,j}$ are the true value and the predicted value, N is the number of sensors we select in the road network, and Q is the total number of predicted time steps.

D. EXPERIMENTAL RESULTS

1) Prediction Performance Comparison

Table 2 displays our model and all baseline models on the METR-LA and PEMS-BAY datasets for 15 minutes, 30 minutes and 60 minutes ahead prediction of MAE, RMSE and MAPE. As shown in the table, we can observe that graph-based models including STGCN, DCRNN, Graph Wavenet and STSeq2Seq make better predictions than FNN and FC-LSTM. This means that considering the hidden spatial dependencies are critical to prediction performance.

TABLE 2: Performance comparison of different approaches for traffic prediction on METR-LA and PEMS-BAY datasets

Dataset	models	15min			30min			60min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	FNN	3.99	7.94	9.90%	4.23	8.17	12.90%	4.49	8.69	14.00%
	FC-LSTM	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
	STGCN	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	DCRNN	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
	GMAN	2.77	5.48	7.25%	3.07	6.34	8.35%	3.40	7.21	9.72%
	Graph Wavenet	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
	STSeq2Seq	2.64	5.10	**6.72%**	3.02	6.18	8.16%	3.47	7.36	9.96%
	DGLSTNet	**2.64**	**5.01**	6.73%	**2.98**	**6.00**	**8.05%**	**3.38**	**7.07**	**9.57%**
PEMS-BAY	FNN	2.20	4.42	5.19%	2.30	4.63	5.43%	2.46	4.98	5.89%
	FC-LSTM	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	STGCN	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	DCRNN	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	GMAN	1.34	2.82	2.81%	1.62	3.72	3.63%	**1.86**	4.32	**4.31%**
	Graph Wavenet	1.30	2.74	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
	STSeq2Seq	1.30	2.73	2.72%	1.62	3.72	3.61%	1.92	4.48	4.42%
	DGLSTNet	**1.29**	**2.70**	**2.68%**	**1.60**	**3.59**	**3.54%**	1.88	**4.30**	4.35%

STGCN and DCRNN are the earliest two GCNs-based methods for traffic prediction. STGCN employs K-order Chebyshev graph convolution and complete convolution structure on traffic data, while DCRNN combines diffusion graph convolution with recurrent neural networks in encoder-decoder manner. STGCN performs poorly compared to DCRNN, because STGCN is designed for one-step prediction and does not suit multiple steps ahead prediction scenario. However, the two model focus on modeling the spatial dependency by utilzing GCNs throughout a fixed weighted graph, which neglects the complexity and dynamic traffic condition over time. To overcome the challenges, GMAN proposed an complete attention-based encoder-decoder framework without GCNs, mainly improving the long-term prediction performance but performing poor short-term performance. Nevertheless, GMAN calculate multi spatial and temporal attention score from all vertices and time steps represently, which the time and memory consumption is more heavy.

DGLSTNet achieves the best performance for almost all forecasting horizons on all metrics and both datasets. There are some similarities among the three model GraphWavenet, STSeq2Seq and DGLSTNet. They all adopt the diffusion convolution to capture local spatial correlations and then integrate it with the dynamic global spatial correlations. The main difference is the constructe way of dynamic global adjacency matrix. GraphWavenet introduces an adaptive adjacency matrix and STSeq2Seq constructs patten-aware adjacency matrix by feature embedding. Both non-local adjacency matrix only be calculated once at the beginning of the model. Additionally, the adjacency matrix calculated by softmax function results in dense fully connected adjacency matrices, which introduce lots of noise into the learned spatial correlations. Such dynamic global spatial information derived from GraphWavenet and STSeq2Seq is relatively weak. By contrast, our model learned the dynamic adjacency matrices per-sample as well as per-GConv layer by the immediate features and sparsemax function, which can provide a better comprehension of traffic data and are crucial for modeling complicated dependencies. A phenomenon worth considering that our model achieves small improvement on 15-minute horizons (short-term) over GraphWavenet and STSeq2Seq, while large improvement on 60-minute horizons(long-term). As the long-term forecasting is inherently more uncertain than short-term forecasting, we consider our model is more capable to model complicated dependencies.

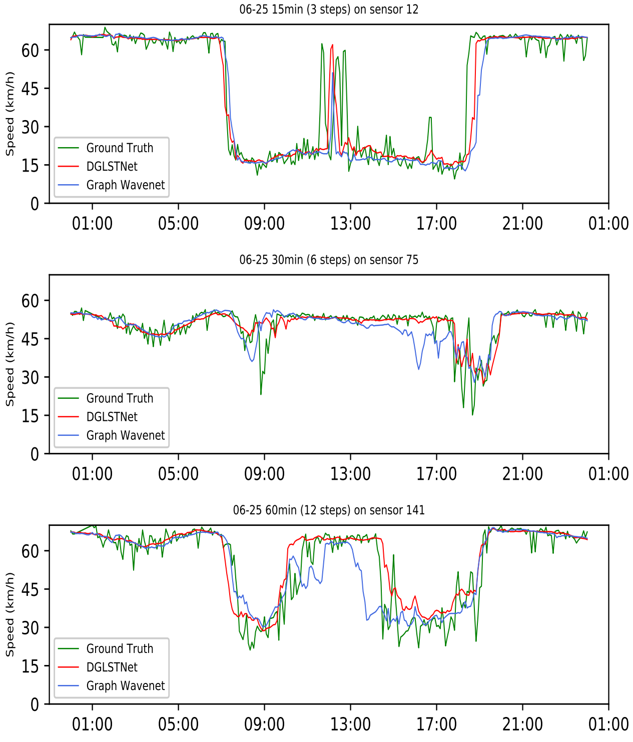


FIGURE 4: speed prediction of the dataset METR-LA on three random sensors

To better illustrate the model's forecasting ability, we randomly select three sensors and visualize in Figure 4 one-day predicting results of DGLSTNet and GraphWavenet under 15-min, 30-min and 60-min horizons, respectively. As can be seen, the curve of DGLSTNet is closer to the ground truth and predicted more accurately than GraphWavenet when the ground truth changes rapidly. Especially, the accuracy becomes more obvious as the predicting horizon increases. Moreover, DGLSTNet can capture the trends of peak hous better, this can be observed throughout the prediction of sensor 12 around 18:00 and sensor 141 around 9:00. Although both DGLSTNet and GraphWavenet are spatial-temporal deep learning network which consider global-local spatial dependencies simultaneously and based solely on convolution architecture, DGLSTNet is more effective than GraphWavenet in modeling the complex traffic conditions. We believe that the stable performance of DGLSTNet remains due to the dual role of dynamic graph learning in each STM and global temporal attention block.

2) Parameters Analysis

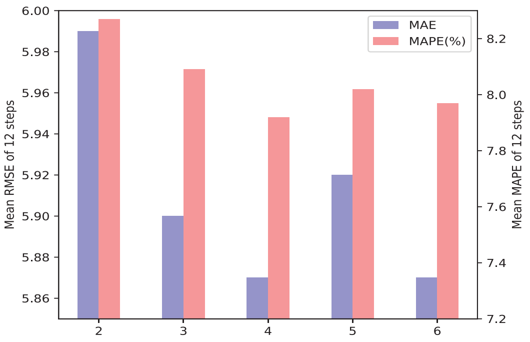


FIGURE 5: Prediction performance of DGLSTNet with different of L on METR-LA.

To investigate how different hyperparameter affect the model performance and also to choose the optimal settings of our model, we conduct analysis on two main hyperparameter: the number L of spatial-temporal modules ranges from 2 to 6; the parameter β of DGLB controlling the contribution of the static adjacency matrix ranges from 0 to 2.0. We use the mean RMSE and MAPE of 12 steps prediction results as the comparison metrics. The experimental results about parameter L are presented in Figure 5. As the number of STM increases, the prediction performance of the model improves. However, after the number of STM reaches 4, the accurary of the model was not improved or even becomes worse, and the training time and GPU memory of the model also increases greatly. Therefore, we chose $L = 4$ for our model as a tradeoff between performance and running efficiency.

After L was determined, we conduct analysis to select the approprite β, as shown in Figure 6. It can be observed that too larger or too smaller β degrades the model performance significantly. The RMSE metric was not sensitive to these parameters which were within the area of from 0.75 to 1.25, while the impact of MAPE metric was greater. On the whole, the best result was obtained when parameter $\beta = 1.0$. Finally, parameter $\beta = 1.0$ was used in our model.

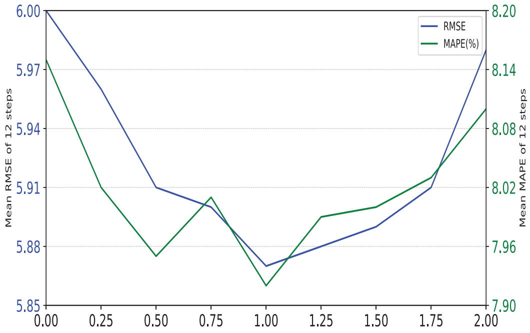


FIGURE 6: Prediction performance of DGLSTNet with different of β on METR-LA.

3) Ablation Studies

As described, there are three important components of our approach: 1) GTAB, which extract global temporal features by dense synthesizer attention; 2) static GCN branch in each spatial sub-module, which extract local spatial dependencies by diffusion graph convolution; 3) dynamic GCN branch in each spatial sub-module,which extract dynamic glocal spatial dependencies. In this section, in order to verify the effectiveness of every compoments on our model , we compare the following four variants of our model on METR-LA data.

- No-GTAB, whcih removes the global temporal attention block.
- No-StaticGCN, which removes static branch and retain the dynamic GCN branch.
- No-DyGCN, which removes dynamic GCN branch and retain the static GCN branch.
- Dense-DGLSTNet, which replace the sparsemax with softmax function.

Table 3 shows the RMSE and MAPE prediction performance of every variant over different prediction interval. We can find that DGLSTNet achieves the best prediction performance. The predicting results of excels the No-GTAB model, which verifies that it is necessary to model global dependencies in the temporal domain. The DGLSTNet are superior to model No-StaticGCN and No-DyGCN, which proves that capturing local and global spatial-temporal features simultaneously are important and effective for traffic prediction. Compared with DGLSTNet, Dense-DGLSTNet has poor prediction precision indicating that the softmax function resulting in dense fully connected adjacency matrix introduce lots of noise into the learned spatial correlations. In summary, the DGLSTNet can achieve the best results regardless of the prediction horizons, and each component of our model make sense.

In order to further prove the effectiveness of our proposed dynamic graph learning(DGL) in this paper, we evaluate two new variants based on our model:

TABLE 3: Performance of variants of DGLSTNet on different predicting intervals on METR-LA dataset

Models	15min		30min		60min		all mean	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
No-GTAB	5.06	6.84%	6.08	8.20%	7.15	9.81%	5.94	8.06%
No-StaticGCN	5.06	6.76%	6.07	8.17%	7.15	9.86%	5.95	8.08%
No-DyGCN	5.13	7.06%	6.10	8.42%	7.17	10.04%	5.98	8.32%
Dense-DGLSTNet	5.08	6.81%	6.12	8.14%	7.19	9.71%	6.00	8.00%
DGLSTNet	**5.01**	**6.73%**	**6.00**	**8.05%**	**7.07**	**9.57%**	**5.87**	**7.92%**

– Self-adapadj, similar to GraphWavenet, we constructe a self-adaptive global adjacency matirx at the beginning of DGLSTNet to replace DGL and apply the global matrix in each STM.

– PAM, we apply the same patten-aware matirx from STSeq2Seq to replace DSL in each STM.

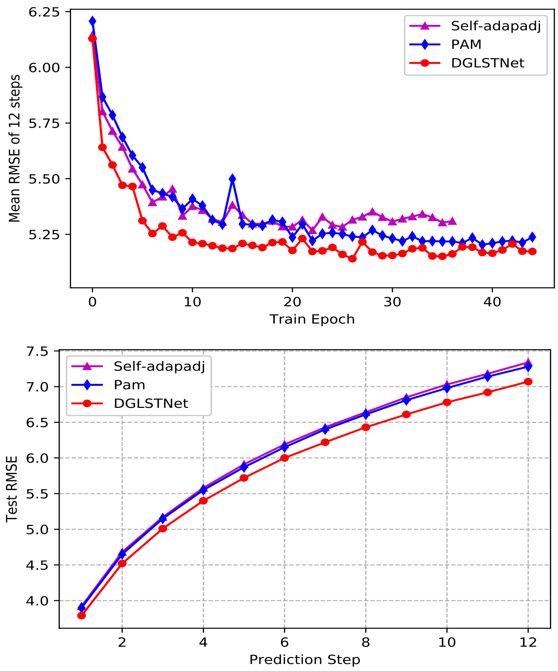


FIGURE 7: (a) validation Mean RMSE of 12 steps versus the number of training epoch on METR-LA dataset. (b) RMSE of each prediction interval of DSTGCN and two variants on METR-LA dataset.

Fugure 7 illustrates the RMSE comparison of the DGLSTNet and its two variants during training and inference phase on METR-LA. It can be seen from the figure that DGLSTNet has fast convergence rate and consistently outperforms better than Self-adapadj and PAM, indicating that the DGL is more effective than Self-adapadj and PAM in term of the dynamic global spatial correction modeling. Therefore, it is also proved that the dynamic and unique global spatial matrix learned by DGL per-STM can extract more informative global spatial dependencies than global spatial matrix only be learn once at the beginning of model.

4) Training Efficiency

Table 4 Table presents the computation time of DCRNN, Graph Wavenet, STSeq2Seq and DGLSTNet on the METR-LA dataset. We reccorded the average training time cost of each epoch and inference time for validation. It can be observed that during the training phase, GraphWavenet is the most efficient, followed by STSeq2Seq and DGLSTNet. DCRNN runs much slower than other methods due to the time-consuming sequence learning in complete recurrent networks. STSeq2Seq and DGLSTNet spend almost the same amount of time in the training phase, while differ greater in the inference phase. This is because STSeq2Seq use recurrent networks to predict step by step, while DGLSTNet generate all predictions in one run. Our model DGLSTNet runs about two times slower than Graph Wavenet due to compute the dynamic global spatial correction layer by layer. To summarize, although DGLSTNet improves prediction performance, it also increases computation time and there is still room for further improvement.

TABLE 4: The computation time on the METR-LA datasets

Model	Computation Time	
	Training(s/epoch)	Inference(s/epoch)
DCRNN	320.13	37.64
Graph Wavenet	79.48	2.60
STSeq2Seq	159.12	20.69
DGLSTNet	162.37	9.46

V. CONCLUSION

We propose a novel spatial-temporal deep learning network called DGLSTNet to focus on network-wide multiple steps ahead traffic speed predicting. In the spatial dimension, a dynamic graph learning block is introduced for learning the dynamic sparse spatial correction in a global way. When construcing the global adjacency matrix, the model considers not only the similarities of the nodes, but also the underlying pairwise relationships between nodes. By integrating the pre-defined and the adaptively learned global adjacency matrices into graph convolution operation to capture both local and global spatial dependencies simultaneously. In the temporal dimension, a dynamic temporal module considering the short-term local neighboring and the long-term global trend correlations is proposed to effectively explicit informative temporal dependencies. Experiments on two real-world datasets showed that the predicting accurary of our model is better than existing models. In the further works, we plan to explore more complex spatial correlations to further improve the prediction accurary, in addition, we can apply our approach to other spatial-temporal forecasting, such as ride-hailing demand prediction.

REFERENCES

[1] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors, 17(4):818, 2017.

[2] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors, 17(7):1501, 2017.

[3] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM international Conference on Data Mining, pages 777–785. SIAM, 2017.

[4] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[5] Shengdong Du, Tianrui Li, Xun Gong, and Shi-Jinn Horng. A hybrid method for traffic flow forecasting using multimodal deep learning. arXiv preprint arXiv:1803.02099, 2018.

[6] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 5668–5675, 2019.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in neural information processing systems, pages 3844–3852, 2016.

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[9] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875, 2017.

[10] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121, 2019.

[11] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, 2017.

[12] Zhengchao Zhang, Meng Li, Xi Lin, Yinhai Wang, and Fang He. Multistep speed prediction on traffic networks: A graph convolutional sequence-to-sequence learning approach with attention mechanism. arXiv preprint arXiv:1810.10237, 2018.

[13] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 2019.

[14] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Transactions on Intelligent Transportation Systems, 2019.

[15] Xinglei Wang, Xuefeng Guan, Jun Cao, Na Zhang, and Huayi Wu. Forecast network-wide traffic states for multiple steps ahead: A deep learning approach considering dynamic non-local spatial correlation and non-stationary temporal dependency. arXiv preprint arXiv:2004.02391, 2020.

[16] Marco Lippi, Matteo Bertini, and Paolo Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Transactions on Intelligent Transportation Systems, 14(2):871–882, 2013.

[17] Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. Modeling financial time series with S-PLUS®, pages 385–429, 2006.

[18] Steven I-Jy Chien and Chandra Mouly Kuchipudi. Dynamic travel time prediction with real-time and historic data. Journal of transportation engineering, 129(6):608–616, 2003.

[19] Jian Wang, Wei Deng, and Yuntao Guo. New bayesian combination method for short-term traffic flow forecasting. Transportation Research Part C: Emerging Technologies, 43:79–94, 2014.

[20] Yan Qi and Sherif Ishak. A hidden markov model for short term prediction of traffic conditions on freeways. Transportation Research Part C: Emerging Technologies, 43:95–111, 2014.

[21] Pinlong Cai, Yunpeng Wang, Guangquan Lu, Peng Chen, Chuan Ding, and Jianping Sun. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transportation Research Part C: Emerging Technologies, 62:21–34, 2016.

[22] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Transactions on Intelligent Transportation Systems, 15(5):2191–2201, 2014.

[23] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2):865–873, 2014.

[24] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies, 54:187–197, 2015.

[25] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143, 2018.

[26] Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pages 324–328. IEEE, 2016.

[27] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2020.

[28] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 922–929, 2019.

[29] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 1234–1241, 2020.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[31] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention in transformer models. arXiv preprint arXiv:2005.00743, 2020.

[32] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. 2016. cite arxiv:1609.03499.

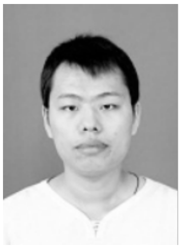
[33] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. arXiv preprint arXiv:1801.03226, 2018.

[34] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. Hierarchical graph pooling with structure learning. arXiv preprint arXiv:1911.05954, 2019.

[35] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

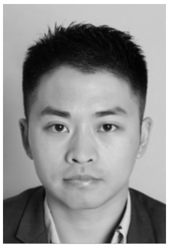
[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.



DONG FENG received the M.S. degree from the School of information and computer, Anhui Agricultural University, in 2013 . He is currently pursuing the PH.D degree with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, and also with the University of Science and Technology of China (USTC). His current research interests include data mining on the Internet of Vehicles, graph convolution networks, and deep learning.

ZHONGCHENG WU received the Ph.D. degree from Plasma Physical, Chinese Academy of Science(ASIPP), in 2001. From 2001 to 2004, he did his Postdoctorral research with the University of Science and Technology of China (USTC). He was a Visiting Professor research with the Computer Science Department, Hong Kong Baptist University, in 2005. He has been a Professor with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, and a Doctoral Supervisor with the University of Science and Technology of China and the University of Chinese Academy of Sciences, since 2008. He has published more than 140 papers in journals and international conference. His current research interests include standardization of the sensor interface, sensor technology, machine perception, pen computing and pen inferencem, and natural hunman-computer interaction.

JUN ZHANG received the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2020. He is currently an engineer with the Hefei Institutes of Physical Science, Chinese Academy of Sciences. His current research interests include the Internet of Things, machine learning, and patten recognition.



ZIHENG WU received the Ph.D. degree in University of science and Technology (USTC), Hefei, China, in 2017. Now he is a lectuer in AnHui University of Technology. His research interests include machine learning, artificial intelligence, grey systems theory, medical informatics, science engineering and intelligent control.