APADGCN: Adaptive Partial Attention Diffusion Graph Convolutional Network for Traffic Flow Forecasting

Bowen Zhang[1], Bohan Li[1,2,3](✉), Jinzhan Wei[4], and Hao Wen[5]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
bhli@nuaa.edu.cn

[2] Key Laboratory of Safety-Critical Software, Ministry of Industry and Information Technology, Beijing, China

[3] National Engineering Laboratory for Integrated Aero-Space-Ground Ocean Big Data Application Technology, Xi'an, China

[4] School of Electronic Information and Automation, Guilin University of Aerospace Technology, Guilin 541010, China

[5] College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Abstract. Traffic flow forecasting is a core task of urban governance and plays a vital role in the development of ITS. Because of the complexity and uncertainty of traffic patterns, it is of great challenge to capture spatial-temporal correlations. Recent researches mainly focus on the pre-defined adjacency matrix based on prior knowledge as the basis of spatial-temporal correlation modeling, but the fixed graph structure cannot adequately describe the dependency between traffic sensors. To tackle this issue, a novel deep learning model framework is proposed in this paper: Adaptive Partial Attention Diffusion Graph Convolutional Network(APADGCN), which consists of three main parts: 1) the Multi-Component module that divides the historical traffic flow into recent, daily-periodic, and weekly-periodic, to capture the traffic patterns of different periodic; 2) the spatial correlation modeling which can dynamically capture node relationships and model spatial dependency, and enhance the aggregation ability of low-order information; 3) the temporal correlation modeling which models long-term time dependencies using convolution and gating. The final result is obtained by the weighted fusion of the results of the multi-components. We compared our APADGCN with various baseline models in the four real datasets from the Caltrans Performance Measurement System (PeMS). The experimental results show that the prediction accuracy of APADGCN is better than that of the baseline models.

Keywords: Traffic flow forecasting · Spatial-temporal · Graph convolution networks · Attention networks

X. Meng et al. (Eds.): SpatialDI 2023, LNCS 13887, pp. 3–20, 2023.
https://doi.org/10.1007/978-3-031-32910-4_1

1 Introduction

Traffic flow forecasting is a classic spatial-temporal task, which aims to simulate the road conditions of a certain traffic area for a period of time in the future. At present, the number of cars is growing rapidly. The growth of traffic flow has led to lots of problems, which makes the urban transportation system unbearable. Predicting the future traffic flow quickly and accurately for traffic control, road transportation, and public convenience means a lot. Figure 1 shows the spatial-temporal correlation of traffic flow. From the time dimension, the flow at different historical moments will affect the flow at other moments. Different observation points will also influence each other. A spatial association may occur even if the two nodes are far apart (this kind of spatial dependency is indicated by a dashed line). How to correlate and mine the information in traffic data needs careful consideration thoroughly. However, traffic flow is highly random and uncertain, also many other factors, such as unexpected events and weather, can affect traffic conditions, which makes it more challenging to forecast traffic flow.

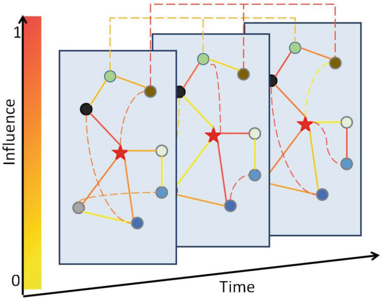

Fig. 1. The spatial-temporal correlation diagram of traffic flow.

Existing methods mainly utilized mathematical statistics, such as Kalman filter, fuzzy theory, and k-Nearest Neighbor(KNN) [1]. These algorithms achieved good results at first, but these models could not model complex traffic data nonlinearly and could not handle the spatial-temporal correlation, so most of these models relied on feature engineering. The growth of data volume and data types has increased the error of prediction results of traditional methods. In recent years, deep learning has gained attention for its ability to model high-dimensional nonlinearities for data, and it has good results in traffic flow forecasting. Recurrent Neural Network(RNN) [2] is the mainstream model for the ability to mine temporal features. However, these models are unable to extract features from the spatial attributes, leading to poor effects in traffic flow forecasting. Convolutional neural network(CNN) [3] is introduced for this. Historical traffic information is represented as a matrix, and the spatial topological links of the traffic data can be extracted by convolutional kernels. Therefore, to solve this problem, temporal correlation and spatial correlation need to be considered together. Combining CNN and RNN is a classical approach.

However, CNN is suitable for capturing spatial correlation in regular grids, which means that it is not applicable to realistic non-grid networks such as traffic networks. To address this problem, recently, spatial-temporal forecasting has been viewed as a problem of modeling, and data is usually regarded as a graph. Graph Convolutional Network (GCN) is used to discover spatial correlations in non-grid traffic networks due to its applicability to non-Euclidean spatial structures.

GCN can acquire and aggregate representations of neighbors in the vicinity of nodes, giving GCN the advantage of handling graph structures. However, there are many challenges. First, most models with GCN use the pre-defined adjacency matrix as the representation of nodes relationship, which can not truly represent the spatial relationship between nodes to a large extent. This situation is very common in traffic, where nodes are not only affected by their neighbors (such as traffic emergencies). The adjacency matrix cannot represent the dependency of nodes, which means some actual node associations are not represented in the adjacency matrix. Guo et al. [4] assigned weights to the adjacency matrix, which optimized the spatial correlation to a certain extent but still did not take into account the implicit dependency between nodes. Bai et al. [5] used points of interest(POI) to compute node similarity to represent the spatial association. Similarly, Geng et al. [6] encoded spatial associations using multigraph convolution. However, the pre-defined graph is still unable to represent node dependencies well. Because these methods rely on prior knowledge and are not available in other contexts. Wu et al. [7] embed dynamic learning of the spatial relationship between nodes and achieved good results. But relying only on the adaptive dependence matrix may ignore the attention of some inevitable node relationships. Second, when GCN aggregates information many times, the information of lower-order neighbors of nodes will be overwritten by higher-order nodes, resulting in inaccurate association. Our proposed APADGCN enhances the representation of nearby nodes in aggregation and reduces the loss of low-order neighbor information. Third, in the modeling of time correlation, previous studies mostly used recurrent neural networks such as GRU and LSTM to deal with time sequence relations. But there are some algorithm defects in the application, such as high model complexity, unstable gradients, difficulty in parallel, and so on.

To solve the above challenges, we propose Adaptive Partial Attention Spatial-Temporal Graph Convolutional Network(APADGCN). Different from previous studies, APADGCN can model the implicit spatial relationship between nodes dynamically. The problem of multiple convolution information loss is also taken into account, which enhances the aggregation ability of low-order neighbors. It also captures long-term time correlations and traffic patterns over different periods. The main contributions are as follows.

- We propose a new deep learning framework APADGCN for traffic flow prediction, which captures spatial-temporal correlation by stacking spatial-temporal layers.
- We design a new adaptive node relation matrix APA, which uses an adaptive matrix based on node embedding to capture the implicit association of nodes and propose partial attention mechanisms to enhance the aggregation ability of low-order information. Diffusion convolution is used to supplement the implied transition process and aggregate information with GCN.
- An improved gated temporal diffusion convolution is proposed, which uses diffusion convolution for long-term time dependency modeling, and incorporates the gated mechanism to control information transmission. The Multi-Component structure is used to model the traffic patterns of different periods.
- We compare our model with multiple baseline models in four real datasets, and the results of our proposed models are all better than the baseline models.

2 Related Works

2.1 Traffic Flow Forecasting

In previous studies, mathematical models were often used for traffic flow forecasting. For example, ARIMA is a classical model for forecasting [8]. Moreover, Chien et al. [9] used Kalman filtering algorithm to predict how long trips would take. Nikovski et al. [10] used Linear Regression (LR) to forecast travel time. Hou et al. [11] proposed a double-layer k-nearest neighbor (KNN) to predict the short-term traffic flow, and improved the efficiency of the model. However, these early prediction methods are mostly based on mathematics and statistics, which can not capture the intrinsic correlation between data only by relying on low-dimensional processing. This leads to the unsatisfied effect of these methods. Recently, deep learning has shown better modeling results in exploring spatial-temporal correlation [12].

Recently, deep learning has brought new solutions. Traffic flow can be modeled with long-term spatial and time dependencies. Different neural networks can be constructed to realize the learning of multidimensional representation of data. In terms of dealing with temporal correlations, early deep learning mainly uses RNN for temporal modeling. Long Short-Term Memory Neural Network (LSTM) is proposed to forecast traffic speed [13]. Cui et al. [14] proposed an SBU-LSTM framework with a data imputation mechanism, which achieved excellent prediction results for traffic data with different patterns of missing values. These methods based on RNN have some defects, such as complex parameters, low efficiency, and difficulty in parallelism. Many studies use Convolutional Neural Network(CNN) to deal with time series. Lea et al. [15] proposed a Temporal Convolutional Network(TCN) to mine spatial-temporal features in the framework of Encoder-Decoder on the temporal dimension. Liu et al. [16] proposed SCINet, which conducts sample convolution for recursive downsample to model time series effectively.

In terms of dealing with spatial correlations, CNN is often used for spatial modeling, Study [17,18] applied CNN to predict traffic speed. Wu et al. [19] construct the traffic flow prediction framework CLTFP with CNN and RNN. However, CNN is usually used for regular Euclidean graph, and the topological structure of many traffic networks is non-Euclidean. So, CNN does not apply to many traffic networks. The appearance of Graph convolutional networks (GCN) makes the study of non-Euclidean space further. GCN uses adjacency matrix to aggregate nearby nodes to achieve information dissemination. Li et al. [20] designed a diffusion graph convolution layer and completed the aggregation of information after K-hops.

2.2 Graph Neural Networks

Graph Neural Networks (GNN) was first proposed in [21], which is used to obtain topological information of non-Euclidean data. Subsequently, GCN emerged, which is one of the mainstream graph neural networks. At present, GCN is widely used because the traffic network can be represented by graph structure [22–26]. GCN is mainly divided into two categories: spectral graph convolution and spatial graph convolution [27]. In the field of spectral graph convolution, Bruna et al. [28] extended CNN to more common domains and proposed spectral convolution based on the graph Laplacian. ChebNet [29] used Chebyshev polynomials to expand and calculate the graph convolution, which avoided the calculation of the eigenvalues of the Laplacian matrix to optimize the high computational complexity of the original spectral convolution. In the field of spatial graph convolution, Micheli et al. [30] add the contextual data of the graph vertices through the traversal. The method is simple but has achieved good results. Graph Attention Network(GAT) is proposed in [31], which attached attention weight to the relationship between nodes and selectively aggregated the information of associated nodes. Wu et al. [7] proposed a Graph WaveNet that uses the node embedding algorithm to replace the pre-defined adjacency matrix with adaptive learning of the matrix, which improves the prediction accuracy of the model and training efficiency. Recently, A new adaptive matrix is proposed in [22], in which a Network Generator model is generated using the Gumbel-Softmax technique to explore the node associations.

2.3 Attention Mechanism

The attention mechanism was initially used in natural language processing to focus on the context of a word. It has since been used in many areas. At present, attention mechanism has been widely used in works, such as recommendation systems, computer vision, spatial-temporal prediction, video processing, and so on. Xu et al. [32] proposed a dual attention mechanism to classify image nodes. Liang et al. [33] added a multilevel attention network to the time series prediction, but due to a large number of parameters, the training takes a long time. In the field of graph data, there are also relevant studies to introduce the attention mechanism into the graph, through the construction of attention matrix, to achieve the function of dynamic correlation nodes. Guo et al. [4] proposed an ASTGCN, which uses the attention mechanism to dynamically compute the spatial attention weights between nodes. Zheng et al. [34] proposed a multi-attention neural network to model time steps of historical and future, based on an encoder-decoder framework. Xu et al. [35] proposed a STTN based on transformer, with the addition of spatial-temporal embedding, using attention mechanisms for time and space respectively. Jiang et al. [36] employed attention mechanism and convolution components to process long sequences.

3 Methodology

3.1 Preliminaries

In this study, we consider a traffic network as a graph $G = (V, E, A)$, where $V \in R^N$ is a set of nodes(e.g., traffic sensors) in the road network, and $E \in R^{N \times N}$ is a set of edges(e.g., the spatial connectivity between nodes); G is represented by adjacency matrix $A \in R^{N \times N}$, where $A_{i,j}$ represents the spatial connection of node i and node j, and $A_{i,j} = 1$ if $v_i, v_j \in V$ and $(v_i, v_j) \in E$.

Traffic flow forecasting can be regarded as a time series prediction task. Each node in graph G has F features at each time, and each node has the same sampling frequency. We donate $x_t^i \in R$ as the features of node i at time t. The characteristic data of all nodes at time t is expressed as $X_t = (x_t^1, x_t^2, ..., x_t^N)^T$. The historical observation traffic data are expressed as $H = (X_1, X_2, ..., X_T)$, which represents the data in T time steps of history. Our purpose is to predict the traffic flow data of the future T_{pre} time slices based on historical data H. Our task can be represented as:

$$X_{(T+1):(T+p)} = \mathcal{F}_\theta(H; G) \tag{1}$$

where $\mathcal{F}$ represents the transformation function, and θ represents all the learnable parameters in the training of the whole model.

3.2 Overview of Model Architecture

To effectively model the spatial and temporal traffic conditions, we propose a variant GCN model named APADGCN. Figure 2 depicts our proposed APADGCN, which consists of three modules: spatial Correlation module, temporal Correlation module, and Multi-Component Fusion module. We use the same association module for the recent period, daily period, and weekly period, and output the final prediction results by fusion. The spatial-temporal module is composed of APAGCN and GTCN. APAGCN is used to capture spatial correlation, and GTCN is used to explore temporal correlation.

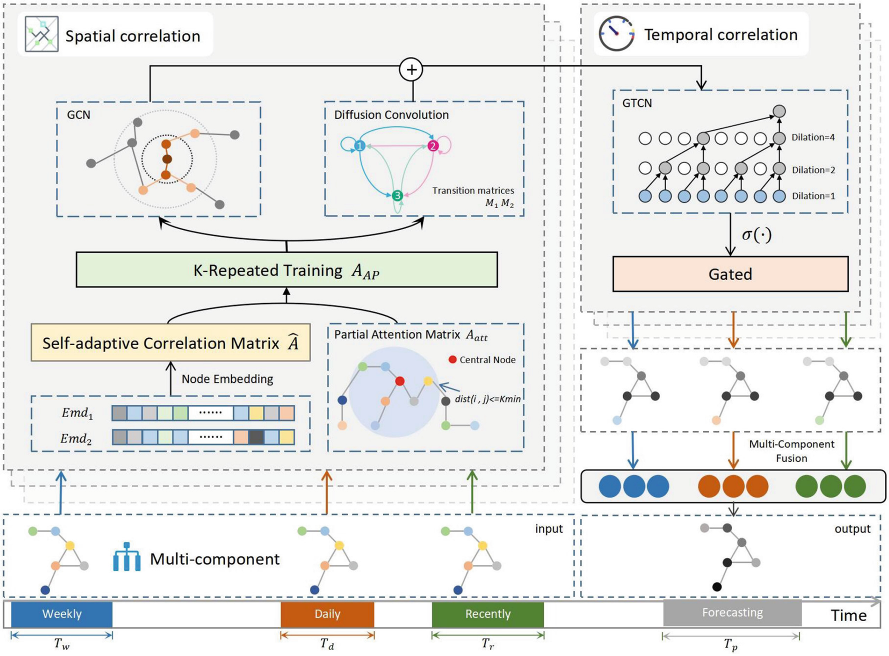

Fig. 2. Detailed framework of APADGCN.

3.3 Multi-component

Due to the strong historical periodicity of traffic flow, traffic flow often has similar patterns to the flow in history. Therefore, in this study, to explore the periodic patterns among the data, traffic data are divided into three time periods. Inspired by [4], we use T_{resent}, T_{day}, and T_{week} to denote the length of time in different period. Assume that the daily sampling frequency is T_q, the current time is $T_{current}$, and the prediction window size is T_{pre}. The detailed representation of the three periods is as follows:

(1) Recent Periodicity: This is the moment in history that is closest to and closely related in time to the forecast period. The traffic at this time has an important impact on the next time. This period of time is denoted as: $H_{recent} = (X_{T_0-T_{recent}+1}, X_{T_0-T_{recent}+2}, ..., X_{T_0}) \in R^{N\times F\times T_{recent}}$.

(2) Daily Periodicity: This period refers to the data at the same time one day before, and is a segment of the same time interval as the forecast period the day before. In a fixed road section, people usually have a certain daily life pattern, which means that traffic may show similar patterns. For example, in the morning and evening of weekdays, there will be morning and evening peaks, which is an obvious repeated pattern. But there are still many traffic characteristics and patterns that we cannot intuitively recognize. So we select the daily periodicity

to capture the daily hidden features. This period of time is denoted as:$H_{day} = (X_{T_0-T_q+1}, X_{T_0-T_q+2}, ..., X_{T_0-T_q+T_{pre}}) \in R^{N \times F \times T_{day}}$.

(3) Weekly Periodicity: This period is the same as the forecast period in the last few weeks. In general, traffic patterns are similar every week. For example, there are similar traffic conditions every Friday, but there are big differences in traffic patterns on weekends. Therefore, we expect to model and study the weekly traffic patterns through the weekly Periodicity module. This period of time is denoted as: $H_{week} = (X_{T_0-7*T_q+1}, X_{T_0-7*T_q+2}, ..., X_{T_0-7*T_q+T_{pre}}) \in R^{N \times F \times T_{week}}$.

In this study, these three time period modules are modeled with a learning network and enter the network for learning respectively. Finally, the three output results are merged through the fusion module to obtain the final prediction result.

3.4 Spatial Correlation Modeling

Partial Attention Self-adaptive Correlation Matrix. The core idea of graph convolution is to aggregate the information of nodes in the graph, by which information can be updated. The basic GCN representation is as follows:

$$X^{(h)} = Conv(X^{(h-1)}) = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X^{(h-1)} W^{(h)}) \tag{2}$$

where h represents the number of convolution executions, and the more h, the more information nodes aggregate. $X^{(0)} \in R^{N \times d}$ is the input feature matrix (i.e., the traffic signal data at time t_i), $\widetilde{D}$ is a diagonal matrix, $\widetilde{D}_{i,i} = \sum_j \widetilde{A}_{i,j}$. $\widetilde{A} = A + I_N \in R^{N \times N}$, where A is the adjacency matrix, I_N is the identity matrix. The matrix $W \in R^{N \times d}$ is a learnable parameter. Function $\sigma(\cdot)$ is the activation function (e.g., sigmoid or ReLU). $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$ is a normalized adjacency matrix, which is to aggregate the information of the adjacency nodes of a node. The significance of GCN for a node is to transform the features. The data of each node in the input data are F feature signals. The function of GCN is to aggregate information and increase the features of nodes to high dimensions and discover hidden spatial features.

Traditional GCN can aggregate node information through an adjacency matrix, but it is one-sided to judge node association. Aggregation based on spatial geographical adjacency cannot reflect the real association relationship between nodes. At present, there are many pre-defined methods for adjacency matrices, but these methods are intuitive and cannot represent the real spatial association between nodes, which will lead to deviation in the forecasting results of the model. Relying on the pre-defined adjacency matrix to represent the spatial correlation makes the pre-defined method only suitable for the specific environment, which leads to poor prediction results in other models.

To discover the real spatial correlation between nodes, we design an adaptive adjacency matrix module, which can autonomously and adaptively explore the dependency relationship of nodes from data without relying on prior knowledge. We use $Emd_1, Emd_2 \in R^{N \times Emd_C}$ to represent the node embedding dictionary and initialize them randomly. The adaptive matrix is formulated as follows [7]:

$$\widehat{A} = SoftMax(ReLu(Emd_1 Emd_2^T)) \tag{3}$$

where the function of $SoftMax$ is to normalize the embeddings. $ReLu$ is the activation function, which is used to eliminate the embedded weak connection between Emd_1 and Emd_2, which will skip the calculation of the Laplace matrix to speed up the training. In addition, the adaptive adjacency matrix is also used for the data of unknown graph structures, which can mine the potential connection relationship.

Partial Attention. The spatial variation of traffic flow has great correlations. We use an adaptive adjacency matrix to dynamically model the spatial correlation. But in pure using an adaptive adjacency matrix would produce some problems, in Eq. 2, h represents the number of convolution layers. When the value of h is large, it means that the central node has been aggregated many times. Although the data of remote points are aggregated, it will also lead to the loss of low-order information, that is, the information of neighboring nodes of the central node is overwritten. To ensure that the nodes can fully obtain the high-order information without losing the association of nearby nodes, we propose an adaptive adjacency matrix with a partial attention mechanism.

Inspired by the modeling of road network distance in Gaussian kernel [37]. To strengthen the ability of the model to associate information of nearby nodes after multiple aggregations, we propose a partial attention mechanism, which only imposes attention weights on nodes within a certain range of distance from central nodes. The formula is as follows:

$$A_{att} = \begin{cases} V_s \cdot \sigma((\chi^{h-1} W_1) W_2 (W_3 \chi^{h-1})^T + b_s), & dist(v_i, v_j) \leq k_{min} \\ 0, & otherwise \end{cases} \tag{4}$$

$$A^{'}_{atti,j} = SoftMax(A_{atti,j}) = \frac{exp(A_{atti,j})}{\sum_{j=0}^{Tr} exp(A_{atti,j})} \tag{5}$$

where $\chi^{h-1} = (X_1, X_2, X_3, ..., X_{T_r}) \in R^{N \times C \times T_r}$ is the input of h^{th} layer. $V_s, b_s \in R^{N \times N}$, $W_1 \in R^{T_r}$, $W_2 \in R^{C \times T_r}$, $W_3 \in R^C$ are the parameters to be learned. The matrix $A_{att} \in R^{N \times N}$ is the weight matrix of partial attention, $A_{atti,j}$ represents the associated value between nodes i and j, and the larger the value of $A_{atti,j}$ is, the stronger spatial connection between nodes i and j. We only apply attention weights to nearby nodes of the central node to strengthen the aggregation of information of nearby nodes. If attention weight is applied to all nodes, it will also lead to the loss of information of nearby nodes after multiple convolutions. It also speeds up the training process of the model by omitting many unnecessary modeling. Subsequently, we use $SoftMax$ function to normalize the attention matrix to ensure that the sum of the weights of relational nodes of node i is 1. The matrix $A^{'}_{att}$ is the normalized attention weight matrix.

After getting the partial attention matrix, we integrate it into the adaptive adjacency matrix. To ensure the stationarity of modeling learning, we use the average value of K training results after K repeated training as the final adjacency matrix. The formula of the adaptive adjacency matrix with partial attention mechanism is as follows:

$$A_{AP} = \frac{\lambda}{K} \sum_{i=0}^{K} \widehat{A}^{i} + (1-\lambda) A_{att}^{'} \tag{6}$$

where λ is a hyperparameter, which represents the fusion degree of the adjacency matrix with attention weight. When λ approaches 1, it means that the local attention matrix is not adopted. When λ approaches 0, it means that the local attention matrix is completely used as the node correlation matrix. A_{AP} is the partial attention adaptive adjacency matrix. The graph convolution formula with partial attention adaptive adjacency matrix is as follows:

$$X^{(h)} = Conv\_AP(X^{(h-1)}) = \sigma(\widetilde{D}^{-\frac{1}{2}} A_{AP} \widetilde{D}^{-\frac{1}{2}} X^{(h-1)} W^{(h)}) \tag{7}$$

Diffusion Convolution. The process of normalized adaptive adjacency matrix can be regarded as a transition matrix of a hidden diffusion process and can be used as a supplementary form of diffusion convolution [38]. Therefore, we introduce diffusion convolution and fuse the convolution layer with the diffusion convolution layer. The formula is as follows:

$$X^{(h)} = Conv\_AD(X^{(h-1)}) = \sigma(\widetilde{D}^{-\frac{1}{2}} A_{AW} \widetilde{D}^{-\frac{1}{2}} X^{(h-1)} W^{(h)} + Q_D) \tag{8}$$

$$Q_D = \sum_{k=0}^{R} (\theta_0 M_0^k X^{(h)} W_0 + \theta_1 M_1^k X^{(h)} W_1) \tag{9}$$

where $M_0^k = A / \sum_j A_{i,j}$ and $M_1^k = A^T / \sum_j A_{i,j_T}$ are the forward and backward transition matrices in the diffusion process, θ_0, θ_1, W_0, W_1 are the parameters matrices to learn. $M_0^2 = M_0 \cdot M_0$. The function of M_0^k and M_2^k is to represent the transition probability between nodes, and K is the number of diffusion steps. The diffusion process of convolution is simulated by the multiplication of the transition matrix. Matrix Q_D can also further enhance the ability to aggregate the information of nearby nodes to weaken the disadvantages caused by multi-layer convolution.

3.5 Temporal Correlation Modeling

GTCN. After the temporal attention layer, we have related traffic information at different moments. In this subsection, we will further merge the signals on the time slice. Recurrent neural networks, such as RNN and LSTM, have been widely used in temporal data, but there are some algorithm defects in the application.

Therefore, we follow [7] and use a dilated temporal convolutional mechanism to update the information.

$$Y^{(h)} = \begin{cases} X, & l = 0 \\ g(\theta_1 *_{d^l} Y^{(h-1)} + b) \bigodot \sigma(\theta_2 *_{d^l} Y^{(h-1)} + c), & l = 1, 2, 3, ..., l \end{cases} \tag{10}$$

where X is the input of DTCN, $Y^{(h-1)}$ is the input of l^{th} layer. θ_1, θ_2 are the convolution kernels. b and c are model parameters to be learned. $\bigodot$ is the element-wise product. $g(\cdot)$ and $\sigma(\cdot)$ are the activation function. $d^l = 2^l - 1$ is an exponential dilation rate. We use $\sigma(\cdot)$ to control how much information can be retained. We use dilated convolution to expand the receptive field on time series, which enhances the ability to model long-time series data.

3.6 Multi-component Fusion

In this section, we integrate the results of the three time periods to get the final forecasting results. For the period to be predicted, the three periods have different impacts on it. For example, the morning peak traffic patterns on weekdays are similar, so they are greatly influenced by daily and weekly periods, so we need to pay more attention to these two periods. However, if an emergency occurs, which leads to abnormal traffic conditions, it is necessary to pay more attention to the recent period. Therefore, combined with the attention mechanism, we attach different attention weights to the forecasting results of the three time periods to achieve the purpose of different attention to the period data. The final result after the fusion of features is:

$$\widehat{Y} = Linear(Concat(\widehat{Y}_{recent}, \widehat{Y}_{day}, \widehat{Y}_{week})) \tag{11}$$

where *Linear* is linear layer, *Concat* means concatenation operation. $\widehat{Y}_{recent}$, $\widehat{Y}_{day}$and $\widehat{Y}_{week}$ represent the results of the recent period, daily period, and weekly period, respectively.

4 Experiments

4.1 Datasets

To evaluate the effect of our proposed APADGCN model, we selected real highway datasets (PEMSD3, PEMSD4, PEMSD7, and PEMSD8) collected from California as experimental data. The dataset was produced by Caltrans Performance Measurement System(PeMS), which is real data on California highways and includes more than 39,000 physical sensors that integrate data every five minutes. The specific descriptions of datasets are shown in Table 1.

Table 1. Detailed information on datasets.

Dataset	PeMSD3	PeMSD4	PeMSD7	PeMSD8
Data type	Traffic flow	Traffic flow	Traffic flow	Traffic flow
Nodes(Sensors)	358	307	883	170
Edges	547	340	866	295
Time steps	26,208	16,992	28,224	17,856
Features	3	3	3	3
Data frequency	5 min	5 min	5 min	5 min

4.2 Settings

We use Z-Score normalization for the datasets we use to ensure that the inputs are of the same order of magnitude, and we divided datasets into the training set, validation set, and test set with the ratio of 6:2:2. Consecutive time slices are separated by 5 min, and a day is divided into 288 time slices. We set different data windows according to the selected period, namely T_r=24, T_d=12, T_w=24. For the three time periods, we predict the traffic flow for the next day, so the prediction window size is the same, that is, T_p=12. In APADGCN, we set the hidden dimension of graph convolution as 64, the repetition part K=6, λ=0.5. The threshold for partial attention $K_{min} = 0.12$, and the number of diffusion hops R=2. We superimposed three spatial association modules. Each TCN layer uses 64 convolution kernels. In this study, we use mean square error (MSE) as the loss function. In the stage of training, the batch size is 64 and the learning rate is 0.0001. We use the adamoptimizer and set the number of epochs to 100.

4.3 Baseline Methods

We used the following seven baselines to compare with our proposed APADGCN model.

VAR [39]: Vector Auto-Regression, which is a classical model for time series modeling.

ARIMA [40]: Autoregressive Integrated Moving Average model, which is one of the classical time series forecasting analysis methods.

LSTM [41]: Long Short Term Memory network, which is based on RNN to model timing relationships.

FC-LSTM [42]: FullConnection-LSTM, which combines the fully connected layer and LSTM layer to predict traffic flow.

TCN [43]: Temporal Convolution Network, which uses convolution kernel to aggregate information of time dimension for prediction.

STGCN [44]: Spatio-Temporal Graph Convolution Network, which combines GCN and convolution to model spatial-temporal dependency.

DCRNN [20]: Diffusion Convolutional Recurrent Neural Network, which introduces dilated convolutional to capture spatial-temporal correlation.

GraphWaveNet [7]: Graph WaveNet, which combines adaptive convolution and dilated convolution layers using a node embedding algorithm.

4.4 Comparison and Result Analysis

Results on the PEMS Dataset. In Table 2, we compare our proposed model with the baseline on the four PEMS datasets using MAE, MAPE, and RMSE metrics. It can be seen that our APADGCN has achieved the best results in the four indicators. This shows that our model can capture the spatial and temporal dependence of traffic flow data well. In addition, we can observe that compared with other models, ARIMA and LSTM show larger prediction errors, because ARIMA and LSTM only take the temporal correlation of nodes into account and ignore the spatial correlation. Although VAR considers the spatial correlation, it is not able to capture the hidden information, so it also has a poor effect. Other models using deep learning consider the spatial-temporal correlation, thus the results of forecasting are far better than the previous two.

TCN, STGCN, DCRNN, and GraphWaveNet achieved good results on the four datasets, but the prediction accuracy was not as good as our proposed model. These baseline models use GCN to model spatial association, in which TCN and STGCN only take the network connection relationship on the real map as the adjacency matrix, and cannot associate the possible spatial relationship between nodes. Although DCRNN and GraphWaveNet use extended convolution and adaptive adjacency matrix to expand spatial correlation, their temporal correlation processing method cannot model long-term temporal dependence. Our proposed APADGCN can capture the implicit spatial correlation and the long-term temporal information. Therefore, APADGCN can better model the spatial-temporal dependency.

Table 2. Prediction Errors on the PeMSD3, PeMSD4, PeMSD7 and PeMSD8 Datasets.

Dataset	Metric	VAR	ARIMA	LSTM	FC-LSTM	TCN	STGCN	DCRNN	WaveNet	**Ours**
PEMSD3	MAE	23.65	33.51	20.62	21.33	19.32	17.49	17.99	19.85	**16.85**
	MAPE(%)	24.51	33.78	33.54	23.33	19.93	17.15	18.34	19.31	**16.57**
	RMSE	38.56	47.59	28.94	35.11	33.55	30.12	30.31	32.94	**29.34**
PEMSD4	MAE	23.75	33.73	26.81	27.14	23.22	22.70	21.22	25.45	**20.43**
	MAPE(%)	18.09	24.18	18.74	18.20	15.59	14.59	14.17	17.29	**13.46**
	RMSE	36.66	48.80	43.49	41.59	37.26	35.55	33.44	39.70	**32.86**
PEMSD7	MAE	75.63	38.17	29.71	29.98	32.72	25.38	25.22	26.85	**24.57**
	MAPE(%)	32.22	19.46	45.32	13.20	14.26	11.08	11.82	12.12	**10.79**
	RMSE	115.24	59.27	14.14	45.94	42.23	38.78	38.61	42.78	**37.91**
PEMSD8	MAE	23.46	31.09	22.19	22.20	22.72	18.02	16.82	19.13	**16.26**
	MAPE(%)	15.42	22.73	33.59	14.20	14.03	11.40	10.92	12.68	**10.48**
	RMSE	36.33	44.32	18.74	34.06	35.79	27.83	26.36	31.05	**25.71**

Ablation Experiment. To verify the validity of each component in our proposed model, we proposed the following variants of APADGCN which removed several modules: (1)RemSA: It removes the Self-adaptive Correlation Matrix in the APADGCN. (2)RemPA: It removes Partial Attention in the APADGCN. (3)RemAPD: It removes the Self-adaptive Correlation Matrix and Partial Attention and replaces them with a normal adjacency matrix. (4)RemDC: It removes Diffusion Convolution and replaces it with normal GCN. We compare these four variants with our proposed APDGCN model on PEMS04. We used MAE, MAPE, and RMSE as metrics. In Fig. 3, the comparison results of the models are shown in detail.

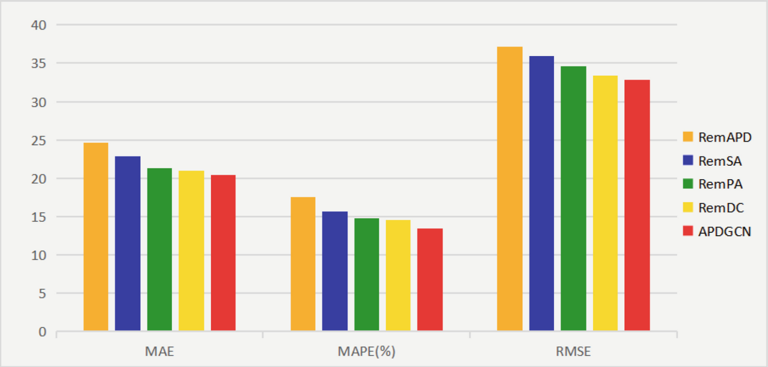

Fig. 3. Details of Ablation experiment.

Figure 3 shows the prediction accuracy of each model. It can be seen that the accuracy of the four variant models is lower than APADGCN. RemAPD has the worst prediction effect. It can be found that our proposed SEPA module has the function of dynamically capturing node association, which indicates the importance of node spatial correlation. By comparing RemPA and RemSA, it can be found that the performance of PA is inferior to SA, which means that the adaptive correlation matrix is better than the attention mechanism in capturing spatial correlation. The performance of RemDC is worse than APADGCN, which indicates that the transition matrix in diffusion convolution can enhance the function of capturing the spatial relationship of nodes.

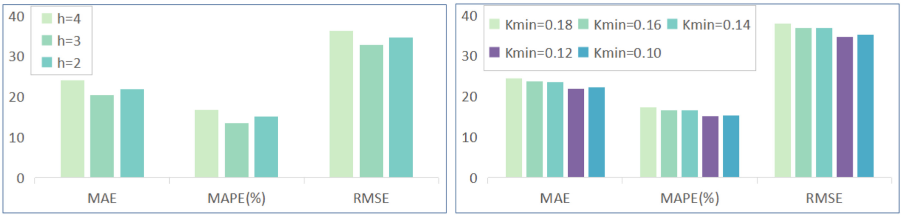

Fig. 4. Network configuration analysis. In these two images, we have different configurations for the hyperparameter. Where h is the number of spatial convolution layers, K_{min} is the distance threshold of partial attention.

Effect of Different Network Configurations. To explore the influence of hyperparameters in the model on the prediction results, we conducted experiments on the networks with different hyperparameters. All parameters are the same as those in 4.2. Only the parameters for comparison are adjusted. Figure 4 shows the experimental results for different configurations of the hyperparameter. It can be seen that (1) When the convolution layer is expanded from two layers to three layers, no information loss is caused because we set part of the attention mechanism, so more node information is aggregated to achieve the best effect; (2) The expansion of the distance threshold of some attention increases the number of nodes aggregated, but decreases the effect. When the threshold is close to 1, it is the global attention mechanism and cannot enhance the near-point representation.

5 Conclusion

In this paper, we propose a novel traffic flow forecasting model APADGCN based on deep learning. We use a node embedding algorithm and partial attention mechanism to build an adaptive node association matrix and combine graph convolution and diffusion convolution to aggregate node information to capture spatial association. This approach can represent the node association without the pre-defined adjacency matrix and enhance the representation of the hidden dependency of nodes and the attention to nearby nodes. We introduce Multi-component to model traffic patterns in the different periods. Therefore, our model can better capture the spatial-temporal correlation of traffic flow. We conduct sufficient comparisons with some baseline models on four public datasets, and the results show that our proposed APADGCN is superior to the baseline model and has good performance. In the future, we will consider adding information such as weather to assist traffic flow forecasting and enhance the versatility of the model in different scenarios.

References

1. Pang, X., Wang, C., Huang, G.: A short-term traffic flow forecasting method based on a three-layer k-nearest neighbor non-parametric regression algorithm. J. Transp. Technol. **6**(4), 200–206 (2016)
2. Laptev, N., Yosinski, J., Li, L.E., Smyl, S.: Time-series extreme event forecasting with neural networks at uber. In: International Conference on Machine Learning, vol. 34, pp. 1–5 (2017)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_13
4. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proceed. AAAI Conf. Artif. Intell. **33**(01), 922–929 (2019)
5. Bai, L., Yao, L., Kanhere, S.S., Yang, Z., Chu, J., Wang, X.: Passenger demand forecasting with multi-task convolutional recurrent neural networks. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11440, pp. 29–42. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16145-3_3
6. Geng, X., et al.: Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. Proceed. AAAI Conf. Artif. Intell. **33**(01), 3656–3663 (2019)
7. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph waveNet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121 (2019)
8. Ahmed, M.S., Cook, A.R.: Analysis of freeway traffic time-series data by using box-Jenkins techniques (1997)
9. Chien, S.I.-J., Kuchipudi, C.M.: Dynamic travel time prediction with real-time and historic data. J. Transp. Eng. **129**(6), 608–616 (2003)
10. Nikovski, D., Nishiuma, N., Goto, Y., Kumazawa, H.: Univariate short-term prediction of road travel times. In Proceedings.: IEEE Intelligent Transportation Systems, vol. 2005, pp. 1074–1079 (2005). IEEE (2005)
11. Xiaoyu, H., Yisheng, W., Siyu, H.: Short-term traffic flow forecasting based on two-tier k-nearest neighbor algorithm. Procedia. Soc. Behav. Sci. **96**, 2529–2536 (2013)
12. Li, Z., Ren, Q., Chen, L., Sui, X., Li, J.: Multi-hierarchical spatial-temporal graph convolutional networks for traffic flow forecasting. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 4913–4919. IEEE (2022)
13. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C: Emerg. Technol. **54**, 187–197 (2015)

14. Cui, Z., Ke, R., Pu, Z., Wang, Y.: Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. Transp. Res. Part C: Emerg. Technol. **118**, 102674 (2020)
15. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 156–165 (2017)
16. Liu, M., Zeng, A., Xu, Z., Lai, Q., Xu, Q.: Time series is a special sequence: forecasting with sample convolution and interaction. arXiv preprint arXiv:2106.09305 (2021)
17. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)
18. Zhang, J., Zheng, Y., Qi, D.: Deep Spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-first AAAI Conference on Artificial Intelligence (2017)
19. Wu, Y., Tan, H.: Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022 (2016)
20. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
21. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, vol. 2, no. 2005, pp. 729–734 (2005)
22. Kong, X., Zhang, J., Wei, X., Xing, W., Lu, W.: Adaptive spatial-temporal graph attention networks for traffic flow forecasting. Appl. Intell. **52**(4), 4300–4316 (2022)
23. Zhang, C., et al.: Augmented multi-component recurrent graph convolutional network for traffic flow forecasting. ISPRS Int. J. Geo Inf. **11**(2), 88 (2022)
24. Wang, Y., Jing, C., Xu, S., Guo, T.: Attention based spatiotemporal graph attention networks for traffic flow forecasting. Inf. Sci. **607**, 869–883 (2022)
25. Zhang, W., Zhu, K., Zhang, S., Chen, Q., Xu, J.: Dynamic graph convolutional networks based on spatiotemporal data embedding for traffic flow forecasting. Knowl.-Based Syst. **250**, 109028 (2022)
26. Zhang, S., Guo, Y., Zhao, P., Zheng, C., Chen, X.: A graph-based temporal attention framework for multi-sensor traffic flow forecasting. IEEE Trans. Intell. Transp. Syst. **23**(7), 7743–7758 (2021)
27. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
28. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
29. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
30. Micheli, A.: Neural network for graphs: a contextual constructive approach. IEEE Trans. Neural Networks **20**(3), 498–511 (2009)
31. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
32. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057. PMLR (2015)
33. Liang, Y., Ke, S., Zhang, J., Yi, X., Zheng, Y.: GeoMAN: multi-level attention networks for geo-sensory time series prediction. IJCAI **2018**, 3428–3434 (2018)

34. Zheng, C., Fan, X., Wang, C., Qi, J.: GMAN: a graph multi-attention network for traffic prediction. Proceed. AAAI Conf. Artif. Intell. **34**(01), 1234–1241 (2020)
35. Xu, M., et al.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)
36. Jiang, S., Zhu, M., Li, J.: Traffic flow forecasting using a spatial-temporal attention graph convolutional network predictor. In: Meng, X., Xie, X., Yue, Y., Ding, Z. (eds.) SpatialDI 2020. LNCS, vol. 12567, pp. 107–121. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69873-7_8
37. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. **30**(3), 83–98 (2013)
38. Qi, J., Zhao, Z., Tanin, E., Cui, T., Nassir, N., Sarvi, M.: A graph and attentive multi-path convolutional network for traffic prediction. IEEE Transactions on Knowledge and Data Engineering (2022)
39. Hamilton, J.D.: Time series analysis. Princeton University Press (2020)
40. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal Arima process: theoretical basis and empirical results. J. Transp. Eng. **129**(6), 664–672 (2003)
41. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
42. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
43. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
44. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)