Optimized Graph Convolution Recurrent Neural Network for Traffic Prediction

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

***Abstract*— Traffic prediction is a core problem in the intelligent transportation system and has broad applications in the transportation management and planning, and the main challenge of this field is how to efficiently explore the spatial and temporal information of traffic data. Recently, various deep learning methods, such as convolution neural network (CNN), have shown promising performance in traffic prediction. However, it samples traffic data in regular grids as the input of CNN, thus it destroys the spatial structure of the road network. In this paper, we introduce a graph network and propose an optimized graph convolution recurrent neural network for traffic prediction, in which the spatial information of the road network is represented as a graph. Additionally, distinguishing with most current methods using a simple and empirical spatial graph, the proposed method learns an optimized graph through a data-driven way in the training phase, which reveals the latent relationship among the road segments from the traffic data. Lastly, the proposed method is evaluated on three real-world case studies, and the experimental results show that the proposed method outperforms state-of-the-art traffic prediction methods.**

***Index Terms*— Graph convolution network, recurrent neural network, traffic prediction.**

Manuscript received August 22, 2018; revised April 29, 2019, July 16, 2019, and November 4, 2019; accepted December 17, 2019. Date of publication January 14, 2020; date of current version February 2, 2021. 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, and in part by the Beijing Talents Project under Grant 2017A24. The Associate Editor for this article was J. Sanchez-Medina.
(Corresponding author: Yongli Hu.)

Kan Guo, Yongli Hu, and Yanfeng Sun are 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; huyongli@bjut.edu.cn; yfsun@bjut.edu.cn).

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

Hao Liu is with the Beijing Transportation Information Center, Beijing Municipal Commission of Transport, Beijing 100073, China (e-mail: hao.liu@bjjtw.gov.cn).

Ke Zhang is with the Beijing Transportation Coordination Center, Beijing Municipal Commission of Transport, Beijing 100073, China (e-mail: zhangke@bjjtw.gov.cn).

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).

Baocai Yin is with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China, and also with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: ybc@dlut.edu.cn).

Digital Object Identifier 10.1109/TITS.2019.2963722

I. Introduction

With the continuous expansion of the modern city and rapid increasing of vehicle ownership, transportation systems are facing many challenging problems on congestion, reliability, and sustainability. New technologies have been studied to construct intelligent transportation systems, which aim to optimize the allocation of vehicles and travelers. In any intelligent transportation systems, traffic prediction is a fundamental problem with a wide spectrum of applications. For example, customers can plan their route in advance, and the traffic operator/manager can guide the traffic systems according to the forecasted traffic states. However, there exist many obstacles for achieving accurate prediction, such as the large scale road infrastructure, an enormous number of vehicles and travelers, the stochastic, noisy and unpredictable traffic data. Moreover, it is difficult to reveal the underlying physics of transportation system even if large-scale transportation data are provided [1]. Thus, researchers have proposed many model-based approaches such as the simulation models and the machine learning models in the past decades.

For establishing Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS), several real-time simulation systems were constructed, such as DynaMIT [2], [3] and DYNASMART-X [4]. In these systems, the traffic state estimation, Kalman Filters(KFs), is the basic function. Then, based on KFs, Wang and Papageorgiou [5], [6] investigated an Extended Kalman Filters (EKFs) algorithm by incorporating a stochastic macroscopic freeway network. To further extend the capability of EKFs, many variants were proposed, such as EKFs with the nonlinear Cell Transmission Model (CTM) [7], the localized EKFs [8] and the probabilistic heterogeneous traffic data fusion based EKFs [9], etc. Different from EKFs, the ensemble KFs was developed by Lighthill Whitham Richards (LWR) [10], [11] and was used for freeway network estimation problems [12], [13].

The forecasting models based on the machine learning theory involve cross-domain knowledge between statistics, artificial intelligence, and traffic theory. In the early 1980s, some traffic prediction methods started to use the past few hours' traffic data to forecast the traffic states in the next few minutes (mostly no more than 5 minutes) by parameterized models [14]. Thereafter, many data-driven techniques were designed to address the short-time traffic prediction problem, such as K-Nearest Neighbor (KNN) [15] and other clustering-based methods [16], Autoregressive Integrated Moving Average Model (ARIMA) and its extensions [17], [18], Support Vector Regression (SVR) [19]–[21], Random Forest Regression [22], [23], Neural Networks(NN) [24], [25], Bayesian networks (BNs) [26], [27], Recurrent Neural Network (RNN) [28], Fuzzy Logic Regression [29], [30] and the hybrid models [1], [31], etc.

With the great progress achieved in the deep learning field, many related methods have made numerous successful applications such as image processing [32], [33] and natural language understanding [34], [35]. Thus, one prospective way is to introduce deep learning methods into traffic prediction.

Deep Neural Network(DNN) generally uses a set of historical traffic data as the training set and learns the complicated nonlinear relation. At present, a set of neural networks based traffic prediction have been proposed, including Feedforward Neural Network (FNN) [36], Convolution Neural Network (CNN) [37], and Gated Recurrent Unit (GRU) [38], etc. Benefiting from the power of nonlinear approximating of DNN, these methods usually outperform the conventional statistic methods. Due to all the data (such as traffic flow, speed, incidents, etc.) having spatial and temporal correlations, a key problem in these methods is how to explore the spatial and temporal information. Thus the spatial relations were modeled by matrices or nonlinear functions [26], for example, CNN [37], [39]. However, CNN demands the regular grid data such as images as input, while the traffic data on the road network is irregular. Therefore, to feed CNN, the traffic data is forced to sample as regular grids in the current methods, such as DSTCN [37], which may lose the spatial information of traffic data. The ideal manner for representing the traffic data is to model it as a network or graph with its original spatial form.

Recently, a new type of neural network based on the graph theory [40] was proposed [41], namely Graph Convolution Network (GCN). Comparing to CNN, GCN aims to process the irregular data by graphs, i.e. the input data of GCN is a set of nodes and their relations as edges. Then, some research fields have successfully utilized GCN to achieve improvements, such as protein interface [42], action recognition [43] and semi-supervised classification [44], etc. This is also verified by the latest exploration of GCN used in processing traffic data [45], [46]. Although GCN sheds light on the traffic data processing, what factors affect traffic prediction is still unclear. Especially, the graph in the current methods is generally fixed and constructed by prior human experience, which cannot represent the complex and large-scale traffic system. Thus, learning an optimized graph from the observed traffic data with a data-driven method is a natural and essential idea.

In this paper, we propose a novel Optimized Graph Convolution Recurrent Neural Network (OGCRNN) for traffic prediction. The main objective of the proposed method is to exploit the proper representation of the spatial and temporal information. In particular, for capturing the spatial features of the traffic data, we adopt the natural graph representation to model the traffic data of actual road networks. Additionally, we learn the spatial-temporal features of the traffic data by the latest variation of RNN, i.e. the Graph Convolution Gated Recurrent Unit (GCGRU) [47]. Different from the fixed graph used in the current graph based neural network, we design a residual graph model to replace the empirical and fixed graph.

The main contributions of this paper are summarized as follows:

1) A novel spatial and temporal neural network is constructed for traffic prediction based on GCN and GRU networks, which aims to understand and explore the complex spatial-temporal nature of traffic evolution among all road segments;
2) A graph updating strategy is proposed to construct an optimized graph matrix in the training step of the graph network;
3) The proposed method is validated and proven more effective than the existing methods through three representative real-world case studies, in which we predict traffic data in future 15 and 30 minutes.

The rest of the paper is organized as follows. In Section II, we summarize the related works of traffic prediction. Section III describes the proposed optimized graph recurrent neural network in detail. In Section IV, the performance of the proposed method is evaluated on three real-world traffic datasets. Finally, conclusions and suggestions for future work are provided.

II. Related Works

A rich set of methods or models have been proposed for traffic prediction. These methods were designed to deal with different traffic types, such as highway, district road and city traffic, etc. Additionally, to improve the prediction accuracy, the other conditions related to the transportation system were considered in some traffic prediction methods, such as weather, construction, and accidents, etc. Base on different traffic types and related conditions, we categorize these related works into different classes, as Table.I shown. Generally, highway traffic prediction has better results than city roads since highway has a simple traffic environment, and the adding of the related transportation condition will improve the accuracy of the traffic prediction.

To further inspect these related traffic prediction methods, we also extract their underlying principles and categorize them into different classes, including ARIMA, SVR, Random Forest Regression (RFR), NN and DNN. We summary these methods in Table.II. It is shown that although the traditional methods, including ARIMA, SVR, and RFR, have a long development history, NN has better performance. Particularly, the DNN based methods have obvious superiority because of its representation power for modeling the complex traffic data. Thus, in this paper, we study the DNN based traffic prediction method and exploit a new GCN based method by representing the traffic data as a graph.

III. Method

In this section, we first give an overview of the proposed method. Then several key components of the proposed method are introduced in detail, including the data representation, the graph convolution, GRU, and OGCRNN. Finally, the loss function and the training procedure of OGCRNN network are given.

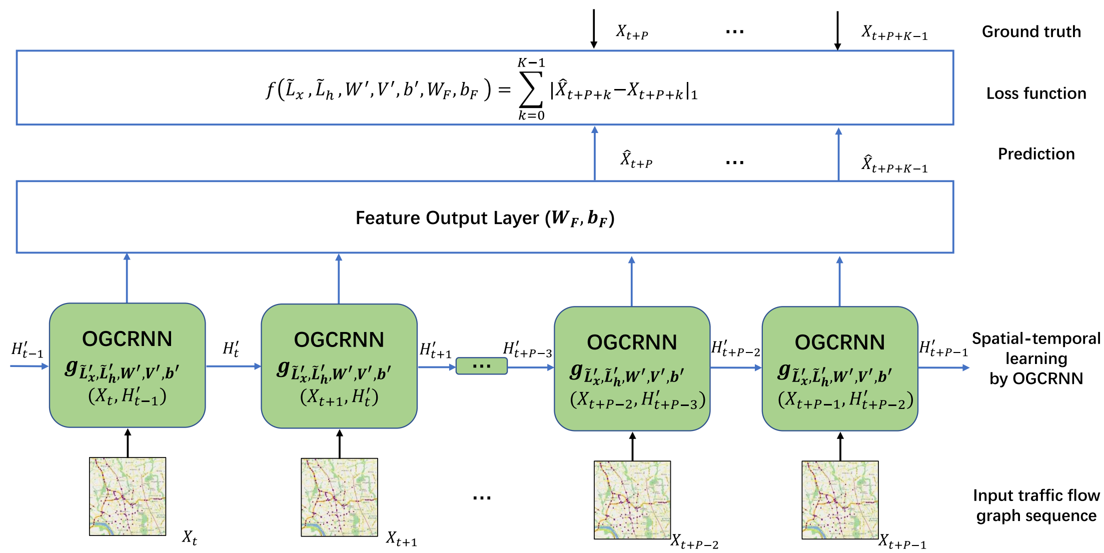


Fig. 1. The framework of the proposed Optimized Graph Convolution Recurrent Neural Network.

TABLE I

THE CATEGORIES OF THE RELATED WORKS ACCORDING TO TRAFFIC TYPES AND CONDITIONS

Traffic types	Conditions	References	Features
Highway	No conditions	[17], [18], [19], [26], [36]	The traditional traffic prediction without considering related conditions. Due to the simple traffic environment of highway, these methods can easily get accurate prediction results.
Highway	Weather and accident	[23],[25]	The relation between transportation system and conditions was exploited. It was shown that using related conditions can get better traffic prediction results.
City road	No conditions	[20], [21], [22], [24],[39]	Compared to highway traffic prediction, city road has complex traffic environment. So there are many studies propose different methods to deal with the complex and dynamic traffic prediction problem in city road.
City road	Weather and accident	[37]	The feature of weather and accident was extracted and combined with the traffic states. A deep learning method was adopted to address the complex traffic prediction problem in the city road network.

A. The Framework of the Proposed Method

Traffic forecasting is a specific problem of time series analysis. If we measure the traffic data in a certain frequency, such as one sample per 5 minutes, we will have the sequence of the traffic data in a time series form, denoted by $\{\ldots, X_t, \ldots, -\infty < t < +\infty\}$, in which $X_t \in \mathbf{R}^{N\times 1}$ and N is the number of road nodes. The core of this problem is how to reveal the spatial and temporal correlation of the traffic data. Thus, we use GCGRU to learning the spatial and temporal property hidden in the data and construct a novel optimized graph convolution recurrent neural network.

The framework of the proposed OGCRNN network is illustrated in Fig.1. There are several steps to implement the proposed method:

1) Traffic data representation: The traffic data input sequence which has P time slices is firstly represented as the form of graphs, denoted by $\{X_t, \ldots, X_{t+P-1}\} \in \mathbf{R}^{P\times N\times 1}$, and $\{X_{t+P}, \ldots, X_{t+P+K-1}\} \in \mathbf{R}^{K\times N\times 1}$ is the traffic prediction sequence with K time slices;

2) Spatial-Temporal feature extraction: The traffic sequence data is pipelined into the proposed OGCRNN network to learn spatial and temporal correlation by two Optimized Graph Matrices (OGMs) $\tilde{L}'_x \in \mathbf{R}^{N\times N}$, $\tilde{L}'_h \in \mathbf{R}^{N\times N}$, which will be defined in (16) and (17). The size of temporal features is $H'_{t+p} \in \mathbf{R}^{N\times F}$, $p = 0, \ldots, P-1$.

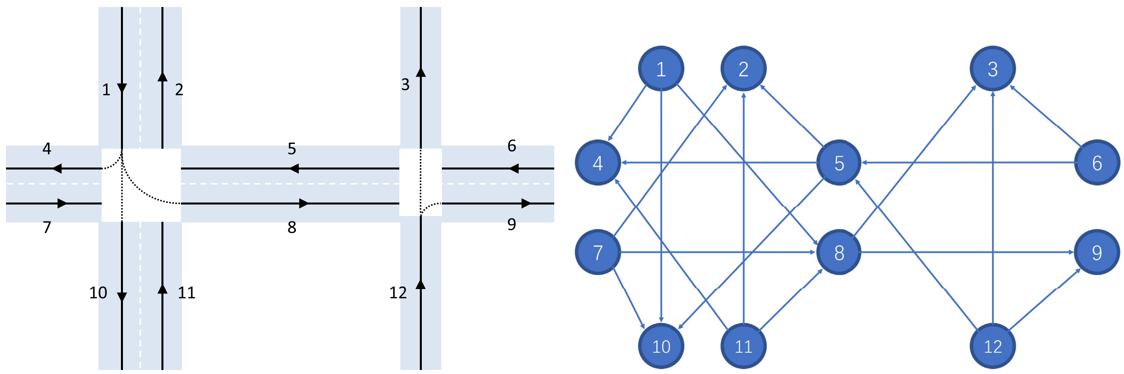
3) Traffic prediction: Based on the output of the OGCRNN network, $\{H'_t, \ldots, H'_{t+P-1}\}$, we can make predictions for the traffic data at future times $\{\hat{X}_{t+P}, \ldots, \hat{X}_{t+P+K-1}\} \in \mathbf{R}^{K\times N\times 1}$ by a feature output layer.

4) Objective definition: To implement the neural network for finally prediction, a proper objective function of the proposed neural network should be defined. Similar to the common neural network, the difference between the prediction $\{\hat{X}_{t+P}, \ldots, \hat{X}_{t+P+K-1}\}$ and its ground truth $\{X_{t+P}, \ldots, X_{t+P+K-1}\}$ is used in the objective function.

For the above OGCRNN network, giving a set of observed traffic data as the training set, we can train the parameters of the network by a common training method: the Back Prorogation (BP) method based on gradient descent algorithm. In the following, we will describe each step of the proposed method in detail.

B. Traffic Data Graph Representation

To preserve the natural spatial structure of the traffic data, we represent the traffic data as graphs. Different from the common graph representation of the road network, which models the road segments as edges and the crossroads as nodes, here we adopt an inverse representation. In this representation, the road segments are regarded as the graph nodes. The edges of these nodes are constructed if there exist connections between pairs of the road segments. For simplification, we only use directed connections to construct the edges of the graph. An example is shown in Fig.2, there are 12 road segments labeled as 1, …, 12 in Fig.2(a) which are the nodes



(a) A road network with 12 labeled road segments. (b) The graph of 12 road nodes with directed connections.

Fig. 2. An example of the graph representation of traffic data.

TABLE II

THE CATEGORIES OF THE RELATED WORKS ACCORDING TO THE UNDERLYING PRINCIPLES

Principle	References	Main idea	Features
ARIMA	[17],[18]	[17] simply utilized ARIMA to predict traffic volume of highway. [18] used the seasonal pattern of traffic data in the first difference of ARIMA on the highway.	ARIMA is a simple but efficient method for traffic prediction. It runs fast, but it has a big prediction error.
SVR	[19],[20],[21]	Due to the good generalization ability and global minima, SVR was proposed for traffic prediction [19],[20]. The fruit fly optimization was adopted to determine the hyper-parameters of SVR for city road traffic prediction [21].	Compared to ARIMA, SVR has greater generalization ability, but it is more easily affected by training data.
RFR	[22],[23]	The Random Forest and Adaboost were combined as the city traffic prediction method, which could deal with missing data [22]. [23] calculated the correlation between different conditions, and they used PCA to remove redundancy and select the key condition, which was utilized in Random Forest for highway traffic prediction.	The large decision tree structure of Random Forest brings good traffic prediction accuracy, but RFR is time-consuming compared with ARIMA and SVR.
NN	[24], [25],[26]	FNN was the simple NN method to capture the linear and nonlinear regularities of traffic prediction in city road [24]. The weather, incidents and other conditions were considered in the NN based traffic prediction [25]. The first order context memory of NN was also used to seek spatio-temporal relations in highway traffic prediction [26].	Because NN can explore nonlinear relation among the traffic data, it obtains better prediction results than the above methods. However, NN has more parameters to be determined in the training step.
DNN	[36],[37],[39]	DNN was more powerful to reveal the complex intrinsic relations hidden in the traffic data. So different DNN based traffic prediction methods were proposed for different traffic types and conditions, such as the stacked Autoencoder for highway [36], the residual convolution units model for city road with considering weather and accidents [37], and Deep Convolution Network(DCN)+LSTM method [39].	DNN has a more complex model structure than the above methods, so it has the best prediction accuracy. But DNN has enormous parameters to be trained and demands large scale training data.

of the graph. If there exists a directed connection between two road segments, there will be an edge. For example, the node of road segment 1, there are three directed-connected nodes of road segments, 4, 10, 8, thus there are edges (1,4), (1,10) and (1,8) for the graph; for the bidirectional traffic of roads, we regard the different directs of a road segment as two nodes in our graph, such as the nodes 1 and 2, 10 and 11 in Fig.2. By this method of finding the edges of the graph, we can construct the final graph of the road network. For the road network in Fig.2(a), the constructed graph is shown in Fig.2(b). Then we use an empirical matrix to represent the weights of the edges, i.e. $A \in \mathbf{R}^{N\times N}$, with element a_{ij} defined as follows,

$$a_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is a directed edge of} \\ & \text{the road network's graph;} \\ 0, & \text{else.} \end{cases}$$

C. Graph Convolution Neural Network

In the theory of GCN [41], the graph data was processed in the spectral domain. For this purpose, we compute the normalized Laplace matrix of the graph from the adjacent matrix A as follows,

$$\bar{L_0} = I_N - D^{(-1)/2} A D^{(-1)/2} \tag{1}$$

where I_N is the identity matrix and $D = diag(\Sigma_j a_{ij}) \in \mathbf{R}^{N\times N}$. Then we decompose $\bar{L_0}$ and get $\bar{L_0} = U \Lambda U^T$, where U is a matrix composing of the eigenvectors of $\bar{L_0}$ and $\Lambda = diag(\lambda)$, in which λ is the eigenvalues of $\bar{L_0}$ in descent order. According to the theory of GCN [41], we can implement convolution operation in the spectral domain of the graph as follows,

$$g_{\boldsymbol{\theta}} * X_t = U g_{\boldsymbol{\theta}}(\Lambda) U^T X_t \tag{2}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ is the convolution parameter and $g_{\boldsymbol{\theta}}(\Lambda) = diag(\boldsymbol{\theta})\Lambda$ is the product operator.

In the graph convolution (2), $U^T X$ means that the input X_t is transformed from the data space to the graph spectral space (Fourier space). Then the product operator $g_{\boldsymbol{\theta}}(\Lambda)$ is implemented on the spectral domain with the parameter $\boldsymbol{\theta}$, and the multiplication in the spectral domain is equivalent to the convolution in the data space, so $g_{\boldsymbol{\theta}}(\Lambda)$ in the Fourier space is the convolution operator. Finally, the result of the spectral domain is transformed back to the data space by the inverse transformation U.

Although it is a good way to perform the graph convolution, the computation complexity of (2) is high due to the decomposition of $\bar{L}_0$. So an approximation model of $g_{\boldsymbol{\theta}}(\Lambda)$ was proposed by utilizing the Chebyshev polynomials as an alternate [48], which has the form as follows,

$$g_{\boldsymbol{\theta}'(\tilde{\Lambda})} = \sum_{m=0}^{M} \theta'_m T_m(\tilde{\Lambda}) \tag{3}$$

where $\boldsymbol{\theta}' = (\theta'_0, \ldots, \theta'_M)$, $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$ and λ_{max} is the max eigenvalue of $\bar{L}_0$, and the Chebyshev polynomials $T_m(\tilde{\Lambda}), m = 0, 1, .., M$ are defined as [49]

$$\begin{aligned} T_m(\tilde{\Lambda}) &= 2\tilde{\Lambda}T_{m-1}(\tilde{\Lambda}) - T_{m-2}(\tilde{\Lambda}), \\ T_1(\tilde{\Lambda}) &= \tilde{\Lambda}, \\ T_0(\tilde{\Lambda}) &= I_N, \end{aligned}$$

Form this, (2) will be changed as follows,

$$\tilde{L} = \frac{2}{\lambda_{max}}\bar{L}_0 - I_N \tag{4}$$

$$\theta'(\tilde{L}) \star X = g_{\theta'}(\Lambda) * X_t \approx \sum_{m=0}^{M} \theta'_m T_m(\tilde{L}) X_t \tag{5}$$

so the graph convolution can be also denoted by $\theta'(\tilde{L}) \star X$.

D. GRU Based Neural Network

The GRU neural network [50], [51] is a variant of the Recurrent Neural Network (RNN) [52]. Different from the conventional RNN, the GRU neural network is characterized by making each recurrent unit to adaptively capture dependencies of different time scales. The structure of GRU is shown in Fig.3. There are several components in the GRU unit:

$$\begin{aligned} R_t &= \sigma(W_r X_t + V_r H_{t-1}) + b_r & (6)\\ \tilde{H}_t &= \upsilon tanh(W_h X_t + V_h(R_t \odot H_{t-1}) + b_h) & (7)\\ Z_t &= \sigma(W_z X_t + V_z H_{t-1}) + b_z & (8)\\ H_t &= (1 - Z_t) \odot \tilde{H}_t + Z_t \odot H_{t-1} & (9) \end{aligned}$$

where the operator, the variables and the parameters are explained as follows,

$\odot$: the elementwise product of two matrices;

X_t: the current traffic data measurement, which is the input of the GRU unit;

H_{t-1}: the output of the previous GRU Unit, which is also the input of the GRU unit;

R_t: the output of the reset gate from X_t and H_{t-1};

$\tilde{H}_t$: the candidate activation controlled by the reset gate R_t, which determines the forgetting degree of the previous state H_{t-1};

Z_t: the output of the updating gate from X_t and H_{t-1}, which determines how many elements of H_{t-1} with its state updated in the GRU unit;

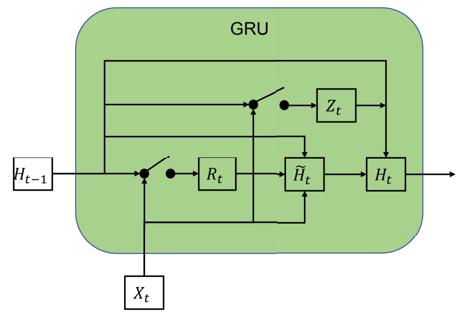


Fig. 3. The structure of a GRU unit with several inner processing components. The inputs of the current GRU unit include the output of the front GRU unit and the current observation. Through inner processing, the output feature is obtained and fed into the next GRU unit.

H_t: the output of the current GRU unit, which will be regarded as the input of the next GRU unit;

$W_r, V_r, W_h, V_h, W_z, V_z$ are the parameters of the GRU unit, which will be determined in the training procedure, and b_r, b_h, b_z are the bias parameters. For convenience, we denote them as W, V and b.

E. Optimized Graph Convolution Recurrent Neural Network

The GCGRU network [47] is an improved version of GRU based on GCN and has been used in traffic prediction [46]. Contracting the above GRU units, GCGRU replaces the feature mappings with the graph convolution in (5). So GCGRU can be formulated as follows,

$$\begin{aligned} R'_t &= \sigma(W'_r(\tilde{L}) \star X_t + V'_r(\tilde{L}) \star H'_{t-1}) + b'_r & (10)\\ \tilde{H}'_t &= tanh(W'_h(\tilde{L}) \star X_t + V'_h(\tilde{L}) \star (R_t \odot H'_{t-1}) + b'_h) & (11)\\ Z'_t &= \sigma(W'_z(\tilde{L}) \star X_t + V'_z(\tilde{L}) \star H'_{t-1}) + b'_z & (12)\\ H'_t &= (1 - Z'_t) \odot \tilde{H}'_t + Z'_t \odot H'_{t-1} & (13) \end{aligned}$$

where the main components $R'_t, \tilde{H}'_t, Z'_t, H'_t$ have the same effect as that in GRU, while the input $X_t \in \mathbf{R}^{N\times 1}$ and $H'_t \in \mathbf{R}^{N\times F}$. The new parameter matrices $W'_r, W'_h, W'_z \in \mathbf{R}^{M\times F}, V'_r, V'_h, V'_z \in \mathbf{R}^{MF\times F}$ can be simply denoted by W', V', which play the graph convolution with the Laplace matrix $\tilde{L}$; the bias parameters b'_r, b'_h, b'_z all with the size of $\mathbf{R}^F$ are simply denoted by b'. By integrating GCN and GRU, the output of GCGRU $\{H'_t, \ldots, H'_{t+P-1}\}$ will include both spatial and temporal features of the traffic data.

However, the Laplace matrix $\tilde{L}$ in GCGRU is fixed, which is considered the main drawback of the current graph convolution neural network as discussed above. So in our proposed OGCRNN method, we propose to explore an optimized graph matrix instead of a fixed one. Additionally, considered that the spatial property of $\{X_t, \ldots, X_{t+P-1}\}$ is different from $\{H'_t, \ldots, H'_{t+P-1}\}$, we adopt different graph matrices for $\{X_t, \ldots, X_{t+P-1}\}$ and $\{H'_t, \ldots, H'_{t+P-1}\}$, denoted by $\tilde{L}_x, \tilde{L}_h$. Similarly, we formulate them as the sum of the initialized Laplace matrix and the parameterized residual matrices as follows,

$$\tilde{L}_x = \tilde{L} + L_x \tag{14}$$

$$\tilde{L}_h = \tilde{L} + L_h \tag{15}$$

where $L_x, L_h \in \mathbf{R}^{N\times N}$ are the parameterized residual matrices. Then we can obtain the normalized-parameterized graph matrices, i.e. OGMs, as follows,

$$\tilde{L}'_x = D^{-1}_{\tilde{L}_x}\tilde{L}_x \tag{16}$$

$$\tilde{L}'_h = D^{-1}_{\tilde{L}_h}\tilde{L}_h \tag{17}$$

where the $D_{\tilde{L}_x}$ and $D_{\tilde{L}_h}$ are the degree matrices of $\tilde{L}_x$ and $\tilde{L}_h \in \mathbf{R}^{N\times N}$. Then we utilize $\tilde{L}'_x, \tilde{L}'_h$ to replace $\tilde{L}$ in GCGRU and get OGCGRU, shown in Algorithm 1.

For convenience, we denote the above OGCRNN unit by $g_{\tilde{L}'_x,\tilde{L}'_h,W',V',b'}(X_t, H'_{t-1})$.

F. Objective Function and Training Procedure

The aim of our proposed OGCRNN is using a sequence of the traffic data in the current interval $X_t, X_{t+1}, \ldots, X_{t+P-1}$ to make predictions at the future times $\hat{X}_{t+P}, \ldots, \hat{X}_{t+P+K-1}$. After the above OGCRNN neural network, we have coded the spatial and temporal feature of the current traffic data, so we design a Feature Output Layer to make the prediction:

$$\hat{X}_{t+P+k} = H'_{t+P+k-K}W_F + b_F, k = 0, \ldots, K-1 \tag{18}$$

where W_F, b_F are the weight matrix. If the output of the OGCRNN unit has F features, i.e. $H'_{t+P+k-K} \in \mathbf{R}^{N\times F}$, then $W_F \in \mathbf{R}^{F\times 1}, b_F \in \mathbf{R}^{N\times 1}$, the prediction $\hat{X}_{t+P+k} \in \mathbf{R}^{N\times 1}$.

Up to now, we have constructed the whole traffic prediction network. Then, similar to the general neural network, we use the difference between the prediction $\hat{X}_{t+P}, \ldots, \hat{X}_{t+P+K-1}$ and its ground truth $X_{t+P}, \ldots, X_{t+P+K-1}$ to establish the loss function as follows,

$$f(\tilde{L}_x, \tilde{L}_h, W', V', b', W_F, b_F) = \sum_{k=0}^{K-1} \|\hat{X}(t+P+k) - X(t+P+k)\|_1 \tag{19}$$

To solve the objective function in (19), we build a training data set of the traffic data, denoted by $\{X_t, t = 1, \ldots, T_s\}$, where $T_s >> P$. Then we build a test data set, which length is T'_s. So we slide the window of $[t, t+P-1]$ on the time axis of the training set and update all parameters in the training phase, especially $\tilde{L}'_x, \tilde{L}'_h$.

In the end, we analyze the complexity of OGCRNN. For the training complexity of OGCRNN, as shown in Algorithm 1, the complexity of step 3 and 4 is $O(EN^2)$, the complexity of step 5 and 6 is $O(EN^3)$, the complexity from step 7 to 12 is $O(EMN^2P) + O(EMN^2FP) + O(ENMF^2P) + O(ENFP) + O(EMNFP) + (EN^2P)$, and the complexity of step 14 is $O(NF) + O(N)$. So the training complexity is $O(EN^2) + O(EN^3) + O(EMN^2P) + O(EMN^2FP) + O(ENMF^2P) + O(ENFP) + O(EMNFP) + (EN^2P) + O(NF) + O(N)$. Considered $E >> N > F > M, P$, the training complexity could be simplified as $O(EN^3) + O(EMN^2FP)$. For the test complexity of OGCRNN, as the test phase of OGCRNN only carries out step 2 to 14, so we have the test complexity of $O(N^3) + O(MN^2FP)$, which is far less than the training complexity of OGCRNN.

Algorithm 1 The training phase of OGCRNN

Input:
The current traffic measurement sequence $X^e_t, \ldots, X^e_{t+P-1} \in \mathbf{R}^{P\times N\times 1}$, the zero initialized state $H'_{t-1} \in \mathbf{R}^{N\times F}$;

Parameter:
$L_x \in \mathbf{R}^{N\times N}, L_h \in \mathbf{R}^{N\times N}, W' = W'_r, W'_h, W'_z \in \mathbf{R}^{M\times F}, V' = V'_r, V'_h, V'_z \in \mathbf{R}^{MF\times F}, W_F \in \mathbf{R}^{F\times 1}, b_F \in \mathbf{R}^{N\times 1}, b' = b'_r, b'_h, b'_z \in \mathbf{R}^F$

1: **for** e-th training iteration in total E training iterations **do**
2: Update $\tilde{L}$ by (4);
3: $\tilde{L}_x = \tilde{L} + L_x$;
4: $\tilde{L}_h = \tilde{L} + L_h$;
5: $\tilde{L}'_x = D^{-1}_{\tilde{L}_x}\tilde{L}_x \rhd D_{\tilde{L}_x} = diag(\sum_j (\tilde{L}_x)_{ij})$;
6: $\tilde{L}'_h = D^{-1}_{\tilde{L}_h}\tilde{L}_h \rhd D_{\tilde{L}_h} = diag(\sum_j (\tilde{L}_h)_{ij})$;
7: **for** p-th X^e_{t+p} in $X^e_t, \ldots, X^e_{t+P-1}$, $p = 0, \ldots, P-1$ **do**
8: $R'_{t+p} = \sigma(W'_r(\tilde{L}'_x) \star X^e_{t+p} + V'_r(\tilde{L}'_h) \star H'_{t+p-1}) + b'_r$;
9: $\tilde{H}'_{t+p} = tanh(W'_h(\tilde{L}'_x) \star X^e_{t+p} + V'_h(\tilde{L}'_h)$ $\star(R_{t+p} \odot H'_{t+p-1})) + b'_h$;
10: $Z'_{t+p} = \sigma(W'_z(\tilde{L}'_x) \star X^e_{t+p} + V'_z(\tilde{L}'_h) \star H'_{t+p-1}) + b'_z$;
11: $H'_{t+p} = (1 - Z'_{t+p}) \odot \tilde{H}'_{t+p} + Z'_{t+p} \odot H'_{t+p-1}$;
12: **end for**
13: **return** $H'_t, \ldots, H'_{t+P-1}$;
14: $\hat{X}_{t+P+k} = H'_{t+P+k-K}W_F + b_F, k = 0, \ldots, K-1$;
15: Calculate parameters gradient by the loss function in (19) and update parameters by Back Propagation algorithm.
16: **end for**

IV. Experiments

To evaluate our proposed OGCRNN method, we undertake traffic prediction experiments on three real-world traffic datasets and compare the results with the conventional methods and the neural network based methods.

A. Experiment Setting

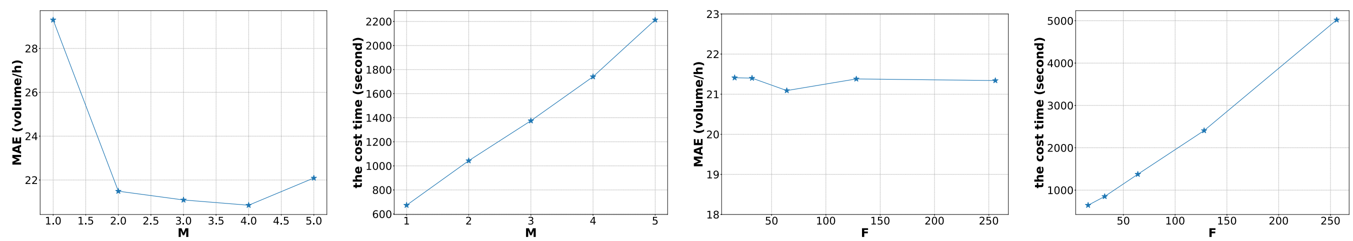
1) The Datasets Used in the Experiments: In our experiments, we use the following three real-world traffic datasets for assessing the proposed method.

- Travel time/speed dataset near the D.C. area

This dataset covers the northwestern part of the D.C. metropolitan area and it is collected in the summer of 2016. There are $N = 370$ road segments in the road network, including roads of I-270, I-495, MD-185, MD-187, MD-335, MD-190, and MD-191, etc. There are 150 highway segments and the others are the main surface streets of the city, and the speed of the 370 road segments is measured every 5 minutes and 24,000 measurements are collected for this experiment.

- Travel time/speed dataset in Philadelphia

This dataset covers the center city of Philadelphia and it is collected in the summer of 2016. As it is the core area of the city, the traffic is busy and complex with crossroads. There are $N = 397$ road segments in the road network, including roads of I-676, I-95, I-76, Kelly DR, Fairmount Ave, RACE St. etc. There are only 35 highway segments and the others are



(a) MAE w.r.t. M (b) The cost time w.r.t. M (c) MAE w.r.t. F (d) The cost time w.r.t. F

Fig. 4. MAE and the cost time of OGCRNN with respect to M, F.

main roads, so Philadelphia dataset is more complex than D.C. dataset. The speed of the 397 road segments is also measured every 5 minutes and 24,000 measurements are collected.

- Traffic flow dataset in PeMSD4

This dataset, namely PeMSD4 [53], [54], covers the San Francisco Bay Area, and it is collected from January to February in 2018. There are $N = 307$ highway segments in the road network. The traffic flow of the 307 road segments is measured every 5 minutes and contains 288 measurements per day, so 15000 measurements are collected.

2) Compared Methods: To verify the effectiveness of our proposed OGCRNN method, we compare our method with the following related methods, including HA, SVR, Random Forest, ARIMA, FNN, GRU, GCGRU. In these methods, SVR, Random Forest, ARIMA, FNN, and GRU use a general vector form of traffic data as input, i.e., X_t. GCGRU and our method use the input with graph structure. It is noted that we do not select LSTM as our compared method because the principle and ability of LSTM is close to GRU. DSTCN is not used for comparison in our experiments, because it demands a resampling image-like input for the traffic data, which is completely different from the graph data representation of our method. Additionally, the output of DSTCN cannot be compared as it is a re-scaled image that is difficult to be transformed into the measurement of each road segment.

We adopt 3-fold cross-validation in our experiments, i.e. we divide a dataset into 3 parts, denoted by $Data = \{D_1, D_2, D_3\}$). In each round of experiment on the dataset, two parts are used as the training set, and the remaining one is used as the test set, i.e. the dataset using for the ith round experiment is denoted by $test_i, i = 1, 2, 3$, which is composed of

$$\begin{aligned} TrainingSet: &\ Data - D_i \\ TestSet: &\ D_i \end{aligned}$$

We calculate the mean error of all rounds of the experiment as the final result for the experiment.

The experiment results of different traffic prediction methods are evaluated by three metrics, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Square Error(MSE), which are given as follows,

$$MAE = \frac{1}{3}\sum_{i=1}^{3}\{\frac{1}{J}\sum_{j=1}^{J}|(X_j - \hat{X}_j)|\}_{test_i}, \hat{X}_j \in test_i$$

$$MAPE = \frac{1}{3}\sum_{i=1}^{3}\{\frac{1}{J}\sum_{j=1}^{J}\frac{|(X_j - \hat{X}_j)|}{|X_j|}\}_{test_i}, \hat{X}_j \in test_i$$

$$MSE = \frac{1}{3}\sum_{i=1}^{3}\{\frac{1}{J}\sum_{j=1}^{J}(X_j - \hat{X}_j)^2\}_{test_i}, \hat{X}_j \in test_i$$

where J is the number of test samples, X_j is the ground truth, and $\hat{X}_j$ is the prediction. In the D.C. and Philadelphia datasets, the unit of MAE is the mile per hour($mile/h$), the unit of MAPE is the percentage(%), and the unit of MSE is the square of a mile per hour($mile^2/h^2$). In PeMSD4, due to the traffic flow could be zeros, we only use MAE and MSE to evaluate the experiments' results, so the unit of MAE is the volume per hour($volume/h$) and the unit of MSE is the square of volume per hour($volume^2/h^2$).

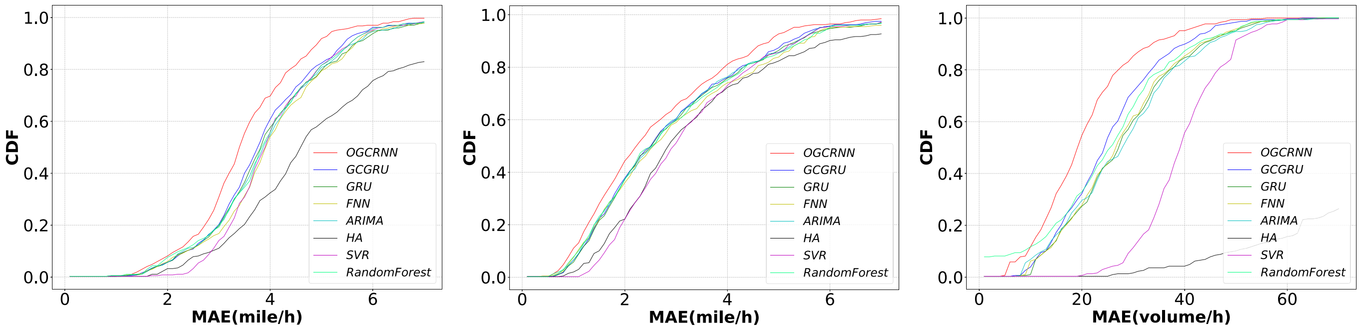
3) The Parameters Setting: To identify the effects of M, F for OGCRNN, we implement model training and prediction experiments with $test_3$ on PeMSD4 with different sets of M, F. The results are shown in Fig.4. Here, the graph convolution Polynomial order $M = 1, 2, 3, 4, 5$ and the feature size $F = 16, 32, 64, 128, 256$. The prediction error and the cost time of the training phase are reported. From Fig.4(a) and Fig.4(b), it is shown that the prediction error is stable when $M > 2$ and the model training time is rising with the increasing of M, so we select $M = 3$. From Fig.4(c) and Fig.4(d), it is indicated that the model is robust to different F. Considering the time complexity, we select $F = 64$.

4) The Development Platform: All the algorithms in our experiments are coded by Python, Tensorflow and the relevant python statistical packages, in which the initial learning rate is 0.001 with decay rates 0.7 every 5 epochs and the whole training epochs are 40. We use the batch trick to train and the size is set as 32 in D.C. and Philadelphia dataset, and 25 in PeMSD4. The algorithms are implemented on a computer with an E5-2630 V4 CPU, 64G RAM and a P5000 GPU with 16G memory.

B. Experiment Results

- The result of D.C. experiment

In the experiment, the size of training data is 16,000, i.e. $T_s = 16,000$, and the remaining part with 8,000 measurements is used as the test set, i.e. $T_s' = 8,000$. We use the current 0.5 hour's measurements to predicate the speed in the next 15 or 30 minutes, i.e. $P = 6$, $K = 3$ or $K = 6$. The traffic



(a) D.C. (b) Philadelphia (c) PeMSD4

Fig. 5. The CDF curves of MAE of different methods on the three datasets.

TABLE III

THE TRAFFIC SPEED PREDICTION MEAN RESULTS OF DIFFERENT METHODS ON D.C. DATASET

Method	15 mins			30 mins		
	MAE $(mile/h)$	MAPE (%)	MSE $(mile^2/h^2)$	MAE $(mile/h)$	MAPE (%)	MSE $(mile^2/h^2)$
HA	5.25	22.56	68.06	5.25	22.56	68.06
SVR	3.67	15.16	28.38	4.07	16.68	36.66
Random Forest	3.49	14.46	28.68	3.94	16.48	37.22
ARIMA	3.55	14.69	28.41	3.97	16.72	36.69
FNN	3.50	14.42	30.67	4.04	16.83	41.42
GRU	3.53	14.64	30.68	3.97	17.16	38.54
GCGRU	3.40	14.09	28.32	3.87	16.53	36.61
OGCRNN	**3.23**	**13.48**	**25.66**	**3.52**	**14.94**	**31.28**

TABLE IV

THE TRAFFIC SPEED PREDICTION MEAN RESULTS OF DIFFERENT METHODS ON PHILADELPHIA DATASET

Method	15 mins			30 mins		
	MAE $(mile/h)$	MAPE (%)	MSE $(mile^2/h^2)$	MAE $(mile/h)$	MAPE (%)	MSE $(mile^2/h^2)$
HA	3.54	26.39	31.94	3.54	26.39	31.94
SVR	3.07	23.97	21.04	3.27	25.34	24.29
Random Forest	2.63	19.65	20.17	2.95	21.90	23.74
ARIMA	2.61	19.27	19.30	2.93	21.76	25.61
FNN	2.59	18.23	22.01	3.03	21.51	26.85
GRU	2.61	18.52	20.98	2.96	21.59	23.98
GCGRU	2.53	18.11	20.01	2.90	21.15	23.14
OGCRNN	**2.40**	**17.21**	**18.35**	**2.64**	**19.04**	**20.79**

speed prediction results of different methods will be averaged at all $test_i, i = 1, 2, 3$, and these results are shown in Table III.

- The result of Philadelphia experiment

In this experiment, the scales of the training and test sets T_s, T_s', the prediction parameters P, K are same as that of D.C. dataset, i.e. $T_s = 16,000$, $T_s' = 8,000$, $P = 6$, $K = 3$ or $K = 6$. Under these setting, the traffic speed prediction mean results are shown in Table IV.

- The result of PeMSD4 experiment

In this experiment, the training and test sets scales T_s, T_s' and the prediction parameters P, K are $T_s = 10,000$, $T_s' = 5,000$, $P = 6$, $K = 3$ or $K = 6$. Under these setting, the traffic volume prediction results are shown in Table V.

TABLE V

THE TRAFFIC FLOW PREDICTION MEAN RESULTS OF DIFFERENT METHODS ON PEMSD4 DATASET

Method	15 mins		30 mins	
	MAE $(volume/h)$	MSE $(volume^2/h^2)$	MAE $(volume/h)$	MSE $(volume^2/h^2)$
HA	104.62	16725.02	104.62	16725.02
SVR	35.76	2035.48	39.93	2502.75
Random Forest	21.40	1175.21	25.53	1616.10
ARIMA	22.61	1217.41	28.28	1797.55
FNN	22.45	1225.39	27.66	1783.85
GRU	22.50	1248.88	29.02	1830.67
GCGRU	21.83	1162.03	25.42	1494.22
OGCRNN	**19.19**	**951.26**	**20.58**	**1085.16**

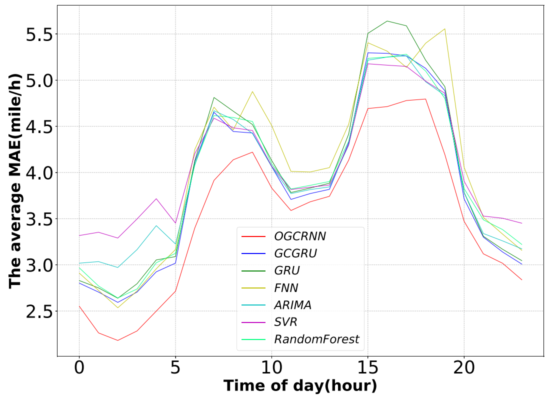
The results in Table III, Table IV and Table V show that OGCRNN has the best performance compared with other methods in all metrics. In all the methods, HA has the worst results, which implies that the traffic data of the three test datasets is unstable and changeable, especially for the data measured from the mixed highway roads and urban roads. Overall, the neural network methods, including FNN, GRU, GCGRU, and OGCRNN, have better performance than the other methods. Particularly, our method gets the best results, which is explained that the optimized graph convolution integrated with GCGRU is suitable to extract the spatial and temporal features of the traffic data.

To illustrate the results clearly, we provide the Cumulative Distribution Function (CDF) curves of MAE errors of all methods, as shown in Fig.5(a), Fig.5(b), Fig.5(c) for D.C., Philadelphia, and PeMSD4 datasets, respectively. It is shown that, in terms of MAE, the proposed OGCRNN method has the best performance at all levels of speed and volume on the three datasets.

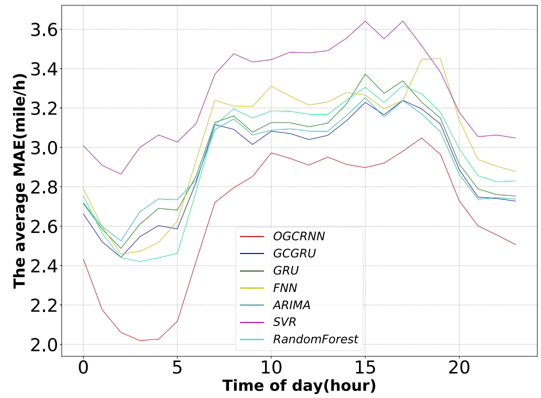
C. Results Analysis and Discussion

The experimental results in Table III, IV and V are the overall prediction errors in the sense of the average of all road segments in all testing times. To further analyze the performances of different methods in spatial and temporal distribution, we explore the prediction results of the highway and main road. Then from the temporal view, we exploit the change of the prediction errors of different methods over time. Finally, we discuss the advantages of OGMs $\tilde{L}_x', \tilde{L}_h'$ compared to the fixed one $\tilde{L}$.

1) The Spatial Distribution of Traffic Prediction Errors: To observe the prediction difference for different road types,



(a) D.C.



(b) Philadelphia

Fig. 6. The average MAE($mile/h$) with respect to time of day of different methods on the D.C. and Philadelphia datasets.

TABLE VI
THE 30 MINUTE AHEAD TRAFFIC SPEED MEAN PREDICTION IN THE D.C. AREA

Method	Highway			Main roads		
	MAE ($mile/h$)	MAPE (%)	MSE ($mile^2/h^2$)	MAE ($mile/h$)	MAPE (%)	MSE ($mile^2/h^2$)
HA	6.42	20.18	112.80	4.46	24.18	37.56
SVR	4.06	9.94	44.61	4.09	21.61	31.24
Random Forest	4.12	10.23	46.12	3.82	20.64	31.15
ARIMA	4.15	10.44	45.92	3.86	20.95	30.40
FNN	4.13	9.54	51.03	3.99	21.81	34.87
GRU	4.13	11.15	48.63	3.87	21.27	31.66
GCGRU	3.96	10.01	45.26	3.81	20.98	30.71
OGCRNN	**3.57**	**8.76**	**37.11**	**3.50**	**19.16**	**27.32**

TABLE VII
THE COMPARISON BETWEEN THE FIXED ADJACENT MATRIX A AND OGMS $\tilde{L}'_x$ WITH TOP 15 ABSOLUTION VALUES OF D.C. DATASET

OGM $\tilde{L}'_x$	The fixed A	Signal correlation	Ranking
0.57	1	0.94	0.4%
0.51	1	0.92	0.4%
0.49	1	0.99	0.3%
0.47	1	0.86	0.5%
0.45	1	0.94	0.4%
0.42	0	0.27	23.3%
0.42	1	0.94	0.4%
0.42	0	0.23	29.4%
0.40	0	0.86	0.5%
0.39	1	0.97	0.3%
0.39	1	0.94	0.4%
0.38	0	0.34	14.0%
0.38	1	0.94	0.4%
0.37	1	0.94	0.4%
0.37	1	0.87	0.5%

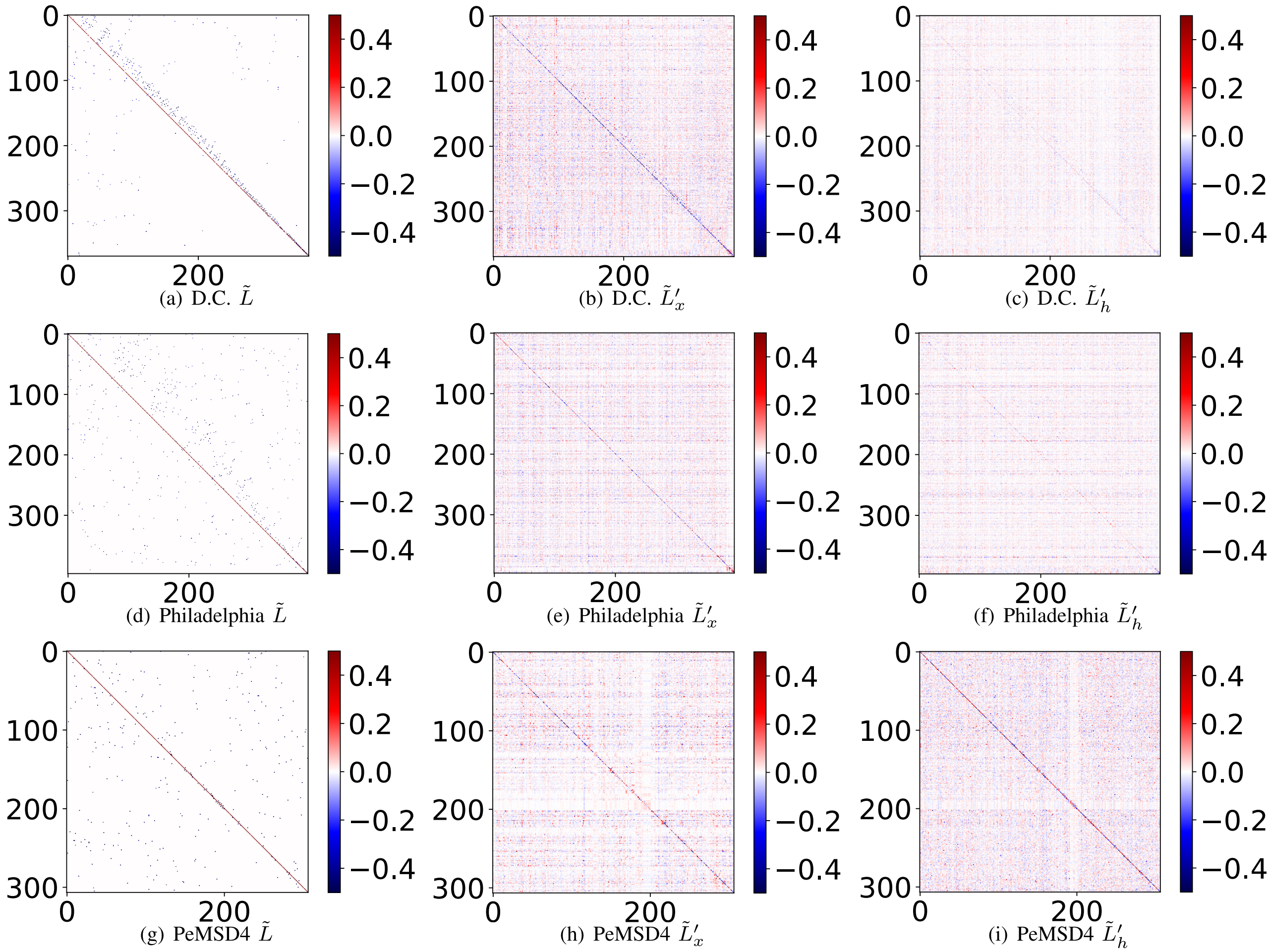
we compute MAE, MAPE, and MSE of highway roads and the main roads separately on the D.C. dataset. As Table VI shown, in MAE metric, there is no obvious difference for different roads, while in MAPE metric, the highway roads have obvious smaller errors compared with main roads. For OGCRNN, whenever on the highway and the main roads, it has the best performance in all metrics, which implies that the proposed method is effective on both road types. Especially, OGCRNN has at least 1.5% improvement on main roads and 0.8% improvement on the highway compared with the other methods. Therefore, OGCRNN is suitable to deal with the complex urban traffic condition without losing the good performance for the highway.

2) The Time-Varying Feature of Traffic Prediction: To exploit the time-varying feature of different methods, we plot MAE at the same time of day for all methods, as shown in Fig.6(a) and Fig.6(b). It is shown that OGCRNN has minimum errors in all time, and GCGRU, GRU, ARIMA and Random Forest are better than FNN and SVR. On the peaking hours, for example, 8 AM to 10 AM and 2 PM to 7 PM, OGCRNN outperforms other methods obviously, which illustrates OGCRNN is robust to the complicated temporal situation of the road network. Because the PeMSD4 dataset has not a detailed illustration of data's start date, we do not draw its time curve.

3) The Fixed Laplace Matrix vs. the Optimized: Different from the fixed Laplace matrix used in the conventional graph convolution neural network, our method is characterized by using OGM in the graph neural network, which is considered more consistent with the intrinsic property of the traffic data. So we discuss the difference between OGM and the fixed.

We draw the fixed Laplace matrix $\tilde{L}$ and OGMs $\tilde{L}'_x$, $\tilde{L}'_h$ and compare them intuitively. As shown in Fig.7, $\tilde{L}$ is sparse, i.e. a small set of elements of $\tilde{L}$ is nonzero, which means each road segment is only related to several other segments. It is consistent with the construction of $\tilde{L}$ in Section III. However, the other figures show that $\tilde{L}'_x$, $\tilde{L}'_h$ have more nonzero elements than $\tilde{L}$. This point is verified by the prior that there exist different correlations among different road segments, which cannot be simply represented like that in $\tilde{L}$. In this sense, OGCRNN has the potential to establish a proper road network graph matrix from the input data and reveal the relations among different road segments.

In order to show the difference between the fixed adjacent matrix A and OGM in detail, we select the top 15 elements of OGM $\tilde{L}'_x$ with large absolute values, which represent the pairs of road segments have large correlations, then we list them in the first column in Table VII with the corresponding values of the fixed-initial graph adjacent matrix A listing



(a) D.C. $\tilde{L}$ (b) D.C. $\tilde{L}'_x$ (c) D.C. $\tilde{L}'_h$

(d) Philadelphia $\tilde{L}$ (e) Philadelphia $\tilde{L}'_x$ (f) Philadelphia $\tilde{L}'_h$

(g) PeMSD4 $\tilde{L}$ (h) PeMSD4 $\tilde{L}'_x$ (i) PeMSD4 $\tilde{L}'_h$

Fig. 7. The visualization of the different matrices $\tilde{L}, \tilde{L}'_x, \tilde{L}'_h$ of the three datasets in the $test_1$. To enhance image contrast, all values in the matrices are set to [-0.5,0.5].

in the second column. We also compute the absolute signal correlation coefficients between pairs of the road segments, as shown in the third column. The ranking of the signal correlations in all elements is also provided in the fourth column. From Table VII, it is shown that if the pairs of road segments are connected, i.e. $a_{ij} = 1$, the corresponding values of $\tilde{L}'_x$ are also large, same to the signal correlations. It is concluded that OGM can preserve the main spatial information of the original road network. Furthermore, OGM can reveal the correlations of road segments without directed connections ($a_{ij} = 0$), while they have relatively large signal correlations. This is the advantage of OGM.

V. Conclusion

In this paper, we propose a novel graph convolution network to extract the temporal and spatial features of traffic data. GCN is used to extract the spatial feature and GRU is used to extract the dynamic feature, then GCN and GRU are integrated seamlessly into one model. Considering the correlations of the road segments and its traffic measurements are not fixed, we also propose an updating strategy to find an OGM by a data-driven way in the training procedure. Although OGCRNN has good traffic prediction results and outperforms the compared methods, the constructed OGM is still fixed in the test phase, which is considered the main limitation of our work. Thus, it is an interesting work to construct a dynamic and adaptive Laplace matrix of the road network. However, it faces the challenge of utilizing the static road network and the observed dynamic traffic measurement at the same time. Due to the complexity of dynamic traffic data, we will try to design an attention mechanism network to construct a dynamic Laplace matrix. Considering the high complexity of OGCRNN, another research direction is to find an efficient algorithm to make the proposed method applicable for large scale road networks.

References

[1] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," *Transp. Res. B, Methodol.*, vol. 46, no. 9, pp. 1097–1122, Nov. 2012.

[2] 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.

[3] M. E. Ben-Akiva, S. Gao, Z. Wei, and Y. Wen, "A dynamic traffic assignment model for highly congested urban networks," *Transp. Res. C, Emerg. Technol.*, vol. 24, pp. 62–82, Oct. 2012.

[4] H. S. Mahmassani, X. Fei, S. Eisenman, X. Zhou, and X. Qin, "Dynasmart-X evaluation for real-time TMC application: Chart test bed," Maryland Transp. Initiative, Univ. Maryland, College Park, MD, USA, Tech. Rep., Jul. 2005, pp. 1–144. [Online]. Available: https://www.academia.edu/23980075/Dynasmart-X_Evaluation_for_Real-Time_TMC_Application_Chart_Test_Bed

[5] 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.

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

[7] C. M. Tampere and L. H. Immers, "An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 209–216.

[8] 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.

[9] 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, Emerging Technol.*, vol. 66, pp. 99–118, May 2016.

[10] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices," in *Proc. 47th IEEE Conf. Decision Control*, 2008, pp. 5062–5068.

[11] D. B. Work, S. Blandin, O. P. Tossavainen, B. Piccoli, and A. M. Bayen, "A traffic model for velocity data assimilation," *Appl. Math. Res. Express*, vol. 2010, no. 1, pp. 1–35, Apr. 2010.

[12] Y. Yuan, J. W. C. Van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real–time lagrangian traffic state estimator for freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 59–70, Mar. 2012.

[13] Y. Yuan, H. Van Lint, F. Van Wageningen-Kessels, and S. Hoogendoorn, "Network-wide traffic state estimation using loop detector and floating car data," *J. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 41–50, Jan. 2014.

[14] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[15] 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.

[16] 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.

[17] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-Jenkins techniques," *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[18] B. L. Smith, B. M. Williams, and R. K. 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.

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

[20] N. Mitrovic, M. T. Asif, J. Dauwels, and P. Jaillet, "Low–dimensional models for compressed sensing and prediction of large-scale traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2949–2954, Oct. 2015.

[21] Y. Cong, J. Wang, and X. Li, "Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm," *Procedia Eng.*, vol. 137, pp. 59–68, Dec. 2016.

[22] M. A. Putra, N. A. Setiawan, and S. Wibirama, "Wart treatment method selection using AdaBoost with random forests as a weak learner," *Proc. World Acad. Sci., Eng. Technol.*, vol. 19, pp. 193–198, Jan. 2007.

[23] S. Yang and S. Qian, "Understanding and predicting roadway travel time with spatio-temporal features of network traffic flow, weather conditions and incidents," in *Proc. Transp. Res. Board*, 2018.

[24] E. Yu and C. Chen, "Traffic prediction using neural networks," in *Proc. GLOBECOM IEEE Global Telecommun. Conf.*, Dec. 2002, pp. 991–995.

[25] L. Florio and L. Mussone, "Neural-network models for classification and forecasting of freeway traffic flow stability," *Control Eng. Pract.*, vol. 4, no. 2, pp. 153–164, Feb. 1996.

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

[27] G. Fusco, N. Isaenko, and C. Colombaroni, "Comparative analysis of implicit models for real-time short-term traffic predictions," *IET Intell. Transp. Syst.*, vol. 10, no. 4, pp. 270–278, May 2016.

[28] C. Zhou and P. Nelson, "Predicting traffic congestion using recurrent nueral network," in *Proc. World Congr. Intell. Transp. Syst.*, 2002.

[29] H. Xiao, H. Sun, B. Ran, and Y. Oh, "Fuzzy-neural network traffic prediction framework with wavelet decomposition," *Transp. Res. Rec.*, vol. 1836, no. 1, pp. 16–20, Jan. 2003.

[30] H. Kanoh, T. Furukawa, S. Tsukahara, K. Hara, H. Nishi, and H. Kurokawa, "Short-term traffic prediction using fuzzy C-means and cellular automata in a wide-area road network," in *Proc. IEEE Intell. Transp. Syst.*, Vienna, Austria, Sep. 2005, pp. 381–385.

[31] H. Wang, L. Liu, S. Dong, Z. Qian, and H. Wei, "A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework," *Transportmetrica B, Transp. Dyn.*, vol. 4, no. 3, pp. 159–186, Sep. 2016.

[32] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance–sensitive fully convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2016, pp. 534–549.

[33] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R–CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[35] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.

[36] 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.

[37] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for city wide crowd flows prediction," in *Proc. 31st Assoc. Adv. Artif. Intell. Conf.*, 2017.

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

[39] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, 2017.

[40] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1992.

[41] J. Bruna, W. Zaremba, A. Szalm, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[42] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2017, pp. 6533–6542.

[43] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," *Proc. 32nd AAAI*, vol. 33, Aug. 2019, pp. 8561–8568.

[44] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolution networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.

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

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

[47] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[48] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolution neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[49] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[50] K. Cho, B. Van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. SSST 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, 2014, pp. 103–111.

[51] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[52] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.

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

[54] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. 33rd AAAI*, vol. 33, Sep. 2019, pp. 922–929.

Kan Guo received the bachelor's degree in mathematics and physics from the School of Beijing University of Posts and Telecommunications of 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, China. 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 Artificial Intelligence Institute, Beijing University of Technology. He is also a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology and the Beijing Advanced Innovation Center for Future Internet Technology. 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, Davis, Davis, CA, USA, in 2011. He was a Post-Doctoral Researcher with the Department of Civil and Environmental Engineering, Stanford University, from 2011 to 2013. He is currently directs the Mobility Data Analytics Center, Carnegie Mellon University. His research interests include intelligent transportation systems and dynamic large-scale network modeling.

Hao Liu received the M.S. degree in transportation planning from the Research Institute of Highway Ministry of Transport, Beijing, China, in 2002, and the Ph.D. degree from the Delft University of Technology, The Netherlands, in 2008. He is currently a Researcher with the Beijing Transportation Information Center. His research interests include transportation planning and intelligent transportation systems.

Ke Zhang received the bachelor's degree in mathematics from the School of Northwest University of China, in 1995, the master dissertation and the M.Sc. degree in mathematics from the School of Beijing University of Technology, in 1998, and the Ph.D. degree in transportation from the School of Beijing University of Technology, in 2001. He then started to work for the Chinese National Intelligent Transport Systems Research Center of Engineering and Technology (ITSC), Research Institute of Highway (RIOH), Ministry of Transport. In 2011, he joined the Beijing Municipal Transportation Operations Coordination Center (TOCC). His research interests include transportation operations surveillance, transportation data analysis, intelligent transport system (ITS) planning, and applications.

Yanfeng Sun 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 Artificial Intelligence Institute, Beijing University of Technology, Beijing, China. She is also a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology and the Beijing Advanced Innovation Center for Future Internet Technology. Her research interests are machine learning and image processing. She is a member of the China Computer Federation.

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

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 Professor of computer science and technology with the Faulty of Electronic Information and Electrical Engineering, Dalian University of Technology. He is also a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology and the Beijing Advanced Innovation Center for Future Internet Technology. He has authored or coauthored more than 200 academic articles in prestigious international journals, including the IEEE T-PAMI, the IEEE T-MM, the IEEE T-IP, the IEEE T-NNLS, the IEEE T-CYB, the IEEE T-CSVT, 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.