

Traffic-GGNN: Predicting Traffic Flow via Attentional Spatial-Temporal Gated Graph Neural Networks

Yang Wang^{ID}, Jin Zheng, *Member, IEEE*, Yuqi Du, Cheng Huang, and Ping Li, *Member, IEEE*

Abstract—Recent spatial-temporal graph-based deep learning methods for Traffic Flow Prediction (TFP) problems have shown superior performance in modeling higher-level spatial interactions and temporal correlations. However, most of these methods suffer from post-fusion efficiency difficulty caused by separate explorations of the spatial communications and the temporal dependencies, which could result in delayed and biased predictions. To address that, we propose a Traffic Gated Graph Neural Networks (Traffic-GGNN) for real-time-fused spatial-temporal representation modeling. Firstly, we adopt bidirectional message passing to capture the location-wise spatial interactions. Secondly, we apply a GRU-based module to explore and aggregate the spatial interactions with the temporal correlations in a real-time fusion way. Lastly, we introduce a self-attention mechanism to reweight the location-based importance and produce the final prediction. Moreover, our proposed model allows end-to-end training thus it is easy to scale to diverse types of traffic datasets and yield better efficiency and effectiveness on three real-world datasets (SZ-taxi, Los-loop, and PEMS-BAY).

Index Terms—Gated graph neural networks, self-attention, spatial-temporal graph, traffic flow prediction.

I. INTRODUCTION

RECENT years have witnessed the boom of Spatial-Temporal Graph Neural Networks (STGNNs), which are efficient variants of GNNs for dealing with spatial-temporal graph-based data [1], [2]. The key idea of STGNNs is to stack multiple modules/layers that iteratively carry out the following two steps at each module/layer: spatial/temporal message passing with non-linear neighborhood transformation,

Manuscript received 12 May 2021; revised 31 January 2022; accepted 30 March 2022. Date of publication 2 May 2022; date of current version 11 October 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002201, in part by the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, and in part by the State Key Program of National Science Foundation of China under Grant 61836006 and Grant 61432014. The work of Ping Li was supported in part by the National Natural Science Foundation of China under Grant 61873218, in part by the Applied Basic Research Foundation of Sichuan Provincial Science and Technology Department under Grant 18YYJC1147, and in part by Southwest Petroleum University (SWPU) Innovation Base Foundation under Grant 642. The work of Jin Zheng was supported in part by the Scientific Research Starting Project of SWPU under Grant 2019QHZ016 and in part by the Higher Education Reform Research Project of SWPU under Grant 2018041. The Associate Editor for this article was S. Pan. (*Corresponding author: Jin Zheng*)

The authors were with the School of Computer Science, Southwest Petroleum University, Chengdu, China, 610500 (e-mail: 798433627@qq.com; zhengjin@swpu.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3168590

and followed by the aggregation to yield a fused representation of spatial embeddings and temporal embeddings. Thus, the higher-order spatial-temporal characteristics of a graph node could be effectively explored [3]. As many real-world data show the spatial-temporal graph structure, these STGNN-based models have been widely applied and expressed competing performance, such as [4]–[8].

In this paper, we focus on applying the STGNN-based method to Traffic Flow Prediction (TFP) tasks. TFP is a typical problem of spatial-temporal data forecasting. The key to solving such problems is how to explore the inherent spatial-temporal patterns from the nonlinear and complex spatial-temporal data and make satisfactory predictions.

Early TFP models are generally developed based on time series analysis methods such as statistical methods [9] or machine learning methods [10]. Yet, it is difficult for them to handle the complex dynamics of the practical traffic data and simultaneously process the coupling relationships of spatial as well as temporal. In recent years, deep learning(DL) based methods for TFP tasks have been widely studied [11]–[13]. To deal with irregularity of the traffic grids, the researchers combine graph neural networks (GNN), which are efficient in capturing non-Euclidean correlations, with convolutional neural networks (CNN) or recurrent neural networks (RNN) and formulate a novel kind of deep neural networks, i.e. STGNN [14].

Specifically, STGNNs often employ a two-staged paradigm: firstly, use GNN-based modules to explore spatial communications of graph-based traffic data and adopt RNN [15] or CNN [14] to effectively extract the temporal correlations of location-based features. Then, use a fusion module, for instance, a Multilayer Perceptron (MLP), to aggregate the spatial embeddings with the temporal embeddings and generate a final prediction. These STGNNs-based models show superior performance and make great progress for TFP tasks.

Despite the relative success of STGNN-based TFP systems, we argue that two important problems in those models still remain unsolved. On one hand, these methods still suffer from post-fusion efficiency difficulty. In the related literature, STGNN-based methods generally explore the temporal/spatial characteristics separately and then combine them to an integrated traffic flow representation. Due to this separately extracting paradigm, the fusion of spatial-temporal embeddings is processed after all explorations accomplished, such

as [16], [17]. The delayed-fusion policy could lead to that the final predictions are commonly short of the global context of traffic dynamic topology and partial to spatial factors or temporal aspects, which could eventually have a negative impact on effective prediction. On the other hand, most of the current STGNN-based TFP models neglect the influence of traffic flow directions as well as significance differences of location-based representations, which could result in remarkable biases of the prediction.

To tackle the aforementioned problems, we propose the Traffic Gated Graph Neural Networks (Traffic-GGNN) which is specifically designed for traffic flow prediction with real-time-fused spatial-temporal representation modeling. In Traffic-GGNN, firstly, we adopt bidirectional message passing to capture the location-wise spatial interactions. Secondly, we apply a GRU-based module to recurrently explore the higher-order of spatial interactions, extract the temporal correlations and aggregate them to a fused spatial-temporal embedding in a real-time fusion way. The number of this recurrence is the product of the step of message passing and the temporal length of traffic data. Lastly, we introduce a self-attention mechanism to reweight the location-based importance and produce the final prediction.

Note that different from the original GGNN [18], the recurrent processing in Traffic-GGNN is not only involved with temporal but also with spatial. Besides, the self-attention calculation does not exist in the original GGNN [18].

Moreover, our proposed model allows end-to-end training and joint tuning of all components via back-propagation, thus it is easy to scale to diverse types of traffic datasets. We conduct extensive experiments on three real-world datasets, SZ-taxi, Los-loop, and PEMS-BAY, and the results clearly demonstrate better efficiency and effectiveness of our proposed Traffic-GGNN.

The main contributions of our work are listed as follows:

- We process message passing in a bidirectional way rather than the traditional undirected style to capture finer-grained location-wise spatial interactions.
- We apply a recurrent aggregating method to learn spatial-temporal embedding in a real-time fusion way rather than a separate extraction and delayed fusion paradigm. The recurrent aggregation and real-time fusion method make it positive to explore spatial interactions and temporal correlations as well as to aggregate them to a fused spatial-temporal representation in real time.
- We adopt a self-attention mechanism to reweight the location-based importance so as to produce more accurate predictions.

The rest of the paper is organized as follows: We first discuss the related works and then give a detailed description of the Traffic-GGNN model. Next, we report experiments in Section IV. We conclude our work and discuss the interesting work that can be done in the future work in Section V.

II. RELATED WORK

In this section, we briefly review previous methods of traffic forecasting and models of gated graph neural networks.

A. Traffic Forecasting

Traffic flow forecasting problem has been attracted to the research society of spatial-temporal data mining. Early traffic prediction models are mainly statistic based (such as historical average [19], Autoregressive Integrated Moving Average (ARIMA) [20], vector autoregressive [21], etc.) or machine learning based methods (including K-nearest neighbor [22], support vector machine (SVM) [23], etc.). However, the aforementioned works generally lack the ability to learn the combined embeddings of spatial-temporal characteristics.

Under this premise, deep learning methods with powerful feature extraction capabilities and nonlinear fitting capabilities have been proposed to traffic prediction tasks [24]–[26]. For instance, [12] and [27] transform the road network into a regular grid, and then adopt CNN to extract spatial features. Yet, they still have some limitations. A key limitation is that the road network is irregular in nature. Therefore, converting forcibly traffic data into standard 2D or 3D grid data is most likely to result in the loss of topological information.

To tackle that, graph neural networks (GNN) are introduced to model the spatial-temporal network data and usually get promising prediction results [14], [17], [28]–[31]. Among them, some works focus on more efficient learning of spatio-temporal representations. For instance, T-GCN [16] combines graph convolutional network (GCN) and gated recurrent unit (GRU) [32] to obtain the spatial-temporal correlation from traffic data. On the other hand, some researchers designed more effective ways to retain and explore the topological information [3], [33], [34]. For instance, Yan *et al.* [35] propose a Traffic Transformer to dynamically extract spatiotemporal features by multi-head attention and masked multi-head attention mechanism and find the influential part of road networks by analyzing the attention weight matrixes.

B. Gated Graph Neural Network

Graph neural networks (GNNs) are proposed to learn graph-structure data and have achieved excellent performance in various tasks [36]–[38]. To deal with sequence problems, Li *et al.* [18] proposed a spatial domain message passing model, Gated Graph Neural Network (GGNN), which utilized fixed-recurrence GRU. Due to good performance in spatio-temporal data mining, GGNN has been applied to various tasks, such as recommendation [39], image classification [40], situation recognition [41], and financial event prediction [6]. For instance, TextING [7] builds individual graphs for each document and then applies GGNN to learn the embeddings of the word nodes. To our knowledge, our Traffic-GGNN is the first GGNN-based model applied in traffic flow prediction (TFP). Meanwhile, for better adaptive to TFP, we design bi-directional message passing which is different from the original in [18].

III. TRAFFIC GATED GRAPH NEURAL NETWORKS

In this section, we first formulate the TFP problem and then introduce the overview of our proposed method, followed by the elaborate detail of each module.

TABLE I
SUMMARY OF NOTATION

Notation	Description
N	Number of locations
T	Length of time span
τ	Lead time of the prediction
$G^t = (V, E)$	Traffic state (a DAG) at a specific time point t , where V denotes collection of locations, E denotes set of connections of V
K	Steps of bi-directional message passing (BMP) at time t
$h_i^t \in R^D$	Traffic flow state at time t corresponding to location i
x_i^t	traffic flow record embedding vector at time t corresponding to location i
$A^t \in R^{N \times 2N}$	connectivity matrix at time t
A_F^t	Forward-directional adjacency matrix
A_R^t	Reversed-directional adjacency matrix
$a^{t(k)}$	Inputs of GRU
$H^{t(k)}$	Traffic flow states at step k corresponding to time t
$Attn^T$	Attention score of the importance of each location
\hat{Y}	Final prediction

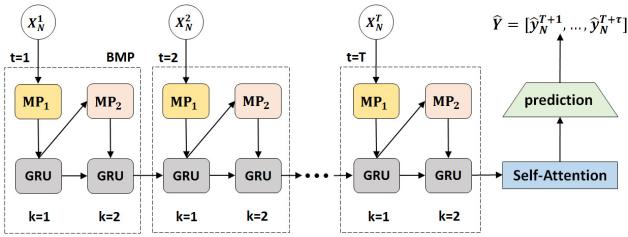


Fig. 1. The framework of Traffic-GGNN. We take the traffic flow profiles as input and produce the final prediction through the bi-directional message passing and GRU module. Note that we only apply one GRU which means all the corresponding parameters of the GRUs shown in this figure are shared.

A. Problem Formulation

In this paper, we formulate the TFP problem as a spatial-temporal regression task. Suppose the traffic records consist of N locations and T time spans. Specifically, the term of location represents a traffic object, which can be a sensor [17], [42], [43], a road segment [14], [44], a road intersection [45]–[47], and the time granularity of time span may refer to a week, a day or a period of random length. Thus, we denote the multi-location traffic flow profile over a time span of size T as $X = [X_N^1, \dots, X_N^t, \dots, X_N^T] \in R^{N \times T}$. The objective of the TFP problem is to take both current traffic statistics and external traffic factors into account and predict a traffic profile at a future time point $T + \tau$ where τ represents the lead time of the prediction.

B. Overview

Distinguished from the previous related works, we organize traffic flow statistics in the form of bi-directional spatial-temporal graph, then model spatial dependencies as well as temporal correlations real-timely in a recurrent aggregating process with Traffic-GGNN, and produce an attentional prediction (see Fig. 1).

To this end, our proposed Traffic-GGNN method comprises four key modules: 1) bi-directional spatial-temporal graph

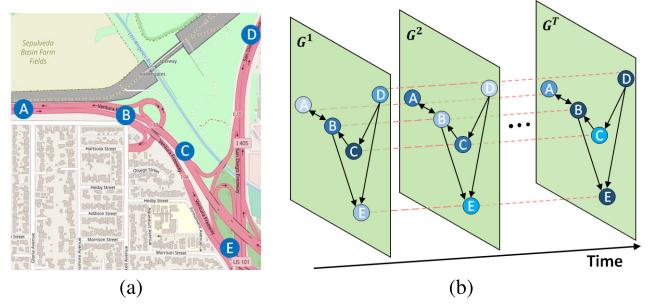


Fig. 2. We define the traffic grids (as shown in (a)) as a set of bi-directional spatial graph along with the time span (as shown in (b)). Different colors indicate changes of traffic flow profiles with time.

construction to represent traffic flow statistics, 2) bi-directional message passing to capture the location-aware traffic correlations, 3) spatial-temporal fusion to update the locational traffic flow states in real time and 4) self-attentional output to produce the final spatial-temporal representations. In the following, we will introduce the details of our proposed method.

C. Bi-Directional Spatial-Temporal Graph

In this study, we represent the traffic flow profiles as a set of graphs denoted as $\mathcal{G} = [G^1, \dots, G^t, \dots, G^T]$ where G^t refers to the traffic state at a specific time point t . Meanwhile, we define $G^t = (V, E)$ as a Directed Acyclic Graph (DAG), where V is the collection of locations and E represents the set of connections of those locations.

Generally, the traffic flow situations are not only dynamic in different time steps, but also direction related in physical connections. If two locations have spatial connecting relationship in the traffic flow records, there are two directed edges in reverse directions between the two corresponding nodes in G^t , since that the interactions in two directions may potentially lead to different impacts on traffic situations in those locations. For instance, the traffic congestion at location A may propagate the jam state to the neighbouring road intersection B of which the initial situation is smooth, but the process in the inverse direction is unlikely the same (see Fig. 2).

Note that in the most of related GNNs-based TFP works, the adjacency of the traffic graph is set in an undirected style. The root cause is that it is normally difficult for those proposed methods to efficiently handle with the scale complexity brought about by the directed graph. In contrast to that, the GGNN framework has a significant advantage of structural simplicity and computing efficiency on the directed graph [18]. The detail will be discussed in the following sections.

D. Bi-Directional Message Passing

In this study, we utilize K steps bi-directional message passing (BMP) at time t to model K -order traffic interactions between locations on G^t (see Fig. 3). Generally, we set the value of K according to the scale of the traffic flow graph and $K = 2$ is enough for most of the cases.

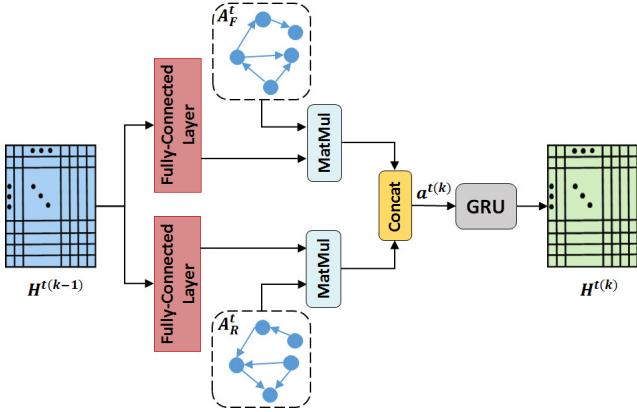


Fig. 3. The overall process of bi-directional message passing and spatial-temporal fusion. Note that for simplicity, only one round of those operations is shown in this figure.

1) *State Initialization*: At the beginning of BMP, each location i is associated with a traffic flow state h_i^t and the overall traffic flow profile at time t is composed of these location-aware states as $H^t = [h_1^t, \dots, h_i^t, \dots, h_N^t]$, where N is the number of locations.

The initial location-wise traffic flow states at the k -step of BMP are that of the input traffic flow graphs, i.e., the traffic flow record embedding vectors $X^t = [x_1^t, \dots, x_i^t, \dots, x_N^t]$. In particular, the D -dimensional embedding vector x_i^t of the i -th location at time t is utilized as the initial state vector of the corresponding location i , which can be formalized as

$$H^{t(k-1)} = X^t, k = 1 \quad (1)$$

where $H \in R^{N*D}$.

2) *Spatial Passing*: For each step k , the location-wise states communicate with their neighbours' in a directed way. Once the k -step process is completed, the location-wise traffic states will be k -th updated. Here, we explore the multi-location traffic flow interactions by BMP operations.

Specifically, we use the connectivity matrix $A^t \in R^{N*2N}$ to represent how those locations in the traffic flow graph G^t could communicate information with each other at time t . Note that A^t is the concatenation of A_F^t and A_R^t (see Fig. 3), where A_F^t refers to the forward-directional adjacency matrix, and A_R^t refers to the reversed-directional adjacency matrix, which is the transposed matrix of A_F^t . In addition, A^t is a public parameter in the whole k steps of BMP and $2N$ means that the interactions in two directions between locations are different. Then the A_F^t is defined as:

$$A_{F(i,j)}^t = \begin{cases} 1 & \text{if } v_i \rightarrow v_j \in E \\ 0 & \text{others} \end{cases} \quad (2)$$

Hence, the traffic flow states $a^{t(k)}$ at step k can be updated as:

$$a^{t(k)} = ((A_F^t H^{t(k-1)}) || (A_R^t H^{t(k-1)})) W_a + b_a, \quad k \in [1, 2, \dots, K] \quad (3)$$

where A_F^t , A_R^t , and $H^{t(k-1)}$ are the interaction matrices at time t and traffic flow states at step $k-1$ respectively. $||$ is

the concatenation operator. W_a is the learnable transformation matrix and b_a is the corresponding bias. Note that W_a can be set to a public parameter in the whole k steps of BMP as well as a series of trainable matrices W_a^K where each W_a^k corresponding to a k step.

3) *Recurrent Spatial Update*: Considering the related works [48]–[51] have proved that the excellent performance of recurrent neural networks (RNNs) to combine the low-order and high-order interactions together, we then employ an RNN to update the local traffic flow representations upon each location i . In this work, we follow the conventional GGNN [18] and update the spatial traffic flow representations via GRU [32] (see Fig. 1). Thus, the recurrent spatial update can be defined as:

$$H^{t(k)} = GRU(H^{t(k-1)}, a^{t(k)}), \quad k \in [2, 3, \dots, K] \quad (4)$$

The recurrence can be formalized in detail as:

$$z^{t(k)} = \sigma(W_z a^{t(k)} + U_z H^{t(k-1)} + b_z) \quad (5)$$

$$r^{t(k)} = \sigma(W_r a^{t(k)} + U_r H^{t(k-1)} + b_r) \quad (6)$$

$$\tilde{H}^{t(k)} = \tanh(W_h a^{t(k)} + U_h(r^{t(k)} \odot H^{t(k-1)}) + b_z) \quad (7)$$

$$H^{t(k)} = \tilde{H}^{t(k)} \odot z^{t(k)} + H^{t(k-1)} \odot (1 - z^{t(k)}) \quad (8)$$

where $z^{t(k)}$ and $r^{t(k)}$ are update gate and reset gate respectively at the k -th step of time t to explore what contributes to the current traffic flow representation, σ is the sigmoid function, \odot is element-wise multiplication, all W , U , and b are trainable weights and biases.

After K update steps, i.e., one round of bi-directional message passing accomplished, we can obtain the location-wise traffic flow state $H^{t(k=K)}$ at time t .

E. Spatial-Temporal Fusion

Besides the spatial correlations, the traffic flow statistics commonly have sequential dependencies over time. To handle with this, we still employ the same GRU module used in the BMP module to explore the temporal dependency of corresponding historical traffic flow states upon each location i and produce an aggregated representation of spatial and temporal embeddings in the meantime (see Fig. 1). We denote this recurrent process as spatial-temporal fusion. Specifically, we design that an update process contains T steps and each single t -step includes one round of BMP containing K steps. Therefore spatial-temporal fusion can be defined as:

$$H^{(t_k=1)} = GRU(H^{(t-1)(k_{t-1}=K)}, a^{(k_t=1)}), \quad t \in [1, 2, \dots, T] \quad (9)$$

Particularly, when $t = 1$, $H^{(t-1)(k_{t-1}=K)} = X^1$.

Then the whole processes in the spatial-temporal fusion can follow Fig. 1. In this way, the spatial aggregation and the temporal integration are combined effectively in one single GRU module, which could bring about good performance to represent the spatial-temporal characteristics of traffic flow and excellent efficiency for model training.

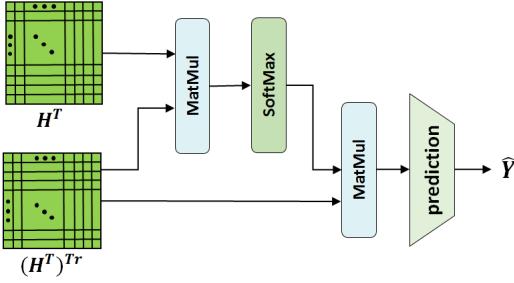


Fig. 4. The overall process of self-attention operation. Note that in this work, this attentional process only takes the H^T as inputs.

F. Self-Attentional Output and Optimization

Along with the T steps of spatial-temporal fusion processing, we can obtain a sequential of traffic flow profile representations $H = [H^1, \dots, H^t, \dots, H^T], t \in [1, 2, \dots, T]$. Here we introduce a self-attention mechanism to calculate the importance of each location and produce the final output prediction. In this work, this attentional processing only takes the H^T as inputs. Specifically,

$$(H_{att}^T) = (H^T)^{Tr} \cdot Attn^T \quad (10)$$

and

$$Attn^T = \frac{\exp(H^T \cdot (H^T)^{Tr})}{\sum_1^N \exp(H^T \cdot (H^T)^{Tr})} \quad (11)$$

Then we feed the H_{att}^T into a multi-layer perceptron to produce the final prediction \hat{Y} , which is defined as:

$$\hat{Y} = W_y H_{att}^T + b_y \quad (12)$$

where H_{att}^T is the attentional embedding, W_y is the learnable mapping matrix and b_y is the corresponding bias.

Considering traffic flow prediction problem is a regression task, we use RMSE as the loss function, which can be defined as follow:

$$\mathcal{L} = \sqrt{\frac{1}{S} \sum_{i=1}^S (y_i - \hat{y}_i)^2} \quad (13)$$

where S is the number of training samples and i indexes the training samples. All model parameters are updated via minimizing the RMSE Loss using Adam algorithm [52].

IV. EXPERIMENTS

In this section, we first describe the datasets, baseline methods, and experiment settings. Then we evaluate the performance of the proposed model, Traffic-GGNN, show the influences of different network configurations. Next, we analyze the effectiveness of the self-attention mechanism. At last, we visualize the prediction results of our model.

A. Datasets

In order to evaluate the performance of our model, we conduct experiments on three real-world datasets, SZ-taxi, Los-loop, and PEMS-BAY. All three datasets contain traffic speed as traffic information. The SZ-taxi dataset is the traffic

TABLE II
SUMMARY STATISTICS OF THREE PUBLIC TRAFFIC NETWORK DATASETS

Datasets	Nodes	Edges	Time Steps	Max	Mean
SZ-taxi	156	532	2976	86.4	12.2
Los-loop	207	2833	2016	70.0	58.9
PEMS-BAY	325	2369	52116	85.1	62.6

data of Luohu District, Shenzhen from Jan. 1 to Jan. 31, 2015, including 156 major roads in this area. The Los-loop dataset is collected from 207 loop detectors in the highway of Los Angeles County from Mar. 1 to Mar. 7, 2012. The PEMS-BAY dataset contains 6 months of data recorded by 325 traffic sensors ranging from Jan. 1 to June. 30, 2017 in the Bay Area. In SZ-taxi dataset, the traffic speed is aggregated every 15 minutes, while in the other datasets, it is aggregated every 5 minutes. Finally, in the experiments, we choose 80% of the data as the training set and the remaining 20% as the test set. Table II shows the detailed statistics of the three datasets.

B. Experiment Settings

Our model is implemented based on Pytorch framework. Following the previous works [16], we use the data of the previous 60 minutes as input to predict the traffic speed in the next 15, 30, 45, and 60 minutes. In addition, we train all models on three datasets for 1000 epochs with a learning rate of 0.001. During the training phase, the batch size is 32 and we adopt the Adam optimizer to train our model. All experiments are conducted on a single NVIDIA Tesla T4 GPU with 16GB Ram.

We compare our model with the following six baseline methods:

(1) ARIMA [20]: Autoregressive Integrated Moving Average model, which is a classical model for predicting the future values in a time series.

(2) SVR [53]: Support Vector Regression model, which uses a linear support vector machine for regression tasks.

(3) ASTGCN [17]: Attention based Spatial-Temporal Graph Convolutional Networks, which designs the spatial-temporal attention and the spatial-temporal convolution to capture the dynamic spatial-temporal correlations in traffic data.

(4) T-GCN [16]: Temporal Graph Convolutional Network, which is combined with the graph convolutional network and the gated recurrent unit for traffic forecast.

(5) A3T-GCN [54]: Attention Temporal Graph Convolutional Network, which uses the attention mechanism and T-GCN to capture global temporal dynamics and spatial correlations.

(6) GWN [33]: Graph WaveNet, which combines a learnable adaptive dependency matrix and a stacked dilated 1D convolution component for traffic prediction.

In order to better verify the effectiveness of Traffic-GGNN, we use the following three metrics for evaluation:

(1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{\tau N} \sum_{j=1}^{\tau} \sum_{i=1}^N (y_i^j - \hat{y}_i^j)^2} \quad (14)$$

(2) Mean Absolute Error (MAE):

$$MAE = \frac{1}{\tau N} \sum_{j=1}^{\tau} \sum_{i=1}^N |y_i^j - \hat{y}_i^j| \quad (15)$$

(3) Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{\tau N} \sum_{j=1}^{\tau} \sum_{i=1}^N \frac{|y_i^j - \hat{y}_i^j|}{|y_i^j|} \times 100\% \quad (16)$$

where τ is the number of time slots; N is the number of roads; y_i^j and \hat{y}_i^j respectively represent the real traffic value and predicted value of the i -th road at the j -th time. Y and \hat{Y} represent the set of real traffic information and predicted traffic information of all roads at different times. The smaller the value of RMSE and MAE is, the better the prediction effect is.

C. Comparison and Result Analysis

Table III shows the performance of Traffic-GGNN and other baseline models for 15, 30, 45, and 60 minutes traffic forecasting on three datasets. We have the following observations from Table III:

(1) The approaches based on deep learning almost have higher prediction precision than the traditional time series analysis methods and machine learning methods (e.g., ARIMA and SVR) on three datasets in terms of all evaluation metrics. The reason for these results is the limited nonlinear modeling abilities of the ARIMA and SVR models. In particular, ARIMA has a high requirement for the stability of data, but the traffic data is complex with many influencing factors. Hence, ARIMA presents the least ideal prediction results. In contrast, these deep learning models can not only model nonlinear data but also obtain more spatiotemporal features by introducing traffic graphs. Therefore, the deep learning methods have a better performance.

(2) Traffic-GGNN achieves more advanced prediction performance than deep learning-based methods on the long-term and short-term predictions of the three datasets. For example, for the 15-minute traffic prediction task on the Los-loop dataset, Traffic-GGNN shows a 3.43% improvement over GWN and an increase of 6.65% over ASTGCN on RMSE. Similar results also appear in the comparison of other time spans prediction tasks and evaluation metrics. The architectural factors of our model facilitate these improvements. Firstly, other deep learning-based methods ignore the importance of the direction of traffic flow, while Traffic-GGNN uses the bidirectional message passing module, which can capture more complex spatial relationships in real traffic networks. Moreover, our model applies a recurrent aggregating method to learn spatial-temporal embedding in a real-time fusion way rather than a separate extraction and delayed fusion paradigm. This helps to learn more fine-grained dynamic spatial-temporal relationships. Finally, the self-attention mechanism can capture the importance of the road after spatial-temporal fusion, instead of focusing only on the importance of global time like A3T-GCN, which can effectively improve the prediction accuracy of the model.

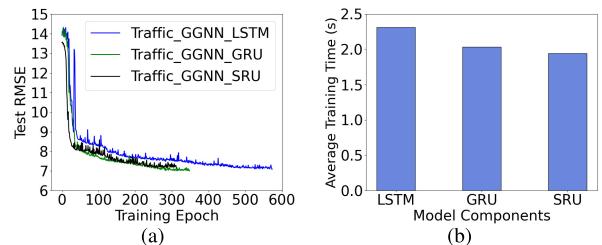


Fig. 5. Comparison of test RMSE and average training time cost of different model components. (a) Test RMSE versus the number of training epochs. (b) Average training time per epoch using different model components.

D. Ablation Study

1) *Model Component*: For traffic forecasting tasks, modeling the temporal dependence of sequence data is a significant method to improve the prediction precision. Recurrent neural networks (RNNs) have advantages in learning the nonlinear characteristics of sequence data due to their model structure. However, the traditional recurrent neural network is not suitable for long-term prediction because it easily causes gradient disappearance and gradient explosion. The variants of RNNs, such as LSTM [55], GRU [32], SRU [56], etc. can solve the above problems well, so they can be used to replace the RNN module.

We plot the test RMSE versus the training epochs and average training time per epoch to further investigate the performance of the three deep learning models. As shown in Fig. 5(a), it is obvious that GRU and SRU are easier to converge than LSTM, and GRU has the lowest RMSE. In addition, it can be seen that the average training time per epoch of LSTM is longer than that of GRU and SRU from Fig. 5(b). This is mainly because LSTM has a more complex model structure, more parameters, and slower training ability. Therefore, after comparing the training time cost and convergence effect of LSTM, GRU, and SRU, we choose GRU to obtain the temporal dependence of traffic sequence data.

2) *Hidden Units*: Usually, the number of hidden units is the direct cause of overfitting, so the number of hidden units is a crucial hyperparameter. In order to verify the influence of the number of hidden units on Traffic-GGNN and select the optimal value, we conduct a comparison experiment on the three datasets. In this experiment, we select the number of hidden units from [16, 32, 64, 96, 128, 160] to analyze its effect on prediction precision.

As shown in Fig. 6, the horizontal axis represents the number of hidden units and the vertical axis represents different metrics. Fig. 6(a), Fig. 6(b), and fig. 6(c) respectively show the changes in RMSE and MAE for different hidden units on SZ-taxi, Los-loop, and PEMS-BAY datasets. We can find that when the number of hidden units in the three datasets is 96, 32, and 64, the prediction error is the lowest. At the same time, it can be seen from Fig. 6 that when the number of hidden units exceeds a certain value, the prediction error will become larger. This is mainly because the hidden units are too large, the model is prone to overfitting.

3) *Bidirectional Message Passing*: In this part, we test the effectiveness of message passing in a bidirectional way

TABLE III
PERFORMANCE COMPARISON OF TRAFFIC-GGNN AND OTHER BASELINE METHODS ON SZ-TAXI, LOS-LOOP AND PEM-S-BAY DATASETS

Datasets	Models	15min			30min			45min			60min		
		RMSE	MAE	MAPE									
SZ-taxi	ARIMA	6.8044	4.6801	29.73%	6.8042	4.6798	29.72%	6.8040	4.6796	29.72%	6.7963	4.6758	29.71%
	SVR	4.1521	2.6271	21.09%	4.1678	2.6887	21.83%	4.1932	2.7371	22.35%	4.2203	2.7762	22.76%
	ASTGCN	4.0899	2.6898	21.38%	4.1130	2.7228	22.03%	4.1040	2.7478	22.45%	4.0875	2.7654	22.89%
	T-GCN	4.1372	2.8245	23.08%	4.1605	2.8450	23.49%	4.2114	2.8838	23.66%	4.2007	2.8873	23.95%
	A3T-GCN	4.0715	2.7489	22.27%	4.0762	2.7312	22.58%	4.0685	2.7298	22.30%	4.0812	2.7414	22.44%
	GWN	4.2564	2.7335	21.82%	4.3022	2.7823	22.28%	4.3444	2.7971	22.34%	4.3785	2.8291	22.67%
	Traffic-GGNN	4.0312	2.6485	21.36%	4.0516	2.6812	21.64%	4.0572	2.7012	22.01%	4.0668	2.7090	22.05%
Los-loop	ARIMA	10.0542	7.7060	20.84%	10.0591	7.7110	20.86%	10.0626	7.7138	20.88%	10.0659	7.7166	20.90%
	SVR	5.6506	3.8639	9.33%	6.7343	4.6472	11.51%	7.3033	3.8422	13.21%	8.0274	4.2193	14.65%
	ASTGCN	5.2510	3.2714	8.05%	6.1439	3.7637	9.68%	6.7546	3.9823	10.69%	7.4662	4.4346	12.29%
	T-GCN	5.2098	3.2745	8.34%	6.1169	3.8069	9.96%	6.7829	4.1837	11.36%	7.2660	4.4182	12.11%
	A3T-GCN	5.1176	3.1748	8.08%	6.1033	3.7329	9.98%	6.7860	4.1711	11.28%	7.2329	4.5552	12.21%
	GWN	5.0759	2.8587	7.42%	6.0304	3.4098	9.06%	6.7060	3.7314	10.33%	7.1535	4.0811	12.01%
	Traffic-GGNN	4.9018	2.8494	7.29%	5.8397	3.3744	9.08%	6.5748	3.7032	10.22%	7.0873	4.0705	11.40%
PEMS-BAY	ARIMA	8.6429	6.0465	14.31%	8.6431	6.0465	14.31%	8.6432	6.0465	14.31%	8.6433	6.0465	14.31%
	SVR	4.9051	4.3504	7.57%	5.6174	4.8502	8.60%	6.1754	5.2290	9.42%	6.6162	5.5201	10.10%
	ASTGCN	2.4559	1.2457	2.55%	3.1657	1.5252	3.29%	3.6649	1.8286	3.95%	4.1187	2.0889	4.51%
	T-GCN	2.7228	1.4828	3.05%	3.3142	1.7776	3.70%	3.6851	1.9682	4.16%	3.9653	2.1080	4.51%
	A3T-GCN	2.6493	1.4354	2.94%	3.4066	1.8443	3.86%	3.7131	1.9793	4.26%	4.1884	2.2425	4.79%
	GWN	2.5278	1.2108	2.40%	3.3391	1.4881	3.19%	3.9117	1.7238	3.75%	4.3515	1.8870	4.39%
	Traffic-GGNN	2.3116	1.1833	2.38%	3.0247	1.4795	3.15%	3.4918	1.7048	3.75%	3.7706	1.8506	4.15%

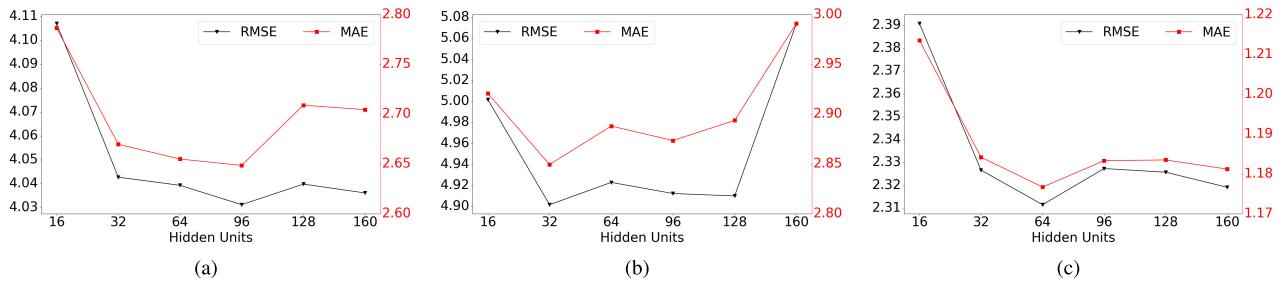


Fig. 6. Infuence of different hidden units in the test set on three datasets. (a) SZ-taxi dataset. (b) Los-loop dataset. (c) PEMS-BAY dataset.

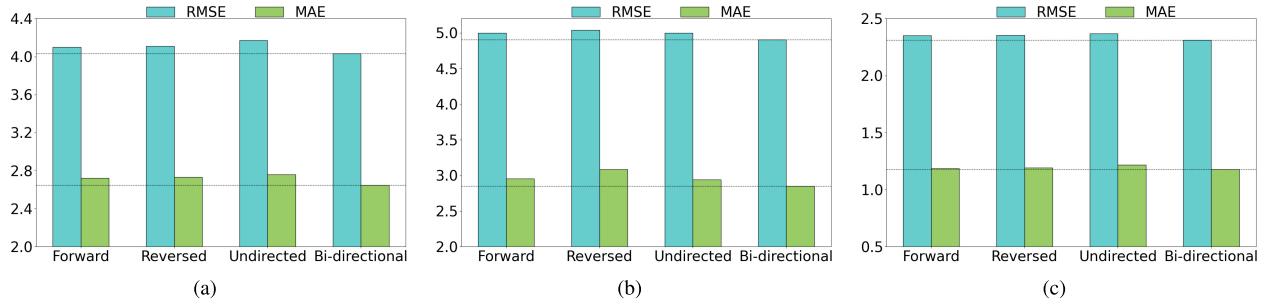


Fig. 7. The impact of different message passing ways on traffic forecasting results. (a) SZ-taxi dataset. (b) Los-loop dataset. (c) PEMS-BAY dataset.

through comparative experiments. As shown in Fig. 7, we use two metrics to evaluate the prediction performance of unidirectional message passing, bidirectional message passing, and undirected message passing on SZ-taxi, Los-loop and PEMS-BAY datasets. In detail, unidirectional message passing uses the forward-directional adjacency matrix or the reversed-directional adjacency matrix to represent the spatial relationships between nodes. Bidirectional message passing uses the forward-directional adjacency matrix and the reversed-directional adjacency

matrix to jointly transmit spatial information. Undirected message passing uses the traditional symmetric adjacency matrix.

The vertical axis represents the metrics and the horizontal axis represents the Traffic-GGNN model with different message passing methods. It can be observed that the model with bidirectional message passing achieves the best predictive performance on both datasets. This is mainly because the interaction in different directions may potentially lead to different impacts on traffic conditions, and the use of bidirectional

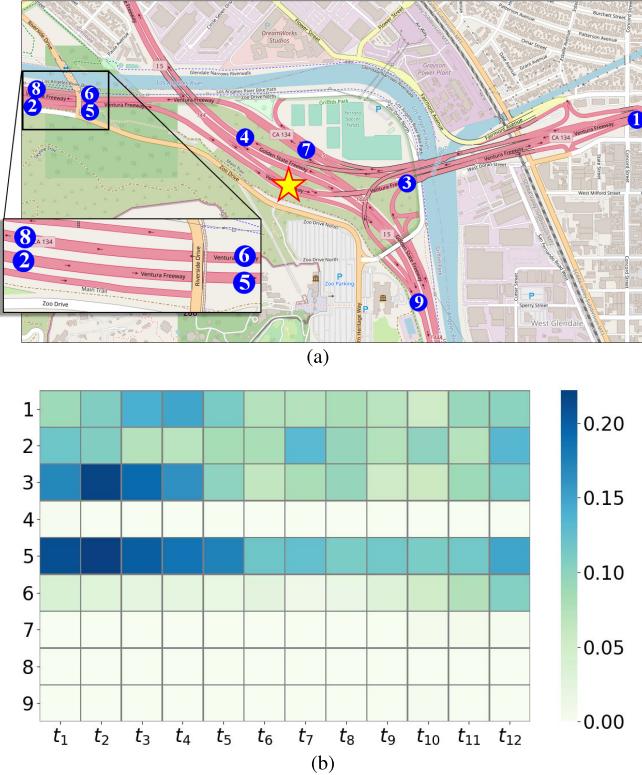


Fig. 8. Visualization of the geographic location of nodes and heat map of attention weight in Los-loop dataset. (a) The geographic location of some nodes on the map. The star denotes the center node. (b) Heat map of attention weights on different time slots.

message passing can capture more spatial dependence on different directions.

E. Evaluation on Attention

In this part, we intuitively explain the role of the attention mechanism through the visualization of the geographic location of partial nodes and heat map of attention weight. As shown in Fig. 8(a), we extract a sub-graph with 10 nodes from the Los-loop dataset and mark the nodes on the map. The star represents the center, and its neighboring nodes are labeled 1-9. Fig. 8(b) shows the correlation between the neighboring nodes and the center at different time slots through the heat map. The number on the vertical axis represents the i -th neighboring node and the horizontal axis represents the i -th time slots.

As can be seen from Fig. 8(b), the influence of neighboring nodes on the center is dynamic at different moments. Furthermore, we can find that nodes 1, 2, 3, and 5 are extremely relevant and important when predicting the traffic speed of the center. In particular, the traffic speed at the center is closely related to that at node 3 and node 5. Because these nodes are close to the center and the direction of traffic flow is consistent in space, as shown from Fig. 8(a). Hence, the self-attention method in our model can capture the spatial-temporal dynamic correlation between nodes in the real traffic network. Meanwhile, compared with similar research such as in [35], our proposed Traffic-GGNN could learn better node correlations over different time slots based on a more excellent

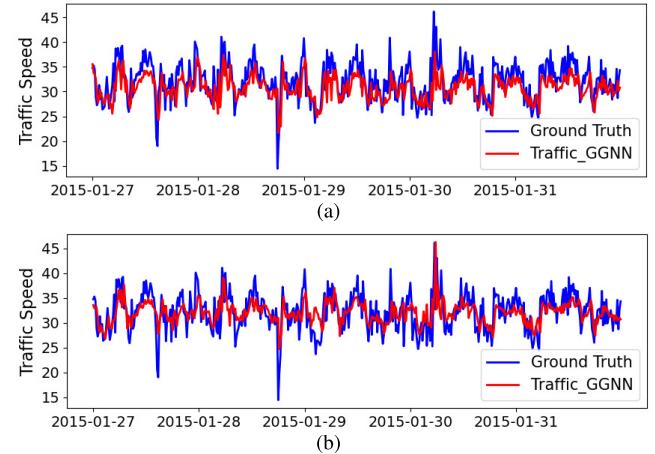


Fig. 9. The visualization results for different prediction tasks. (a) The visualization results for prediction horizon of 15 minutes. (b) The visualization results for prediction horizon of 60 minutes.

TABLE IV
THE COMPUTATION TIME ON THE PEMBS-BAY DATASET

Models	Computation Time	
	Training (s/epoch)	Testing (s)
ASTGCN	263.24	21.98
T-GCN	38.62	4.33
A3T-GCN	42.03	4.60
GWN	563.27	44.27
Traffic-GGNN	113.35	10.56

attention mechanism. Furthermore, the attention method in Traffic-GGNN can help us find the key nodes in the traffic network, so as to take effective measures to alleviate the traffic problems.

F. Visualization Results

In order to better understand the Traffic-GGNN model, we visualize the prediction result of one road on the SZ-taxi dataset from January 27 to 31, 2015. Visualization results for prediction horizons of 15 and 60 minutes are shown in Fig. 9. The horizontal axis represents time and the vertical axis represents traffic speed.

From Fig. 9 we can find that our model can achieve good results whether it is a short-term forecast or a long-term forecast. Moreover, the Traffic-GGNN model can predict the start and the end of the peak hours and capture the trend of traffic information. This is because Traffic-GGNN can obtain the spatial-temporal features and capture the spatial-temporal dynamic correlation between nodes. Therefore, the prediction result of the Traffic-GGNN model is similar to the real traffic pattern.

G. Computation Cost

To further discuss the performance of real-timely fusion in Traffic-GGNN and post-fusion mechanism in related works, we compare their computation time (including training and testing phases) on the PEMBS-BAY dataset as shown in Table IV. During training, Traffic-GGNN is almost five times

faster than GWN, almost twice faster than ASTGCN, yet more than twice slower than T-GCN and A3T-GCN. Similar results are also presented during the testing phase. This is mainly because that Traffic-GGNN has a more complex model structure than T-GCN and A3T-GCN. However, considering the prediction accuracy of the models in Table III, the rank of Traffic-GGNN in Table IV is acceptable and the real-timely fusion in Traffic-GGNN is relatively more excellent than the post-fusion mechanism.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose Traffic-GGNN model for the urban traffic flow forecasting problem. Experiments on three real-world datasets illustrate its superior performance. Note that, the Traffic-GGNN employs only one GRU to extract the spatial-temporal correlations in a real-time fusion style rather than in post-fusion way (In post-fusion way, the fusion module is generally processed after all spatial and temporal modules separately accomplished). That is the most different aspect from the existing related works. For future works, we plan to extend Traffic-GGNN to feed more complex traffic flow profiles such as traffic density [57], environmental factors [58], and social events. Furthermore, we would also focus on model capability and performance stability, for instance, over-fitting problem.

REFERENCES

- [1] X. Wang *et al.*, “Traffic flow prediction via spatial temporal graph neural network,” in *Proc. Web Conf.*, Apr. 2020, pp. 1082–1092.
- [2] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14412–14420.
- [3] B. Lu, X. Gan, H. Jin, L. Fu, and H. Zhang, “Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting,” in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1025–1034.
- [4] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in *Proc. AAAI*, vol. 32, no. 1, Apr. 2018, pp. 7444–7452.
- [5] J. Yang, W.-S. Zheng, Q. Yang, Y.-C. Chen, and Q. Tian, “Spatial-temporal graph convolutional network for video-based person re-identification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3289–3299.
- [6] Y. Yang, Z. Wei, Q. Chen, and L. Wu, “Using external knowledge for financial event prediction based on graph neural networks,” in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2161–2164.
- [7] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, “Every document owns its structure: Inductive text classification via graph neural networks,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 334–339.
- [8] A. Kapoor *et al.*, “Examining COVID-19 forecasting using spatio-temporal graph neural networks,” 2020, *arXiv:2007.03113*.
- [9] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal ARIMA model with limited input data,” *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 1–9, Sep. 2015.
- [10] K. Kumar, M. Parida, and V. Katiyar, “Short term traffic flow prediction for a non urban highway using artificial neural network,” *Proc. Social Behav. Sci.*, vol. 104, pp. 755–764, Dec. 2013.
- [11] N. G. Polson and V. O. Sokolov, “Deep learning for short-term traffic flow prediction,” *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.
- [12] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, “Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction,” *Sensors*, vol. 17, no. 4, p. 818, 2017.
- [13] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. 31st AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 1655–1661.
- [14] 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.*, Stockholm, Sweden, Jul. 2018, pp. 3634–3640.
- [15] W. Chen, L. Chen, Y. Xie, W. Cao, and X. Feng, “Multi-range attentive bicomponent graph convolutional network for traffic forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 3529–3536.
- [16] L. Zhao, Y. Song, C. Zhang, and Y. Liu, “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
- [17] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, Jul. 2019, pp. 922–929.
- [18] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, “Gated graph sequence neural networks,” in *Proc. ICLR*, Apr. 2016, pp. 1–19.
- [19] J. Liu and W. Guan, “A summary of traffic flow forecasting methods,” *J. Highway Transp. Res. Develop.*, vol. 21, no. 3, pp. 82–85, Mar. 2004.
- [20] M. Ahmed and A. Cook, “Analysis of freeway traffic time series data by using box-jenkins techniques,” *Transp. Res. Rec.*, vol. 773, pp. 1–9, Jan. 1979.
- [21] E. Zivot and J. Wang, “Vector autoregressive models for multivariate time series,” in *Modeling Financial Time Series With S-PLUS*. Springer, 2006, pp. 385–429.
- [22] X. L. Zhang, H. E. Guo-Guang, and L. U. Hua-Pu, “Short-term traffic flow forecasting based on K-nearest neighbors non-parametric regression,” *J. Syst. Eng.*, vol. 24, no. 2, pp. 178–183, Feb. 2009.
- [23] Y.-S. Jeong, Y.-J. Byon, M. Mendonca Castro-Neto, and S. M. Easa, “Supervised weighting-online learning algorithm for short-term traffic flow prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1700–1707, Dec. 2013.
- [24] N. Laptev, J. Yosinski, L. E. Li, and S. Smly, “Time-series extreme event forecasting with neural networks at uber,” in *Proc. ICML*, vol. 34, 2017, pp. 1–5.
- [25] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, “Deep learning: A generic approach for extreme condition traffic forecasting,” in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2017, pp. 777–785.
- [26] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, Dec. 2015, pp. 802–810.
- [27] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1. Palo Alto, CA, USA: AAAI Press, 2017, pp. 1655–1661.
- [28] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.
- [29] C. Chen *et al.*, “Gated residual recurrent graph neural networks for traffic prediction,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 485–492.
- [30] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, “Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3074–3082.
- [31] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive graph convolutional recurrent network for traffic forecasting,” in *Proc. Adv. Neural Inf. Process. Syst. 33: Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 17804–17815.
- [32] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.
- [33] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial-temporal graph modeling,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.
- [34] C. Zheng, X. Fan, C. Wang, and J. Qi, “Gman: A graph multi-attention network for traffic prediction,” in *Proc. AAAI*, vol. 34, no. 1, 2020, pp. 1234–1241.
- [35] H. Yan, X. Ma, and Z. Pu, “Learning dynamic and hierarchical traffic spatiotemporal features with transformer,” *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 20, 2021, doi: [10.1109/TITS.2021.3102983](https://doi.org/10.1109/TITS.2021.3102983).
- [36] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, “MGAE: Marginalized graph autoencoder for graph clustering,” in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 889–898.

- [37] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 5171–5181.
- [38] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–14.
- [39] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 346–353, Jul. 2019.
- [40] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 20–28.
- [41] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation recognition with graph neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4183–4192.
- [42] L. Ge, H. Li, J. Liu, and A. Zhou, "Temporal graph convolutional networks for traffic speed prediction considering external factors," in *Proc. 20th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2019, pp. 234–242.
- [43] Z. Diao, G. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 890–897.
- [44] Y. Zhang, S. Wang, B. Chen, and J. Cao, "GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
- [45] J. Hu, C. Guo, B. Yang, and C. S. Jensen, "Stochastic weight completion for road networks using graph convolutional networks," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1274–1285.
- [46] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.
- [47] Q. Zhang, Q. Jin, J. Chang, S. Xiang, and C. Pan, "Kernel-weighted graph convolutional network: A deep learning approach for traffic forecasting," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1018–1023.
- [48] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Sep. 2016, pp. 7–10.
- [49] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 255–262.
- [50] W. Song *et al.*, "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1161–1170.
- [51] S. Deng, S. Wang, H. Rangwala, L. Wang, and Y. Ning, "Cola-GNN: Cross-location attention based graph neural networks for long-term ILI prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 245–254.
- [52] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [53] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
- [54] J. Bai *et al.*, "A3T-GCN: Attention temporal graph convolutional network for traffic forecasting," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 7, p. 485, Jul. 2021.
- [55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [56] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4470–4481.
- [57] A. Anand, G. Ramadurai, and L. Vanajakshi, "Data fusion-based traffic density estimation and prediction," *J. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 367–378, Jul. 2014.
- [58] Y. Zhuang, Z. Pu, J. Hu, and Y. Wang, "Illumination and temperature-aware multispectral networks for edge-computing-enabled pedestrian detection," *IEEE Trans. Netw. Sci. Eng.*, early access, Dec. 31, 2021, doi: [10.1109/TNSE.2021.3139335](https://doi.org/10.1109/TNSE.2021.3139335).



Yang Wang is currently pursuing the Graduate degree with the School of Computer Science, Southwest Petroleum University, Chengdu, China. His research interests include computer vision and graph neural networks.



Jin Zheng (Member, IEEE) is currently pursuing the Ph.D. degree with the School of Computer, Sichuan University. He is also a Lecturer with Southwest Petroleum University, Chengdu, China. His research interests include machine learning and graph neural networks.



Yuqi Du is currently pursuing the Graduate degree with the School of Computer Science, Southwest Petroleum University, Chengdu, China. Her research interests include graph neural networks and natural language processing.



Cheng Huang is currently a Professor with Southwest Petroleum University, Chengdu, China. His current research interests include intelligent development of oil-gas field and data mining.



Ping Li (Member, IEEE) is currently a Full Professor with Southwest Petroleum University, Chengdu, China. Her current research interests include network science, machine learning, and their application in natural language processing.