



Multi-attribute Graph Convolution Network for Regional Traffic Flow Prediction

Yue Wang¹ · Aite Zhao¹ · Jianbo Li¹ · Zhiqiang Lv¹ · Chuanhao Dong¹ · Haoran Li¹

Accepted: 13 September 2022 / Published online: 22 October 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In recent years, traffic flow prediction has been extensively explored in Intelligent Transportation Systems, which is beneficial for reducing traffic jams and accidents as well as optimizing traffic network resources. Most of the previous methods divide cities into equal-sized grids and predict flows within one grid. However, we believe that each area is not independent, and there are interactions between areas. And the interaction between areas belonging to different attributes is more regular. Therefore, we propose a Multi-Attribute Graph Convolutional Network (MAGCN) for regional traffic flow prediction. Based on the attributes to which the areas belong, we divide cities into unequal-sized grids, and then a matrix is constructed using the flow of Functional area-based Origin-Destination pairs. GCN and dilated causal convolution allow the model to capture the spatial correlation and temporal dependence between functional regions while overcoming the under-fitting of local peaks. Extensive experimental results and evaluation metrics on two real-world datasets show that the MAGCN outperforms the baselines and has a higher accuracy for traffic flow prediction.

Keywords Traffic flow prediction · Spatio-temporal dependence · Unequal-sized grids · Functional area-based Origin-Destination pairs · Four-dimensional structure

1 Introduction

By September 2020, China had 365 million motor vehicles nationwide. A large number of motor vehicles will inevitably bring a series of problems such as traffic congestion. ITS [1] has emerged and will become the main development direction of the transportation system in the future. Traffic flow forecast [2–4] is a crucial part of the ITS, which can provide a strong traffic decision-making basis for traffic managers [5]. If accurate traffic conditions can be communicated to the masses in advance, traffic congestion and traffic accidents will be greatly reduced, and road network resources will be fully utilized.

✉ Jianbo Li
lijianbo@qdu.edu.cn

¹ College of Computer Science & Technology, Qingdao University, Qingdao 266071, China

In early traffic flow forecasting, time series analysis methods (e.g., autoregressive integrated moving average (ARIMA) [6], Holt-Winters (HW) model [7] and Kalman filtering [8]) were mainly used. Due to their linear property, which limits the representation of highly nonlinear traffic flows, they cannot capture the spatio-temporal dependence of traffic flows well. Machine learning models can achieve more sophisticated data modeling as well as higher prediction accuracy. Chen and Sun [9] achieves traffic speed prediction by Bayesian temporal factorization on multidimensional time series in the presence of missing values. Zhao and Sun [10] performs traffic flow prediction by joint modeling of the fourth-order Gaussian process dynamical model and weighted k-NN of multiple intersections. However, the above methods are not ideal in capturing complex nonlinear spatio-temporal dependencies. And because some traditional methods need to solve the constrained optimization and the eigenvalues of the decomposition matrix, making the time complexity extremely high when dealing with large amounts of data.

Powerful big data characterization and computational capabilities have enabled deep learning methods to achieve great success in many challenging tasks [11–15], providing new research directions for learning tasks in the field of traffic prediction. Recurrent Neural Network (RNN) [16] is mainly used for the prediction of sequential data. As variants of RNN, Long Short-Term Memory (LSTM) [17] and Gated Recurrent Units (GRU) [18] alleviate the gradient disappearance or explosion in RNN, and can effectively capture semantic associations between long sequences as well as model nonlinear temporal dependencies. Many researchers have employed the powerful feature extraction capability of Convolutional Neural Networks (CNN) to model nonlinear spatial dependencies. The combination of CNN and LSTM [19, 20] is used to jointly model spatio-temporal information and obtain more accurate results without much domain knowledge to further effectively solve the traffic prediction problem. However, due to the inability to process non-Euclidean structured data, CNN cannot accurately extract the spatial correlation of inter-regional traffic flows in the city. RNN and LSTM are computationally expensive and difficult to train. The parameters of GRU are relatively small, however, its prediction effect is not satisfactory when the amount of data is large. And the above methods need many layers to capture longer sequences. Therefore, we propose a Multi-Attribute Graph Convolutional Network (MAGCN) for regional traffic flow prediction. The cores of MAGCN are GCN, dilated causal convolution, and Functional area-based Origin-Destination pairs (FOD). GCN extends the traditional convolution to the graph [21] structure data and integrates node features with the topology of the graph for modeling spatial correlation in traffic prediction. The dilated causal convolution allows the receptive field to grow exponentially with the increase of hidden layers, which is used to describe the dependencies of adjacent time steps in the long term. Compared with the flow within one area, FOD reflects the directional traffic interaction between functional areas, which is more conducive to the scheduling of vehicles in urban traffic. Our main contributions are as follows:

- Different from the previous division of cities into equal-sized grids, we divide cities into unequal-sized grids according to the attributes to which the areas belong.
- For the first time, we construct a matrix consisting of the values of the Functional area-based Origin-Destination pairs (FOD), i.e., the edges in a weighted directed graph. Thus, taking into account the computation time, we change the output from the previous three-dimensional to four-dimensional.
- Consisting of the unique spatio-temporal convolutional module, a Multi-Attribute Graph Convolutional Network (MAGCN) is designed to predict FOD-based traffic flows while capturing the spatio-temporal correlation of traffic data.

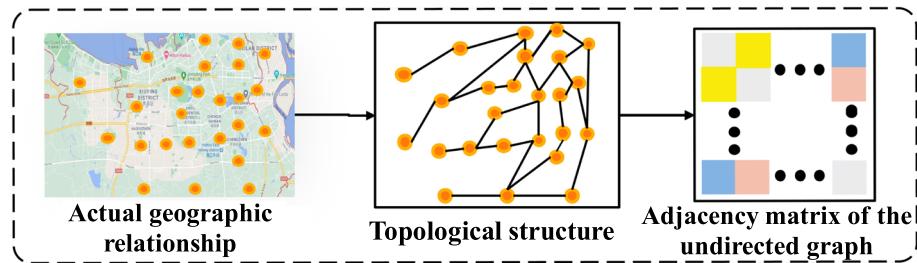


Fig. 1 Distance adjacency matrix. The spatial distance is converted to topological structure and then converted to the adjacency matrix of the undirected graph

- We evaluate our model on two real road traffic datasets. After conducting ablation experiments and comparing with several state-of-the-art baselines, MAGCN has better predictive performance and evaluation metrics.

2 Related Work

2.1 Grid Division Method

For fixed-point traffic flow prediction based on road sensors [22–24], it is not suitable for global traffic prediction and route planning because of the small distribution and fixed location of road sensors, especially when data are collected only from major highways, streets and bus stops [25]. Therefore, we choose to divide the city into grids. The existing grid division method divides spatial maps into equal-sized grids by latitude [25–34]. However, this causes the attributes of all locations included within each grid to be random and scattered. We divide the city into unequal-sized grids based on the regions formed by the aggregation of attributes of locations (i.e., functional areas). Each grid is then labeled with the serial number and the attribute to which it belongs for follow-up data statistics and analysis.

2.2 Constitution of Adjacent Matrix

CNN is designed for structured data, mainly for Euclidean correlation modeling between different regions. However, most things in the real world are represented as unstructured sequences [35] or grid data, such as images [36], speech [37], and natural language sequences [38]. Therefore, we adopt GCN that can process Non-Euclidean structure data. The calculation of the adjacency matrix [39] is greatly important in GCN. The spatial correlation between nodes conforms to the topological relationship, which can be converted into an adjacency matrix for the calculation process of GCN, as shown in Fig. 1. The spatial correlation in the T-GCN [40] model is represented by the connectivity between nodes, and the adjacency matrix is constructed by 0 and 1. Similar to STGCN [26] and ASTGCN [24], the spatial correlation of MAGCN is represented by the distance between nodes, and the adjacency matrix of the undirected graph is constructed by the final calculated real distance.

2.3 Functional Area-based Origin-Destination Pairs (FOD)

In works that predict the traffic flow within one grid [25, 26, 33, 34, 40–42], the interaction between grids can only be represented implicitly through the adjacency matrix. The limitation of such a calculation is that it lacks the interaction of traffic flow in different functional areas in the time sequence. In the MAGCN model, we replace the prediction of point-value corresponding to a single region with the prediction of two directed edges corresponding to pairs of regions. Therefore, it is necessary to add one more dimension to the output in the point-value prediction to distinguish the inflows and outflows. By multi-step calculation of the edge values, we can also obtain the flow within one region. In actual urban planning, the prediction of edge values can be useful for rationalizing vehicle routes as well as relieving traffic congestion between areas. For example, during the morning rush hour, the prediction of edge values can be used for community-to-business traffic scheduling.

3 The Proposed Method

3.1 Data Modeling

Firstly, we select an area with the densest trajectory data on the map and opt for all the data belonging to that area.

Four types of areas are defined based on attributes, i.e., the residential area, the commercial area, the school district, and the leisure area. Then we reproduce the dataset in the case of binary classification and ternary classification. The binary classification includes the residential area and the commercial area. The ternary classification includes the residential area, the commercial area, and the school district.

The city is divided into unequal-sized grids according to attributes to make it graph-structured. The average of the distance between the pick-up and drop-off locations in all opted data is used as the approximate standard to determine the grid size on the map. When the grid size is too large, the total number of grids will decrease, and the number of traffic interaction data to be discarded in the same grid will also increase; when the grid size is too small, the total number of grids will increase, and the feature matrix will be too sparse.

Taking out the latitude and longitude of the four vertices of each grid is used to determine the range of the grid. And according to the latitude and longitude of the original data and the location of the grid, mark the attributes and serial numbers of the functional areas in the pick-up and drop-off positions. Then, the adjacency matrix Tl of the directed graph is constructed, consisting of enumeration values that distinguish the type of interaction, which is determined by the attributes to which the two regions that constitute the interaction belong (e.g., there are $4 * 4$ types of interactions in quaternary classification, so 16 values are used to distinguish the interactions).

We need to calculate two more matrices A and V . A is a distance-based adjacency matrix (i.e., the adjacency matrix of an undirected graph), representing the distance between different grids. V represents the total flow values of the FOD pairs in time steps. In the calculation of matrix A , we use the distance between the center points of the grid to represent the distance between each region. Since we divide the map into unequal-sized grids, it is impossible to get the coordinates of all center points by only one calculation. According to the coordinates of the four vertices of the obtained grid, we use formulas to find the coordinates of each center point. The calculation processes of the latitude and the longitude of the grid center are shown

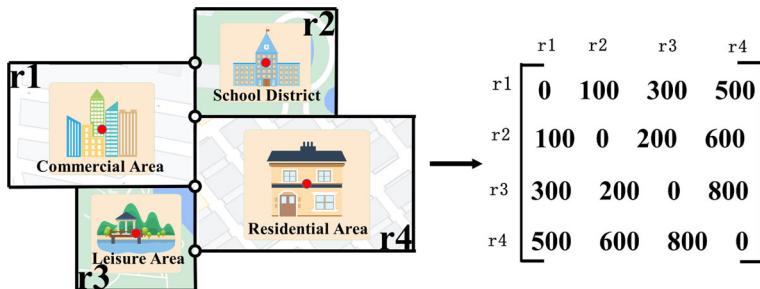


Fig. 2 An example of transforming spatial position into an adjacency matrix of an undirected graph

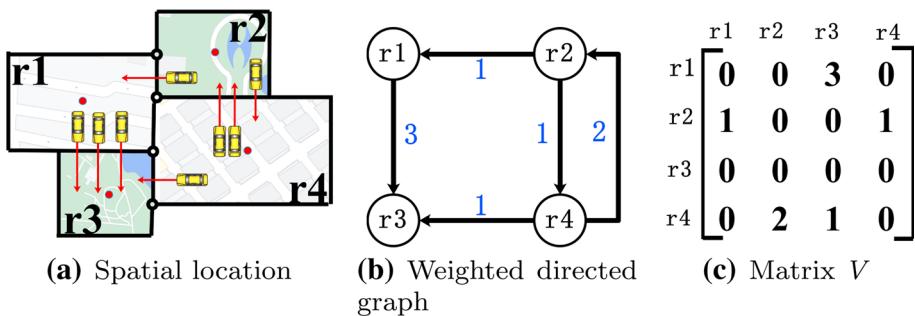


Fig. 3 An example of converting data in spatial location into a weighted directed graph and then into a matrix

in Eqs. (1, 2). The calculation process of the distance between two locations on the earth by latitude and longitude is demonstrated in Eq. (3),

$$\varphi = \frac{|\varphi_2 - \varphi_1|}{2} \quad (1)$$

$$\lambda = \frac{|\lambda_2 - \lambda_1|}{2} \quad (2)$$

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (3)$$

where d represents the distance between two regions, φ and λ represent the latitude and the longitude of the center point, respectively, r is the radius of the earth, φ_1 and φ_2 represent the latitude of two vertexes, λ_1 and λ_2 represent the longitude of two vertexes. The distances between all regions constitute adjacency matrix A , as shown in Fig. 2.

As shown in Fig. 3, in the matrix V , the first row stores the total flow values of the interaction from $r1$ to the rest of the grids in one time step, and so on until we complete the construction of the matrix V . Each value of matrix V represents not the flow within the node but the edge value between the nodes. The flow in the spatial location Fig. 3a is abstracted into a weighted directed graph Fig. 3b and then converted into a matrix Fig. 3c. We can see that the flow value from $r2$ to $r4$ is 1, and the flow value from $r4$ to $r2$ is 2. We designed the output data into four dimensions, which are $[batch_size, num_nodes, channels, num_nodes]$, where *channels* is the product of the number of time steps and 1 (1 is the number of features, and the feature is traffic flow).

After completing the production of the dataset, we use the *Z-Score* method to standardize data. The mean value of the data processed by *Z-Score* is 0, and the standard deviation is 1.

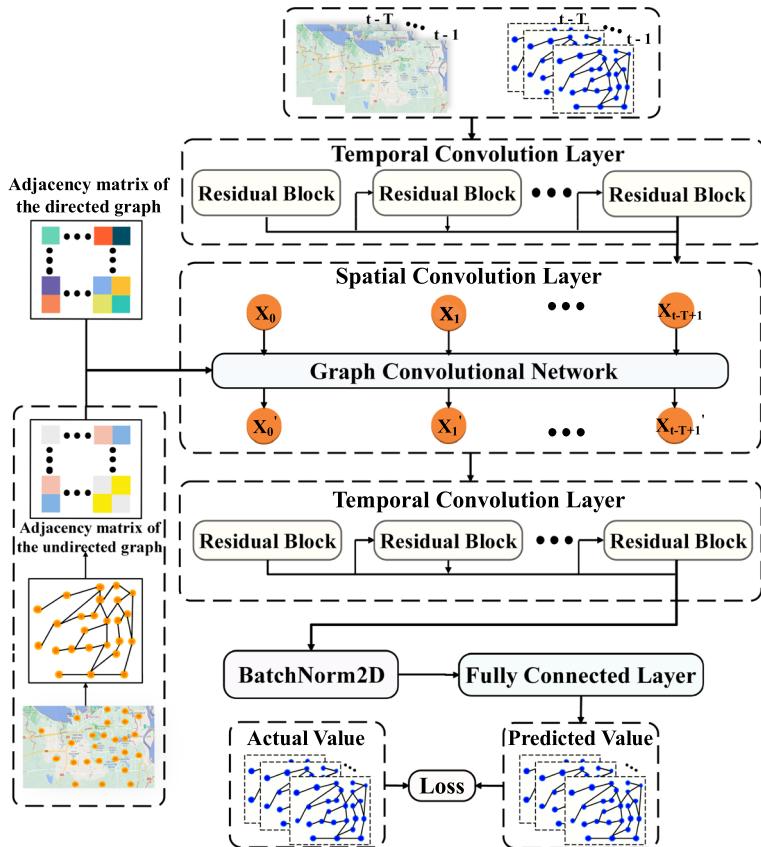


Fig. 4 The structure of the MAGCN model

The computation is shown in Eq. (4),

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

where z is the converted z-score, x is the original value, μ is the average value of the overall sample space, and σ is the standard deviation of the overall sample space. The main purpose of using *Z-Score* is to uniformly convert data of different magnitudes into the same magnitude and use the calculated *Z-Score* value to measure uniformly to ensure the comparability of data.

3.2 Multi-Attribute Graph Convolution Network

3.2.1 Overview

The structure of the MAGCN model is shown in Fig. 4. $\{x_0, x_1, \dots, x_{t-T+1}\}$ represents the time series data of traffic flow processed by the first Temporal Convolution Network. \hat{Y} represents the predicted value of the MAGCN model for traffic flow and Y represents the real value of the traffic flow.

The Multi-Attribute Graph Convolution Network (MAGCN) model is mainly composed of two Temporal Convolution Layers, a Spatial Convolutional Layer in between, *BatchNorm2D* [43], and a Fully Connected Layer. The input data are fed into the first Temporal Convolution Layer in the format of a graph [44], instead of simple linearization and splicing. The distance relationship between the grids is mapped to an adjacency matrix of the undirected graph, which is normalized to make the data comparable while maintaining the relative relationship between the data. The normalized adjacency matrix A and Tl , and the output of the first Temporal Convolution Layer are fed into the Spatial Convolution Layer for capturing the topological characteristics of the road network to obtain spatial correlation. The second Temporal Convolution Layer calculates the temporal dependence of the fused data. Since MAGCN is entirely composed of convolution networks, it will cause gradient disappearance and gradient explosion. Therefore, we add *BatchNorm2D* to solve these problems in the process of reverse propagation. By using Fully Connected Layer to map the learned distributed feature representation to the sample markup space, the best parameters of the model are selected by using Adam optimizer, and then the error between the predicted values and the real values is calculated by *MSELoss*.

3.2.2 Spatial Convolution Layer

Road networks are often represented as topological graphs on graph theory [45] with corresponding relationships established by vertices and edges. CNN enables the extraction of spatial features, but it considers the data to be translation invariant, leading it to be mainly utilized to deal with neatly aligned non-Euclidean data and cannot be directly applied to graphs with different degrees at each node. Therefore, we adopt GCN that can be used directly on graph-structured data to extract highly valuable spatial correlations between regions.

Graph Convolution Networks (GCN) is a neural network that operates on graph data and extracts spatial features of topological graphs. GCN extends the convolution operation from traditional data (such as images and videos) to graph data. It links Spectral graph theory with the deep neural network, in which the simple generalization of Spectral graph theory is to study the properties of graphs utilizing the eigenvalues and eigenvectors of the Laplacian matrix of graphs [46]. GCN learns the function mapping $f(\cdot)$ by using the node v_i in the mapping graph, which can aggregate its feature x_i and its neighbor feature $x_j (j \in N(v_i))$ to generate a new representation of node v_i . That is to use the edge information to aggregate the node information to generate a new node representation. N vertices in the graph will always be affected by their neighboring or farther vertices and constantly change their state until they reach equilibrium.

The convolution is relatively simple to compute in the Fourier domain and to perform the Fourier transform on the graph, we use the eigenvectors of the Laplacian matrix. The Laplacian matrix is the basic tool of spectral analysis that can define the derivative and depict the smoothness of the signal on the graph. To introduce its own degree matrix and solve the self-transfer problem, we use the improved Laplacian matrix. The computation is shown in Eq. (5),

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (5)$$

where $D \in R^{N \times N}$ is a degree matrix and A is the adjacency matrix that considers the state of neighboring nodes while introducing the state of the node itself.

However, the eigendecomposition of the Laplacian matrix is inefficient in the decomposition process of large graph structures. Therefore, the computational complexity of the Laplacian matrix can be reduced by using Chebyshev polynomial approximation [47]. More-

over, the previous 2D-Conv can only capture local features but ignore the connectivity and globality of the traffic network. In this paper, we directly use data in graph structure to extract high order features in the spatial domain and use Chebyshev polynomial to approximate its value. The definition of Chebyshev graph convolution based on K -order Chebyshev polynomial is shown in Eq. (6),

$$g_\theta *_G x = g_\theta(L)x \approx \sum_{k=1}^K \theta_k T_k(\tilde{L})x \quad (6)$$

where $g_\theta *_G x$ is the spectral convolution of the input data $x \in R^N$ with the filter $g_\theta = \text{diag}(\cdot)$ in the Fourier domain, $*_G$ denotes the convolution operation of a graph, and N is the number of nodes. k is the kernel size of the graph convolution. $\theta \in R^K$ denotes the vector of Chebyshev coefficients. $L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ represents the Laplacian matrix, $\tilde{L} = \frac{2L}{\lambda_{\max}} - I_n$ is the scaled eigenvector matrix. The purpose of scaling is to satisfy the condition of truncated expansion of the K^{th} order of Chebyshev polynomial $T_k(x)$. The maximum is the maximum eigenvalue (i.e., spectral radius) of the Laplace matrix, I_n is the unit matrix, and the Chebyshev polynomial is defined recursively as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, $T_0(x) = 1$, $T_1(x) = x$. Eventually, the complexity of the whole operation decreases from $O(n^2)$ to $O(K|E|)$, which is linear only in the number of edges E .

3.2.3 Temporal Convolution Layer

Causal Convolution Different from the traditional convolutional neural network, causal convolution only focuses on historical information without considering future information. It is a strict one-way time constraint model, i.e., for the output data y_t , its input can only be the data at time t and before time t . The purpose of adding causal convolution is to ensure that historical data are not missed. The computation of causal convolution is shown in Eq. (7):

$$(F * X)(x_t) = \sum_{k=1}^K f_k x_{t-K+k} \quad (7)$$

Dilated Convolution If we consider the long ago variable x , the convolution layer must be increased. However, the increase of convolution layers leads to the disappearance of gradients, complicated training, and poorer fitting results. To solve these problems, the dilated convolution appears. Dilated convolution is used to expand the receptive field of the convolution process and capture multi-scale context information. The convolution kernel dilation is to expand the convolution kernel to the scale of expansion scale constraint and fill the area not occupied in the original convolution kernel with zero. Since the information obtained from a long distance is not highly relevant, we change the dilation factor according to an exponential function of 2. The computation for dilated convolution is shown in Eq. (8):

$$y_t = \sum_{i=0}^{K-1} f_i x_{t-id} \quad (8)$$

The structure of the dilated causal convolution [48, 49] is shown in Fig. 5. k is the size of the convolution kernel, which defines the receptive field of the convolution operation. The larger the value of k , the more historical information can be traced and the more layers are hidden. *Dilation* is the dilation rate, which controls the number of intervals between the points of the kernel. Increasing *dilation* or k increases the range of the receptive field.

Fig. 5 Dilated causal convolution with the kernel size (3,3)

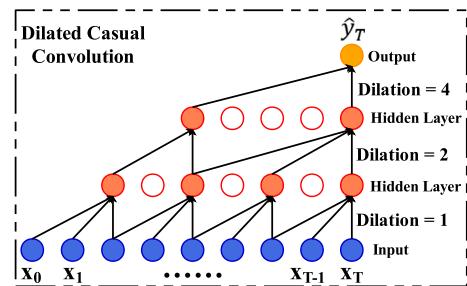
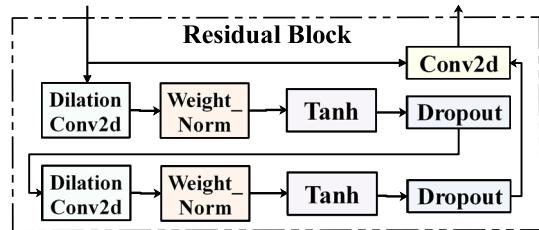


Fig. 6 Structure of the residual block



Residual Structure Even if we use the dilated causal convolution, sometimes the model may still be very deep. Deeper networks are prone to gradient disappearance and gradient explosion. In order to deepen the network while reducing complexity and ensuring accuracy, we added the residual structure [50], which also allows the temporal convolution layer to be more generic, and the structure of the residual block is shown in Fig. 6. The residual block mainly introduces two layers of dilated causal convolution, activation, normalization, regularization, and nonlinear mapping methods. Each residual block is responsible for two main aspects, i.e., tracking historical information over a longer period and acquiring features over a larger receptive field. Since our experimental data are distributed in four-dimensional space, we convert the dilated convolution to Conv2d. Weight Normalization [51] is performed to normalize the inputs of the hidden layer as well as to accelerate the convergence of the model. After clipping the padding part, Tanh is applied to introduce nonlinearity and compress the data to the range of $[-1,1]$. Dropout is 0.2, which effectively mitigates overfitting and achieves regularization to some extent.

Ultimately, the temporal convolution layer integrates the modeling capability of the time-domain and the feature extraction capability of the low-parameter convolution, which is well implemented to capture long-term dependent information in a non-recursive form.

3.2.4 Post-processing

Data normalization can improve the convergence speed and accuracy of the model. After two-layer temporal convolution and one-layer spatial convolution, the *BatchNorm2D* layer is added for data normalization, which can accelerate the convergence rate of the network so that the data will not be too large to cause network instability before ReLU activation [52]. The computation of *BatchNorm2D* is shown in Eq. (9),

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \epsilon}} * \gamma + \beta \quad (9)$$

where x is the input data that need to be normalized, $E(x)$ is the mean value of the input data x , $\sqrt{Var(x)}$ is the variance of the input data x , ϵ is the variable added to prevent the denominator of zero, and its value is equal to 1e-5. γ and β are the affine of input values, i.e., linear transformation. γ and β are the learning parameters of the model, and their default values are 1 and 0 respectively. The affine contains the results without affine so that the introduction of Batch Normalization at least does not reduce the efficiency of the model. During training, this layer calculates the mean and variance of each input and uses the moving average. The $\frac{x-E(x)}{\sqrt{Var(x)}}$ obtained during training is used to standardize validation data. The fully connected layer maps the learned distributed feature to the sample markup space. Adam optimizer is chosen to obtain the optimal parameters by calculating the adaptive learning rate. Then *MSELoss* is computed to measure the error between the predicted value \hat{y}_i and the real value y_i , which is shown in Eq. (10):

$$\text{loss}(y_i, \hat{y}_i) = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 \quad (10)$$

4 Experiment

4.1 Datasets & Settings

4.1.1 Datasets

In the Haikou city taxi order dataset, the orders are dated from May 1, 2017, to October 31, 2017, including the pick-up and drop-off dates/time and latitude and longitude of the pick-up and drop-off locations. The range of data we selected is shown in Fig. 7a, with a longitude range of [110.20, 110.40] and a latitude range of [19.95, 20.05]. The time of the selected data was from 08:00 to 22:00 from May 1 to July 31, and the time step was set to 2 hours. The number of grids divided on the map is 24, 21 and 18 in quaternary, ternary, and binary classification, respectively.

In the Chengdu city taxi order dataset, the orders are dated from November 1, 2016, to November 30, 2016, including the pick-up and drop-off dates/time and latitude and longitude of the pick-up and drop-off locations. We selected the data range as shown in Fig. 7b, with a longitude range of [103.95, 104.15] and a latitude range of [30.65, 30.75]. The time of the selected data is from 8:00 to 22:00 on 30 days, and the time step is set to 2 hours. The number of grids divided on the map was 60, 48, and 42 in quaternary, ternary, and binary classification, respectively.

4.1.2 Settings

All experiments were implemented in Python and Pytorch 1.9.0 on NVIDIA GeForce RTX 2080Ti GPU and Intel Xeon W-2133 CPU. During training, the batch size was set to 64, and the running epoch was 200. We repeat all the experiments 5 times and report the mean of the different runs. We adopt the Adam optimizer to optimize the parameters with an initial learning rate of 0.001. *MSELoss* is then utilized to train the models.



(a) Haikou City, Hainan Province, (b) Chengdu City, Sichuan Province, China

Fig. 7 Geographical scope selected in the dataset

4.2 Baselines

To evaluate the overall performance of our model, including the traditional recurrent neural networks GRU and LSTM, spatio-temporal neural network STDN, graph convolution network GCN, and graph spatio-temporal neural networks T-GCN, STGCN, and ASTGCN. All the models are trained and evaluated on the same datasets. The structures of baselines in the experiment are as follows:

- *Long Short-Term Memory Network (LSTM)* [17] It is an improved recurrent neural network that can learn long-term dependent information.
- *Gated Recurrent Unit (GRU)* [18] Like LSTM, it can capture the semantic association between long sequences and solve the problem of gradient disappearance in reverse propagation.
- *Spatial-Temporal Dynamic Network (STDN)* [27] It uses CNN to deal with local spatial dependence and LSTM to obtain the temporal dependencies.
- *Graph Convolution Networks (GCN)* [53] It can process data from Non-Euclidean Structure and extract spatial features from topological graphs.
- *Temporal Graph Convolutional Network (T-GCN)* [40] It consists of GCN and GRU.
- *Spatio-Temporal Graph Convolutional Networks (STGCN)* [26] It is a spatio-temporal graph convolution model based on the spatial method, which consists of multiple spatio-temporal convolution modules.
- *Attention Based Spatial-Temporal Graph Convolutional Networks (ASTGCN)* [24] It integrates three modules to process recent, daily, and weekly fragments for capturing multi-scale temporal correlation.

4.3 Evaluation Metrics

When we deal with more data, it is unrealistic to check each value to see if there are one or some outliers. Therefore, we use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to determine the presence of large but uncommon errors, and R-Square to quantify the variance of the model error and to measure the fit of the model. MAE represents the mean of absolute error between predicted and real values, which can better reflect the actual situation of the predicted error. MAE is a linear fraction, and all individual differences have equal weights on average. RMSE represents the sample standard deviation of the difference between the predicted and real values, which can well reflect the precision of predicted results.

RMSE shows the discreteness of the sample since it enlarges the gap between large errors. The range of R-Square is $[-\infty, 1]$. The closer the value is to 1, the stronger the interpretation of the variables of the equation is for y .

The computations of MAE, RMSE, and R-square are shown in Eqs. (11, 12, 13),

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (11)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (12)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (13)$$

where \hat{y}_i is the predicted value, y_i is the ground truth and \bar{y}_i is the mean of y_i . Although MAE can obtain an evaluation value, because it is based on the calculation process of absolute error, we cannot completely know the pros and cons of the model performance represented by this evaluation value. Only by further comparing the models can we get a more comprehensive understanding of the performance of the models.

4.4 Unequal-Sized Grids & FOD

Figure 8 shows the flow variation within one grid after dividing Haikou city into equal-sized grids. Figure 9 illustrates the flow variation between grid pairs after dividing Haikou City into equal-size grids. Figure 10 presents the flow variation of one kind of FOD after dividing Haikou City into unequal-sized grids, from the residential area to each other functional area, where every four units represent one day, for a total of six days. Taking one day as an example, R-C, R-L, R-S, and R-R are the interactions between the residential area and commercial area, residential area and recreational area, residential area and school area, and residential area and residential area, respectively. To reflect the overall trend of the interactions, we use the mean value to represent the flow. Through the intuitive comparison between Figs. 8, 9 and 10, we can see that the flow trends of the four interaction types for six consecutive days in Fig. 10 are generally consistent, and the dataset can be extracted as a regularity, which is helpful to improve the prediction accuracy of the model and simply proves that MAGCN has certain correctness and validity. Finally, the visualization of the lower triangle of the distance-based symmetric adjacency matrix is shown in Fig. 11a, and the visualization of the interaction type-based asymmetric adjacency matrix is shown in Fig. 11b.

4.5 Results

4.5.1 Sequential Forecasting Method

Considering that flow changes are affected by the recent traffic state, we first use a sequential forecasting method. The data of the first 12 time steps are used as the input to predict the flow of the next 4 time steps, i.e., 24 hours of historical data are used to predict the next 8 hours of data. The computation of the input data S is shown in (14),

$$S = \{I_d^{t+u} \mid u = 0, 1, \dots, n-1, n\} \quad (14)$$

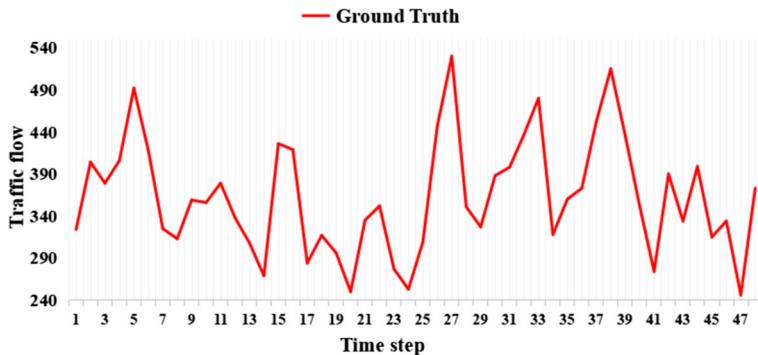


Fig. 8 Traffic flow example within one grid in equal-size grids

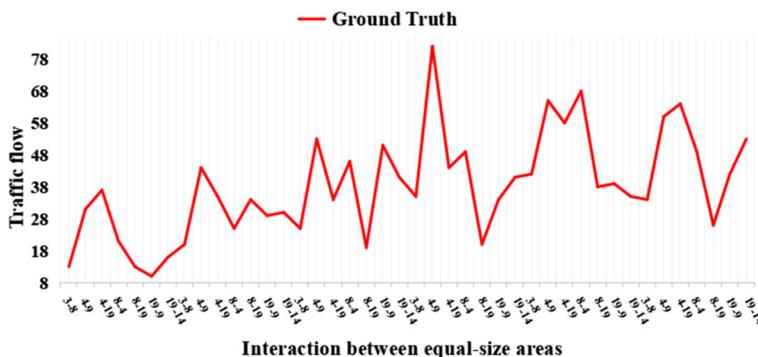


Fig. 9 Example of traffic flow between grid pairs in equal-sized grids

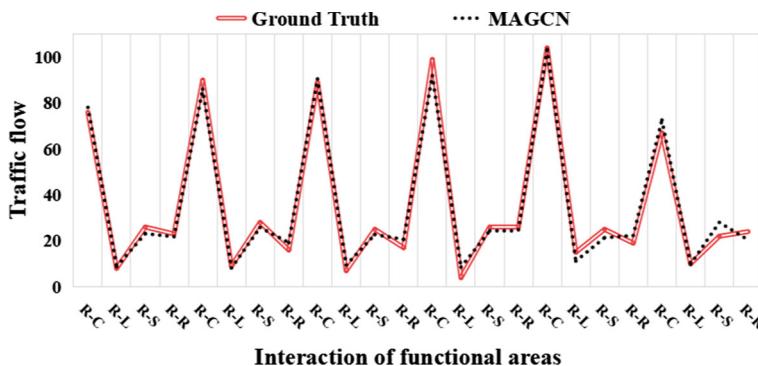


Fig. 10 Example of traffic flow from the residential area to each functional area in unequal-sized grids

where t is the current time step, d is the d^{th} day, and n is the number of required time steps.

The MAE, RMSE, and R-square for the quaternary classification w.r.t. sequential forecasting method are shown in Table 1. From Table 1, we can tell that the MAGCN has the lowest error and the best fit among all models, followed by STGCN and ASTGCN.

There are $4 * 4$ types of interaction in quaternary classification. Since multiple regions belong to the same attribute, the interaction includes the interaction between the same

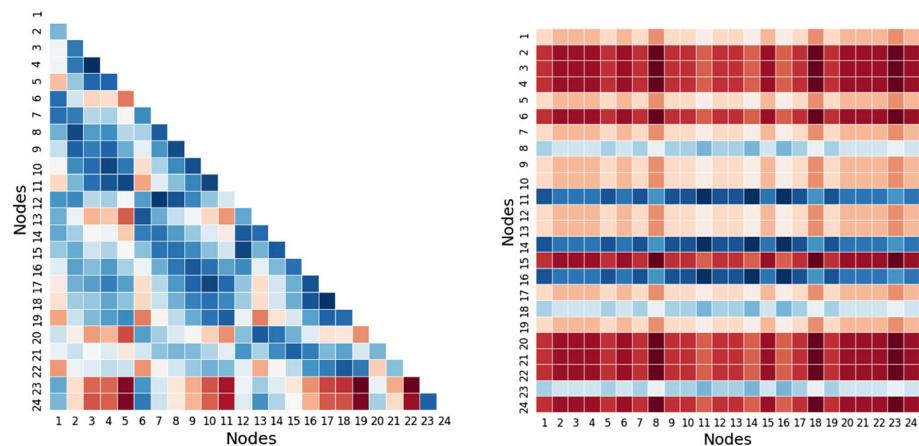


Fig. 11 Visualizations of the adjacency matrices

(b) Adjacency matrix TI of the directed graph

Table 1 Comparisons in quaternary classification w.r.t. sequential forecasting method

Model	Haikou Dataset			Chengdu Dataset		
	MAE	RMSE	R	MAE	RMSE	R
GRU [18]	2.461	3.896	0.877	5.316	6.764	0.816
LSTM [17]	2.459	3.859	0.877	5.303	6.728	0.816
STDN [27]	2.330	3.205	0.890	5.071	6.454	0.841
GCN [53]	2.095	3.493	0.884	4.641	6.038	0.832
T-GCN [40]	1.992	3.078	0.889	4.735	6.124	0.835
ASTGCN [24]	1.930	2.996	0.894	4.679	6.024	0.833
STGCN [26]	1.893	2.857	0.911	4.537	5.791	0.854
MAGCN	1.818	2.751	0.917	4.362	5.495	0.863

The best prediction results, i.e., those of our proposed method, are in bold to distinguish our from other experimental results

attributes (e.g., R–R). To verify that the flow values differ for different interaction situations, we select the average of the traffic flow of 16 interaction situations in two same periods for comparison.

From the spatial perspective, Fig. 12 shows the predicted and real values of 16 kinds of regional interactions in two certain periods, and Fig. 13 illustrates the visualizations of the entire matrix V at a certain moment. Taking the C–R interaction as an example, the predicted and real values are the averages of the flow between commercial areas and residential areas. The x-axis shows 16 kinds of interaction: C–C, C–R, C–L, C–S, L–C, L–R, L–L, L–S, R–C, R–R, R–L, R–S, S–C, S–R, S–L, S–S (C: commercial area, L: leisure area, S: school district, R: residential area). The ordinate represents the flow value for the same period of the day. The red and blue lines represent the change of the flow values from 16: 00 to 18: 00 and from 8: 00 to 10: 00, respectively.

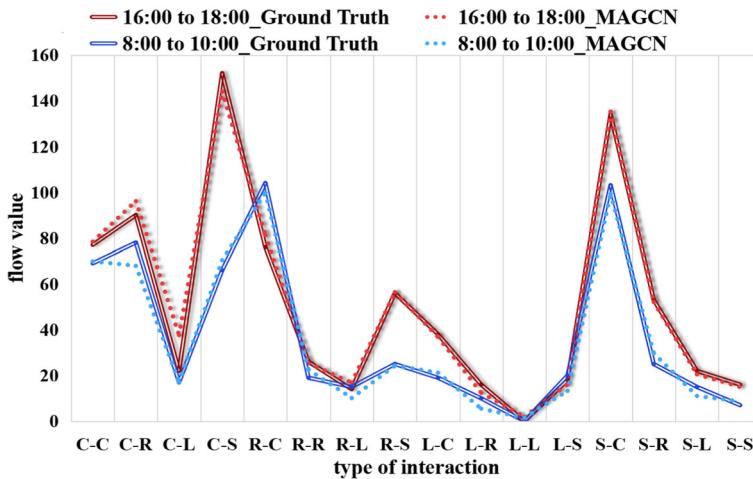


Fig. 12 Prediction example. Sixteen regional interactions in two periods w.r.t. sequential forecasting method

It can be seen in Fig. 12 that there is not much traffic flow for interactions involving leisure areas, which is in line with reality. For students without self-study at night, they mainly go to the commercial area to eat something or shop with friends after school from 16: 00 to 18: 00, so there is a peak at S–C. For students with self-study at night, they return to school after dinner in the commercial area or at home, so there are peaks at R–S and C–S. From 16: 00 to 18: 00 is the rush hour, so there is a peak at C–R. From 8: 00 to 10: 00 is the peak period for work and school, so there are peaks at R–C and R–S. The R–S value is not very large because the schools have started classes around 8: 00, and the peak at S–C is because parents send their children to school and then go to work. From 8: 00 to 10: 00, evening workers came home, so the C–R peak appeared.

The change of flow in the same interaction is related to time. The following Figs. 14 and 15 show the results of our model compared with STGCN and ASTGCN. Figures 14 and 15 are the predicted and real values in 85 time steps from the commercial area to the residential area and from the commercial area to the school district, respectively. The x-axis represents 85 time steps. We take every 2 hours as a time step and every 8 time steps as a whole day. The ordinate is the flow value. The first 56 time steps in Fig. 15 represent the flow value for 7 days. Since fewer students go to school on weekends, the flow value for the first 16 time steps is slightly smaller. We select three models with smaller errors to compare the degree of the fitting. According to Figs. 14 and 15, compared with STGCN and ASTGCN, the fitting degree between the predicted values and the real values of MAGCN is the highest. This is because we make full use of the dilated convolution and causal convolution in the temporal convolution layer. However, the fitting results of MAGCN on local peaks remain to be improved due to the fact that the period of data we selected is from 8: 00 to 22: 00, which is not a complete day and the time series are not completely continuous.

4.5.2 Periodic Forecasting Method

In this section, we utilize the periodic forecasting method, i. e., the flow data of a certain time step in each of the first seven days as input to predict the flow at the same time step on the

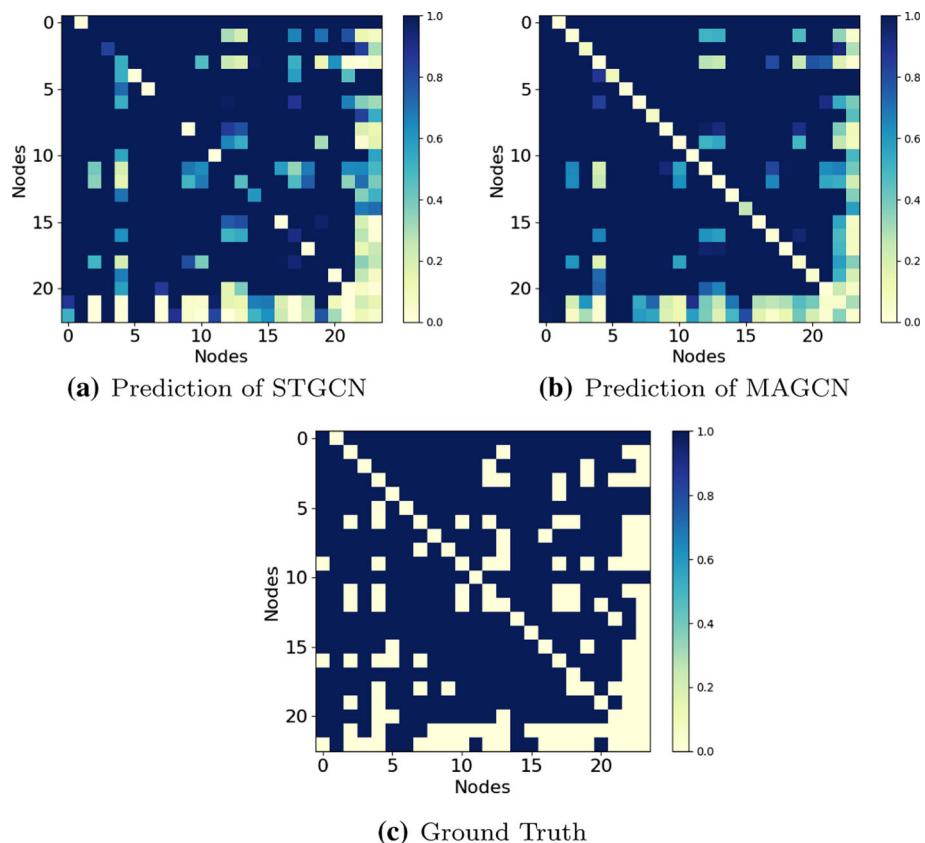


Fig. 13 Visualizations of prediction examples in quaternary classification w.r.t. sequential forecasting method

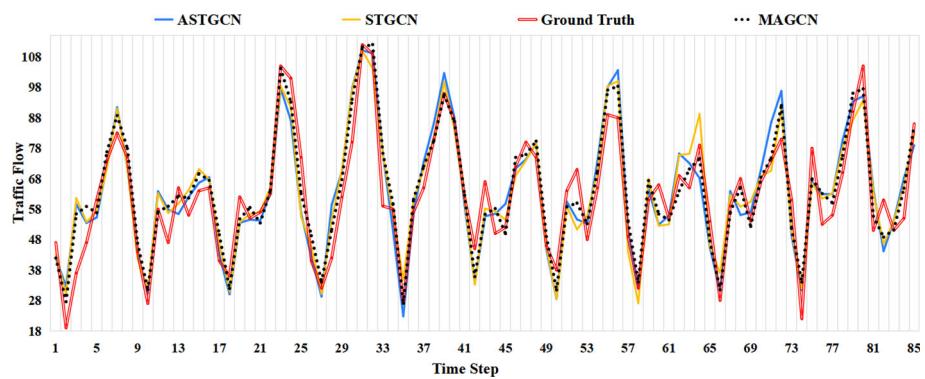


Fig. 14 Prediction example. C-R interaction in quaternary classification w.r.t. sequential forecasting method

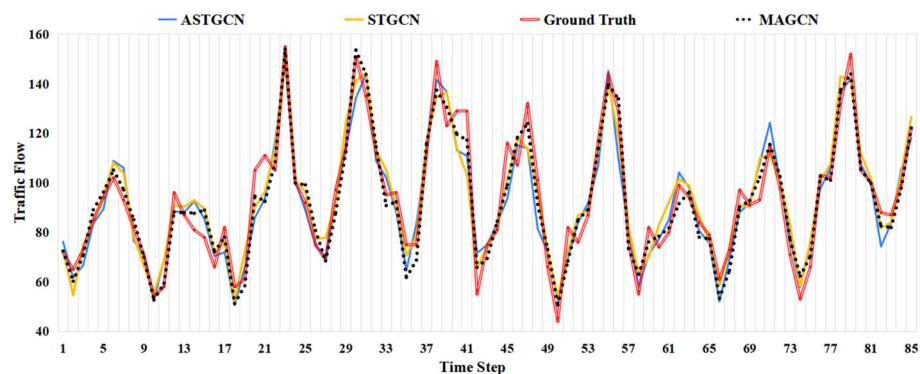


Fig. 15 Prediction example. C-S interaction in quaternary classification w.r.t. sequential forecasting method

Table 2 Comparisons of different models in quaternary classification w.r.t. periodic forecasting method

Model	Haikou Dataset			Chengdu Dataset		
	MAE	RMSE	R	MAE	RMSE	R
GRU [18]	2.370	3.511	0.884	5.159	6.592	0.825
LSTM [17]	2.362	3.448	0.885	5.143	6.588	0.826
MAGCN-w/o-2L	2.359	3.442	0.893	5.126	6.582	0.837
STDN [27]	2.211	3.292	0.900	4.898	6.243	0.849
GCN [53]	1.946	2.865	0.897	4.482	5.808	0.844
T-GCN [40]	1.841	2.848	0.901	4.587	5.919	0.844
ASTGCN [24]	1.826	2.780	0.908	4.473	5.822	0.841
MAGCN-w/o-1L	1.813	2.674	0.916	4.441	5.616	0.851
STGCN [26]	1.788	2.655	0.922	4.397	5.561	0.865
MAGCN-w/o-TI	1.731	2.588	0.929	4.196	5.393	0.871
MAGCN	1.728	2.559	0.932	4.115	5.201	0.879

The best prediction results, i.e., those of our proposed method, are in bold to distinguish our from other experimental results

eighth day. The computation of the input data P is shown in (15),

$$P = \{I_{d+v}^t \mid v = 0, 1, \dots, m-1, m\} \quad (15)$$

where m is the number of days of input data.

Quaternary Classification To further verify the performance of MAGCN, we conduct ablation studies.

- *MAGCN-w/o-2L* remove the first two layers of MAGCN (i.e., the first temporal convolution layer and the spatial convolution layer are removed).
- *MAGCN-w/o-1L* remove the first layer of MAGCN (i.e., the first temporal convolution layer is removed).
- *MAGCN-w/o-TI* remove the type of the interaction as an influencing factor in MAGCN (i.e., the adjacency matrix TI is removed).

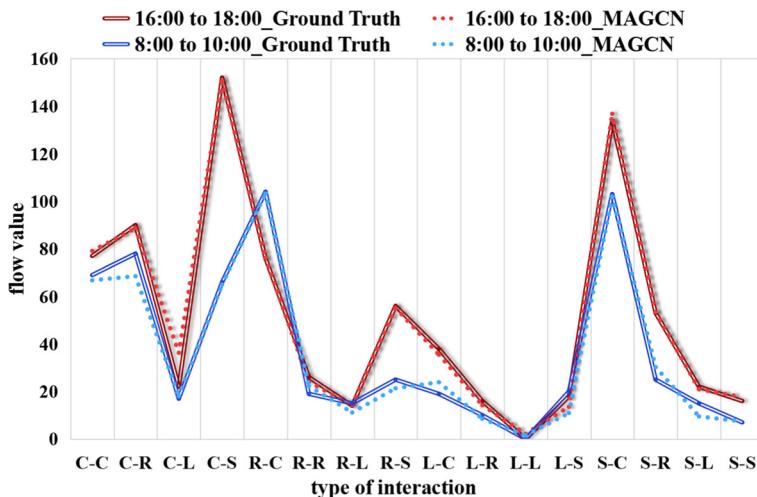


Fig. 16 Prediction example. Sixteen regional interactions in two periods w.r.t. periodic forecasting method

We compared MAGCN with other methods on two datasets separately and the results are summarized in Table 2. Our proposed method achieves the highest R-square and the lowest MAE and RMSE.

In the Haikou dataset, MAE and RMSE of MAGCN-w/o-1L are 0.7 and 3.8% lower than those of ASTGCN, respectively. MAE and RMSE of STGCN are 1.4 and 0.7% lower than those of MAGCN-w/o-1L, respectively. MAE and RMSE of MAGCN-w/o-TI are 3.2 and 2.5% lower than those of STGCN, respectively. MAE and RMSE of MAGCN are 0.2 and 1.1% lower than those of MAGCN-w/o-TI, respectively. In the Chengdu dataset, MAE and RMSE of MAGCN-w/o-1L are 0.7 and 3.5% lower than those of ASTGCN, respectively. MAE and RMSE of STGCN are 1.0 and 1.0% lower than those of MAGCN-w/o-1L, respectively. MAE and RMSE of MAGCN-w/o-TI are 4.6 and 3.0% lower than those of STGCN, respectively. MAE and RMSE of MAGCN are 1.9 and 3.6% lower than those of MAGCN-w/o-TI, respectively.

We can see that the metrics derived from the experiments on the Haikou dataset are better overall than those derived from the experiments on the Chengdu dataset. The reason for this is that there are fewer data in the Chengdu dataset compared to Haikou, and there are more leisure areas in Chengdu, which are more dispersed, thus leading to a more random distribution of vehicles.

From a spatial perspective, Fig. 16 shows the predicted and real values of 16 kinds of regional interactions in two certain periods, and Fig. 17 illustrates the visualizations of the entire matrix V at a certain moment w.r.t. periodic forecasting method. In Fig. 16, the x-axis represents 16 kinds of interaction, and the ordinate represents the traffic flow of the same period on a certain day. The red and blue lines represent the change of the traffic flow from 4 p.m. to 6 p.m. and from 8 a.m. to 10 a.m., respectively. There is the same trend in Figs. 12 and 16, but after comparing the prediction and metric, we can see that the predictions of the periodic forecasting method are much closer to the real values as a whole.

From a temporal perspective, the variation of the flow in the same interaction is time-dependent. Figures 18 and 19 respectively show the predicted and real values of the flow in 44 time steps from the commercial area to the residential area and from the leisure area to

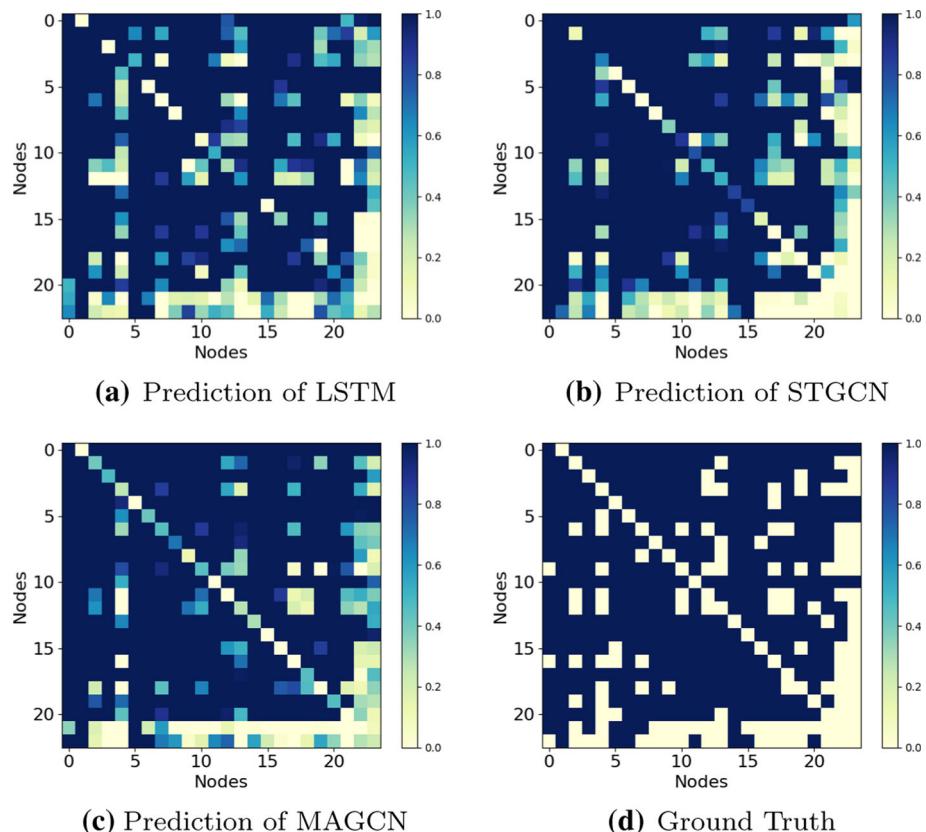


Fig. 17 Visualizations of prediction examples in quaternary classification w.r.t. periodic forecasting method

the leisure area. The x-axis represents 44 time steps. We take every 2 hours as a time step and every 8 time steps as a whole day. The ordinate is the traffic flow. In Fig. 18, there are two peaks in a day, which are after working on the night shift and the day shift. Due to the different working days, the night shift can change from 8: 00 to 12: 00, and the day shift can change from 16: 00 to 20: 00. In Fig. 19, there is not much traffic flow from the leisure area to the leisure area so it is difficult for the model to obtain the regularity of data. When the flow value changes near 1 for a long time, the model is difficult to predict the sudden peak, leading to under-fitting.

Figure 20 shows that the overall performance of the periodic forecasting method is better than that of the sequential forecasting method. The MAE and RMSE of MAGCN in the periodic prediction method are 4.9 and 6.9% lower than those of MAGCN in the sequential prediction method, respectively because the period we choose is from 8:00 to 22:00, which is not a complete day and the time series are not fully continuous. In predicting the traffic from 8:00 a.m. to 10:00 a.m. on the third day, 8 time steps are needed on the second day and 4 time steps on the first day. Due to the long interval, the traffic on the first night has little effect on the traffic on the second day, and the temporal dependence between the data is reduced.

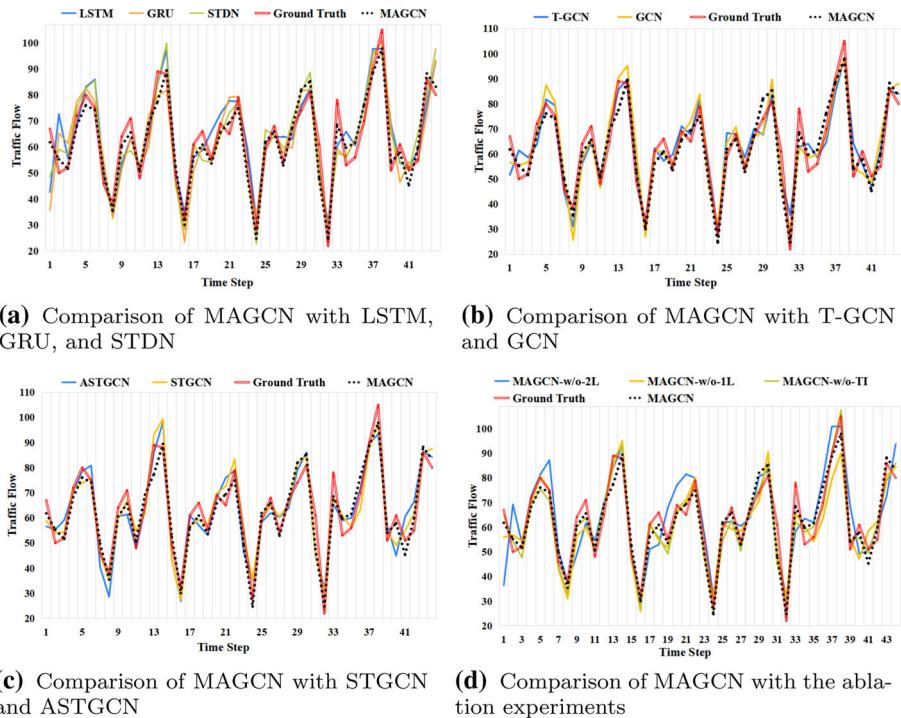


Fig. 18 Prediction examples. C-R interaction in quaternary classification w.r.t. periodic forecasting method

Ternary & Binary Classification Since the classification design of the data has an impact on the prediction accuracy of the model, we regenerated the dataset with binary classification and ternary classification to validate the effectiveness of our proposed method. The binary classification is to exclude the school district and leisure district from the quaternary classification, and the ternary classification is to exclude the leisure district from the quaternary classification. MAE, RMSE, and R of all models in the binary classification and ternary classification calculated by the periodic forecasting method are shown in the following tables.

Haikou dataset in Table 3, MAE and RMSE of MAGCN are 2.9 and 3.6% lower than those of MAGCN-w/o-TI. MAE and RMSE of MAGCN-w/o-TI are 3.1 and 1.5% lower than those of STGCN, respectively. Chengdu dataset in Table 3, MAE and RMSE of MAGCN are 4.9 and 5.6% lower than those of MAGCN-w/o-TI. MAE and RMSE of MAGCN-w/o-TI are 6.9 and 2.1% lower than those of STGCN, respectively.

Haikou dataset in Table 4, MAE and RMSE of MAGCN are 6.5 and 5.3% lower than those of MAGCN-w/o-TI. MAE and RMSE of MAGCN-w/o-TI are 3.3 and 0.5% lower than those of STGCN, respectively. Chengdu dataset in Table 4, MAE and RMSE of MAGCN are 2.8 and 7.8% lower than those of MAGCN-w/o-TI. MAE and RMSE of MAGCN-w/o-TI are 8.1 and 2.2% lower than those of STGCN, respectively. The results demonstrate that MAGCN performs optimally in all the above three classifications.

In Fig. 21a, the x-axis represents 2×2 kinds of interaction, and the ordinate represents the flow value for the same period in a day. The red and blue lines represent the change of the flow values from 16: 00 to 18: 00 and from 8: 00 to 10: 00, respectively. 8: 00 to 10: 00 is the peak period from the residential area to the commercial area, 16: 00 to 18: 00 is the

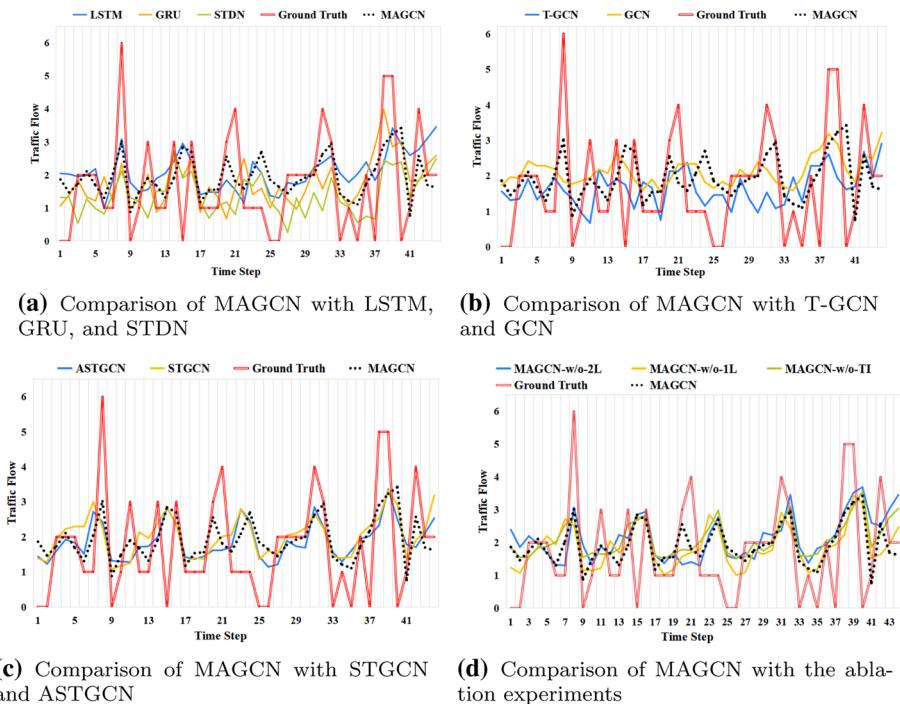


Fig. 19 Prediction examples. L-L interaction in quaternary classification w.r.t. periodic forecasting method

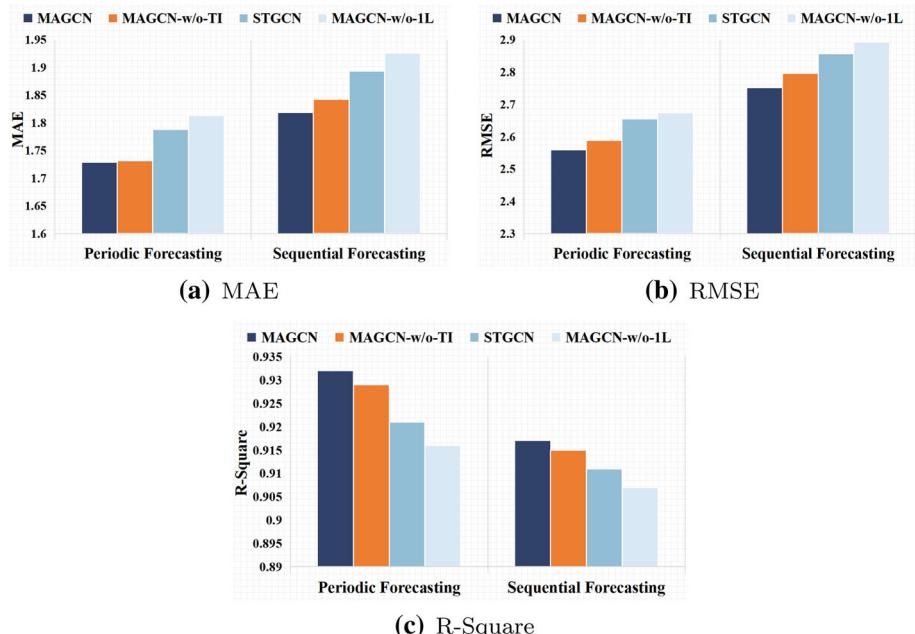


Fig. 20 Metrics in quaternary classification w.r.t. periodic forecasting method and sequential forecasting method

Table 3 Metrics in ternary classification w.r.t. periodic forecasting method

Model	Haikou Dataset			Chengdu Dataset		
	MAE	RMSE	R	MAE	RMSE	R
GRU [18]	2.348	3.415	0.890	5.172	6.564	0.823
LSTM [17]	2.324	3.372	0.891	5.137	6.558	0.823
MAGCN-w/o-2L	2.362	3.437	0.894	5.125	6.576	0.831
STDN [27]	2.184	3.311	0.906	4.891	6.892	0.835
GCN [53]	1.926	2.822	0.901	4.477	5.818	0.844
T-GCN [40]	1.827	2.751	0.902	4.575	5.902	0.843
ASTGCN [24]	1.789	2.699	0.904	4.467	5.824	0.842
MAGCN-w/o-1L	1.812	2.671	0.916	4.439	5.641	0.846
STGCN citeRefJ26	1.784	2.627	0.922	4.393	5.501	0.865
MAGCN-w/o-TI	1.729	2.588	0.929	4.109	5.386	0.862
MAGCN	1.678	2.494	0.931	4.089	5.086	0.873

The best prediction results, i.e., those of our proposed method, are in bold to distinguish our from other experimental results

Table 4 Metrics in binary classification w.r.t. periodic forecasting method

Model	Haikou Dataset			Chengdu Dataset		
	MAE	RMSE	R	MAE	RMSE	R
GRU [18]	2.303	3.376	0.895	5.161	6.539	0.835
LSTM [17]	2.291	3.369	0.897	5.118	6.521	0.838
MAGCN-w/o-2L	2.357	3.436	0.896	5.806	6.499	0.838
STDN [27]	2.150	3.355	0.909	4.893	6.125	0.842
GCN [53]	1.895	2.843	0.911	4.472	5.809	0.846
T-GCN [40]	1.782	2.738	0.904	4.575	5.856	0.843
ASTGCN [24]	1.730	2.706	0.905	4.466	5.822	0.843
MAGCN-w/o-1L	1.812	2.669	0.927	4.592	5.685	0.847
STGCN [26]	1.673	2.597	0.922	4.342	5.493	0.875
MAGCN-w/o-TI	1.730	2.585	0.930	4.104	5.374	0.874
MAGCN	1.618	2.449	0.932	3.989	4.957	0.881

The best prediction results, i.e., those of our proposed method, are in bold to distinguish our from other experimental results

period from the commercial area to the residential area. In Fig. 21b, the x-axis represents 3 * 3 kinds of interaction. From 8: 00 to 10: 00, the peak at S-C is because parents send their children to school and then go to work. From 16: 00 to 18: 00 is the peak period of students returning to school for self-study at night, and it is also the peak period of going home for students without self-study at night. It can also be seen from Tables 3 and 4 and Figs. 22a, 22b, and 18c that the fitting result of the MAGCN model is optimal in all classifications.

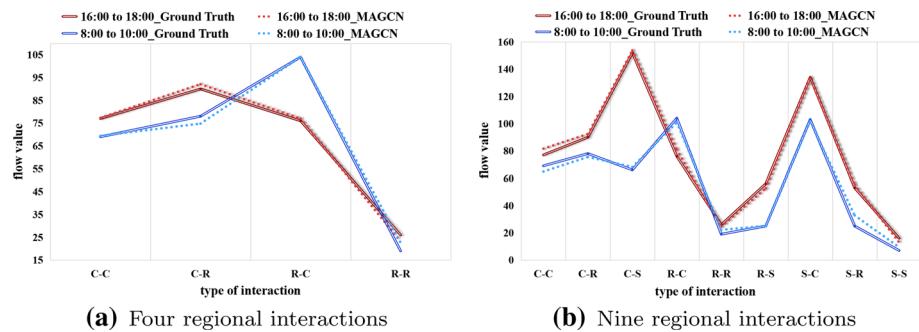


Fig. 21 Prediction examples in two periods w.r.t. periodic forecasting method

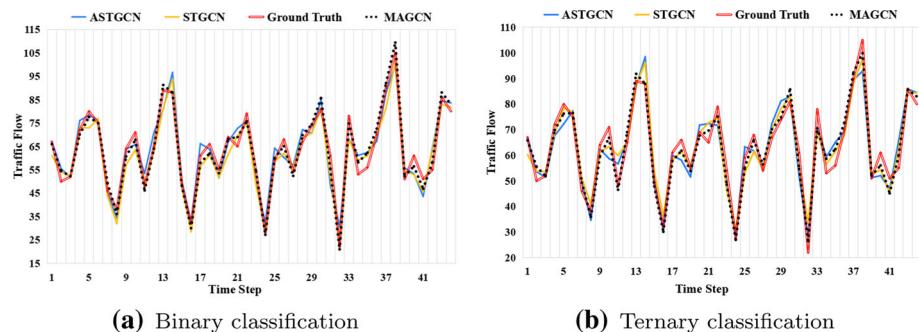


Fig. 22 Prediction examples of C-R interaction w.r.t. periodic forecasting method

5 Discussion

5.1 Predictive Performance

As can be seen from all the Tables and Figures, traditional time series analysis methods are ineffective for prediction because only the temporal variation of traffic states is considered and the calculation of spatial correlation is ignored, indicating that these methods have limited ability to model nonlinear complex traffic data. In contrast, deep learning-based methods can usually obtain better prediction results than traditional time series analysis methods. Based on the LSTM model, CNN is added to constitute the STDN model that realizes the processing of spatio-temporal information. The accuracy of the STDN model has been improved. However, the effect is still poor in this model because it is not applicable to graph structures. GCN is suitable for calculating the spatial relationship of the graph structure, but its predictions are not optimal because of its lack of dependencies on temporal features. T-GCN uses GCN and recurrent neural network to calculate the spatial and temporal relationships, respectively, and its experimental results are better than those of LSTM, GRU, STDN, and GCN, but the shortcomings of recurrent neural network limit the improvement of its prediction accuracy. STGCN, ASTGCN, and MAGCN are composed entirely of convolutions, so the parameters are less and the performance is higher than the model with the recurrent neural network. Compared with STGCN and ASTGCN, we can see that the prediction of MAGCN is closer to the true value at the local peaks caused by rapid changes, mainly because MAGCN makes full use of the dilated causal convolution in the temporal convolution layer based on the

Table 5 Running time (s/epoch) of different models w.r.t. 4D data

STDN	T-GCN	LSTM	GRU	ASTGCN	STGCN	MAGCN	GCN
5.817	5.426	5.030	4.927	2.426	0.158	0.321	0.081

extraction of spatial features. Since the size of the receptive field of standard 1D convolution grows linearly with the number of hidden layers, it makes many layers necessary for capturing longer sequences. However, the receptive field size of the dilated causal convolution grows exponentially with the number of hidden layers, thus achieving a larger field of perception using fewer layers. The dilated convolution can obtain the information of the flow change on the grid, and reduce the influence of data fluctuation in adjacent regions on the accuracy, which produces more stable results of MAGCN in learning spatial characteristics. Causal convolution means that there is no missed historical information or future data, and it can enhance the weight of historical data and the transmission of spatial stability in time sequence. MAGCN-w/o-2L is less capable of modeling spatial features, MAGCN-w/o-1L is unable to extract further temporal features compared to MAGCN, and MAGCN-w/o-TI lacks the type of interaction between functional areas as an influencing factor, so MAGCN is more effective for traffic flow prediction.

5.2 Computation Cost

In addition to the predictive performance of the model, the computation cost also needs to be considered. Taking Haikou Dataset as an example, when the output data is three-dimensional, the parameters of MAGCN are 71,258, occupying 3834.5 MB of GPU memory, and the running time per epoch is 0.297 (s/epoch). When the output data is four-dimensional, the parameters of MAGCN are 72,321, occupying 4159.9 MB of GPU memory, and the running time per epoch is 0.321 (s/epoch). Therefore, compared to the three-dimensional data, the four-dimensional data we adopt has a slight increase in time and spatial complexity. We also compare the running time per epoch for the 8 models in Table 5. Since LSTM and GRU are time series models, they are computed with serial dependencies and cannot be computed in parallel, so the running time is relatively long. T-GCN and STDN introduce GCN and CNN respectively based on recurrent neural networks, so the parameters and computation time are increased. ASTGCN, STGCN, and MAGCN consist entirely of convolution. However, since ASTGCN is designed as a multiplexed network consisting of three kinds of periods (recent, day, and week) and the three are not computed in parallel, it ends up with a relatively long computation time. MAGCN is capable of parallel computation and thus greatly reduces the computation time. In summary, the computation cost of MAGCN is moderate, and increasing the dimension of the output will not make the complexity of the model increase sharply.

6 Conclusion

In this paper, we proposed a Multi-Attribute Graph Convolution Network (MAGCN) for Regional Traffic Flow Prediction. The areas where the order data of taxis in Haikou and Chengdu are concentrated are divided into unequal-sized grids. The traffic flow on the road was regarded as the edge value, so the output data was expanded from three-dimension to four-dimension. In light of the topological characteristics of the road network, we devel-

oped GCN to capture spatial correlation and then designed a Temporal Convolution Layer to capture temporal correlation. Compared to other state-of-the-art models and ablation experiments with the same experimental conditions, the MAGCN model has always obtained the best prediction performance of all metrics in two datasets, which proved the effectiveness and superiority of the MAGCN model in the traffic flow prediction task. While the dilated convolution allows the perceptual field to grow exponentially with the increase of hidden layers, the number of hidden layers grows linearly with the length of the input sequence. Therefore, in the future, we will consider solving the prediction problem for longer-term traffic flow. Meanwhile, we will further optimize the time step and other parameters as well as adding a variety of implicit and external factors to provide a more comprehensive real scene for the model.

Acknowledgements This research was supported in part by National Key Research and Development Plan Key Special Projects under Grant No. 2018YFB2100303, Shandong Province colleges and universities youth innovation technology plan innovation team project under Grant No. 2020KJN011, Shandong Provincial Natural Science Foundation under Grant No. ZR2020MF060, Program for Innovative Postdoctoral Talents in Shandong Province under Grant No. 40618030001, National Natural Science Foundation of China under Grant No. 61802216, and Postdoctoral Science Foundation of China under Grant No. 2018M642613.

References

1. Haydari A, Yilmaz Y (2020) Deep reinforcement learning for intelligent transportation systems: a survey. *IEEE Trans Intell Transp Syst* 23:11–32
2. Zhou T, Han G, Xu X, Han C, Huang Y, Qin J (2019) A learning-based multimodel integrated framework for dynamic traffic flow forecasting. *Neural Process Lett* 49(1):407–430
3. Luo C, Huang C, Cao J, Lu J, Huang W, Guo J, Wei Y (2019) Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm. *Neural Process Lett* 50(3):2305–2322
4. Lv Z, Li J, Dong C, Xu Z (2021) DeepSTF: a deep spatial-temporal forecast model of taxi flow. *The Comput J* bxab178:1–16
5. Cai Z, Zheng X, Yu J (2019) A differential-private framework for urban traffic flows estimation via taxi companies. *IEEE Trans Ind Inf* 15(12):6492–6499
6. Schaffer AL, Dobbins TA, Pearson SA (2021) Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. *BMC Med Res Methodol* 21(1):1–12
7. Hansun S, Charles V, Gherman T, Subanar Indrati CR (2021) A tuned Holt-Winters white-box model for COVID-19 prediction. *Int J Manag Decis Mak* 20(3):241–262
8. Chen G, Long T, Bai Y, Zhang J (2019) A forecasting framework based on Kalman filter integrated multivariate local polynomial regression: application to urban water demand. *Neural Process Lett* 50(1):497–513
9. Chen X, Sun L (2021) Bayesian temporal factorization for multidimensional time series prediction. *IEEE Trans Pattern Anal Mac Intell* 44:4659–4673
10. Zhao J, Sun S (2016) High-order Gaussian process dynamical models for traffic flow prediction. *IEEE Trans Intell Transp Syst* 17(7):2014–2019
11. Zhao A, Wang Y, Li J (2022) Transferable self-supervised instance learning for sleep recognition. *IEEE Trans Multimedia* 2022:1
12. Lv Z, Li J, Li H, Xu Z, Wang Y (2021) Blind travel prediction based on obstacle avoidance in indoor scene. *Wirel Commun Mob Comput* 1–14
13. Wang Y, Lv Z, Sheng Z et al (2022) A deep spatiotemporal meta-learning model for urban traffic revitalization index prediction in the COVID-19 pandemic. *Adv Eng Inf* 53:101678
14. Hu Z, Sun R, Shao F et al (2022) Traffic station classification based on deep spatio-temporal network. *Comput Electr Eng* 97:107558
15. Lv Z, Li J, Dong C, Li H, Xu Z (2021) Deep learning in the COVID-19 epidemic: a deep model for urban traffic revitalization index. *Data Knowl Eng* 135:101912
16. Miebs G, Mochol-Grzelak M, Karaszewski A, Bachorz RA (2020) Efficient strategies of static features incorporation into the recurrent neural network. *Neural Process Lett* 51(3):2301–2316

17. Li D, Sun L, Xu X, Wang Z, Zhang J, Du W (2021) BLSTM and CNN stacking architecture for speech emotion recognition. *Neural Process Lett* 53(6):4097–4115
18. Zhao K, Shao H (2020) Intelligent fault diagnosis of rolling bearing using adaptive deep gated recurrent unit. *Neural Process Lett* 51(2):1165–1184
19. Liu L, Qiu Z, Li G, Wang Q, Ouyang W, Lin L (2019) Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Trans Intell Transp Syst* 20(10):3875–3887
20. Xu Z, Lv Z, Li J, Sun H, Sheng Z (2022) A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. *IEEE Intell Transp Syst Mag* 2–25
21. Maji D, Ghorai G (2019) A novel graph invariant: the third leap Zagreb index under several graph operations. *Discr Math, Algorithms Appl* 11(05):1950054
22. Zhu J, Han X, Deng H, Tao C, Zhao L, Tao L, Li H (2020) Kst-gen: a knowledge-driven spatial-temporal graph convolutional network for traffic forecasting. arXiv preprint [arXiv:2011.14992](https://arxiv.org/abs/2011.14992)
23. Li Z, Xiong G, Chen Y, Lv Y, Hu B, Zhu F, Wang, FY (2019) A hybrid deep learning approach with gcn and lstm for traffic flow prediction. In: 2019 IEEE Intelligent Transportation Systems Conference, pp 1929–1933
24. Guo S, Lin Y, Feng N, Song C, Wan H (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence 33(01): 922–929
25. Zhang H, Liu J, Tang Y, Xiong G (2020) Attention based graph convolution networks for intelligent traffic flow analysis. In: 2020 IEEE 16th International Conference on Automation Science and Engineering, pp 558–563
26. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint [arXiv:1709.04875](https://arxiv.org/abs/1709.04875)
27. Yao H, Tang X, Wei H, Zheng G, Li Z (2019) Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence 33(01): 5668–5675
28. Zhang J, Zheng Y, Sun J, Qi D (2019) Flow prediction in spatio-temporal networks based on multitask deep learning. *IEEE Trans Knowl Data Eng* 32(3):468–478
29. Guo G, Zhang T (2020) A residual spatio-temporal architecture for travel demand forecasting. *Transp Res Part C: Emerg Technol* 115:102639
30. Jin G, Cui Y, Zeng L, Tang H, Feng Y, Huang J (2020) Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. *Transp Res Part C: Emerg Technol* 117:102665
31. Ye J, Sun L, Du B, Fu Y, Xiong H (2020) Coupled layer-wise graph convolution for transportation demand prediction. arXiv preprint [arXiv:2012.08080](https://arxiv.org/abs/2012.08080)
32. Fu H, Wang Z, Yu Y, Meng X, Liu G (2021) Traffic flow driven spatio-temporal graph convolutional network for ride-hailing demand forecasting. In: PAKDD (1), pp 754–765
33. Fang S, Zhang Q, Meng G, Xiang S, Pan C (2019) GSTNet: global spatial-temporal network for traffic flow prediction. In: IJCAI, pp 2286–2293
34. Zhou X, Shen Y, Zhu Y, Huang L (2018) Predicting multi-step citywide passenger demands using attention-based neural networks. In: Proceedings of the Eleventh ACM international conference on web search and data mining, pp 736–744
35. Prada F, Kazhdan M, Chuang M, Collet A, Hoppe H (2016) Motion graphs for unstructured textured meshes. *ACM Trans Graph* 35(4):1–14
36. Chung JH, Kim DW, Kang TK, Lim MT (2020) Traffic sign recognition in harsh environment using attention based convolutional pooling neural network. *Neural Process Lett* 51(3):2551–2573
37. Koundinya S, Karmakan A (2021) Online speech enhancement by retraining of LSTM using SURE loss and policy iteration. *Neural Process Lett* 53(5):3237–3251
38. Delecras S, Becerra-Bonache L, Favre B, Nasr A, Bechet F (2020) Multimodal machine learning for natural language processing: disambiguating prepositional phrase attachments with images. *Neural Process Lett* 53(5):3095–3121
39. Ghorbani M, Li X, Zangi S, Amraei N (2021) On the eigenvalue and energy of extended adjacency matrix. *Appl Math Comput* 397:125939
40. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Li H et al (2019) T-gcn: a temporal graph convolutional network for traffic prediction. *IEEE Trans Intell Transp Syst* 21(9):3848–3858
41. Lee D, Jung S Cheon Y, Kim D, You S (2019) Demand forecasting from spatiotemporal data with graph networks and temporal-guided embedding. arXiv preprint [arXiv:1905.10709](https://arxiv.org/abs/1905.10709)
42. Chung JH, Kim DW, Kang TK, Lim MT (2020) Traffic sign recognition in harsh environment using attention based convolutional pooling neural network. *Neural Process Lett* 51(3):2551–2573
43. Kara O, Churamani N, Gunes H (2021) Towards fair affective robotics: continual learning for mitigating bias in facial expression and action unit recognition. arXiv preprint [arXiv:2103.09233](https://arxiv.org/abs/2103.09233)

44. Maji D, Ghorai G (2020) Computing F-index, coindex and Zagreb polynomials of the k th generalized transformation graphs. *Heliyon* 6(12):e05781
45. Maji D, Ghorai G (2021) The first entire Zagreb index of various corona products and their bounds. *J Math Comput Sci* 11(5):6018–6044
46. Ahmadizadeh S, Shames I, Martin S, Nešić D (2017) On eigenvalues of Laplacian matrix for a class of directed signed graphs. *Linear Algebra Appl* 523:281–306
47. Pereira M, Desassis N (2019) Efficient simulation of Gaussian Markov random fields by Chebyshev polynomial approximation. *Spat Stat* 31:100359
48. Wang JH, Lin GF, Chang MJ, Huang IH, Chen YR (2019) Real-time water-level forecasting using dilated causal convolutional neural networks. *Water Resour Manage* 33(11):3759–3780
49. Zhang X, You J (2020) A gated dilated causal convolution based encoder-decoder for network traffic forecasting. *IEEE Access* 8:6087–6097
50. Liu F, Gao M, Liu Y, Lei K (2019) Self-adaptive scaling for learnable residual structure. In: Proceedings of the 23rd conference on computational natural language learning, pp 862–870
51. Wang Y, Li J, Zhao A et al (2021) Temporal attention-based graph convolution network for taxi demand prediction in functional areas. In: International conference on wireless algorithms, systems, and applications, pp 203–214
52. Eckle K, Schmidt-Hieber J (2019) A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Netw* 110:232–242
53. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering 3844–3852. arXiv preprint [arXiv:1606.09375](https://arxiv.org/abs/1606.09375)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.