Dynamic Graph Convolution Network for Traffic Forecasting Based on Latent Network of Laplace Matrix Estimation

Kan Guo, Yongli Hu, *Member, IEEE*, Zhen Qian, Yanfeng Sun, *Member, IEEE*, Junbin Gao, *Member, IEEE*, and Baocai Yin, *Member, IEEE*

***Abstract*— Traffic forecasting is a challenging problem in the transportation research field as the complexity and non-stationary changing of the traffic data, thus the key to the issue is how to explore proper spatial and temporal characteristics. Based on this thought, many creative methods have been proposed, in which Graph Convolution Network (GCN) based methods have shown promising performance. However, these methods depend on the graph construction, which mainly uses the prior knowledge of the road network. Recently, some works realized the fact of the road network graph changing and tried to construct dynamic graphs for GCN, but they do not fully exploit the spatial and temporal properties of the traffic data in the graph construction. In this paper, we propose a novel dynamic graph convolution network for traffic forecasting, in which a latent network is introduced to extract spatial-temporal features for constructing the dynamic road network graph matrices adaptively. The proposed method is evaluated on several traffic datasets and the experimental results show that it outperforms the state of the art traffic forecasting methods. The website of the code is https://github.com/guokan987/DGCN.git.**

***Index Terms*— Dynamic graph convolution network, Laplace matrix latent network.**

I. Introduction

ESTABLISHING Intelligent Transportation System (ITS) is becoming a pivotal aspect of the modern transportation research field, in which traffic forecasting plays a critical role as it has extensive applications, such as optimizing the allocation of road usage, planing customers' travel route in advance, and guiding the road building, etc.

With the heavy usage of traffic detectors and sensors on the city road network, the modern traffic system accumulates enormous historical data. There exist rich information and regularity hidden in the big data produced by the dynamically changeable traffic system. Thus, many models based on the historical road network information are proposed and researched, in which the main research point is how to establish time series model and exploit spatial relation of road segment nodes through the novel methodology, e.g., the Kalman Filter models [1]–[3], the statistics [4]–[8], and the artificial intelligence [9], [10].

For a real-world transportation system, there are too many causes [11] that impact traffic forecasting, such as the nonlinear and non-stationary traffic data, weather, and incidents, etc., and it brings some difficulties on excavating the spatial and temporal features. With the rapid rise of research in artificial intelligence and the developing of various deep neural network models, many creative methods are proposed to explore complicated temporal-spatial characteristics of the traffic data, such as Space State NN (SSNN) [12], which was designed to seek a kind of temporal-spatial relation based on First Order Context (FOC) memory; Deep Spatio-Temporal Convolution Network (DSTCN) [13], which explores the spatial relation by Convolution Neural Network (CNN) and exploits the temporal information with Recurrent Neural Network (RNN). Although some models establish spatial-temporal relationships, they ignore the natural topology structure of the road network in space and even destroy this structure.

Recently, Graph Convolution Network (GCN) [14] was investigated to represent the relationship between the irregular traffic data as a graph. For example, the studies [15], [16] combined CNN or RNN with GCN to learn the spatial-temporal feature of traffic data. However, the performance of these works was degraded because of using a fixed and empirical graph, in which the dynamic properties of traffic data were not considered. Thus, a data-driven method [17] was proposed to optimize a parameterized global-temporal-sharing Laplace matrix in the network training phase, and it obtained richer space connections compared to the empirical one [15], [16]. But, this graph's Laplace matrix is still fixed in the prediction phase, which cannot capture the dynamic information of the graph to improve the forecasting accuracy. To exploit the complicated and non-stationary changing of traffic data,

Manuscript received November 17, 2019; revised May 30, 2020 and July 8, 2020; accepted August 13, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant U19B2039, Grant 61632006, Grant 61672071, Grant U1811463, Grant 61772048, Grant 61806014, and Grant 61906011; in part by the Beijing Natural Science Foundation under Grant 4172003, Grant 4184082, and Grant 4204086; in part by the Beijing Talents Project under Grant 2017A24, and in part by the Beijing Outstanding Young Scientists Projects under Grant BJJWZYJH01201910005018. The Associate Editor for this article was P. Wang. *(Corresponding author: Yongli Hu.)*

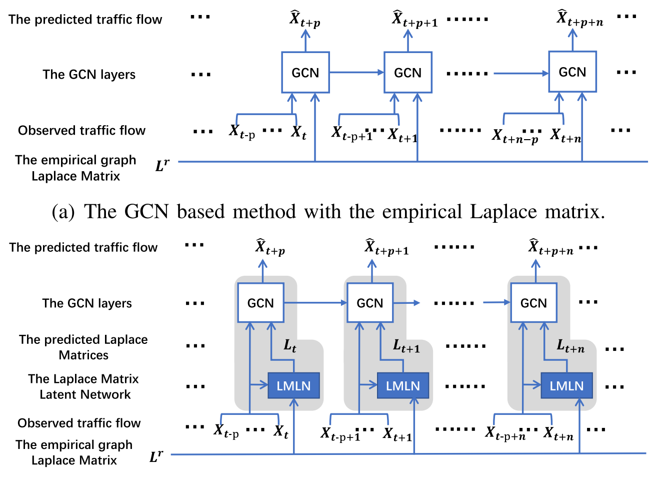
Kan Guo is with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: guokan@emails.bjut.edu.cn).

Yongli Hu, Yanfeng Sun, 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: huyongli@bjut.edu.cn; yfsun@bjut.edu.cn; ybc@bjut.edu.cn).

Zhen Qian is with the Civil and Environmental Engineering and H. John Heinz III College, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: seanqian@cmu.edu).

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

Digital Object Identifier 10.1109/TITS.2020.3019497



(a) The GCN based method with the empirical Laplace matrix.

(b) The proposed method based on the latent Laplace Matrix.

Fig. 1. The framework of the proposed method compared with the traditional GCN based method.

attention mechanism based Spatial-Temporal Graph Convolutional Network (ASTGCN) [18] was presented, in which a dynamic Laplace matrix of the graph was constructed at each input sequence data by the spatial attention mechanism [19]. However, the main limitation of ASTGCN is that it still utilized the empirical Laplace matrix [15], [16] as a mask matrix in the dynamic Laplace matrices of the road network, which could filter out some elements of the empirical Laplace matrix. Furthermore, ASTGCN ignored the inner temporal connection between Laplace matrices of the adjacent periods.

Thus, in this paper, we propose a novel dynamic graph convolution network for traffic forecasting, as shown in Fig.1(b). In contrast to the fixed and empirical Laplace matrix in the traditional GCN based methods, as shown in Fig.1(a), the proposed method introduces a Laplace Matrix Latent Network (LMLN) to adaptively represent the spatial-temporal relationship, then it feeds this relationship to GCN and forms a dynamic graph convolution network. Compared with the closely related work, ASTGCN, the proposed LMLN utilizes a latent spatial-temporal network to estimate a sequence of Laplace matrices. At last, the proposed network of a one-time slice, the gray block in Fig.1(b), will be detailed in Section 3. The main contributions of the paper are summarized as follows,

- A novel Dynamic Graph Convolution Network (DGCN) is proposed for traffic forecasting based on dynamic graph Laplace matrix;
- A latent network of graph Laplace matrix is proposed to represent the spatial-temporal connection of the traffic data adaptively;
- Traffic forecasting experiments are conducted on several real-world traffic datasets and the validity of our model is verified.

II. Preliminaries and Related Work

A. Traffic Forecasting

Traffic forecasting is to predict the future traffic flow or speed by using the current or past observed traffic data. The input of traffic forecasting is a temporal sequence data, so the traffic forecasting is a specific problem of time series analysis and utilises the moving window technique to implement forecasting. For convenience, we define the input data of the $i-th$ road node at t time as the $x_t^i \in \mathbf{R}^F$, in which the F is the feature size of x_t^i. For example, $F = 3$ represents the model will accept three different kinds of road traffic data such as traffic flow, average speed and so on. Then, the traffic data of the road network at time t is denoted by $X_t = (\ldots, x_t^i, \ldots), i = 1, \ldots, N$, where N is the number of road segments, and we will forecast the traffic data for the next T_1 times by using T_2 input data, i.e. the length of moving window is T_2. Thus, the input data is $(X_1, \ldots, X_t, \ldots, X_{T_2}) \in \mathbf{R}^{N \times T_2 \times F}$, and one usually predicts a kind of traffic data such as: traffic flow or speed, so the forecasting results is $(\hat{X}_{(T_2+1)}, \ldots, \hat{X}_{(T_2+T_1)})_{[:,:,j]} \in \mathbf{R}^{N \times T_1 \times 1}$ and the ground truth is $(X_{(T_2+1)}, \ldots, X_{(T_2+T_1)})_{[:,:,j]} \in \mathbf{R}^{N \times T_1 \times 1}$. To represent the graph structure of traffic network, we define graph as $G = (V, E, A)$, where $V \in \mathbf{R}^N$ is the road segment set, E is the edge set, and $A \in \mathbf{R}^{N \times N}$ is the adjacent matrix of G.

In the past decades, there are enormous traffic forecasting methods presented by researchers, and they can be classified as model-driven methods and data-driven methods. In respect of model-driven methods, DynaMIT [20] and DYNASMART-X [21] were proposed to conduct real-time traffic simulation system, and they are based on state estimators like Kalman Filters(KFs). Then, more advanced KFs state estimators were proposed, such as Extended Kalman Filters (EKFs) [1], which combined KFs with a stochastic macroscopic freeway network; the localized EKFs [3] and the probabilistic heterogeneous traffic data fusion based EKFs [22]. The data-driven methods begin at the statistical theory, and the simplest statistical method is Historical Average (HA), in which the mean of the historical traffic data is regarded as the predicted value in the future. Then, the classic linear regression method: Autoregressive Integrated Moving Average Model (ARIMA) [4] was proposed, and the seasonal ARIMA [5] was utilized on the highway datasets. With the research of machine learning, most non-linear methods were utilized to explore the relationship of the input traffic data, Wu *et al.* [8] introduced SVM [23] into the traffic short-time prediction method. Different from SVM, Yu and Chen [9] proposed a single hidden layer NN for traffic forecasting, and Lint *et al.* [12] presented SSNN to reveal the Spatio-temporal relation of traffic data. Recently, with the rapid development of artificial intelligence technology, researchers [24] began to deal with the traffic forecasting problem through deep learning methods. Most works are CNN or RNN based methods, such as LSTM [25], the Gated Recurrent Unit (GRU) [26], DSTCN [13] which uses residual convolution units model and considers the impact of weather and accidents, GeoMAN [27] which utilizes multi-layer Attention mechanism, etc.

B. Graph Convolution Network

Derived from the theory of graph spectrum, GCN can process irregular graph data in the spectral domain [14]. For the input data x with graph structure G, the GCN operator

$g_\theta \star G$ can be given as follows,

$$g_\theta \star G(x) = g_\theta(L)x = U g_\theta(\Lambda) U^T x \tag{1}$$

where g_θ represents the trainable parameter of GCN, $L = U\Lambda U^T$, U is the Fourier basis of G, and $\Lambda = diag([\lambda_1, \ldots, \lambda_N]) \in \mathbf{R}^{N\times N}$. Due to the computational complexity of decomposition of Laplace matrix L, Fast-GCN [28] was proposed as follows,

$$g_\theta \star G(x) = g_\theta(L)x = \sum_{m=0}^{M-1} \theta_m C_m(\tilde{L})x \tag{2}$$

where $\tilde{L} = \frac{2}{\lambda_{max}} L - I_N$, $I_N \in \mathbf{R}^{N\times N}$ is an identity matrix, λ_{max} is the max eigenvalue of L, and M is the order of the Chebyshev Polynomials: $C_m = 2\tilde{L}C_{m-1} - C_{m-2}$ and $C_1 = \tilde{L}, C_0 = I_N$.

Up to now, GCN has been successfully used in many applications, such as Semi-Supervised Classification [29] and Spatial-Temporal Graph Convolution Network (STGCN) [30] for action recognition. In the traffic forecasting field, there are also many GCN based works, such as Graph Convolutional Recurrent Networks(GCRN) [31], Gated Spatial Temporal Graph Convolution Network (Gated-STGCN) [15] and ASTGCN [18] which utilizes spatial attention block to learn dynamic graph matrices, and we define these dynamic matrices as S'.

C. Attention Mechanism

Due to the capacity of learning wide range dependency among related signals, attention mechanism was introduced into Auto-encode, deep convolution, and generation network, and it had achieved successful applications [32], [33]. Then the self-attention was proposed to reveal the interdependency of input signals or features. Compared with the self-attention mechanism [33], another attention method was proposed to calculate the relationship of different axis of feature dependently by a weighted matrix [19]. Recently, Graph Attention Network (GAT) [34] firstly utilized the attention mechanism and Neural Network (NN) to adaptively learn the dynamic adjacent matrix of the graph and obtain high performance in the semi-supervised tasks.

III. Methodology

To implement the proposed DGCN with the framework in Fig.1(b), we give one unit of the network at a certain time concretely, as shown in Fig.2. For exploiting the effect of periodic time data on the forecasting task, different from other works which only use recent data, we sampled three different periods consisting our model's input data, as shown in Fig.2, the recent-period data X_{T_h}, the daily-period data X_{T_d} which is the sampled data at the same forecasting moment in the past few days, the weekly-period data X_{T_w} which is the sampled data at the same forecasting moment in the past few weeks.

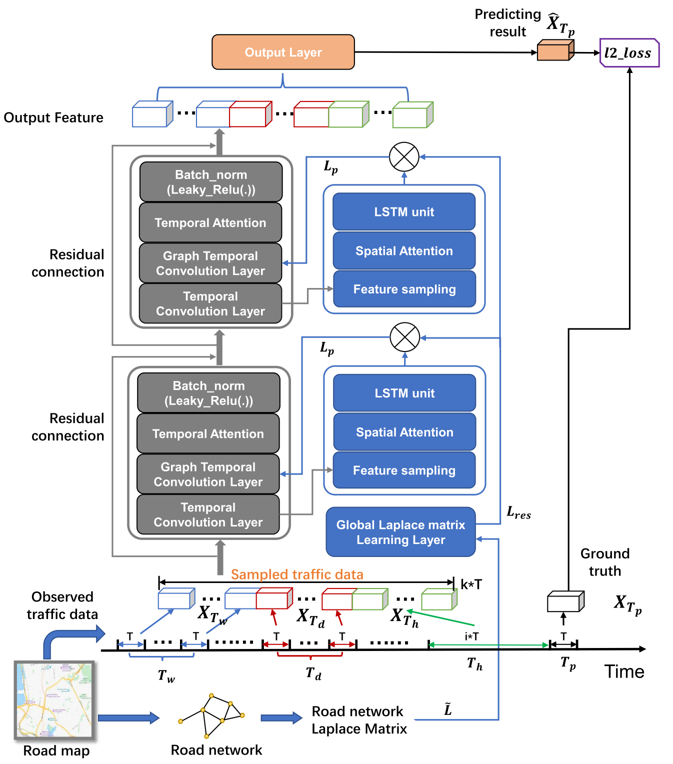


Fig. 2. The proposed DGCN based on LMLN and Spatial-Temporal Unit, where ⊗ is the Hadamard Product, $\tilde{L}$ is scaled road network's Laplace matrix, L_{res} is the global Laplace matrix, and L_p is the estimated Laplace matrix from the current traffic data.

The extracting spatial-temporal feature structure of the proposed DGCN has two main components: the Laplace matrix latent network component (the right part with the blue background of Fig.2) for estimating the Laplace matrix of Observed traffic data; the GCN based traffic forecasting component (the left part with the gray background of Fig.2) for capturing the spatial and temporal feature of Observed traffic data by graph convolution. In the following, we will describe these two components in detail.

A. Laplace Matrix Latent Network

In LMLN, the scaled road network's Laplace matrix $\tilde{L}$ is firstly processed by the global Laplace matrix learning Layer. Then the output global Laplace matrix is transferred to several Laplace matrix prediction units. At last, the dynamic Laplace matrix L_p was transmitted to Graph Temporal Convolution Layer. As shown in Fig.3, the Laplace matrix prediction unit has three blocks: the feature sampling, the spatial attention, and LSTM unit. The main components and their blocks are presented as follows.

1) Global Laplace Matrix Learning Layer: The global Laplace matrix learning layer has a similar function as the parameterized global-sharing Laplace matrix in [17], but it has a different construction manner. In [17], the authors utilized the mask method to generate the new Laplace matrix by $L_{Mask} = L_{par} * L^r$, where $L_{par} \in \mathbf{R}^{N\times N}$ is a trainable parameter matrix, $*$ is Hadamard Product, and $L^r = \prod_1^r L + I_N \in \mathbf{R}^{N\times N}$ is the $r-$hop of L. However, L_{Mask} suffers from the flaw that the connections in the location of zero values in the L^r are omitted. For this purpose, we design a scaled 1-hop residual global Laplace matrix as

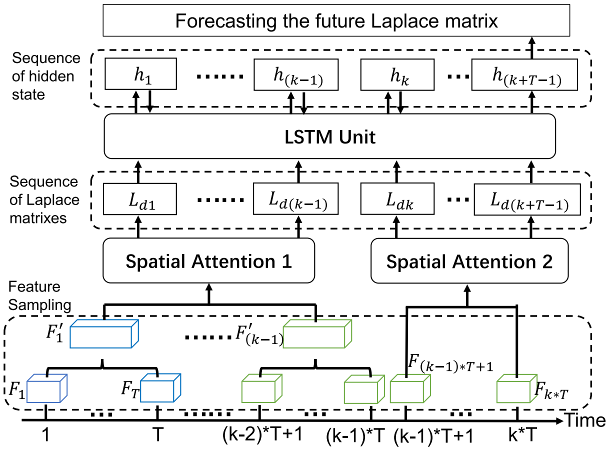

Forecasting the future Laplace matrix
Sequence of hidden state
h_1 ...... $h_{(k-1)}$ h_k ... $h_{(k+T-1)}$
LSTM Unit
Sequence of Laplace matrixes
L_{d1} ...... $L_{d(k-1)}$ L_{dk} ... $L_{d(k+T-1)}$
Spatial Attention 1
Spatial Attention 2
Feature Sampling
F_1' ...... $F_{(k-1)}'$
$F_{(k-1)*T+1}$
F_1 F_T F_{k*T}
1 T (k-2)*T+1 (k-1)*T (k-1)*T+1 k*T
Time

Fig. 3. The diagram of Laplace Matrix Prediction Unit.

follow,

$$\begin{aligned} L_{res1} &= L_{par} + \tilde{L} \\ D_{res\,ii} &= \sum_{j} L_{res1\,ij}, \quad i,j = 1,\dots,N \\ D_{res} &= diag(1/(D_{res\,ii} + 0.0001)), \quad i = 1,\dots,N \\ L_{res} &= D_{res} L_{res1} \end{aligned} \tag{3}$$

where L_{res1} is a parameterized Laplace matrix, $D_{res\,ii}$ is the degree of node i, D_{res} is the inverse matrix of degree matrix with adding 0.0001 to avoid NaN problem, and L_{res} is the normalized global Laplace matrix.

2) Feature Sampling: The input features are contacted by the recent, daily-period, and weekly-period observed traffic data. If the original data are used as the input, there will be too much time and space complexity when the time length $k * T$ increases. Thus, we propose a feature sampling scheme to reduce the data dimension of the traffic feature according to the importance of different time interval, i.e. the recent traffic feature is more important than the other time-period features. Here, we fuse every T length features into one new feature except the recent T features, as shown in Fig.3. The changing of dimension for the sampling features can be represented by $(F_1, \dots, F_t, \dots, F_{(k-1)*T}) \in \mathbf{R}^{N\times(k-1)*T\times F} \rightarrow (F_1', \dots, F_t', \dots, F_{(k-1)}') \in \mathbf{R}^{N\times(k-1)\times T*F}$. As the result, we have the new input feature for the Laplace matrix prediction unit in form of $(F_1', \dots, F_t', \dots, F_{(k-1)}', F_{(k-1)*T+1}, \dots, F_{(k)*T}) \in \mathbf{R}^{N\times(k+T-1)\times F}$. Then the first $k-1$ and the last T features will transfer to the two different spatial attentions respectively in Fig.3.

3) Spatial Attention: To construct the spatial relationship of the road network at every moment, we adopt the attention mechanism [33] to estimate the current period's adjacent matrix $\vec{L}_d$ of the road network. So, if the input feature of the attention mechanism is the feature $\vec{F}_{(1:\tau)} = (F_1, \dots, F_\tau)$, as shown in the Fig.4, the calculation of the the attention process is given by,

$$\begin{aligned} \vec{\tilde{L}}_d &= W_1(\vec{F}_{(1:\tau)})(W_2(\vec{F}_{(1:\tau)}))^{Tr} \\ \vec{L}_d &= Sigmoid(\vec{\tilde{L}}_d) \end{aligned} \tag{4}$$

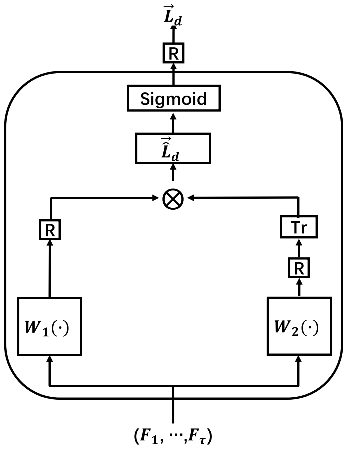


Fig. 4. The structure of spatial attention mechanism, the R is Reshape, the Tr is Transpose and the $\otimes$ is the Matrix inner product.

where $W_1(\cdot)$ and $W_2(\cdot)$ are embedding functions and Tr is the matrix transpose. We use the matrix inner product as the estimate method of an adjacent matrix of the road network. To explore more relationship between nodes, we adopt multi-head attention structure, so we can obtain K dynamic matrices $\vec{L}_d^i, i = 1, \dots, K$ and get the mean of them.

In our model, owing to the input futures being sampled as describing above, one input feature is $(F_1', \dots, F_t', \dots, F_{(k-1)}')$ which produces the sequence of adjacent matrices $(L_{d1}, \dots, L_{d(k-1)}) \in \mathbf{R}^{(k-1)\times N\times N}$, and the other features are $(F_{(k-1)*T+1}, \dots, F_{(k)*T})$ which produces the other sequence of adjacent matrices $(L_{dk}, \dots, L_{d(k+T-1)}) \in \mathbf{R}^{(T)\times N\times N}$. At last, we combine these two sequences as the output of spatial attention $\vec{L}_d = (L_{d1}, \dots, L_{d(k-1)}, L_{dk}, \dots, L_{d(k+T-1)}) \in \mathbf{R}^{(k+T-1)\times N\times N}$.

4) LSTM Unit: To explore the inner relation between the sequence of the adjacent matrices $\vec{L}_d$, we adopt LSTM to learn the temporal correlation. The LSTM unit can be represented as follows,

$$\begin{aligned} f_t &= \sigma\left(W_f(\left[h_{t-1}, L_{dt}\right] + b_f)\right) \\ i_t &= \sigma\left(W_i(\left[h_{t-1}, L_{dt}\right] + b_i)\right) \\ \tilde{C}_t &= \tanh\left(W_C(\left[h_{t-1}, L_{dt}\right] + b_C)\right) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma\left(W_o(\left[h_{t-1}, L_{dt}\right] + b_o)\right) \\ h_t &= o_t * \tanh\left(C_t\right) \end{aligned} \tag{5}$$

where $\sigma = sigmoid$, $tanh$ are activation functions, $L_{dt} \in \mathbf{R}^{N\times N}, t = 1, \dots, k+T-1$ is the input adjacent matrices, $h_{t-1}, h_t \in \mathbf{R}^{N\times N}$ are hidden features, thus $\left[h_{t-1}, L_{dt}\right] \in \mathbf{R}^{N\times 2N}$. f_t, i_t, o_t are forget gate, updating gate and output gate. $C_{t-1}, C_t \in \mathbf{R}^{N\times N}$ are the LSTM unit states, $W_f(\cdot), W_i(\cdot), W_o(\cdot), W_C(\cdot)$ are Linear embedding functions, in which their parameters' size is $\mathbf{R}^{2N\times N}$, and $b_f, b_i, b_o, b_C \in \mathbf{R}^N$ are bias.

From LSTM, we get a future adjacent matrix $h_{k+T-1} \in \mathbf{R}^{N\times N}$. Combining with the global Laplace matrix L_{res}, the output of the Laplace matrix prediction network L_p can

Algorithm 1 LMLN

```
Input:
    The current input feature $\vec{F}_{(1:k*T)} = F_1,\dots,F_t,\dots,F_{k*T} \in \mathbf{R}^{N\times k*T\times F}$;
Update $L_{res}$ by (3);
Get the sampled features from $\vec{F}_{(1:k*T)}$: $(F'_1,\dots,F'_t,\dots,F'_{(k-1)},F_{(k-1)*T+1},\dots,F_{(k)*T}) = (\vec{F'}_{(1:(k-1))},\vec{F}_{(((k-1)*T+1):k*T)})$;
Get $(L_{d1},\dots,L_{d(k-1)})$ from $\vec{F'}_{(1:(k-1))}$ by (4);
Get $(L_{dk},\dots,L_{d(k+T-1)})$ from $\vec{F}_{(((k-1)*T+1):k*T)}$ by (4);
$\vec{L}_d = (L_{d1},\dots,L_{d(k-1)},L_{dk},\dots,L_{d(k+T-1)})$;
for $t$-th $L_{dt}$ in $\vec{L}_d, t = 0,\dots,k+T-1$ do
    Get $h_t$ by (5)
end for
Get $h_1,\dots,h_{k+T-1}$;
Get $L_p$ by (6)
return $L_p$
```

be obtained by,

$$\begin{aligned} L_d &= h_{k+T-1} \\ L_p &= L_d * L_{res} \end{aligned} \tag{6}$$

$L_p \in \mathbf{R}^{N\times N}$ will be transferred to graph temporal convolution layer of GCN component.

From the above, the algorithm of LMLN can be summarized as follow:

B. GCN Based Traffic Forecasting

As shown in Fig.2, GCN Based Traffic Forecasting component has the following four blocks, which are contracted to extract the Spatial-Temporal feature of the input traffic data.

1) Temporal Convolution Layer (TCL): This temporal convolution layer is designed to extract the high-dimensional local temporal information from the original traffic data. The temporal convolution on a segment of the traffic data $\vec{X}_{(1:k*T)} = (X_1,\dots\dots,X_t,\dots\dots,X_{k*T}) \in \mathbf{R}^{N\times k*T\times F}$ can be represented as follows,

$$TC = \Phi \star \vec{X}_{(1:k*T)} = Conv_{1\times t_s}(\vec{X}_{(1:k*T)}) \tag{7}$$

where $Conv_{1\times t_s}$ represents the 2-D convolution operator and its kernel size is $1 \times t_s$.

2) Graph Temporal Convolution Layer (GTCL): Usually, in the traffic forecasting domain, this layer can be realized based on the GCN [15], [18], for example, one can stack GCN and TCL as a Spatial-Temporal Block to abstract the space-time feature TC from the output of TCL as follow:

$$\begin{aligned} \vec{Fea}1 &= g_\theta \star G(TC) \\ \vec{Fea}2 &= \Phi \star Relu(\vec{Fea}1) \end{aligned} \tag{8}$$

However, there are too many computations in the above model. Thus, we propose to integrate these two functions as a single GTCL by replacing the GCN operation g_θ with Φ:

$$\Phi \star G(TC) = \sum_{m=0}^{M-1} \Phi_m \star (C_m(\tilde{L})TC) \tag{9}$$

Algorithm 2 GCN Based Traffic Forecasting Component

```
Input:
    The current input feature $\vec{X}_{(1:k*T)} = (X_1,\dots\dots,X_t,\dots\dots,X_{k*T}) \in \mathbf{R}^{N\times k*T\times F}$;
$TC = \Phi \star \vec{X}_{(1:k*T)}$;
Get dynamic Laplace matrix $L_p$ from $TC$ by Algorithm.1;
$GC = \Phi_{gate} \star L_p(TC)$;
$TA = Tatt(GC)$;
$\vec{F}^1_{(1:k*T)} = Bathc\_norm(Leaky\_Relu(TA))$;
return $\vec{F}^1_{(1:k*T)}$
```

Additionally, we further design a gate mechanism [35] to explore the local temporal feature as follows,

$$\begin{aligned} (\vec{\beta}_1,\vec{\beta}_2) &= split(\Phi \star G(TC)) \\ \Phi_{gate} \star G(TC) &= sigmoid(\vec{\beta}_1) * Leaky\_Relu(\vec{\beta}_2) \end{aligned} \tag{10}$$

where $split$ represents the operator of equally dividing the input feature; $sigmoid$ and $Leaky\_Relu$ are activation functions. So we obtain the final result of GTCL based on the dynamic Laplace matrix L_p as follows,

$$\begin{aligned} (\vec{\beta}_1,\vec{\beta}_2) &= split(\Phi \star L_p(TC)) \\ GC &= \Phi_{gate} \star L_p(TC) \\ &= sigmoid(\vec{\beta}_1) * Leaky\_Relu(\vec{\beta}_2) \end{aligned} \tag{11}$$

3) Temporal Attention: Except for TCL and GTCL, we also need a method to explore the long-range time relation, so we adopt the Temporal Attention in [19] to adaptively capture the large scale temporal correlation of traffic data.

$$\begin{aligned} E &= V_e\sigma((GC)^{Tr}U_1)U_2((GC)U_3)^{Tr} + b_e) \\ E'_{i,j} &= \frac{exp(E_{i,j} + Mas)}{\sum_{j=1}^{k*T} exp(E_{i,j} + Mas)} \end{aligned} \tag{12}$$

where $V_e, b_e \in \mathbf{R}^{k*T\times k*T}$, $U_1 \in \mathbf{R}^N$, $U_2 \in \mathbf{R}^{F\times N}$, $U_3 \in \mathbf{R}^F$ are trainable parameters, and $Mas \in \mathbf{R}^{k*T\times k*T}$ is a mask matrix for keeping the dependence between the discontinuous periods of time and making the value of relationship $E'_{i,j} \in \mathbf{R}^{k*T\times k*T}$ between discontinuous time periods as zero. Thus, in GCN Based Traffic Forecasting, we denote the temporal attention as follow,

$$\begin{aligned} TA &= Tatt(GC) \\ &= E'GC \end{aligned} \tag{13}$$

4) Batch_norm(Leaky_Relu(.)): This part is the activation function of GCN Based Traffic Forecasting, and we utilize $Batch\_norm(Leaky\_Relu(.))$ to activate the output feature of the Temporal Attention as follows,

$$\vec{F}^1_{(1:k*T)} = Bathc\_norm(Leaky\_Relu(TA)) \tag{14}$$

From the above, we summary the algorithm of **GCN based traffic forecasting component** as the following Algorithm.2.

5) Loss Function: After the above DGCN processing, we obtain the output spatial and temporal features. From this, we establish the output layer and construct the model's loss function. As shown in Fig.2, the recent data is contiguous and its length is $i * T$, and the daily-period and weekly-period data are sampled and cut up by $k - i$ unit which length is T, so this $k - i$ input unit is discontinued in the time axis. Therefore, we design a special output layer: $Conv_{1\times i}$ to deal with the recent data, $Conv_{1\times 1}$ to reflect the $k - i$ feature unit dependently in the daily-period and weekly-period. Finally, we sum all convolution output as our model's $prediction$ and adopt the $l2\_loss$ to measure the difference between the prediction and its ground truth to obtain the model's loss function as follows,

$$loss = l2\_loss(prediction, truth) \tag{15}$$

where $l2\_loss(prediction, truth)$ denotes the loss of the model's prediction and truth.

IV. Experiment Settings and Results Analysis

A. Experiment Settings

1) Datasets: In our experiments, we use three real-world traffic datasets: PeMSD4, PeMSD8, and PHILADELPHIA for evaluating the proposed method. The former two are collected from California highway by the Caltrans Performance Measurement System(PeMS) [36] in the rate of one sampling every 30 seconds [18]. Here, we resample these datasets with one sample per 5 minutes. PeMSD4 has traffic data of 307 road segments recorded from January to February in 2018. PeMSD8 has traffic data of 170 road segments collected from July to August in 2016. Both of them contain three traffic measurements, i.e. traffic flow, average speed, and road occupancy, thus $F = 3$. In our model, we will forecast traffic flow as output for the two datasets. PHILADELPHIA [37] is a speed dataset ($F = 1$) sampled every 5 minutes at the city center of Philadelphia in 2016 summer. It has 397 road segments with 35 highway segments, so it has urban traffic features different from the former two datasets. As there is one sample per 5 minutes, i.e. each hour has 12 data samples for all the datasets, we set $T = 12$. We also divide three datasets into three parts: training-set, validation-set, and test-set with the ratio of 60%, 20%, and 20% in the time direction. The details of three parts of the three datasets as shown in Table.I. In our experiments, we select the best model's parameter on the validation-set and evaluate the proposed model on the test-set. In addition, the data samples of each road segment are normalized by $x' = \frac{x - mean(x)}{std(x)}$.

2) Parameters Setting: The proposed model is implemented by Pytorch 1.2.0 on a virtual workstation with a 24G memory Nvidia RTX Titan GPU. We set the order of Chebyshev polynomial $M = 3$ in the GTCL according to the previous works [18], [29]. The size of the temporal kernel $t_s = 3$. The head-number of multi-head attention $K = 4$. Similar to ASTGCN, the size of the output feature GCN Based Traffic Forecasting is 64. The lengths of the three traffic data segments are also set same as that of ASTGCN, i.e. $T_h = 24$, $T_d = 12$, $T_w = 24$. The forecasting time interval $T_p = 12$, i.e. one hour in the future. Thus, we adopt 60 samples to train and predict 12 samples in the next one hour. In this paper, the batch size is 8, and we use $l2\_loss$ as our model's loss function, then Adam Optimization is utilized. The original learning rate is 0.0005 with decay rates 0.92 every epoch. We train 40 epochs in the training phase.

TABLE I

The Details of Data-Set Segmentation on Three Datasets

Dataset	training-set		validation-set		test-set	
	Ratio(%)	Amount	Ratio(%)	Amount	Ratio(%)	Amount
PeMSD8	60	8287	20	2763	20	2763
PeMSD4	60	7769	20	2590	20	2590
PHILADELPHIA	60	12089	20	4030	20	4030

3) Comparison Methods: The proposed method is compared with seven traffic forecasting methods: HA, ARIMA, LSTM, GRU, GCRN, Gated-STGCN, ASTGCN. In these methods, except for ASTGCN using the sampled traffic data as input feature, the other methods use recent data for forecasting. For estimating the impact of different input, we also make a version of our model using the recent data, denoted by DGCN_R. To further evaluate the efficiency of different Laplace matrix for GCN, especially GAT [34], we compare our method with four GCN based methods: ASTGCN [18], in which the attention Laplace matrix is used; DGCN_Mask, a revised version of our model using the mask Laplace matrix; DGCN_Res, a revised version of our model using the residual Laplace matrix; and DGCN_GAT, a method by replacing the spatial feature layer-GTCL of our model with GAT and so being comparable with the others.

The performances of all methods are measured by two metrics, i.e. Root Mean Square Error (RMSE) and Mean Absolute Error(MAE), defined as follow:

$$\begin{aligned} MAE &= \frac{\sum_{i=1}^{T_p}\sum_{j=1}^{N}|(X_{ij} - \hat{X}_{ij})|}{T_p * N} \\ RMSE &= \sqrt{\frac{\sum_{i=1}^{T_p}\sum_{j=1}^{N}(X_{ij} - \hat{X}_{ij})^2}{T_p * N}} \end{aligned} \tag{16}$$

B. Results

The average one-hour traffic forecasting accuracies on PeMSD4, PeMSD8, PHILADELPHIA of different methods are shown in Table.II. It is shown that our proposed DGCN has the best performance compared with other methods in all metrics. The traditional HA and ARIMA have the worst results. Compared with the two baseline methods, the temporal neural network based methods, LSTM and GRU have better results. Then, the GCN based methods, GCRN, and Gated-STGCN have obvious improvements in contrast to the above methods, which should be benefited from the abilities to capture spatial and temporal features of traffic data. The close related method of our method, ASTGCN, has the third position in all metrics, which is explained that its adaptive and dynamic graph Laplace matrix by the mask operator on the empirical Laplace matrix brings the results. Compared with other methods, our method achieves at least 8 percent on two

TABLE II
THE TRAFFIC FORECASTING RESULTS OF DIFFERENT METHODS ON PEMSD4, PEMSD8, AND PHILADELPHIA

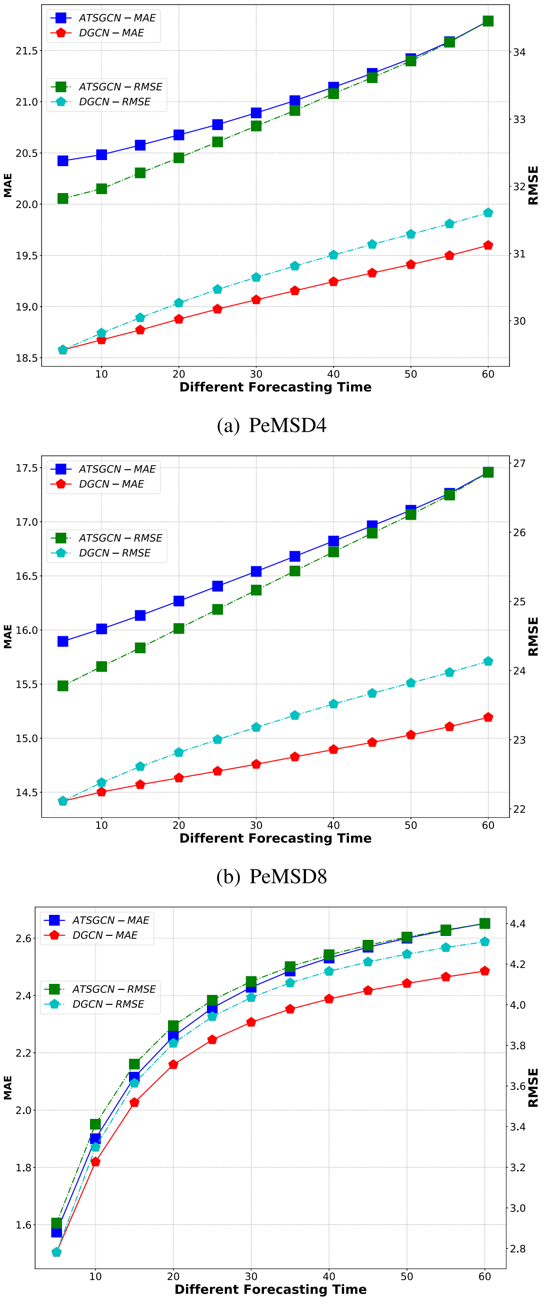
Method	PeMSD4		PeMSD8		PHILADELPHIA	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	107.62	132.18	89.11	112.21	3.51	5.59
ARIMA	30.66	46.25	24.64	37.18	2.78	4.63
LSTM	29.10	43.75	22.72	34.85	2.84	4.76
GRU	29.28	43.80	23.76	36.40	2.86	4.78
GCRN	25.62	38.24	19.99	29.84	2.78	4.62
Gated-STGCN	27.50	40.85	21.49	32.40	2.79	4.66
ASTGCN	21.79	34.46	17.46	26.87	2.65	4.40
DGCN_R(ours)	**20.55**	**32.14**	**16.19**	**25.04**	**2.56**	**4.31**
DGCN(ours)	**19.60**	**31.61**	**15.19**	**24.13**	**2.49**	**4.31**

TABLE III
THE TRAFFIC FORECASTING MEAN RESULTS OF FOUR METHODS ON PEMSD4 AND PEMSD8

Method	PeMSD4/PeMSD8			
	MAE	RMSE	training time(s/epoch)	test time(s/sample)
ASTGCN	21.79/17.46	34.46/26.87	42.85/28.16	0.0018/0.0010
DGCN_Mask	20.81/16.59	32.80/25.73	54.17/30.41	0.0020/0.0010
DGCN_Res	20.70/15.89	32.99/24.93	55.18/30.47	0.0020/0.0011
DGCN_GAT	21.16/16.57	34.27/26.05	50.36/29.89	0.0019/0.0011
DGCN(ours)	19.60/15.19	31.61/24.13	122.44/52.77	0.0056/0.0022

highway datasets(PeMSD4 and PeMSD8) and 5 percent on city road dataset(PHILADELPHIA) compared with ASTGCN, which is considered a significant improvement. We think this is the result of our latent network of Laplace matrix having the advantage of fully revealing the intrinsic relations in the traffic data. Especially, DGCN_R obtain the second position in all metrics, this further confirms that LMLN can exploit a better dynamic spatial relation of the road network. Compared with DGCN, it also verifies that the input of more periodic data can obtain better prediction accuracy.

To further analyze the results of different forecasting time, we compare the forecasting results of ASTGCN and DGCN on the 12 different future times. As shown in Fig.5, In all datasets, DGCN is better than ASTGCN in all forecasting time sizes on both the two metrics, and this further shows that our method is robust on different prediction tasks and different forecasting time.



(a) PeMSD4

(b) PeMSD8

(c) PHILADELPHIA

Fig. 5. Results of ASTGCN and DGCN for different forecasting time.

C. The Impact of Different Laplace Matrices for GCN

For GCN based methods, it is critical to construct a proper graph Laplace matrix for the graph convolution operator. The ideal Laplace matrix should reveal the intrinsic relation of the traffic data. To evaluate the impact of different Laplace matrices, we do traffic forecasting experiments by the GCN based methods with different Laplace matrices, ASTGCN, DGCN_Mask, DGCN_Res, DGCN_GAT, and our DGCN method. The results are shown in Table.III. Except for the forecasting accuracies, the training time (one epoch's training time) and the test time(the average test time of one sample in test-set) are reported in Table.III. It is shown that the two mask methods, i.e. ASTGCN and DGCN_Mask get the worse results. It is explained that the mask Laplace matrix has a limitation of representing the complicated spatial correlation of the traffic data, compared to ASTGCN, DGCN_Mask get better results, this further verifies the effectiveness of our model despite without using the dynamic matrices. DGCN_Res outperforms ASTGCN and DGCN_Mask inaccuracies with little increasing of training time. This owns to the replacement of the Global-optimized-residual Laplace matrix with the empirical Laplace matrix. It also verifies that the global Laplace matrix learning layer in our method is necessary. Our DGCN model obtains the best accuracies compared with the other four methods. It is conducted that LMLN can construct a valid dynamic graph Laplace matrices sequence for the GCN network. Particularly, our method outperforms DGCN_GAT

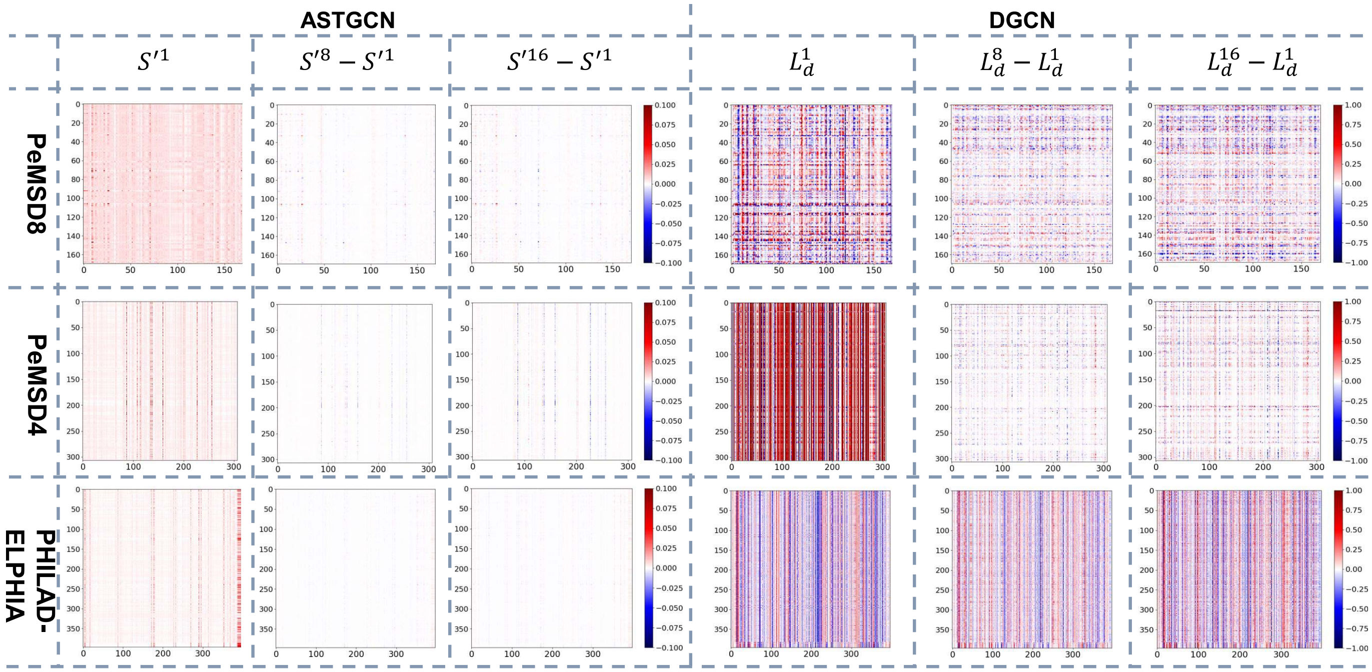


Fig. 6. The estimated Laplace matrices and its residual matrices of ASTGCN and DGCN on three datasets in three certain time intervals.

TABLE IV
THE VARIANCES OF THE DYNAMIC LAPLACE MATRICES AND THE TRAFFIC DATA OF THE THREE DATASETS

Datasets	ASTGCN			DGCN			Traffic Data		
	t_1	$t_8 - t_1$	$t_{16} - t_1$	t_1	$t_8 - t_1$	$t_{16} - t_1$	t_1	$t_8 - t_1$	$t_{16} - t_1$
PeMSD8	1.31×10^{-5}	1.67×10^{-6}	2.79×10^{-6}	5.56×10^{-2}	4.68×10^{-2}	4.63×10^{-2}	6.88×10^{-2}	9.39×10^{-2}	1.08×10^{-1}
PeMSD4	4.60×10^{-5}	1.31×10^{-6}	6.16×10^{-6}	2.78×10^{-1}	2.77×10^{-1}	2.73×10^{-1}	6.52×10^{-2}	1.33×10^{-1}	1.19×10^{-1}
PHILADELPHIA	2.52×10^{-5}	6.33×10^{-7}	9.94×10^{-7}	5.57×10^{-2}	4.12×10^{-2}	4.01×10^{-2}	1.71×10^{-1}	2.64×10^{-1}	3.80×10^{-1}

in both MAE and RMSE, which means the GTCL layer is better than the original graph attention mechanism. However, due to the added complexity of the Laplace matrix latent network, the training time and the test time of our method increase compared with the other methods. But the difference is acceptable. As shown in the last column in Table.III, the impact of the difference of test time is negligible for general real time application.

To intuitively show the dynamic Laplace matrices constructed by our method, we draw the residual matrices of the dynamic Laplace matrices sequence and compare it with that of ASTGCN. As shown in Fig.6, we can see that the residual matrices become more obvious with time interval increasing on all datasets. It is reasonable because the difference in the traffic data will become larger when the time interval increasing. This also verified that the proposed method can capture the changing of the spatial and temporal correlation of the traffic data. For the close related method, ASTGCN, though it also estimates the Laplace matrix adaptively by an attention mechanism, the demonstration of its residual matrices sequence does not have obvious changes, which illustrates the ability of its representing of the dynamic Laplace matrices is limited. Thus, DGCN can further explore the underlying changing of Laplace matrices compared to ASTGCN.

At last, for further exploiting the difference on dynamic Laplace matrices between ASTGCN and DGCN, we first calculate the variances of each row in these matrices, which represents the variety of the magnitude linking to other nodes of the current node, and a bigger value means there are more different patterns of links for the current node, i.e. the links are non-uniform and selective. Then the mean of the variances of all rows is calculated and shown in Table.IV, which reflects the whole variety of links of all nodes in the matrices. However, our latent network aims to reveal the variety of Laplace matrices caused by the observed dynamic traffic data. So we also compute the corresponding variances of the original traffic data (we construct the road nodes' correlation matrices as dynamic graph matrices), as shown in the last three columns in Table.IV. It is indicated that the variances of the traffic data at the beginning time (t_1) are quite large for the three datasets, and even its resident matrices at $t_8 - t_1$, $t_{16} - t_1$ are also considerable, which means the traffic data have a big variety for different nodes at different time. Under this situation, the variances of our dynamic Laplace matrices are consistent with that of traffic data and larger than that of ASTGCN, as shown in Table.IV. This further confirms that our dynamic Laplace matrices are more powerful to reveal the latent spatial-temporal relationship among the observed traffic dynamic data from different road segments than that of ASTGCN, which finally contributes to the superior traffic forecasting performance of our DGCN method.

V. Conclusion

In this paper, a novel graph convolution network, namely DGCN, was proposed to forecast traffic data. Different from most of the current GCN based methods, which generally used empirical graph Laplace matrix in graph convolution, we propose a latent network to estimate the dynamic Laplace matrix adaptively, which is verified with good ability to extract spatial-temporal correlation of the traffic data. The proposed method is evaluated on three real-world traffic data. The experimental results show that the proposed method outperforms the state of the art traffic forecasting methods. However, considering the complexity of DGCN, especially LMLN, if we want to forecast large scale road network, a fast and efficient method to dynamically abstract Laplace matrices is a critical and interesting work in the future.

References

[1] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, Feb. 2005.

[2] Y. Wang, M. Papageorgiou, and A. Messmer, "RENAISSANCE—A unified macroscopic model-based approach to real-time freeway network traffic surveillance," *Transp. Res. C, Emerg. Technol.*, vol. 14, no. 3, pp. 190–212, Jun. 2006.

[3] C. P. I. J. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen, "Localized extended Kalman filter for scalable real-time traffic state estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 385–394, Mar. 2012.

[4] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box–jenkins techniques," *Transp. Res. Rec.*, vol. 722, pp. 1–9, Apr. 1979.

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

[6] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 89–107, Sep. 2013.

[7] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[8] 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.

[9] E. S. Yu and C. Y. R. Chen, "Traffic prediction using neural networks," in *Proc. IEEE Global Telecommun. Conf.*, 1993, pp. 991–995.

[10] C. Zhou and P. Nelson, "Predicting traffic congestion using recurrent nueral network," in *World Congr. Intell. Transp. Syst.*, Dec. 2002, pp. 1–9.

[11] S. Yang and S. Qian, "Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 3, pp. 12–28, Dec. 2019.

[12] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transp. Res. Part C: Emerg. Technol.*, vol. 13, nos. 5–6, pp. 347–369, Oct. 2005.

[13] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for city wide crowd flows prediction," in *Proc. Assoc. Advance Artif. Intell. Conf. (AAAI)*, 2017, pp. 1–8.

[14] J. Bruna, W. Zaremba, A. Szalm, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–8.

[15] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–9.

[16] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–5.

[17] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies," *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 297–322, Aug. 2019.

[18] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. Assoc. Advance Artif. Intell. Conf.(AAAI)*, 2019, pp. 922–933.

[19] X. Feng, J. Guo, B. Qin, T. Liu, and Y. Liu, "Effective deep memory networks for distant supervised relation extraction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4002–4008.

[20] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani, "Dynamit: A simulation-based system for traffic prediction," in *Proc. DACCORD Short Term Forecasting Workshop*, Delft, The Netherlands, 1998, pp. 1–9.

[21] H. S. Mahmassani, X. Fei, S. Eisenman, X. Zhou, and X. Qin, *Dynasmart-x Eval. for Real-Time TMC Application: Chart Test Bed*. College Park, MD, USA: Univ. of Maryland, 2005.

[22] A. Nantes, D. Ngoduy, A. Bhaskar, M. Miska, and E. Chung, "Real-time traffic state estimation in urban corridors from heterogeneous data," *Transp. Res. C, Emerg. Technol.*, vol. 66, pp. 99–118, May 2016.

[23] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 2008.

[24] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[25] Z. Cui, R. Ke, and Y. Wang, "Deep stacked bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," in *6th Int. Workshop Urban Comput.*, 2016, pp. 1–4.

[26] A. Fred Agarap, "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data," 2017, *arXiv:1709.03082*. [Online]. Available: http://arxiv.org/abs/1709.03082

[27] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "GeoMAN: Multi-level attention networks for geo-sensory time series prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3428–3434.

[28] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolution neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3842–3852.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolution networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–5.

[30] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14333–14342.

[31] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 362–373.

[32] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: http://arxiv.org/abs/1805.08318

[33] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Systems(NIPS)*, 2017, pp. 5998–6008.

[34] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–5.

[35] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," 2017, *arXiv:1705.03122*. [Online]. Available: http://arxiv.org/abs/1705.03122

[36] 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.

[37] W. Ma and S. Qian, "Dynamic network analysis, traffic prediction and optimal dynamic messages for the philadelphia region," Pennsylvania Dept. Transp., Wellsboro, PA, USA, Tech. Rep. FHWA-PA-2016-014-CMU WO 04, 2016.

Kan Guo received the bachelor's degree in mathematics and physics from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree in control science and engineering with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing. His research interests include intelligent transportation systems, deep learning, and artificial intelligence.

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

Zhen (Sean) Qian received the Ph.D. degree in civil engineering from the University of California at Davis in 2011. He was a Post-Doctoral Researcher with the Department of Civil and Environmental Engineering, Stanford University, from 2011 to 2013. He directs the Mobility Data Analytics Center, Carnegie Mellon University. His research interests include intelligent transportation systems and dynamic large-scale network modeling.

Yanfeng Sun (Member, IEEE) received the Ph.D. degree from the Dalian University of Technology in 1993. She is currently a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. She is also a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, and with the Beijing Artificial Intelligence Institute. Her research interests include machine learning and image processing. She is a member of the China Computer Federation.

Junbin Gao (Member, IEEE) 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 is currently a Professor of big data analytics with The University of Sydney Business School, The University of Sydney. Prior to this, he was a Professor in computer science with the School of Computing and Mathematics, Charles Sturt University, Australia. He was a Senior Lecturer and a Lecturer in computer science with 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 with the Department of Mathematics, HUST. His main research interests include machine learning, data analytics, Bayesian learning and inference, and image analysis.

Baocai Yin (Member, IEEE) received the M.S. and Ph.D. degrees in computational mathematics from the Dalian University of Technology, Dalian, China, in 1988 and 1993, respectively. He is currently a Director with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology and with the Beijing Artificial Intelligence Institute. He has authored or coauthored more than 200 academic articles in prestigious international journals, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and top-level conferences, such as CVPR, IAAA, INFOCOM, IJCAI, and ACM SIGGRAPH. His research interests include multimedia, image processing, computer vision, and pattern recognition.