Multi-type parameter prediction of traffic flow based on Time-space attention graph convolutional network

Guoxing Zhang[1], Haixiao Wang[2]*, Yuanpu Yin[2]
[1]Baidu Inc, Autonomous Driving Unit (ADU),
Beijing, 100080
China.
[2]Inner Mongolia Agricultural University, College of Energy and Transportation Engineering,
Hohhot, 010018
China

Received: March 7, 2021. Revised: July 23, 2021. Accepted: August 6, 2021. Published: August 11, 2021.

Abstract—Graph Convolutional Neural Networks are more and more widely used in traffic flow parameter prediction tasks by virtue of their excellent non-Euclidean spatial feature extraction capabilities. However, most graph convolutional neural networks are only used to predict one type of traffic flow parameter. This means that the proposed graph convolutional neural network may only be effective for specific parameters of specific travel modes. In order to improve the universality of graph convolutional neural networks. By embedding time feature and spatio-temporal attention layer, we propose a spatio-temporal attention graph convolutional neural network based on the attention mechanism of the neural network. Through experiments on passenger flow data and vehicle speed data of two different travel modes (Hangzhou Metro Data and California Highway Data), it is verified that the proposed spatio-temporal attention graph convolutional neural network can be used to predict passenger flow and vehicle speed simultaneously. Meanwhile, the error distribution range of the proposed model is minimum, and the overall level of prediction results is more accurate.

Keywords—Spatiotemporal attention graph; Convolutional Neural Network; Traffic flow; Prediction

I. INTRODUCTION

Convolutional neural networks have performed very well on Euclidean Space data such as image classification, target detection, and semantic segmentation. This has prompted many studies to extend convolution operations to non-Euclidean spaces, such as graph structure data and point cloud data.

However, since the number of neighbor nodes of each node on the graph is uncertain, the convolution operation cannot be used directly. Traffic network is a typical non-Euclidean space, taking the subway and highway networks for example, the degree of correlation between traffic nodes in the network is proportional to the distance between them, the change of traffic flow at a certain node will lead to the corresponding changes of the nodes closest to this node, and the energy of the changes is decreasing, this rule is consistent with the basic characteristics of graph neural network. Therefore, convolutional neural network cannot be well used to analyze spatial data of traffic network.

Based on the above principles, graph convolutional network which is more suitable for non- Euclidean space is proposed and applied in traffic network. Bruna *et al.* defined graph convolution network for the first time by using spectral graph theory and the Fourier transform of the graph [1]. Since the Fourier transform requires a lot of calculations, Defferrard *et al.* proposed to use Chebyshev polynomials to approximate the convolution, which greatly reduces the computational complexity [2]. The convolution method is further simplified by introducing the hypothesis Kipf, and the obtained approximation is proved to perform optimally on various tasks [3]. This has played an important role in promoting research on traffic flow prediction by Yu *et al.* [4]-[6]. However, at present, the improved method of graph neural network can only be used for a single parameter, and still cannot predict multiple traffic flow parameters simultaneously. In this paper, we have conducted experiments on spatiotemporal attention graph neural networks on different tasks, and the main contributions are as follows: (1) Optimize the attention mechanism to learn the time dimension characteristics of the data better. (2) A new network structure is designed to capture spatio-temporal information more effectively. (3) Experiments verify that the spatio-temporal attention graph convolutional neural network is effective in the prediction of the two traffic flow parameters: speed and flow.

II. Related Work

Before deep learning was used for traffic prediction, researchers mainly used traditional prediction methods such as ARIMA, space-time ARIMA, KARIMA, SARIMA, etc. However, these models only consider the information of a single node and ignore the spatial information between nodes, which makes the prediction effect unsatisfactory.

The methods of deep learning to predict traffic flow parameters mainly include fully connected neural network, grid convolutional neural network and graph convolutional neural network.

Du *et al.* used neural network multi-modal data fusion for the first time to make predictions [7]. However, in addition to the unsatisfactory effect of this forecasting method, it still cannot meet the needs of multi-node forecasting.

Lv *et al.* and Jia *et al.* synthesized each traffic node into a multi-dimensional vector to realize multi-node prediction and used multiple fully connected layers for training [8][9]. Although this method solves the problem of simultaneous prediction of multiple nodes, nodes that are far apart in space also establish connections, resulting a higher computational burden. And the order of the input nodes is random, and the process of organizing all the nodes into a multi-dimensional vector also loses the node space information.

Wang *et al.* and Ma *et al.* obtain spatio-temporal information for prediction by constructing a spatio-temporal matrix so as to preserve the spatial information between nodes [10][11]. But the spatiotemporal matrix method essentially synthesizes the nodes into a vector, so the order of the nodes will have a great impact on the prediction results. If the convolution kernel with fixed size is used for convolution on the spatio-temporal matrix, only the first-order adjacent nodes to the central node can be considered, which limits the use range of the model.

Aiming to overcome the shortcomings of fully connected neural networks. Zhang *et al.* divide the graph into multiple independent squares according to the spatial distance so as to enable the model to convolve on non-sequentially connected nodes [12][13]. Then, the traditional convolution method is used to extract the spatio-temporal features while ensuring the spatial structure is unchanged.

Yao *et al.* proposed a local convolution method to extract spatial features based on the assumption that the traffic flow is only related to neighboring areas [14]. The cyclic neural network is used to learn the meteorological data of the area, and the learning results are spliced with the functional characteristics of the current area to make predictions.

Wang *et al.* concluded that the grid method has two shortcomings [15]. First of all, the geometric structure of the urban road nodes is not regular, and the grid representation cannot reflect the real situation of the urban transportation network. Secondly, urban roads are sparse, and most of the weights learned by the convolution operation are zero.

To solve the shortcomings of the grid model, graph convolutional neural networks have become a new research method.

Li *et al.* used graphs to represent traffic data for the first time, using GCNN to replace the fully connected layer in the recurrent network, so that there is no need to compress the extracted spatial features into one-dimensional vectors, and they can be directly trained through the network [5].

Yu *et al.* simplified the speedometer on the road as a node and used the distance between the detectors to represent the weight between the nodes [4]. It also proposes a module that first convolves in time and then convolves in space, and finally convolves in time to extract spatio-temporal features. Replace the commonly used recurrent neural network to avoid cumulative errors in the training process.

Lee *et al.* input more features to each node to obtain better results [6].

Yao *et al.* used LSTM units to extract temporal features from the results of spatial information extraction aiming to solve the problem of rough and poorly effective extraction process [16]. The difference is that Yao *et al.* use grids to process spatial information and introduced an attention mechanism based on LSTM to extract time information, while Chai *et al.* used graph convolution to process Spatial information [17][18].

Since LSTM can only be used to process vectors, spatial information will be compressed, resulting in complete loss of features.

Jain *et al.* further proposed a method to structure a recurrent neural network, that is, each node and edge are connected by a recurrent neural network [19][14]. The recurrent neural network not only needs to consider the characteristics of the node but also the information on the edge.

Vaishali *et al.* proposed the user association and optimal pricing of heterogeneous wireless railway network based on RAP [19][20].

III. Method

In this chapter, we mainly introduce the innovations and contributions of the model. Firstly, temporal features and spatial features are taken as input features. Then, the graph attention mechanism is extended to the feature level of the node to realize the feature extraction of time dimension. Finally, a new network structure is designed to make the model get better results on subway passenger flow data and road speed data.

IV. Time Feature Embedding

Traffic flow data is generally cyclical, and its time characteristics are very important prior information for prediction problems. We do not regard the prediction problem as a sequence problem, but embed the current prediction time as a feature, and merge it with the historical data of the current node as the new input feature of the node. This means that temporal features are abstracted into features with the same latitude as spatial features to participate in prediction. We divide the time of day into 288 numerical values at 5-minute intervals to represent the current forecast moment. Therefore, the input of a single node can be represented as shown in (1), where i is the current node number, t is the current time point, $\mathrm{T} \in [0,288)$ is the current time feature, and h is the current node historical data.

$$F_i = [h_{t-12}, h_{t-11}, h_{t-10}, \cdots, h_t, T_t] \tag{1}$$

V. Spatio-temporal Attention Layer

First, we will define a spatio-temporal attention layer, which will be used as the graph convolution layer used in our proposed spatio-temporal attention graph convolutional neural network (STAGCN). We think that the graph convolution attention mechanism proposed by Veličković *et al.* is a more special form [21]. We promote it on the original basis and propose an attention mechanism for traffic application scenarios. We think that considering the characteristics of different neighbor nodes at different moments and assigning different attention coefficients is more in line with the actual situation than just considering the relationship between nodes.

The input of the spatio-temporal attention layer is shown in formula (2), where N is the number of nodes, and F is the number of features of each node. The output of the spatio-temporal attention layer is shown in formula (3) as the output of this layer, where F' may be different from F .

$$h = \{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\}, \vec{h}_i \in R^F \tag{2}$$

$$h' = \{\vec{h}_1', \vec{h}_2', \cdots, \vec{h}_N'\}, \vec{h}_i' \in R^{F'} \tag{3}$$

Firstly, linear transformation is carried out to obtain strong expression ability. The parameterized weight matrix $W \in R^{F' \times F}$ is shared in each node. Then, the attention coefficient e_{ijf} is calculated through the attention mechanism.

$$e_{ijf} = a(W\vec{h}_i, (W\vec{h}_j)_f) \tag{4}$$

e_{ijf} indicates the importance of the feature f of node j to node i . We use the Softmax function to normalize all j node features f , so that the coefficients can be easily compared between different nodes:

$$a_{ijf} = soft\max{}_{jf}(e_{ijf}) = \frac{\exp(e_{ijf})}{\sum_{k \in N_i \times N_f} \exp(e_{ik})} \tag{5}$$

The attention mechanism a is a feedforward neural network using the LeakyRelu function with $a = 2$. When the attention mechanism and the correlation coefficient are fully expanded, it is expressed as formula (6).

$$a_{ijf} = \frac{\exp(Leakyrelu(\vec{a}^T [W\vec{h}_i \| (W\vec{h}_j)_f]))}{\sum_{k \in N_f} \exp(Leakyrelu(\vec{a}^T [W\vec{h}_i \| (W\vec{h}_j)_k]))} \tag{6}$$

$$a_{ij} = \{a_{ij1}, ai_{j2}, \cdots, \ ai_{jf}\}, f \in N_f \tag{7}$$

We combine calculated a_{ijf} into a_{ij} in sequence. And use the new attention coefficient to calculate the linear combination of their corresponding features, which is used as the final output feature of each node after nonlinear activation. Use K independent attention mechanisms to perform the conversion, and then connect their features to get the output features, where || represents the connection operation.

$$\vec{h}_i' = \sigma(\sum_{j \in N_i} a_{ij} W\vec{h}_j) \tag{8}$$

$$\vec{h}_i' = \overset{k=1}{\overset{k}{\|}} \sigma(\sum_{j \in N_i} a_{ij}^k W^k \vec{h}_j) \tag{9}$$

a_{ij}^k represents the attention coefficient calculated by k attention mechanisms. And W^k is the corresponding linear transformation matrix. In the end, the output h′ of each node will contain kF′ features. The multi-head attention mechanism is shown in Fig. 1.

$$\vec{h}_i' = \overset{k=1}{\overset{k}{\|}} \sigma(\sum_{j \in N_i} a_{ij}^k W^k \vec{h}_j) \tag{10}$$

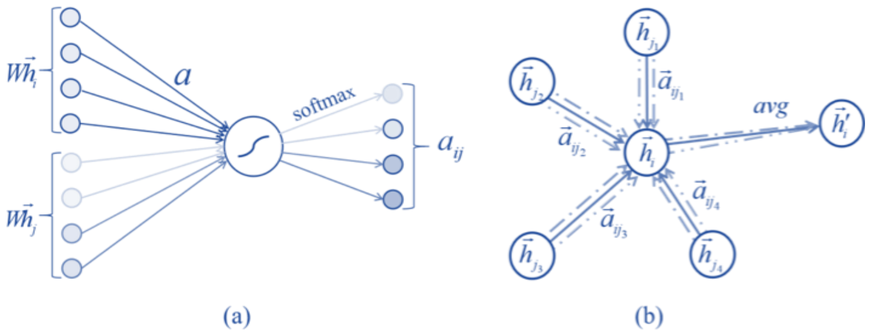

Fig. 1 The optimized attention factor is composed against the dollar

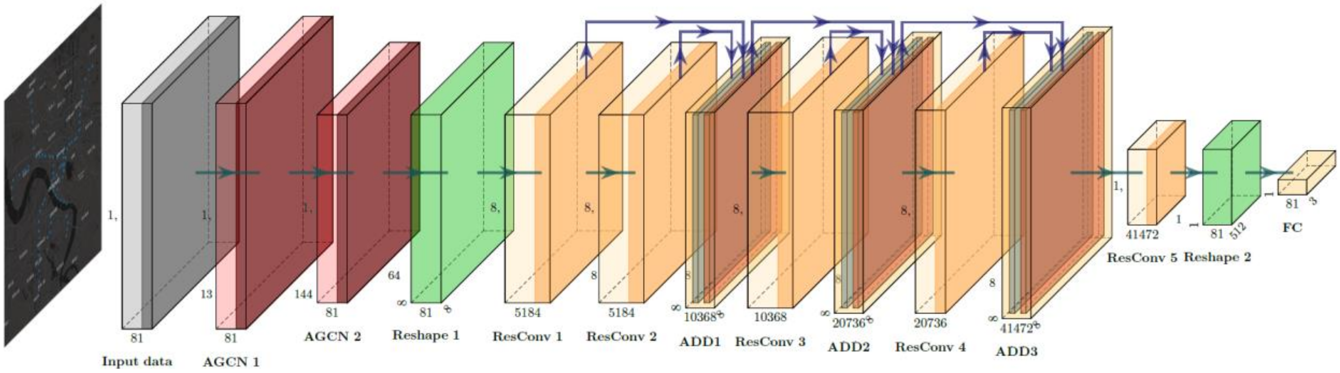

Fig. 2 STAGCN network structure

VI. Network Structure Settings

We set up a simple graph convolution module and residual network module as shown in Fig. 2. The residual network module performs further convolution on the result of the graph convolution module. Both AGCN and AGCN2 are graph convolution modules designed by us. The ResConv1 to ResConv5 layers obey the residual connection, and the network finally uses the fully connected layer to output the prediction results. STAGCN internal network parameter settings are shown in Table Ⅰ.

Table Ⅰ STAGCN network parameter settings

Name	Layer	Input size	Output size	k	p	S
AGCN_1	GAT	81*13	12			
	Relu	81*12*12	81@12*12			
AGCN_2	GAT	81*12*12	8			
	Relu	81*8*8	81@8*8			
Reconv1	Conv2d	1	64	3	1	1
	Maxpool2d	1	1	3	1	1
	BN+Relu	1	1			
Reconv2	Conv2d	64	64	3	1	1
	Maxpool2d	1	1	3	1	1
	Bn+Relu	1	1			
Resconv3	Conv2d	64	128	3	1	2
	Maxpool2d	1	1	3	1	1
	Bn+Relu	1	1			
Resconv4	Conv2d	128	256	3	1	2
	Maxpool2d	1	1	3	1	1
	Bn+Relu	1	1			
Resconv5	Conv2d	256	512	3	1	2
	Maxpool2d	1	1	3	1	1
	Bn+Relu	1	1			
FC		512	3			

VII. Experiment

We compared the proposed spatio-temporal attention graph convolutional neural network with other state-of-the-art traffic prediction benchmarks. The experimental results are compared and analyzed from the perspective of the accuracy of sequence prediction and the distribution range of prediction errors.

VIII. Data Preprocessing

The experiment selects Hangzhou subway card swiping data and US highway detector data METR_LA as the test data. As two different traffic flows, subways and roads have different characteristics. The prediction object of this experiment includes the two most important parameters among the three parameters of traffic flow: flow and speed. Both data sets are divided according to the time interval of 5min, so each node on the graph has 288 values per day. Use linear interpolation to fill in missing values.

Hangzhou Metro Data (HZSW): Hangzhou Public Security Bureau and Alibaba Cloud Intelligence launched the first Global Urban Computing AI Challenge, with the title of

"Metro Passenger Traffic Forecast". The data contains 25-day subway card swiping data records from January 1st, 2019 to January 25th, 2019, involving about 70 million data from 81 subway stations on 3 lines. In the experiment, we divide the data into training and validation sets and test sets according to 19:3:3.

California Highway Data (METR_LA) consists of real-time vehicle speed data collected by sensors deployed on the Los Angeles Ring Road System, covering major metropolitan areas. The data set is the result of aggregating 30-second data samples into 5 minutes. We chose a medium-sized area with 207 detectors. The time range is from March 1st to March 25th, 2012. We divided the training, validation, and test sets according to the ratio of 19:3:3. The data provides the coordinates between the various detectors, and we calculated the spherical distance between the detectors based on these coordinates to generate a distance matrix.

IX. Initialization Of Weights Between Nodes

The sensor distribution of METR_LA is shown in Fig. 4. The site distribution of HZSW is shown in Fig. 3. To make the network better obtain the spatial relationship of the complex transportation network, we design and define the calculation method of the edge weight between nodes according to the knowledge in the transportation field. Establish a suitable weighted adjacency matrix as the object of the graph convolution model.

Distance matrix (W_d): Points closer in space are more relevant than points that are farther apart, so we define the spherical short distance between two points as the distance between two points. A threshold Gaussian kernel function is used to define the matrix W_D , where $w_{ij} \in W_D$ is the edge weight determined by d_{ij} (the distance between node i and node j). σ^2 and ε are the thresholds for the distribution and sparsity of the control matrix W_d , which are set to 10 and 0.5, respectively.

$$\omega_{ij} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), & i \neq j \ \ and \ \ \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \varepsilon \\ 0, & otherwise. \end{cases} \tag{11}$$

X. Evaluation Method

We use STCGN and DCRNN as reference objects to compare performance with STAGCN on the same test set.

Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root mean square error (RMSE) commonly used in traffic forecasting are used to evaluate the effect of the method.

$$MAE = \frac{1}{m}\sum_{i=1}^{m} | x_i^{'} - x_i | \tag{12}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m} \frac{| x_i^{'} - x_i |}{x_i + c} \tag{13}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m} (x_i^{'} - x_i)^2} \tag{14}$$

$x_i^{'}$ represents the predicted value of the model at time i , and x_i represents the true value at time i . m represents the total number of moments.

XI. Experiment Condition setting

We use a grid search strategy to determine the optimal network training parameters. During the training process, all models use 60 minutes as the historical time window, that is, 12 observation data points (H=12) are used to predict traffic conditions in the future 15, 30, and 45 minutes (H=3, 6, 9). The experimental model is trained on the computer platform of two-way E5-2665 and four-way TAITAN-XP. The parameter settings during training are shown in Table Ⅱ.

Table Ⅱ Experimental parameter settings

Name	Learn rate	Learn strategies	epoch	Early stop
STGCN	$1e^{-3}$	RMSProp	150	Yes
DCRNN	$1e^{-2}$	Adam	800	Yes
STAGCN	$1e^{-2}$	Adam	800	Yes

XII. Discussion

Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) are commonly used for evaluation. Table Ⅲ shows the test results of different methods on HZSW and METR_LA.

Table ⅢThe performance of the model

Model		HZSW			MRTE_LA		
	TIME	MAPE	MAE	RMSE	MAPE	MAE	RMSE
STGCN	15min	13.58%	8.99	17.26	18.72%	5.51	8.07

	30min	17.70%	10.06	18.66	20.76%	6.61	10.30
	45min	22.20%	11.90	22.26	22.49%	6.66	12.03
DCRNN	15min	15.75%	11.60	19.99	23.88%	5.36	12.10
	30min	16.70%	12.41	21.02	24.50%	6.63	12.40
	45min	17.83%	13.66	23.48	24.18%	6.72	12.41
STAGCN	15min	12.82%	8.95	16.04	16.96%	5.28	9.59
	30min	13.76%	9.70	18.06	19.61%	6.00	10.87
	45min	14.77%	10.71	21.40	22.34%	6.37	11.77

It can be seen from Table Ⅲ: (1) Methods based on graph neural networks, including STAGCN, DCRNN and the proposed STAGCN, can model nonlinear spatio-temporal objects. (2) Most indicators of STAGCN on these two data sets have achieved better performance. It shows the effectiveness of the proposed spatio-temporal correlation modeling method; (3) Compared with other deep learning models, STGAT shows better results on both HZSW and METR_LA.

The line chart shows the prediction results of the model in the time dimension, in which two control models (STGCN/DCRNN) are represented by dashed lines, the real values and the predicted values of our model are represented by solid lines, and different models are distinguished by different colors as shown in Fig. 7. In the peak period, the true value changes less with time, and our model better fits the true value; because our model has higher model complexity, the true value changes drastically over time during peak hours, and several models show greater fluctuations, while our model is more stable. When the predicted step size increases, the other two models participating in the control experiment show obvious fluctuations, and the prediction results become unstable, while our model fluctuates less. Our model predicts the start time and end time of the peak more accurately than other models and has better stability.

The prediction error is defined here as the predicted value minus the true value. When the predicted value is greater than the true value, the prediction error is positive, otherwise it is negative. It should be noted that the most important thing in the prediction error kernel density graph is the distribution range. The smaller the distribution range, the higher the stability of the model, and the error is more concentrated around zero. When the prediction error distribution range is basically consistent, the prediction is more accurate when the prediction error is more concentrated at zero. By comparing the prediction error kernel density graphs of different models on HZSW data under the same time step, it can be clearly observed that our model has the smallest error distribution range, indicating that the overall level of prediction results is better. In contrast, the prediction error distribution range of STGCN and DCRNN on the HZSW data decreases successively, as shown in Fig. 8. On the METR_LA data, there is no obvious difference in the distribution range of the three models, but our model has more prediction results in the position where the prediction error is zero, as shown in Fig. 9. The experimental results on the MRTE_LA data are as follows, in which STGCN, DCRNN and STAGCN are in turn from left to right. From top to bottom, the forecast error distributions at 15 minutes, 30 minutes, and 45 minutes are shown in sequence.

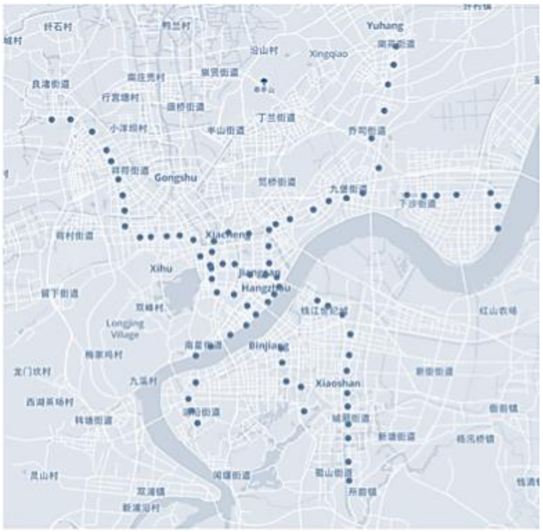

Fig. 3 The spatial location distribution of subway station points

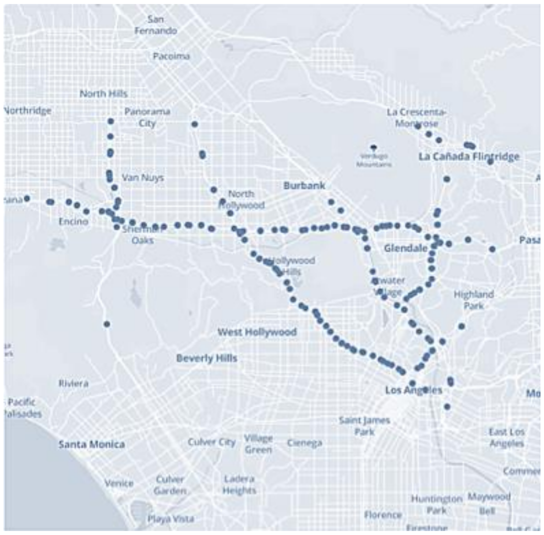

Fig. 4 METR_LA detector spatial distribution

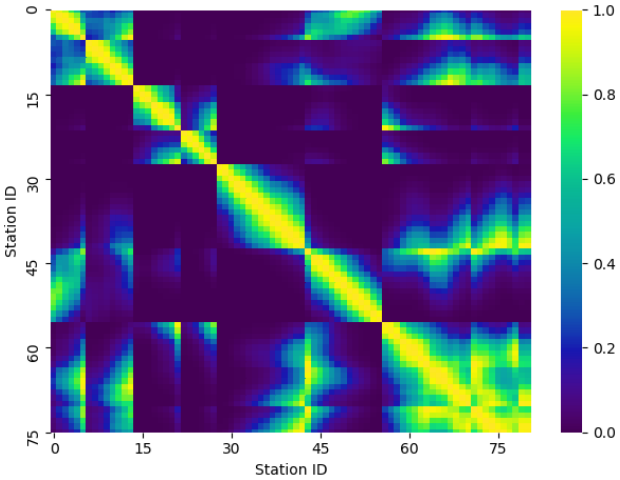

Fig. 5 Connection weights among Hangzhou urban rail transit

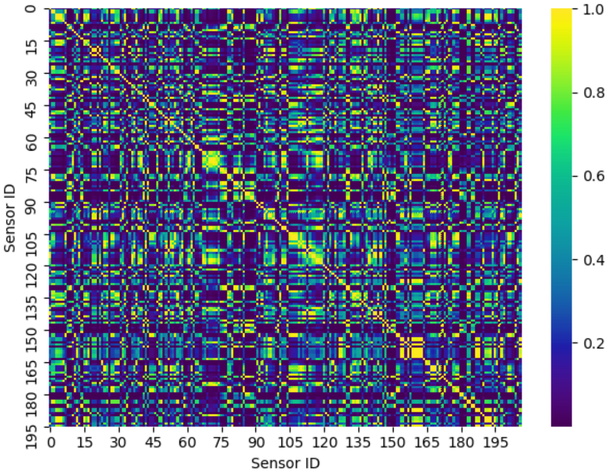

Fig. 6 Connection weights among MRTE_LA road sensor

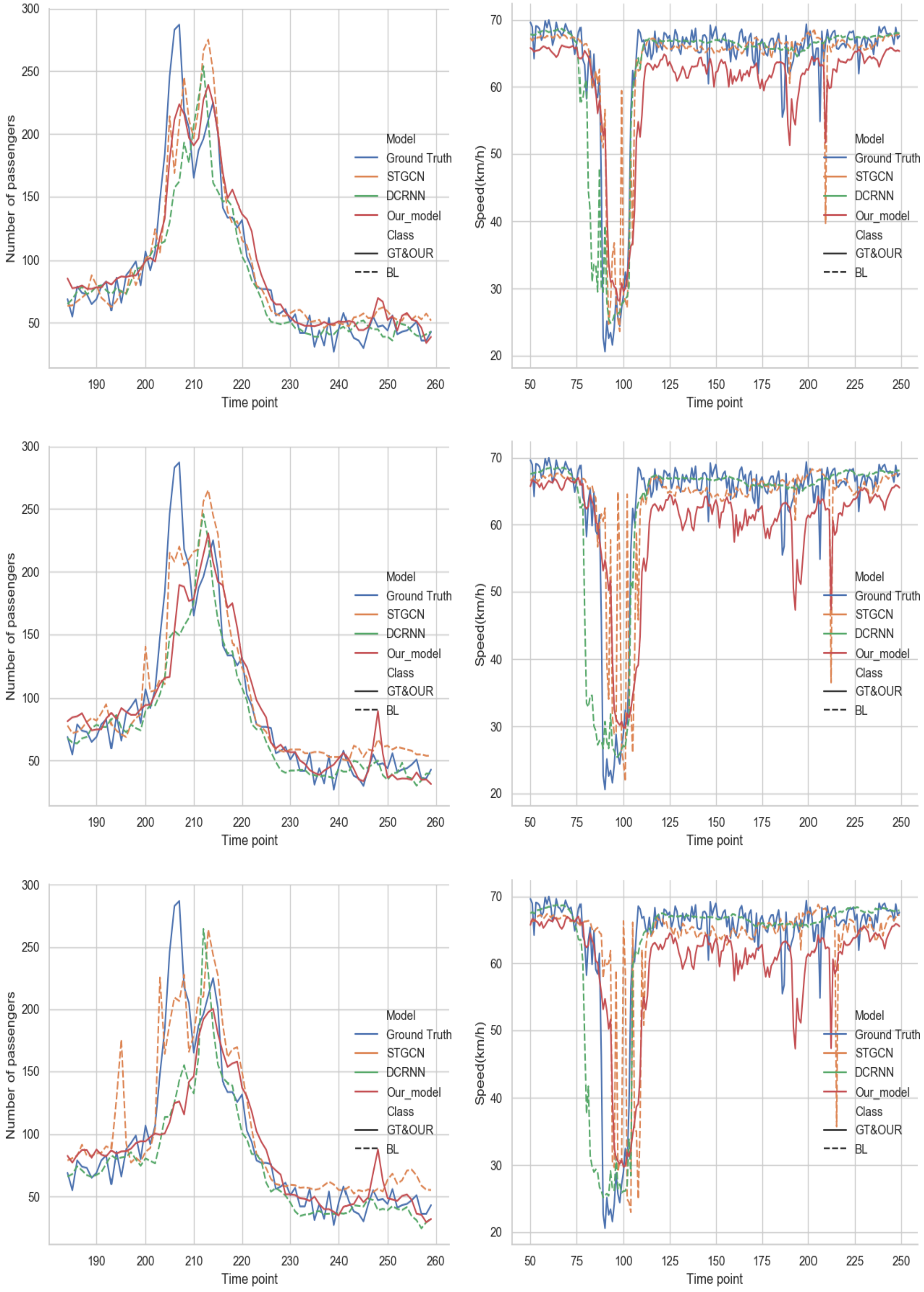

Fig. 7 Predict the parameter line chart

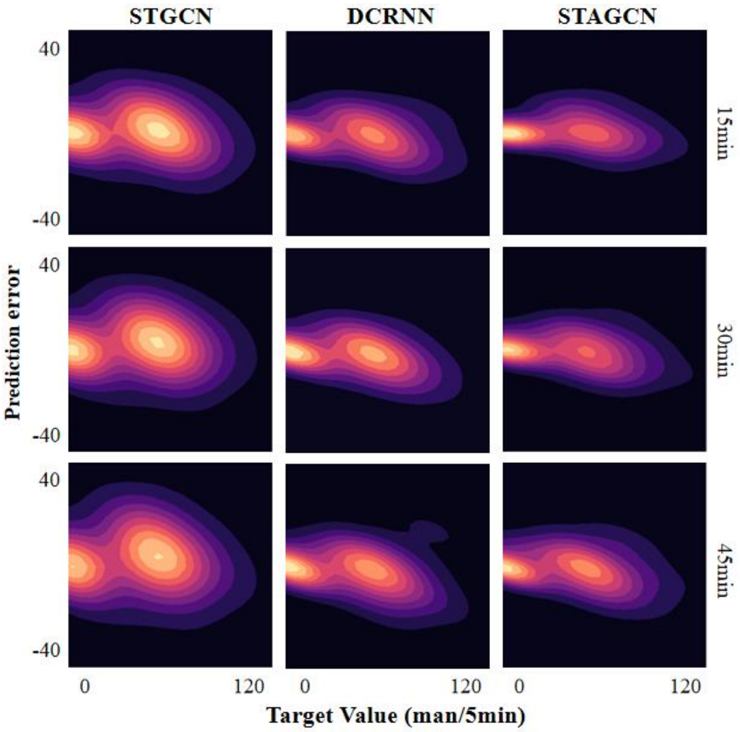

Fig. 8 Subway traffic forecast error distribution

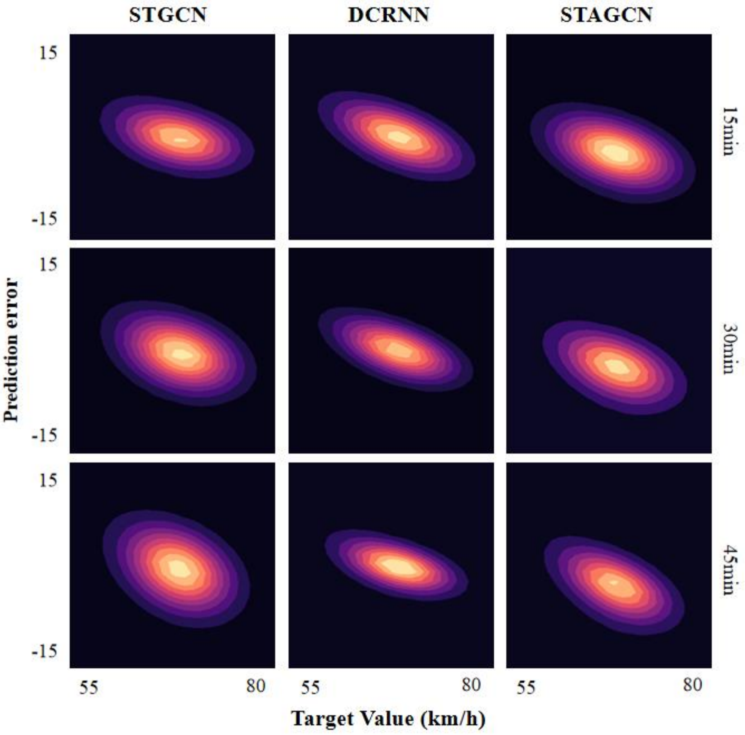

Fig. 9 The error distribution of vehicle speed prediction

XIII. Conclusion

Most graph convolutional neural networks are only used to predict one type of traffic flow parameter. In order to improve the universality of graph convolutional neural networks. This paper proposes a new STAGCN model based on deep learning. Through method of the temporal feature embedding, spatial features and temporal features are used as node features to learn, then we further extend the attention mechanism and consider the relationship between node features on the basis of focusing on the relationship between nodes, making it more suitable for traffic flow feature prediction. More effectively, multiple important parameters of traffic flow can be predicted simultaneously in this way. The new model has achieved better results than the best current benchmark model, solving the existing graph network in the performance of insufficient traffic prediction, and the experimental results show that the model is equally effective in the prediction of speed and flow.

In future work, we will conduct research from the following aspects: (1) Consider that the relationship between nodes changes dynamically with time, not only distance, but also other traffic impedance characteristics. (2) Extend the multi-step sequence method so that it will not be limited to a fixed number of time step predictions.

Acknowledgment

The authors acknowledge the Natural Science Foundation of Inner Mongolia (Grant: 2019MS05005); Science and Technology Project of Inner Mongolia (Grant:2021GG0082).

References

[1] J. Bruna, W. Zaremba, and A. Szlam, “Spectral Networks and Locally Connected Networks on Graphs,” Computer Science, 2013.

[2] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering,” Advances in neural information processing systems, 2016, 29, pp. 3844-3852.

[3] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” arXiv preprint arXiv:1609.02907, 2016.

[4] B. Yu, H. Yin, and Z. Zhu, “Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting,” In: 27th International Joint Conference on Artificial Intelligence, Stockholm, 2017, pp. 3634-3640.

[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” In: 6th International Conference on Learning Representations, Vancouver, 2017, pp. 1-16.

[6] K. Lee and W. Rhee, “Ddp-gcn: Multi-graph convolutional network for spatiotemporal traffic forecasting,” arXiv preprint arXiv:1905.12256, 2019.

[7] S. Du, T. Li, X. Gong, and S. J. Horng, “A hybrid method for traffic flow forecasting using multimodal deep learning,” International Journal of Computational Intelligence Systems, 13(1), 2018.

[8] Y. Lv, Y. Duan, and W. Kang, “Traffic Flow Prediction With Big Data: A Deep Learning Approach,” IEEE Transactions on Intelligent Transportation Systems, 2015, 16(2), pp. 865-873.

[9] Y. Jia, J. Wu, and Y. Du, “Traffic speed prediction using deep learning method,” 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) IEEE, 2016, pp. 1217-1222.

[10] J. Wang, G. Qian, J. Wu, G. Liu, and X. Zhang, “Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method.” 2016 IEEE 16th International Conference on Data Mining (ICDM) IEEE, 2016, pp. 499-508.

[11] X. Ma, D. Zhuang, Z. He, J. Ma, and Y. Wang, “Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction,” Sensors, 17(4), 2017, pp. 818.

[12] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” Thirty-first AAAI conference on artificial intelligence, 2017.

[13] Z. Xian, Y. Shen, Y. Zhu, and L. Huang, “Predicting Multi-step Citywide Passenger Demands Using Attention-based Neural Networks,” the Eleventh ACM International Conference, ACM, 2018, pp. 736-744.

[14] H. Yao, W. Fei, J. Ke, X. Tang, and J. Ye, “Deep multi-view spatial-temporal network for taxi demand prediction,” Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1).

[15] B. Wang, X. Luo, Zhang. F, B. Yuan, A. L. Bertozzi, and P. J. Brantingham, “Graph-based deep modeling and real time forecasting of sparse spatio-temporal data,” arXiv preprint arXiv:1804.00684, 2018.

[16] H. Yao, X. Tang, W. Hua, G. Zheng, and Z. Li, “Modeling spatial-temporal dynamics for traffic prediction,” In: The Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, 2018, pp. 5668-5675.

[17] D. Chai, L. Wang, and Q. Yang, “Bike Flow Prediction with Multi-Graph Convolutional Networks,” Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems, 2018, pp. 397-400.

[18] Y. Jia, J. Wu, and Y. Du, “Traffic speed prediction using deep learning method,” In: IEEE International Conference on Intelligent Transportation Systems, Rio de Janeiro, 2016, pp. 1217-1222.

[19] Vaishali and S. Santosh, “User Association with RAP for Heterogeneous Wireless Railway Networks,” 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) IEEE, 2019.

[20] Vaishali and S. Santosh, “Optimal Pricing for RAP in Heterogeneous Wireless Railway Networks,” 2020.

[21] P. Veličković, G. Cucurull, and A. Casanova, “Graph attention networks,” In: 6th International Conference on Learning Representations, Vancouver, 2018, pp. 1-12.

Author's brief introduction: Guoxing Zhang was born on February 20, 1994, from Baoji, Shaanxi, China. In 2020, he obtained a master's degree in engineering from Shenzhen University, Shenzhen, Guangdong Province, China, majoring in transportation engineering. The main research areas are computer vision and autonomous driving systems.

He worked as a computer vision algorithm researcher in Shenzhen Tencent Computing System Co., Ltd. from 2019 to 2020. In 2021, he joined Beijing Baidu Technology Co., Ltd. in China as an autonomous driving computer 3D vision algorithm engineer. At present, his main research interests are computer vision and autonomous driving digital twin systems.

Author Contributions: Guoxing Zhang propose the methodology and writes this paper; Haixiao Wang is corresponding author, she collects data and validates the proposed method; Yuanpu Yin reviews and edits the manuscript.

Conflicts of interest: The authors declare no conflict of interest.