ST-MGAT: Spatial-Temporal Multi-Head Graph Attention Networks for Traffic Forecasting

Kelang Tian[1,2], Jingjie Guo[1], Kejiang Ye[1,*], Cheng-Zhong Xu[3]

[1]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences*
[2]*University of Science and Technology of China*
[3]*State Key Lab of IoTSC, Faculty of Science and Technology, University of Macau*
kelang@mail.ustc.edu.cn,{jj.guo, kj.ye}@siat.ac.cn, czxu@um.edu.mo

***Abstract*—Graph Neural Networks (GNNs) have attracted increasing attention due to the significant representation learning capacity for graphs. The traffic forecasting is a typical graph representation learning task, but it is challenging to model the complex spatial and temporal relationships in traffics. Traditional spectral approaches get filters based on the eigendecomposition, which depends on the Laplacian matrix of the graph. However, these approaches have expensive matrix operation on graph convolutions neural networks and are insufficient to tackle the spatial dependency. In this paper, we propose a novel graph neural network - Spatial-Temporal Multi-head Graph ATtention network (ST-MGAT), to deal with the traffic forecasting problem. We build convolutions on the graph directly. We consider the features of neighborhood nodes and the weights of the edges to generate new node representation. More specifically, there are two main modules: i) Temporal convolution blocks to capture the dynamic time correlations; ii) Graph attention networks to capture the dynamic spatial relations between nodes. Experimental results show that our model achieves up to 13% improvement over the state-of-the-art approaches in short-term, medium-term, and long-term highway traffic forecasting.** [1]

***Index Terms*—Graph Convolutional Networks, Spatial-Temporal Model, Traffic Forecasting**

I. Introduction

In recent years, the Intelligent Transportation System (ITS) [1] has attracted the increasing attention of governments. Transportation services, such as navigation, rely on traffic condition assessments. Traffic flow forecasting is one of the core functions in ITS. If accurate and reliable predictions can be made in advance, the traffic management department can guide traffic control and make traffic networks run more efficiently. Highway flow or speed forecasting is the main problem in traffic forecasting. Vehicle sensors on urban roads can be represented as a graph where the nodes' Euclidean distance measures edge weights. Measures of traffic conditions, such as speed, flow, and occupancy, are also selected as the nodes' features in the graph.

According to the forecasting horizon, the traffic flow forecasting can be divided into two types: short-term (<30 min) and long-term (>30 min). Previous studies apply statistical approaches for short-term forecastings, such as linear regression. Nevertheless it is hard to model the spatial and temporal trends due to the high complexity and nonlinearity of data. Efforts have been made to carry out improved strategies, such as Support Vector Regression (SVR) [2] and Auto-Regressive Integrated Moving Average (ARIMA) [3]. However, these approaches are under some assumptions, and highly nonlinear traffic flow is too complex to meet these conditions. Nowadays, due to the large-scale deployment of sensors, researchers have switched their focus to data-driven strategies.

Machine learning approaches have been widely used in traffic forecasting tasks due to higher accuracy. Significant advances have been made in recent work. For example, Convolutional Neural Networks (CNNs) are used to model the adjacent correlation and time features. The traditional convolution methods can work efficiently on Euclidean space data (e.g., images) but are not suitable for non-Euclidean space tasks (e.g., normal graph). For example, Yao et al. [4] introduce a Spatial-Temporal Dynamic Network (STDN) to tackle the dynamic relationship between locations and capture long-term temporal changes. However, this study divides traffic data into several parts and treats the data as images, which can only deal with 2D grid data. Furthermore, the Non-Euclidean domain nodes are linked in different ways, such as the social network or protein structure. Traditional CNN cannot be directly applied to the graph structure because the Non-Euclidean data is irregular and has no translation invariance.

In this paper, we propose a novel structure, Spatial-Temporal Multi-Head Graph ATtention networks (ST-MGAT), to tackle the problems mentioned above to deal with the traffic forecasting problem. We apply multi-head graph attention networks to model the spatial correlations between nodes in a graph and apply a dilated convolution structure with gate mechanisms to capture the traffic data's temporal relationship. We build convolutions on the graph directly. We consider the features of neighborhood nodes and the weights of the edges to generate new node representation. More specifically, we use temporal convolution blocks to capture the dynamic time correlations and use graph attention networks to capture the dynamic spatial relations between nodes. Experimental results show that our model achieves up to 13% improvement over the state-of-the-art approaches in short-term, medium-term, and long-term highway traffic forecasting.

*Corresponding author

[1]Code is available at https://github.com/Kelang-Tian/ST-MGAT

2375-0197/20/$31.00 ©2020 IEEE
DOI 10.1109/ICTAI50040.2020.00114

II. Preliminary

A. Graph Convolution Networks

With the rapid development of graph neural networks [5], Graph Convolution Neural (GCN) networks have been widely applied to the graph-structured data. This neural network performs well on some tasks, such as classification [6], [7], embedding [8], link prediction [9] tasks. Niepert et al. [10] introduced a method to collect neighborhood information to represent the node's feature in a graph that is used in social networks. Generally, there are two kinds of graph convolution: spatial domain methods and spectral domain methods.

1) Spectral Domain Methods: Spectral-based approaches deal with inputs data using spectral filters [6], [11], [12]. In earlier studies, graphs tended to be large in size because there was no parallel library to support node information aggregation. Researchers usually take the transformation of adjacency matrices to complete matrix aggregation and then apply libraries such as PyTorch and TensorFlow to accelerate matrix operations. Inspired by the signal processing, the Fourier transformation is performed on the graph, and a lot of work has been done to prove the validity and rationality of the transformation [11]–[13].

As for the problems with traffic prediction, GCN methods are widely used to solve traffic problems [1], [14], [15]. [16] is the first work to establish a novel graph framework based on spectral-domain methods, but the model has a drawback that dynamic spatial-temporal relations are neglected in traffic data. Nevertheless, methods on traffic forecasting based on fixed adjacency matrices have shortcomings that rarely consider the change of spatial structure, and it applies matrix operation with approximations. Diao et al. [17] advocate a method that replaces the fixed matrix with a dynamic Laplacian matrix to overcome the drawback. Nevertheless, this is just an approximate solution based on the spectral domain. All the above methods are based on one premise that the adjacency matrix is known and fixed. However, traffic accidents or road construction will change the topology of the traffic network, so the traffic network is unstable.

2) Spatial Domain Methods: The spatial methods consider the graph structure spatially: the geometric relationship between the target node and other nodes. The challenge is to generate new features of nodes, which is implemented by collecting and aggregating features of adjacent neighbors [14], [18], [19]. While in this paper, we apply multi-head graph attention networks, which are the spatial domain method of graph convolution networks. Instead of applying the encoder-decoder [15], [20], [21] or spectral-domain method, we construct convolution directly on the graph with an attention mechanism.

B. Spatial-Temporal Models

Generally speaking, the spatial-temporal model can be divided into Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) methods. RNN is generally used to solve time series problems by hidden states and filters passed to neural units [22]. It is worth mentioning that some approaches improve RNN-based method with attention mechanisms [23] and diffusion convolutional [24]. This method applies the one-dimensional convolution structure on the time axis to capture the temporal dynamic behavior of traffic flow [16]. Nevertheless, the RNN-based method faces the challenges that it is ineffective for long sequences, and the gradient may explode. In this paper, we apply a dilated convolution structure with gate mechanisms to model the temporal relationship of the traffic data

C. Attention Mechanism in Graph Convolution

The main idea of the attention-based graph convolution is to generate new node representation by aggregating nodes with edge information. Furthermore, the attention coefficient is the mutual importance of each node in the graph relative to its neighbors. The following two papers are significant and inspiring work. Thekumparampil et al. [25] create an attention-based graph neural network for semi-supervised learning, which products node representations by aggregating edge information. Velickovic et al. [26] propose a novel graph attention networks (GAT). The main idea is to aggregate the node information with the edge information and generate a new node representation without expensive operation on the matrix. The two approaches mentioned above are spatial domain methods and apply convolution directly on the graph without using spectral filters.

In the field of traffic flow prediction, Zheng et al. [21] propose a graph multi-attention network (GMAN) with a transform attention mechanism on the graph to transform the historical traffic features to future representations. Park et al. [20] create an encoder-decoder architecture with a self-attention mechanism on a graph (STGRAT) to model complex correlations. Guo et al. [27] devised an attention-based graph convolutional network (ASTGCN) to study spatio-temporal correlation. The spatial attention mechanism is applied to manipulate the correlation between different locations, and the input adjusted by the attention mechanism is fed into the graph convolution module,

It should be emphasized that the attention-based models mentioned above capture graphical structural information in traffic conditions, which prepare for the next blocks, such as spectral-based graph convolution or encoder-decoder structure. Instead of applying convolution directly on the graph, they are still spectral domain or transform methods, even though they named their approaches attention-based graph models. Methods with similar names have different implementations. In this paper, we apply convolution on the graph directly, and the attention mechanism makes the network focus on the valuable information.

III. Problem Definition

We treat the traffic network as a graph, and the task is to predict the features of nodes in the next several time steps. We abstract a two-way lane into two lanes. We treat the lane as the edge in the graph, and the road detector on the lane

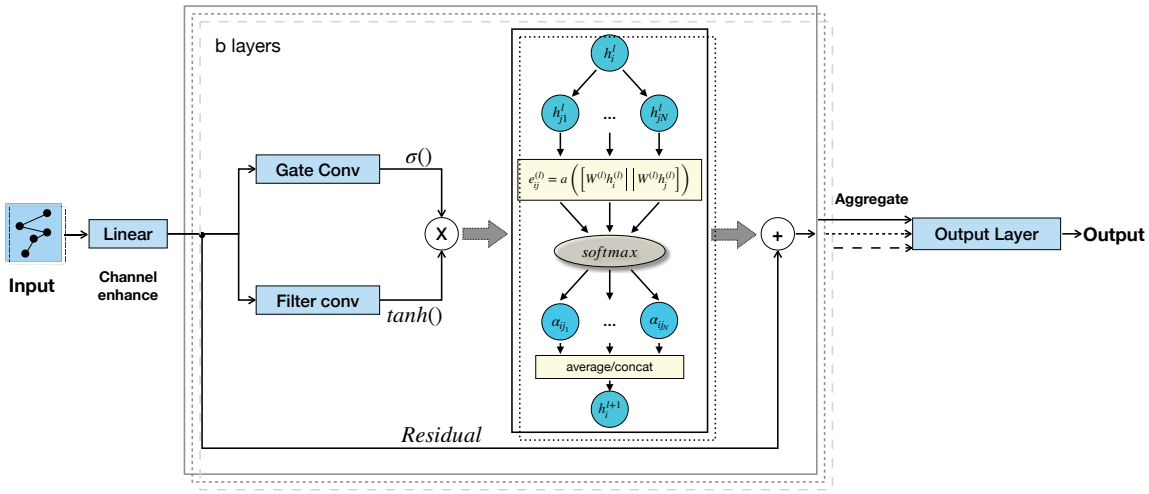

Fig. 1. ST-MGAT Architecture. The input is $X \in R^{N\times T\times F}$, where N is the number of nodes, T is the time steps, F is the features of each node. The output is $Y_{out} \in R^{N\times T}$, which represents the predicted velocity of N nodes in t time steps. The linear for processing input data is to raise the dimension of the original data features. Several two-dimensional convolutions with a convolution kernel of 1 by 1 are adopted for every channel. Filter block adopts two-dimensional convolution. Gate block, residual block, and aggregate block adopt a one-dimensional convolution. There are b layers in our model, and two graph convolution blocks are stacked for every layer.

as the point in the graph. Measures of traffic conditions, such as speed, flow, and occupancy, are also selected as features of the nodes in the graph. It is reasonable to define the network as an undirected graph, G = (V, E, A), V is the finite set of nodes, and E is the set of edges. An adjacency matrix is denoted as $A \in R^{N\times N}$. It should be emphasized that the graph mentioned here is an undirected graph, and the purpose of the article is to predict the features of all nodes on the graph at a certain point in the future. The nodes on the graph are selected from the detectors on the road. Moreover, the data generated by detectors are features of the nodes. What needs to be emphasized is that X is not a single signal of a node but a graph signal with whole nodes. Traffic flow forecasting is to predict traffic flow in the next S time slices using historical measurements (e.g., speed or flow or occupy). The input is $X \in R^{N\times F\times T}$, and the output is $Y \in R^{N\times P}$, where N is the observation station data, F is the features of each node, T is the time steps of the input, and P is the time step of the output. $X_t = (x_t^1, x_t^2, ..., x_t^N)^T \in R^{N\times F}$ represents the vector of nodes at time t. Predict the next P time-steps traffic speed $Y = (y^1, y^2, , ..., y^N) \in R^{N\times P}$, where all nodes on the graph and historical sequences X are taken into consideration.

IV. Methodology

In this section, we introduce the two main parts of our framework. The spatial layer is built up by a graph attention network (GAT) that aggregates nodes features with attention coefficients to generate new node representations. The temporal layer is constituted by a dilated convolution structure with gate mechanisms that captures the temporal features and prevents time-consuming. These layers are stacked to improve the accuracy of predictions, while overfitting is prevented by applying normalization to the layer. Finally, the model produces an output for n nodes in the next t time steps by attaching a fully connected layer. Then we will outline the framework of our approach.

A. Framework

The framework of graph networks is illustrated in Fig. 1. Generally speaking, our model is composed of a dilated convolution with a gate mechanism and a spatial-based graph attention convolution block, followed by a full connection layer for output.

The input data is $X \in R^{N\times T\times F}$, where N is the number of nodes, T is the time steps, F is the features of each node. The temporal layer's preparation work is to implement the feature augments of the input data by two-dimensional convolution. Two identical dilated convolution layers accept data that have been feature-enhanced, in which the dilated kernel size is 1,2 and 4. Hadamard product is applied at element-wise for two parallel convolution layers (Gated Units). Subsequently, two layers of graph attention convolution (GAT) are stacked to deal with the results of the temporal layer. Meanwhile, the residual network layer is set to fuse the unprocessed data with the data processed by graph convolution. The structure of one layer, as described above, and several layers are stacked.

It should be emphasized that the input of the graph attention convolution layer is feature $F \in R^{N\times D}$, where N is the number of nodes, and D is the size of input features. The output is $F_{out} \in R^{N\times D_{out}}$, where D_{out} is the new size of features. However, the output of the gated layer is $X_{gated} \in R^{N\times F\times T}$, where T is the time steps. In this paper, we permute three-dimensional data X_{gated} into two-dimensional data, where $X_{gated} \in R^{N\times(F*T)}$. This method integrates the time information about traffic data into the features of nodes.

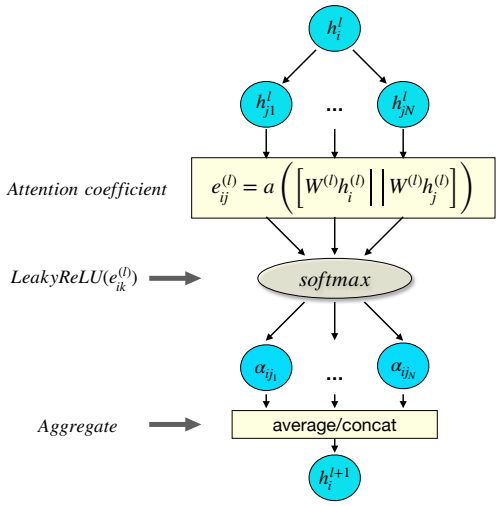

Fig. 2. Graph attention layer. $h^l \in R^{N \times F}$, where h_i^l denotes the feature of node i. h_i^{l+1} is the renewal of hidden feature. N is the number of nodes and F is the number of features.

In brief, the temporal layer is constituted by a gated temporal convolution block that captures the temporal features and prevents time-consuming. The spatial layer is built up by a graph attention network that aggregates nodes features with attention coefficients to generate new node representations. These layers are stacked to improve the accuracy of predictions, while overfitting is prevented by applying normalization to the layer. Finally, the model produces an output for n nodes in the next t time steps by attaching a fully connected layer.

B. Graph Convolution Layer

Graph Attention Network (GAT) [26] proposes a weighted summation of neighboring node features using an attention mechanism. The weight of the featured nodes is completely dependent on the features of the nodes and is independent of the graph structure. This method overcomes the bottleneck of spectral-based graph convolution networks, and it is easy to implement the assignment of different learning weights to different neighbors.

The main difference between the attention-based graph convolution network (GAT) and spectral-based graph convolution network (GCN) is how to collect and sum up the feature representations of neighbor nodes with a distance of one hop. To a certain extent, GAT will be stronger because the correlation between vertex features is better integrated into the model. The most fundamental advantage is that the computation is done on a per-node basis. Every operation needs to loop through all vertices on the graph to aggregate the nodes' features. The vertex-by-vertex operation means that the shackles of the Laplacian matrix are eliminated.

The calculation of GAT is divided into two steps as the other attention mechanisms: calculate the attention coefficient and aggregate the weighted features.

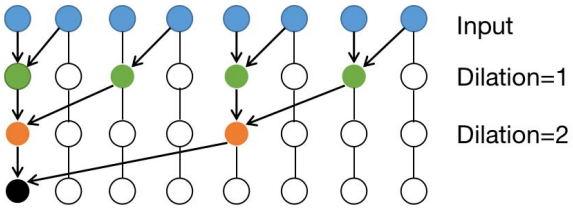

Fig. 3. Dilated convolution. The kernel size is set to 1 and 2.

$$e_{ij}^{(l)} = a\left(\left[W^{(l)}h_i^{(l)} \middle|\middle| W^{(l)}h_j^{(l)}\right]\right), j \in \mathcal{N}_i \tag{1}$$

As we can see in formula (1) that a linear mapping of the shared parameters W augment dimension to vertex features, which are commonly used in feature enhancement. This method concatenates the transformed features of the vertices and maps the high-dimensional features to a real number e_{ij} with $a(*)$. This function is implemented through a single-layer feedforward neural network. Learning the correlation between vertices i and j is made by the learnable parameters W and the mapping function $a(*)$.

Correlation coefficient normalization:

$$\alpha_{ij}^{(l)} = \frac{exp\left(LeakyReLU(e_{ij}^{(l)})\right)}{\sum_{k \in \mathcal{N}_i} exp\left(LeakyReLU(e_{ik}^{(l)})\right)} \tag{2}$$

Attention scores about node's receiving edges are normalized by softmax with the correlation coefficient mentioned above. The features are weighted and aggregated according to the calculated attention coefficient, where $h_i^{(l)}$ is the output of the new feature by GAT for each vertex i which is fused with neighborhood information, and $\sigma(*)$ is the activation function.

$$h_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)}\right) \tag{3}$$

Multi-Head Attentional Layer Same as multiple kernels in a convolutional neural network, multiple heads mechanism of attention is applied to enhance the model's ability and stabilize the training process. Each attention head has its own parameters. K is the number of attention heads. h is the new feature of the GAT that fuses neighborhood information for each vertex and is the activation function. We recommend using concatenate for the middle layer and averaging the last layer.

$$h_i^{(l+1)}(K) = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)k} W^{(l)k} h_j^{(l)}\right) \tag{4}$$

The important learning parameters in GAT are W and $a(*)$. Because of the above vertex-by-vertex calculation method, these two parameters are only related to the vertex features and have nothing to do with the structure of the graph. Therefore, changing the structure of the graph in the test task has little effect on GAT.

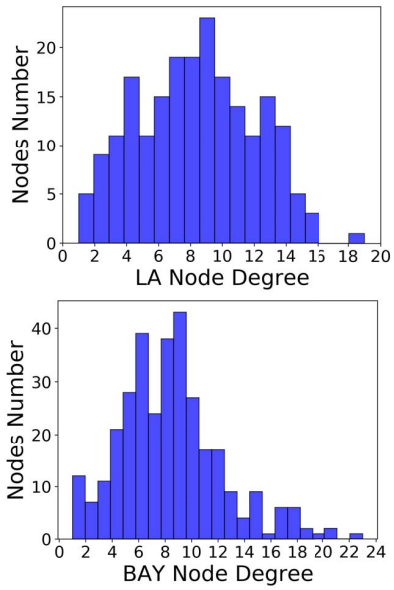

Fig. 4. Degree of Nodes. LA dataset: 1515 edges and 34272 time steps. BAY dataset: 2369 edges and 52116 time steps.

C. Parallel Computing on Graph Structure

A novel practice is to stitch graphs to generate a big graph rather than do graph convolution in the cycle process for every batch. For example, there is a graph g with n nodes and e edges, the input of convolution $I \in R^{n \times f_{in}}$,

$$G(g_i, I) = output \in R^{n \times f_{out}} \tag{5}$$

where f_{in} means nodes' features, $G()$ notes convolution operations, and f_{out} means nodes' new features.

There is no need to run the operation above in the loop for batch size times. Nevertheless, the batching graph faces challenges: graphs can be sparse or large. By contrast, it is batching the graphs as a large graph that processes convolution in parallel. As shown in Fig. 6, the output of the batch operation is still graph, which means the operation for a basic graph still works for the return of the batch process.

$$G(g_{batch}, I_{batch}) = output_{batch} \in R^{batch \times n \times f_{out}} \tag{6}$$

D. Temporal Convolution Layer

RNN-based networks are wildly used in time series tasks to extract temporal features. However, recurrent neural networks consume much training time and are difficult to train. Convolutional methods are applied on a time axis in many models. For example, Yu et al. [28] employ dilated causal convolution to capture features on the time axis. Wu et al. [29] adopt gated CNN to extract complex temporal relationship. Gehring et al. [30] apply convolutional to sequence to the sequence learning task. Inspired by this work, we adopt dilated convolution structures with gate mechanisms in time the axis

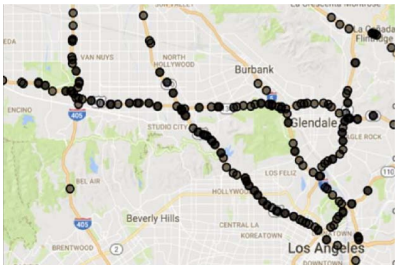

Fig. 5. LA map is abstracted into graph.

to extract temporal correlation, as shown in Fig. 3. The temporal convolution layer set one-dimension convolution, which is followed by gated linear units.

The advantage of dilated convolution is that without the loss of pooling information, the receptive field is increased so that each convolution output contains a larger range of information. Dilated convolution can be applied to the problem that images require global information or speech, and text requires long sequence information, such as image segmentation, speech synthesis, and machine translation. The operation of dilated convolution maintains the relative spatial position of the feature map, which means our model improves receptive fields and take historical information into account. The dilated causal convolution method turns into the following:

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t) \tag{7}$$

where F is a two-dimensional sequence (image), s is domain; k is the kernel function, t is the domain; l is the dilation factor; p is the domain of the dilated convolution. The above formula is not different from the one-dimensional case.

If kernel size is k and the step size of the dilated convolution is r, the receptive field changes from k * k to k + (r-1) * (k-1), the latter part indicates the number of zeros to be inserted.

Gated linear units are applied after dilated convolution to determine information pass through layers [30]. As shown in the Fig. 1 (Gated TCN Block), we set the input as $X \in R^{n \times t \times c}$,which has three dimension (n is spatial, t is temporal sequence and c is channels or features created by dilated convolution layer).The convolution kernel $F \in R^{k \times C_{in} \times C_{out}}$ maps input X to $[Y1, Y2] \in R^{(t-k+1) \times 2C_{out}}$.

$$F *_\gamma X = tanh(\alpha * x + b_1) \odot \sigma(\beta * x + b_2) \tag{8}$$

Where $F *_\gamma X \in R^{(t-k+1) \times C_{out}}$, and $\odot$ means Hadamard product at element-wise. $\sigma()$is the sigmoid gate function that depends on which information spread to the next step. Different types of activation functions are welcomed to be applied and different forms of gated functions are effortlessly applied to our networks.

V. Experiments

In this work, two real-world traffic datasets (METR-LA and PEMS-BAY) are used to evaluate our model, which are

TABLE I
ST-MGAT AND BASELINE MODELS PERFORMANCE

Models	$15min$			$30min$			$60min$		
$Data-BAY/LA$	MAE	$RMSE$	$MAPE(\%)$	MAE	$RMSE$	$MAPE(\%)$	MAE	$RMSE$	$MAPE(\%)$
$ARIMA$	1.62	3.30	3.50	2.33	4.76	5.40	3.38	6.50	8.30
$FC-LSTM$	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.70
$WaveNet$	1.39	3.01	2.91	1.83	4.21	4.16	2.35	5.43	5.87
$DCRNN$	1.38	2.95	2.90	1.74	3.97	3.90	2.07	4.74	4.90
$GaAN$	-	-	-	-	-	-	-	-	-
$GraphWaveNet$	1.30	2.74	2.73	1.63	3.70	3.67	1.95	4.52	4.63
$STGCN$	1.36	2.96	2.90	1.81	4.27	4.17	2.49	5.69	5.79
ST-MGAT (ours)	**1.25**	**2.60**	**2.57**	**1.50**	**3.33**	**3.27**	**1.77**	**3.97**	**4.02**
$ARIMA$	3.99	8.21	9.60	5.15	10.45	12.70	6.90	13.23	17.40
$FC-LSTM$	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	13.20
$WaveNet$	2.99	5.89	8.04	3.59	7.28	10.25	4.45	8.93	13.62
$DCRNN$	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.50
$GaAN$	2.71	5.24	6.99	3.12	6.36	8.56	3.64	7.65	10.62
$GraphWaveNet$	2.69	5.15	6.90	3.07	6.22	8.37	3.53	7.37	10.01
$STGCN$	2.88	5.74	7.62	3.47	7.24	9.57	4.59	9.40	12.70
ST-MGAT (ours)	**2.62**	**5.07**	**6.70**	**2.88**	**5.75**	**7.77**	**3.25**	**6.58**	**9.12**

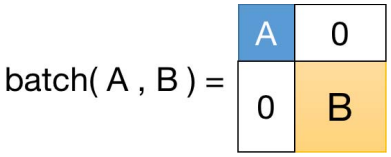

Fig. 6. Join two graphs into a large one.

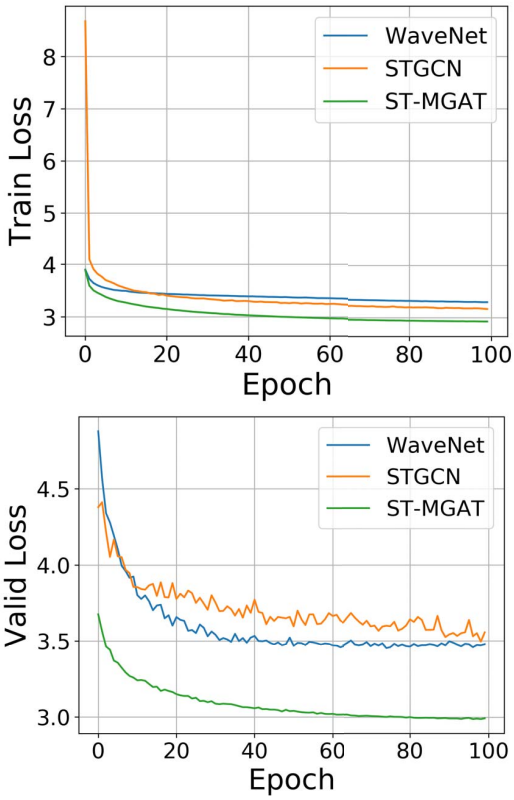

Fig. 7. Comparison with other classical models (WaveNet [32] and STGCN [16]). ST-MGAT is our model. All the above models work on the METR-LA dataset.

also used [24]. METR-LA contains traffic information of 207 detectors for four months in Los Angeles from Mar 1, 2012 to Jun 30, 2012. PEMS-BAY contains traffic information of 325 detectors for six months in the Bay area from Jan 1, 2017 to May 31 2017. The data recorded by the detectors were divided into multiple groups at intervals of 5 minutes. In this experiment, the time of day was divided into 288-time windows.

In spectral domain methods, an adjacency matrix is built using a road network with a Gaussian kernel [31]. $W_{ij} = exp(-\frac{dist(v_i v_j)^2}{\sigma^2})$ calculates the edge weight between nodes, where $dist(v_i v_j)$ means the distance from node v_i to v_j, σ denotes standard deviation. However, The realization of the algorithm is different in different approaches. For example, some methods [16] [29] set threshold about $dist(v_i v_j)$ and set the value of parameter W to 1 instead of a decimal if $dist(v_i v_j)$ is greater than the threshold. In this paper, since our approach is based on the spatial domain, we record the relationships of the nodes in the graph and initialize the equality of relationships.

There are no more than 7% zeros in our dataset, which means no vehicles flow in the last 5 minutes. Lower traffic speeds mean more traffic jams, but a speed of zero means there is no traffic. We replace the zeros with the mean speed to make features of nodes aggregated fluently. To make the comparison fairly, we take the same preprocess on the dataset as in Li et al. [24]. The data set is divided into a training set, validation set, and testing set according to the proportion of 70%, 20%, and 10%.

Three metrics are applied to evaluate the performance of traffic forecasting, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|X_i - \hat{X}_i\right| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2} \tag{10}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i} \times 100\%\right| \tag{11}$$

Experiments are built under a computer environment: Intel(R) Core(TM) i9-9820X CPU@ 4.10GHz, two NVIDIA GeForce 2080Ti GPUs.

A. Experiment Settings

$X \in R^{N\times F\times T}$ is the input data, where N is nodes, T is the timesteps, F is nodes' features. The output is $Y \in R^{N\times P}$, where P are timesteps. We set $T = 12$, $P = 12$ in our program. There are four blocks; each block contains two layers, and the layers is composed by temporal convolution layer and graph convolution layer. The dilation size about temporal convolution is set to 1 and 2. Note that the input of graph attention convolution is $X \in R^{N\times F_{in}}$, where F is the feature. And the output is $Y \in R^{N\times F_{out}}$. The feature size of the convolutional layer are 12, 10, 9, 7,6,4,3 and 1 respectively.

B. Baselines

We compare our model with several other classic models. In order to make the result comparison fair, some models (e.g., STGCN) are reproduced and applied to the same data set. Some results directly cited the data in the paper (e.g., GaAN, GWaveNet). Readers can find some baseline models in the code exposed.

- **ARIMA**: Auto-Regressive Integrated Moving Average [3]
- **LSTM**: Long Short-Term Memory [24]
- **GaAN**: Build an encoder-decoder network with graph convolution blocks and recurrent neural networks, published in ICLR-2018 [23].
- **DCRNN**: Diffusion convolution recurrent neural network, published in ICLR-2018 [24]
- **WaveNet**: A network for time series task [32]
- **Graph WaveNet**: A graph convolution network for time series task, published in IJCAI-2019 [29]
- **STGCN**: Spatial-temporal graph convolution network, published in IJCAI-2018 [16]

C. Results and Analysis

1) Comparison with Baselines: Table I reveals the performance of our model ST-MGAT and baselines on datasets. Our model obtains a better result in three evaluation metrics than others. Compared to spatial-temporal models, traditional methods and machine learning approaches perform badly for long-term prediction because these models are only based on historical records, and there is no structure to capture spatial features. Generally, deep learning approaches, such as GaAN and DCRNN outperform simple models. Specifically, graph-based convolutional network models perform well in simulating real road networks and in capturing spatial-temporal features that lead improvement on 60-minutes horizons.

As shown in Table I, our model performs better than baselines in both short and long-range. In the case of short-term prediction, ST-MGAT model prediction accuracy is slightly higher than other deep learning-based methods, which achieves 3% and 2% accuracy advance on two datasets in 15min traffic flow prediction, respectively. However, MAE improvement over baselines increases significantly in long timesteps. Our model achieves 8% improvement on METR-LA dataset and achieves 13.8% improvement on PEMS-BAY dataset, as compared with the GraphWave Net [29].

By observing and comparing the learning curves of different models, our model converges faster than other models in the training process. The loss curve of the valid set shows that our model can not only achieve smaller loss values but also that the curve is smoother than the previous model. The curves of the WaveNet and STGCN model are uneven. Fig. 7 clearly shows the above results.

2) With/Without Convolutions Analysis: Table II shows the results of the control experiment. We compared the results after removing the graph convolution module from the model with the complete model results. The three indicators of the complete model on two data sets are superior to the results of the other model with the graph attention convolution layer

TABLE II
MODEL COMPARISON WITHOUT GRAPH CONVOLUTION (MEAN OVER 12 HORIZONS)

Datasets	Models	MAE	$RMSE$	$MAPE(\%)$
PEMS-BAY	$ST-MGAT$	1.45	3.16	3.15
	$withoutGconv$	1.77	3.99	4.03
METR-LA	$ST-MGAT$	2.85	5.63	7.62
	$withoutGconv$	3.42	6.89	9.55

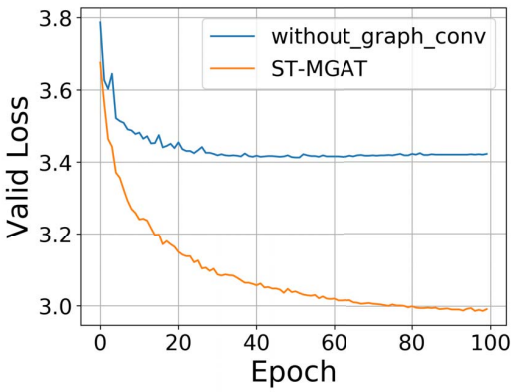

Fig. 8. Loss curve about our model without graph attention convolution. The absence of graph attention convolution leads to higher error. The above results are based on METR-LA dataset.

removed. Fig. 8 illustrates this phenomenon with learning curves.

3) Discussion: We tried many methods, but some did not work. Inspired by multi-component fusion skill [27], which captures recent, daily, and weekly periodic temporal correlation in historical traffic data. We add historical records of the same nodes in the last day and week but does not help. Moreover, as the number of time steps increased, traffic forecasting tends to be improved slightly (horizons are 12-time steps). Finally, it is verified that traffic flow forecasting for the next period based on the current point in time is mainly controlled by the last few hours of traffic flow. Also, changing the loss function from MAE to RMSE improves RMSE, but cannot improve other metrics. As we change the length of timesteps about input and increase the hidden channels of graph convolution, the prediction accuracy was improved slightly, but the model parameters were increased, and the time consumption was increased.

VI. Conclusion and Future Work

In this work, a novel model ST-MGAT is proposed for traffic flow forecasting, which contains an attention-based graph convolution network. To the best of our knowledge, we first apply spatial-based methods in traffic flow forecasting tasks rather than spectral-based approaches, enhanced the generalization ability of the model. The experiment demonstrates that our model surpasses the convolution-based methods and improve the adaptability of dealing with changing road conditions. In future work, we will apply our model on a general graph and add supplementary information such as weather conditions to further improve the model accuracy.

Acknowledgment

This work is supported by the National Key R&D Program of China (No. 2019YFB2102100), National Natural Science Foundation of China (No. 61702492), Science and Technology Development Fund of Macao S.A.R (FDCT) under number 0015/2019/AKP, Youth Innovation Promotion Association CAS (NO. 2019349), and Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence.

References

[1] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] R. Chen, C. Liang, W. Hong, and D. Gu, "Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm," *Applied Soft Computing*, vol. 26, pp. 435–443, 2015.

[3] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering-asce*, vol. 129, no. 6, pp. 664–672, 2003.

[4] H. Yao, X. Tang, H. Wei, G. Zheng, and Zhenhui, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," vol. 33, no. 01, pp. 5668–5675, 2019.

[5] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv: Learning*, 2018.

[6] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv: Learning*, 2016.

[7] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *arXiv: Learning*, 2018.

[8] S. Pan, R. Hu, S. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 1–13, 2019.

[9] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," pp. 5171–5181, 2018.

[10] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," pp. 2014–2023, 2016.

[11] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," *arXiv: Learning*, 2013.

[12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," pp. 3844–3852, 2016.

[13] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[14] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," pp. 1993–2001, 2016.

[15] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," pp. 3428–3434, 2018.

[16] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *international joint conference on artificial intelligence*, pp. 3634–3640, 2018.

[17] Z. L. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," vol. 33, no. 01, pp. 890–897, 2019.

[18] J. Gilmer, S. S. Schoenholz, P. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," *arXiv: Learning*, 2017.

[19] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," pp. 1024–1034, 2017.

[20] C. Park, C. Lee, H. Bahng, T. Won, K. Kim, S. Jin, S. Ko, and J. Choo, "Stgrat: A spatio-temporal graph attention network for traffic forecasting." *arXiv: Learning*, 2019.

[21] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," *arXiv: Signal Processing*, 2019.

[22] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," *arXiv: Machine Learning*, 2016.

[23] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," pp. 339–349, 2018.

[24] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv: Learning*, 2017.

[25] K. K. Thekumparampil, S. Oh, C. Wang, and L. Li, "Attention-based graph neural network for semi-supervised learning," *arXiv: Machine Learning*, 2018.

[26] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv: Machine Learning*, 2017.

[27] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," vol. 33, no. 01, pp. 922–929, 2019.

[28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv: Computer Vision and Pattern Recognition*, 2015.

[29] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," pp. 1907–1913, 2019.

[30] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv: Computation and Language*, 2017.

[31] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[32] A. V. Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv: Sound*, 2016.