sensors

MDPI

Article

ADSTGCN: A Dynamic Adaptive Deeper Spatio-Temporal Graph Convolutional Network for Multi-Step Traffic Forecasting

Zhengyan Cui [1], Junjun Zhang [1], Giseop Noh [2] and Hyun Jun Park [2,*]

[1] Department of Computer Information Engineering, Cheongju University, Cheongju 28503, Republic of Korea; cuizy1017@gmail.com (Z.C.); zjj416320@gmail.com (J.Z.)
[2] Department of Artificial Intelligence Software, Cheongju University, Cheongju 28503, Republic of Korea; kafa46@cju.ac.kr
* Correspondence: hyunjun@cju.ac.kr

Abstract: Multi-step traffic forecasting has always been extremely challenging due to constantly changing traffic conditions. Advanced Graph Convolutional Networks (GCNs) are widely used to extract spatial information from traffic networks. Existing GCNs for traffic forecasting are usually shallow networks that only aggregate two- or three-order node neighbor information. Because of aggregating deeper neighborhood information, an over-smoothing phenomenon occurs, thus leading to the degradation of model forecast performance. In addition, most existing traffic forecasting graph networks are based on fixed nodes and therefore need more flexibility. Based on the current problem, we propose Dynamic Adaptive Deeper Spatio-Temporal Graph Convolutional Networks (ADSTGCN), a new traffic forecasting model. The model addresses over-smoothing due to network deepening by using dynamic hidden layer connections and adaptively adjusting the hidden layer weights to reduce model degradation. Furthermore, the model can adaptively learn the spatial dependencies in the traffic graph by building the parameter-sharing adaptive matrix, and it can also adaptively adjust the network structure to discover the unknown dynamic changes in the traffic network. We evaluated ADSTGCN using real-world traffic data from the highway and urban road networks, and it shows good performance.

Keywords: traffic forecasting; spatio-temporal graph; deep graph convolutional network; adaptive graph construction

Citation: Cui, Z.; Zhang, J.; Noh, G.; Park, H.J. ADSTGCN: A Dynamic Adaptive Deeper Spatio-Temporal Graph Convolutional Network for Multi-Step Traffic Forecasting. *Sensors* **2023**, *23*, 6950. https://doi.org/10.3390/s23156950

Academic Editors: Kai Liu, Xinxiang Zhang, Ye Wang and Feng Guo

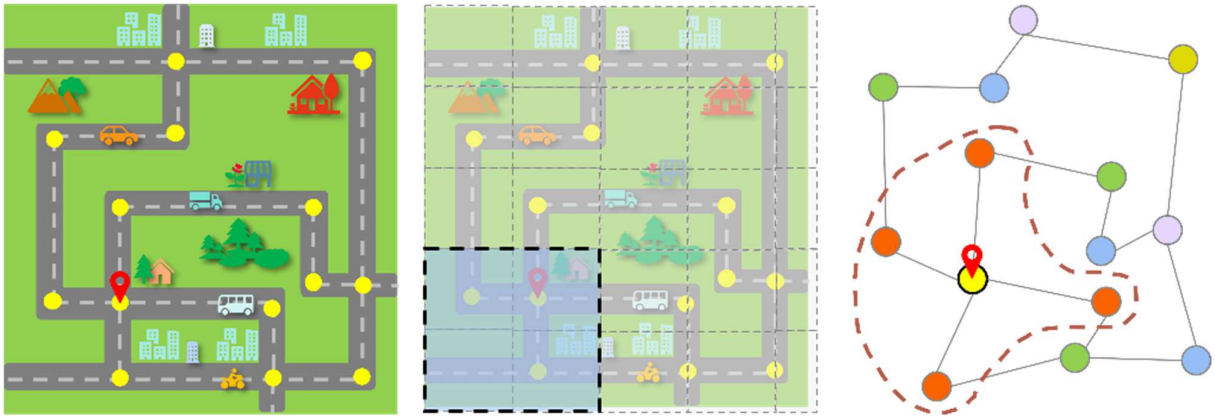
Received: 23 June 2023
Revised: 1 August 2023
Accepted: 3 August 2023
Published: 4 August 2023

1. Introduction

The Intelligent Transportation System (ITS) plays an essential role in urban construction. Reliable and accurate real-time traffic forecasting can help people rationalize travel and ease traffic congestion [1,2]. The development of deep learning has enabled the application of several deep-learning-based forecast models in traffic and transport fields [3,4]. However, traffic conditions have complex, irregular, and nonlinear spatial and temporal relationships [5,6]. The urban road network is complex, irregular, and topological and is challenging to manage conventionally. Graph Convolutional Networks (GCNs) excel in managing non-linear and irregular data, causing them to be extensively applied in traffic forecasting [7,8], as shown in Figure 1. How to construct and optimize graph networks using GCNs to improve traffic forecasting and alleviate traffic congestion is the main problem we address.

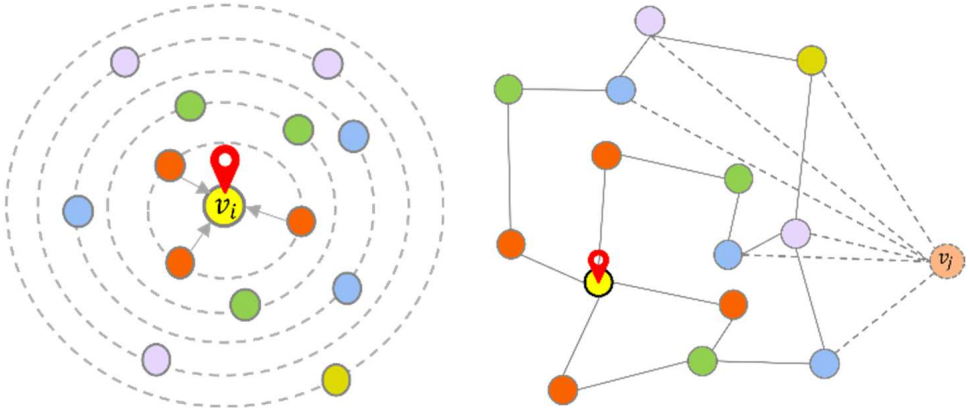
The combination of graph convolution and the Gated Recurrent Unit is the first to have improved traffic forecasting [7]. Initially, a purely convolutional approach using graph convolution and 1D Convolution Neural Networks (CNN) was explored in the field of traffic forecasting [8]. They have shown better results in traffic forecasting. However, they are usually shallow networks that aggregate only two- or three-order node neighbor information [7–11]. Deeper models tend to have superior nonlinear expression abilities and

extract deeper features [12]. The multi-order neighborhood in the traffic graph is shown in Figure 2a. As the network deepens, the adjacent nodes in the graph structure become increasingly similar, creating an over-smoothing problem [13,14]. This leads to a decrease in forecasting performance. In traffic forecasting studies, skip connections [11,15,16] and GRU architectures [7,17] are used to deepen the overall spatio-temporal model level, but GCN is still a shallow network. Divergent from previous studies, to extract deeper and richer spatial relations in the traffic and increase the node receptive field in the traffic graph, we deepen the neighborhood propagation of the graph network and to mitigate the problem of over-smoothing, and we seek to enhance the connectivity between hidden layers.



(**a**) (**b**) (**c**)

Figure 1. (**a**) Urban road network. (**b**) Description of the regular grid structure of urban road network. (**c**) Description of the irregular graph structure of the urban road network, and different colors represent different neighborhood relationships.



(**a**) (**b**)

Figure 2. (**a**) The multi-order neighborhood of node v_i in the traffic graph. (**b**) For v_j nodes that change or are newly added to the graph, the model can adaptively adjust the graph structure and learn its relationship with the surrounding nodes.

The graph construction relies more on the node adjacency matrix. In the traffic forecasting graph, the creation of the adjacency matrix is commonly accomplished by considering the distance, connectivity, or similarity among nodes [7–10]. These fixed pattern-based graph structures are not the best at discovering unknown hidden spatial relationships between nodes. There are also models that use an adaptive matrix to increase the flexibility of the graph [11,18]. However, they create random matrices that adaptively learn node relationships from the perspective of the feature space, ignoring the composite spatial association information with neighbors and similarity. Different from their work,

we propose a parameter-sharing adaptive graph convolution method for traffic forecasting, considering the composite space with near neighbors and similarities and the random feature space in the traffic network. The method discovers unknown dynamic changes in the network by establishing the parameter-sharing adaptive matrix. It can adaptively learn and adjust the spatial dependencies and structures within the traffic according to the changes, as shown in Figure 2b. The main innovative work of this paper is as follows:

1. To address the over-smoothing problem arising from deepening the network layers in multi-step traffic forecasting with Graph Convolutional Networks, we employ a technique of dynamically adjusting hidden layer connections and adaptively modifying the hidden layer weights to prevent model degradation.
2. We propose a parameter-sharing adaptive graph convolution method for multi-step traffic forecasting, which considers the ever-changing complex spatio-temporal relationships within the traffic network. This is able to adaptively learn and adjust the spatial dependencies and structures within the traffic network by building the adaptive matrix for parameter sharing.
3. We propose Dynamic Adaptive Deeper Spatio-Temporal Graph Convolutional Networks (ADSTGCN), a new traffic forecasting model. It uses the diffusion graph convolutional network to obtain spatial dependencies in traffic and the temporal convolutional network to obtain temporal dependencies for better traffic forecasting.
4. We validate our model on two traffic datasets and show better traffic forecasting results than existing advanced baselines.

2. Related Work

Multi-step traffic forecasting involves predicting the traffic conditions at various future time intervals from the spatial and temporal dimensions according to the historical traffic conditions in the traffic road network. Its research focuses on the spatio-temporal correlation between the traffic network structure and traffic time series [19]. Recently, deep learning network models have performed outstandingly in traffic forecasting, and their performance is much better than traditional machine learning models [20–22]. The spatio-temporal dependence of historical traffic data obtained from sensors can be extracted from the two different dimensions of spatial and temporal, respectively, by using a neural network model.

Extraction of spatial dependencies. GCNs [23–25] designed for non-Euclidean data have attracted significant interest in the area of traffic forecasting. According to statistics, most traffic forecasting models since 2019 have used GCNs to model spatial relationships, demonstrating that GCNs research is cutting-edge [1]. The GCNs that are currently available are commonly classified into two categories: spectral domain and spatial domain graph convolution [24–26]. The spectral domain graph convolution uses Fourier transform for convolution operations [23]. However, it is very time-consuming to compute the eigenvalue decomposition of the Laplacian matrix, and the model has sizeable parametric complexity. ChebNet utilizes Chebyshev polynomials in the spectral domain as a substitute for the convolution kernel, aiming to decrease the model's complexity [27]. The GCN simplifies ChebNet by only considering one-order Chebyshev polynomials and only has one parameter per convolution kernel, lowering the model's complexity [23]. The traffic graph's spatial representation is extracted using the one-order approximate graph convolution of the Laplacian matrix, which circumvents the spatial neglect issue encountered in recurrent neural networks [8]. Compared with the complex operation of spectral domain graph convolution, spatial graph convolution operates directly on neighborhood nodes, which is more intuitive and flexible. To obtain heterogeneity in the spatial data, after constructing a local spatio-temporal graph, the spatial representation is extracted through spatial graph convolution [3]. Different from the base spatial graph convolution, which only does a linear transformation of its input feature, the diffusion graph convolution takes the aggregation operation on the input feature of its neighbors. A self-attention-mechanism-based information fusion module utilizes diffusion graph convolution to model and comprehend the

traffic change relationships of various regions, leveraging the global spatial scope of the entire city [4]. It is also used to model the fusion features extracted from road network graphs and regional graphs [11]. Incorporating diffusion graph convolution, modeling the spatio-temporal dependence between main and auxiliary features is achievable through two segmented spatio-temporal modules [16]. Graph Attention Networks consider the importance of different neighbors and employ the attention mechanism to integrate the information from embedded neighboring nodes. It is used to extract the channel, temporal, and spatial embedding relationships between nodes in the traffic graph [28]. K-hop graph convolution obtains spatial dependences on adjacency matrices constructed using road network connections and competing influence relationships [5]. They have shown better results in traffic forecasting. However, they are usually shallow networks that aggregate only two- or three-order node neighbor information. Skip connections [11,15,16] and GRU architectures [7,17] are used in traffic forecasting studies to deepen the overall spatio-temporal model level also achieve better results, but they are also shallow graph networks. Based on these studies, we focus on intensifying the model hierarchy and deepening the graph network to enlarge the receptive fields of graph nodes, thereby capturing deeper and more intricate spatial relationships within the traffic road network.

The graph structure adjacency matrix determines the GCN performance such that it is one of the main research focuses. Existing studies have generally used the distance between nodes [7,11,29], or the similarity between nodes [9,10,16], to construct adjacency matrices. Other studies have utilized external factors such as POI (Point of Interest) to enhance features along with fusion based on local and global adjacency matrices [3,19,30]. However, these models are based on fixed graph structures and lack the flexibility to capture dynamically changing traffic conditions and road network structures. Some other models use adaptive matrices to increase graph flexibility, learning node feature similarity relationships through two random nodes [11,18]. However, they adaptively learn feature spatial relationships from the feature perspective, and do not adaptively learn the association information from the graph spatial adjacency structure at the same time. On the basis of their work, we adaptively learn and adjust the spatial dependencies and structures within the traffic network by building the adaptive matrix for parameter sharing from the feature and spatial perspectives.

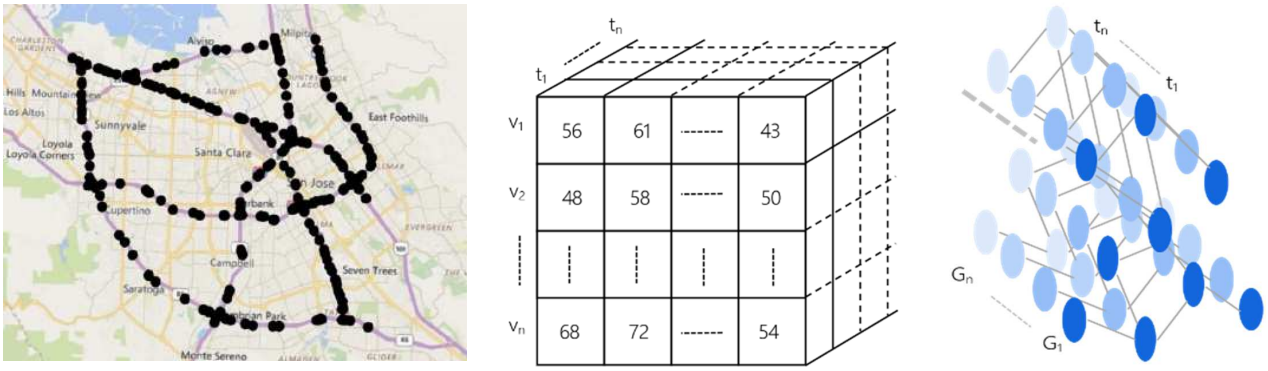
Extraction of temporal dependencies. Traffic forecasting extensively employs recurrent neural networks (RNNs) because of their capacity to memorize and learn both short- and long-term temporal dependencies in sequences [5,7,22,31–33]. However, if the dataset is large, the computational load of gating in the RNNs will be large. During rush hour, capturing fluctuations in large traffic volumes is challenging because RNN calculations often rely on the previous step [8]. In certain research works, Convolutional Neural Networks (CNNs) are employed to capture temporal dependencies in traffic forecasting [8,9,17,20,34,35]. However, CNNs perform convolution through input in a window before and after time t, which leads to information leakage after time t. When the historical sequence is long, CNNs need to increase the convolution size to view additional historical information, leading to less efficient training. Thus, Temporal Convolution Networks (TCNs) [36] which combine dilated and causal convolution, have attracted widespread interest in the field of traffic forecasting. TCNs are simple and effective in processing time series data and cannot see future data. Furthermore, TCNs use dilated convolution to obtain a long receptive field with fewer layers, which is beneficial in capturing long-term periodic dependencies. Experimental TCN results have demonstrated that it outperforms RNN in terms of both accuracy and computational time [36]. Temporal dependence at different temporal levels can be obtained by increasing the model temporal receptive field by stacking 1D and 2D causal dilated TCN [10,11,16,29]. In this paper, we use TCN to extract the time dependence of traffic forecasting.

3. Methodology

3.1. Problem Definition

The primary purpose of multi-step traffic forecasting is to anticipate the traffic conditions for multiple future time steps in the traffic road network, relying on historical traffic data.

Definition 1. *Graph G: In this study, the traffic topology is represented by graph* $G(V, E)$*, as shown in Figure 3. The graph's node set is represented as*$V = \{v_1, v_2, \ldots\ldots v_n\}$*. Then, any node i can be represented as* v_i*.* $E = \{e_1, e_2, \ldots\ldots e_n\}$*represents the set of connection relationships between all nodes in the graphs.*



(**a**) (**b**) (**c**)

Figure 3. (**a**) Urban highway network. (**b**) The traffic speed of each sensor in the time series. (**c**) Traffic spatio-temporal sequence graph.

Definition 2. *Traffic feature matrix X: The traffic conditions of each traffic forecasting sensor are the feature of each node in the graph. In this paper, we mainly study traffic speed as shown in Figure 3b. The traffic speed monitored by all sensors within the road network can be represented by the feature matrix X,* where $X \in \mathbb{R}^{T \times N}$*. The time step is represented by T, and the number of nodes is represented by N. Then, for any node i* in $G(V, E)$*, its eigenvalue can be expressed as* x_i*.*

Definition 3. *Adjacency matrix A: The connectivity among all sensors in the traffic network can be depicted by matrix A, commonly referred to as the adjacency matrix,* $A \in \mathbb{R}^{N \times N}$*. In our work, the connectivity of edges in the graph is represented using the distance and similarity between nodes [37].*

Definition 4. *Multi-step traffic forecasting: We slice the time axis into steps every 5 min, denoted by t, and the total step is denoted by T. In this paper, our objective is to learn a mapping function f, which can effectively transform the traffic conditions X observed over P time steps in the historical data to the predicted traffic conditions* $\hat{Y}$ *over Q future time steps. For any node i, we can define* $\hat{Y}_i$ *as:*

$$\hat{Y}_i = f\left(G; x_i^{P+1}, x_i^{P+2}, \ldots\ldots, x_i^{P+Q}\right) \tag{1}$$

where P is the historical time step and Q is the predicted time step, as shown in Figure 4.

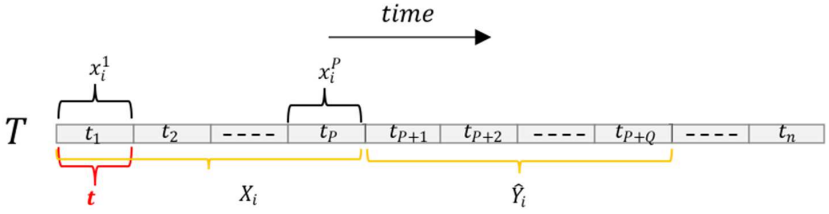


Figure 4. Time series in multi-step traffic forecasting.

3.2. Overall Architecture

Figure 5 shows the overall architecture of the ADSTGCN. The model uses the multi-head attention mechanism [38] to perform multi-strategy fusion transformation on the spatio-temporal dependencies obtained through spatio-temporal convolution and spatio-temporal embedding, respectively. Finally, the forecast results are output after the activation function transformation. In the convolution strategy, TCN convolves the input traffic feature X to obtain the time dependence. Adaptive deep Graph Convolutional Networks obtain spatial dependencies through composite adjacency matrices with distance and similarity relationships. Multiple spatio-temporal layers of the ADSTCN with residual connections [39] are subsequently linked to form the input for the multi-head attention mechanism. In order to further strengthen the spatio-temporal relationship, we integrate the traffic network structure and feature data into E_{st} by embedding and encoding, respectively.

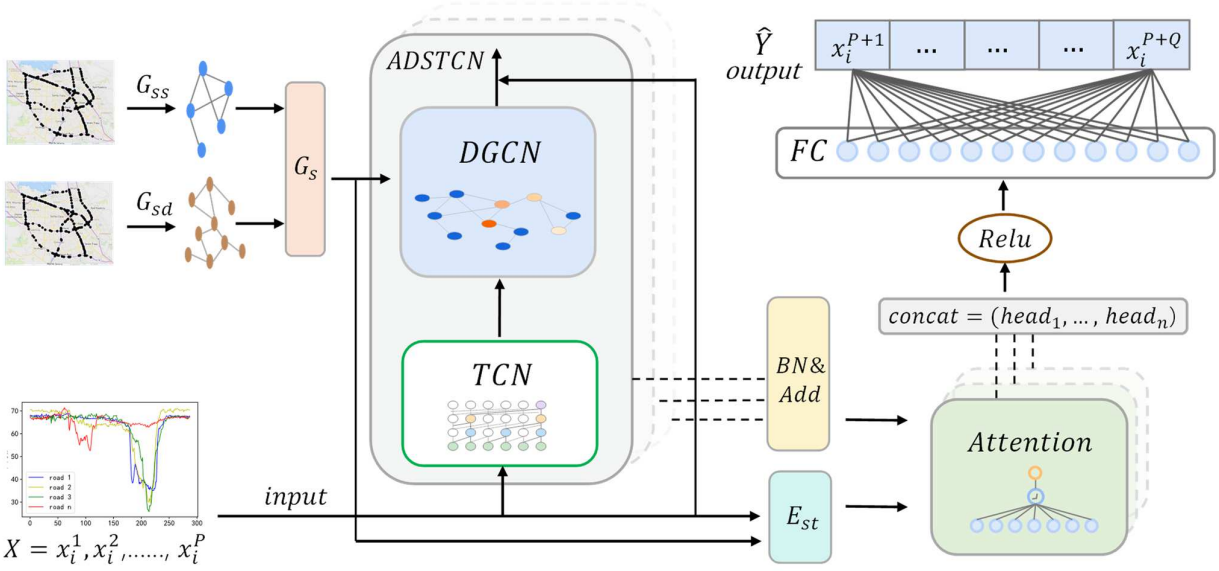


Figure 5. The complete structure of the Dynamic Adaptive Deeper Spatio-Temporal Graph Convolutional Network (ADSTGCN).

3.3. Input Data Processing

Using distances between sensors to create graph adjacencies tends to ignore richer spatial relationships. This paper uses the multi-association graph method in [37] to create graph networks that extract rich spatial dependencies. Spatial static graph G_{ss} represents the neighborhood spatial structure of the traffic network, which is generated based on the distance between road sensors. Spatial dynamic graph G_{sd} is constructed based on the sensors with similar traffic flow in the traffic network with dynamic changes over time. By merging G_{ss} and G_{sd}, we create the spatially fused graph G_s, from which we derive a composite matrix A_s.

In this paper, we use the One-Hot method to encode time series in traffic data, both daily and weekly, to capture fine-grained adjacent temporal traffic features. According to the dynamic time change, we can identify the time step with a similar traffic flow and obtain the similar function dynamic time step, even if the two time steps are not adjacent. The final temporal dynamic and static features are encoded as E_t. To further enhance the feature relationship, we utilize the Node2vec method [40] to perform node embedding on the composite adjacency matrix A_s, resulting in spatial embedding E_s. Ultimately, we combine the two embeddings to obtain the spatio-temporal embedding.

3.4. Deep Diffusion Graph Convolution

Diffusion-Convolutional Neural Networks assume that information propagates continuously between neighboring nodes according to a certain probability of constant dif-

fusion [24]. Usually, GCN has two operation processes, propagation and transformation. Propagation aggregates each node's neighborhood information and transforms the aggregated information through a linear transformation or activation function [41,42]. For the feature matrix X, the propagation in the diffusion graph convolutional network can be defined as follows:

$$Z = f(W \odot P^* X) \tag{2}$$

where $Z \epsilon \mathbb{R}^{N \times C}$ denotes the output, $W \epsilon \mathbb{R}^{C \times C}$ denotes the weight matrix, C denotes the number of input and output channels, $P^* \in \mathbb{R}^{N \times N}$ is the probability transition matrix, and f denotes the mapping function. The symbol $\odot$ indicates element-wise multiplication. In our work, the matrix P^* can be replaced by the composite matrix A_s. We use the hidden layer output as the input of the next layer, so the new propagation is defined as follows:

$$Z^0 = X \tag{3}$$

$$Z = \sum_{k=1}^{i} W^k A_s Z^{k-1} \tag{4}$$

$$A_s = A_{sd} + A_{ss} + I_N \tag{5}$$

Here, k refers to the filter, which also signifies the order of the node neighborhood. Z^0 is the original feature matrix, $A_s \in \mathbb{R}^{N \times N}$ is the composite adjacency matrix, A_{sd} denotes the static distance matrix, A_{ss} denotes the dynamic similarity matrix, and I_N denotes the identity matrix.

If the diffusion order is two in the diffusion graph convolutional network, it means diffusion to the two-order neighbors of the node. For any node v_i, the propagation of its diffusion convolution is expressed as:

$$Z_{v_i} = Z^0 + Z^{(1)} + Z^{(2)} \tag{6}$$

According to the above equations, we define the transformation of diffusion graph convolution as:

$$H = \sigma\left(W^t \odot Z\right) = \sigma(W^t(Z^0 + Z^{(1)} + Z^{(2)}) \tag{7}$$

where σ denotes the activation function and H is the final output of the diffusion graph convolution.

In traffic forecasting, shallow GCNs that aggregate two- or three-order neighborhood information can easily lose the deep spatial dependencies of higher-order neighborhoods. However, GCN is prone to over-smoothing with the increase in the aggregated neighborhood order, resulting in the nodes tending to be consistent and indistinguishable, thus reducing the forecasting performance. The core operations of GCNs are propagation and transformation, which significantly impact network performance. It is verified in [42] that decoupling operations on propagation and transformation can expand the node receptive field. Base on this method, on the basis of Equation (3), we decouple the transformations of the features using MLP operations. Then, the new feature matrix X^0 can be defined as follows:

$$X^0 = \mathrm{MLP(X)} \tag{8}$$

$$Z^0 = X^0 \tag{9}$$

The decoupled GCN neighborhood convolution process is shown in Figure 6. Since the deepening of graph networks can suffer from the problem of over-smoothing, to solve this problem, referring to the residual network approach [39], we connect hidden layers to the network, and their weights are adjusted adaptively. The propagation of the deeper graph convolutional can be defined based on Equations (4) and (8) as:

$$X' = (1 - \alpha)X + \alpha X^0 + \beta\left(X + X^0\right) \tag{10}$$

$$Z = \sum_{k=1}^{i} W^k A_s X' \tag{11}$$

where α and β are hyperparameters, α belongs to the range $(0, 1)$, and β is equal to $1 - k^{-1}$. Here, k represents the node convolution order. The parameter β increases as k grows, and this helps to mitigate model degradation.

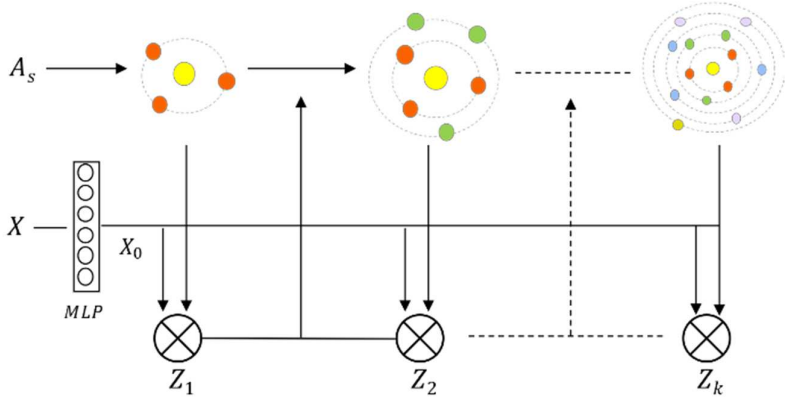


Figure 6. The process of decoupling the feature representation.

3.5. Adaptive Deep Graph Convolution

Although composite adjacency matrices based on node distance and similarity function can simultaneously capture the spatial relationship between adjacent and non-adjacent nodes, they are built based on a fixed structure and are not ideal for discovering the unknown hidden spatial relations between nodes. Traffic flow can change in a complex way depending on various external factors, and a fixed graph structure makes it difficult to extract more information from the challenging changes. We create an adaptive matrix to improve the flexibility of the graph. It can acquire the dependencies in different spaces through parameter sharing and adaptively learns the unknown changing relationships in the network. We set two randomly initialized matrices, fuse them and use a nonlinear activation function to activate, so that the adaptive matrix is defined as follows:

$$A_{adp} = \sigma(A_1 A_2) \tag{12}$$

where A_{adp} is the adaptation matrix, σ is the activation function, and $A_1, A_2 \, \epsilon \, \mathbb{R}^{N \times N}$ are two random initialization matrices representing random sensor nodes in the traffic network. According to the above equation, the propagation of adaptive graph convolution can be defined as:

$$Z_{adp} = W^a \sigma(A_1 A_2) X \tag{13}$$

Adaptive adjacency matrices feature spaces with randomness, and composite adjacency matrices are spaces possessing proximity and similarity. They have some common features, although their parameters are different. By adopting parameter sharing, we extract common features to further strengthen the fusion of spatial and feature information. We can define the spatial graph convolution and adaptive graph convolution with the same shared weights as:

$$Z_{sp} = W^c A_s X \tag{14}$$

$$Z_{adp}{}' = W^c A_{adp} X \tag{15}$$

where Z_{sp} denotes spatial graph convolution, Z_{adp} denotes adaptive graph convolution, and $W^c \, \epsilon \, \mathbb{R}^{C \times C}$ is the shared weight matrix. Then, the shared graph convolution can be defined as:

$$Z_{com} = \left(Z_{sp} + Z_{adp}{}' \right) / 2 \tag{16}$$

According to Equations (13) and (16), we can define the propagation of the parameter-sharing adaptive graph convolution as:

$$Z_{adp\_c} = Z_{adp} + Z_{com} \tag{17}$$

According to Equations (11) and (17), after transformation, as shown in Figure 7, we finally define the adaptive deeper graph convolution as:

$$H_G = \sigma(\sum_{k=1}^{i} W^k A_s X' + Z_{adp\_c}) \tag{18}$$

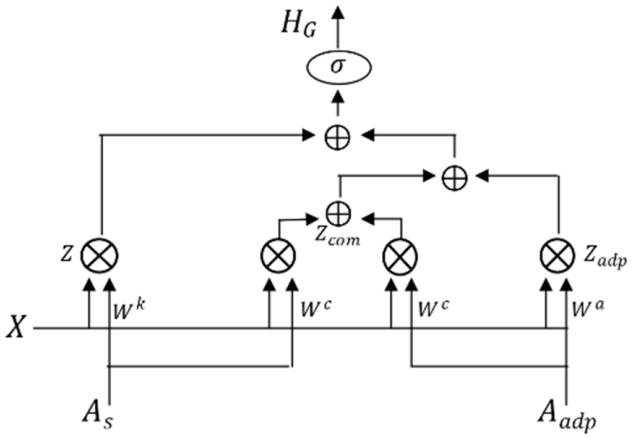


Figure 7. Adaptive graph convolution with parameter sharing.

3.6. Dilated Causal Temporal Convolution

A Temporal Convolution Network (TCN) [36] is widely used in time series research because the inability to see future data during propagation avoids information leakage. It employs dilated convolution to enlarge the receptive field, enabling the capture of longer temporal relationships. In this study, we use a TCN to capture temporal relationships in the traffic flow. It can be defined as:

$$H' = \sum_{i=0}^{k'-1} f \cdot \mathrm{X}_{\mathrm{s-d \cdot i}} \tag{19}$$

where f is the 1-D filter, s is any time step within the set T, d is the dilation factor, and k' is the kernel size. In this paper, we set $k' = 2$, that is, the time convolution on the s-th time step involves convolving the upper layer's time step with the $(s - d)$-th time step, then the above equation can be simplified as:

$$H' = f \cdot \mathrm{X}_s + f \cdot \mathrm{X}_{s-d} \tag{20}$$

To further extract richer time dependencies, we add a gating mechanism:

$$H_T = ReLU\left(sigmoid\left(H'_a\right) * \tanh\left(H'_b\right)\right) \tag{21}$$

where H'_a denotes the 1D temporal convolution operation in the temporal dimension and H'_b denotes the 2D temporal convolution operation in both the spatial and temporal dimensions. The *sigmoid* activation function filters weaken relations in the 1D convolution, and the *tanh* activation function controls the 2D convolution result between $(-1, 1)$. Both activation functions are multiplied to highlight the important information, and the *ReLU* activation function is used to eliminate weak connections in the TCN to obtain the final temporal dependencies. We use double-layer convolution in 2D temporal convolution in both spatial and temporal dimensions to capture additional spatio-temporal relationships, as shown in Figure 8.

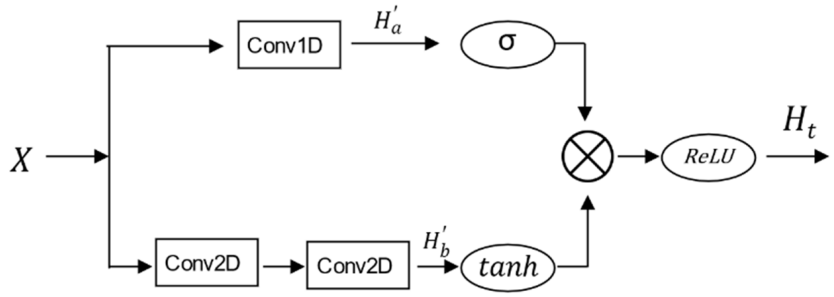


Figure 8. TCN gating mechanism.

3.7. Attention Mechanism

To strengthen the spatio-temporal dependency extraction, we combine the spatio-temporal embedding E_{st} with the spatio-temporal convolutional layer output to perform multi-strategy fusion transformation through the multi-head attention to obtain the forecast result. In this study, we divide the space-time embedding E_{st} into historical spatio-temporal embedding E_{st_h} and predictive spatio-temporal embedding E_{st_p} and acquire the importance weight of the embedding predicted from historical embedding. Referring to the attention mechanism, we define single-head attention as:

$$H' = \sum_{i=1}^{n} \alpha_{st} \cdot V \tag{22}$$

$$\alpha_{st} = softmax\left(\left(E_{st\_p} \cdot {E_{st\_h}}^{T}\right) \cdot h^{-0.5}\right) \tag{23}$$

where α_{st} denotes the importance coefficient of spatio-temporal attention, V denotes the spatio-temporal dependency obtained after stacking ADSTCN layers, H' denotes the output result of single-head attention, $softmax$ is the activation function, and h is the quantity of attention heads.

We concatenate the multi-head attention output to obtain the fusion output result and transform the attention mechanism, which will be converted by the activation function and fully connected layer into the final forecast result. According to Equation (10), the output result after fusion and the multi-head attention mechanism transformation is defined as:

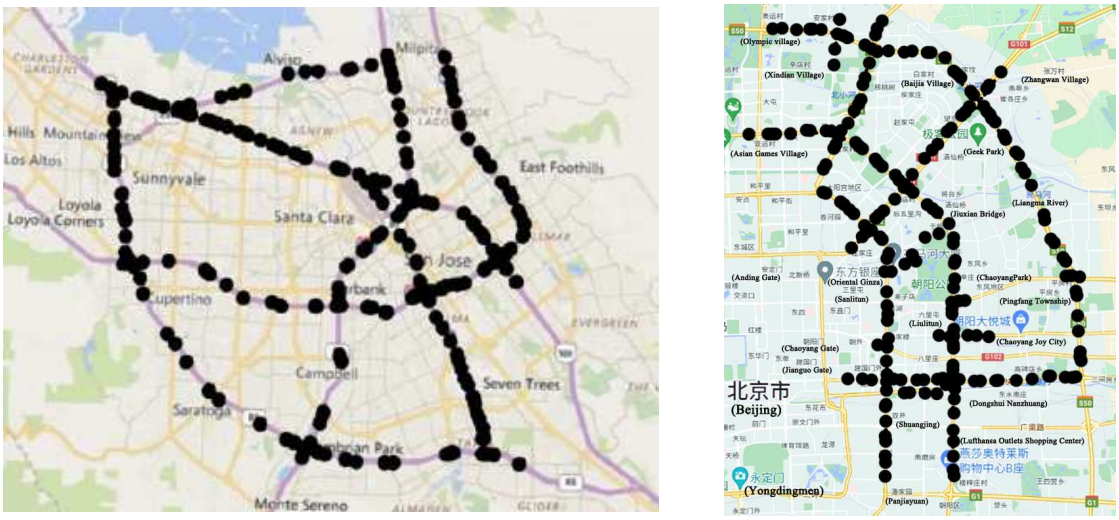
$$H_{att} = concat(H_1, H_2, \ldots\ldots, H_h) \tag{24}$$

4. Experiments

In this section, we assess the performance of the ADSTGCN model using two real datasets, namely the highway network and the urban road network. We compare and analyze our model's experimental outcomes against nine traffic forecasting baseline models to validate its effectiveness. Additionally, we conduct ablation studies and analyze the pivotal components in the model.

4.1. DataSets

In our experiment, we select two real traffic datasets, as shown in Figure 9. One is the highway network dataset PEMS_BAY. The CalTrans Performance Measurement System collects it and has 325 sensors. It collected data for six months, from 1 January 2017 to 31 May 2017. The traffic speed is high, and the traffic situation is comparatively simple as PEMS_BAY involves high-speed road network data. Another dataset used in this study is the NE_BJ road network dataset, comprising 500 sensors, and collected through Navigation data in Northeast Beijing for a duration of one month. It spans between 1 July 2020 and 31 July 2020. The NE_BJ dataset is the real dataset of the main roads within the Beijing urban area. It is more complex and congested than freeway traffic, making it more challenging to forecast traffic. It also has more research value.



(**a**) (**b**)

Figure 9. (**a**) The PEMS_BAY dataset's sensor distribution. (**b**) The NE_BJ dataset's sensor distribution.

Traffic flow data is collected every 30 s, and the unit of speed is km/h. Before the experiment, the collected data were pre-processed and aggregated into 5 min time steps, with one hour of 12 time steps. All data are arranged into time series according to the time step, which is then used as the model's input data. The data is separated into three parts, with proportions of 7:2:1 for the training, test, and validation sets.

4.2. Experimental Settings

We conduct experiments using PyTorch 1.10 on a GeForce RTX 2080Ti GPU. The learning rate is 1×10^{-3}, and the batch size is 16. The order of neighborhood is 8, and the kernel size of the TCN is 2. The time step T is configured to be 12. We use MAE, RMSE, and MAP to evaluate the performance of the models, which are often used in traffic forecasting model evaluation.

4.3. Baselines

During the experiments, we conducted a comparison between ADSTGCN and nine baseline methods. HA [43]: The forecast result is the average of all historical records. VAR [44]: The real-time fluctuation of traffic state can be obtained, and is frequently employed in multivariate time series models. FC-LSTM [45]: A recurrent neural network with LSTM hidden units is fully connected. DCRNN [7]: Graph convolutions are embedded into GRU, and modeled with encoder–decoder architecture for traffic forecasting. STGCN [8]: Spatio-temporal relationships are modelled using pure convolutions to predict traffic with fewer parameters and faster training. GWnet [11]: The use of diffusion graph convolution and an adaptive matrix to obtain better short-term forecast effects. AGCRN [18]: The adjacency matrix is obtained by data-adaptive learning of intrinsic hidden associations between nodes. GMAN [21]: The spatio-temporal representation is extracted according to the random walk of graph nodes and the attention mechanism, and the encoder–decoder architecture is used to model and improve poor medium- and long-term traffic forecasts. MTGNN [46]: Multivariate time series are processed with or without predefined graph structures through a joint framework for modeling learning graph and time series data.

4.4. Experimental Results

We compare the ADSTGCN with the baseline on two real datasets, PEMS_BAY and NE_BJ. The forecasts for each model for the next 15 min, 30 min, and 60 min are presented in Table 1, and all models are evaluated using the MAE, RMSE, and MAPE metrics.

Table 1. Evaluation of traffic forecasting performance of various models on PEMS_BAY and NE_BJ datasets.

	Method	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PEMS_BAY	HA [43]	2.88	5.59	6.80%	2.88	5.59	6.80%	2.88	5.59	6.80%
	VAR [44]	1.74	3.16	3.60%	2.32	4.25	5.00%	2.93	5.44	6.50%
	FC-LSTM [45]	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	DCRNN [7]	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	STGCN [8]	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	GWnet [11]	1.30	2.74	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
	AGCRN [18]	1.37	2.87	2.94%	1.69	3.85	3.87%	1.96	4.54	4.64%
	GMAN [21]	1.34	2.82	2.81%	1.62	3.72	3.63%	1.86	4.32	4.31%
	MTGNN [46]	1.32	2.79	2.77%	1.65	3.74	3.69%	1.94	4.49	4.53%
	ADSTGCN	1.28	2.71	2.70%	1.60	3.63	3.60%	1.86	4.26	4.31%
NE_BJ	HA [43]	6.00	10.95	26.40%	6.00	10.95	26.40%	6.00	10.95	26.40%
	VAR [44]	5.42	8.16	19.28%	5.76	9.07	21.53%	6.14	9.65	23.33%
	FC-LSTM [45]	3.97	7.05	13.05%	4.93	9.04	17.74%	6.06	10.88	23.52%
	DCRNN [7]	3.84	6.84	12.82%	4.51	8.49	15.84%	5.15	9.77	19.08%
	STGCN [8]	5.02	8.34	19.31%	5.10	8.55	19.82%	5.39	9.09	22.14%
	GWnet [11]	3.74	6.54	12.49%	4.41	8.08	15.79%	4.99	9.20	19.45%
	AGCRN [18]	3.84	6.75	13.80%	4.48	8.41	16.70%	4.99	9.44	19.94%
	GMAN [21]	4.08	7.63	14.94%	4.42	8.45	16.51%	4.80	9.18	18.36%
	MTGNN [46]	3.75	6.71	12.91%	4.39	8.33	16.07%	4.90	9.38	19.79%
	ADSTGCN	3.78	6.75	12.95%	4.32	8.16	15.70%	4.73	9.02	18.42%

According to the results presented in Table 1, the non-neural network models, HA and VAR, perform poorly in traffic forecasting, and their learning ability for features is not as strong as that of the neural network models. Conversely, the neural network models achieve better performance in the forecast. After conducting a comprehensive comparison of the two datasets, it is observed that the ADSTGCN model's enhancement of the graph network results in superior performance compared to other baseline models in terms of MAE, RMSE, and MAPE. Through the deepening of the GCN, the ADSTGCN is capable of extracting more profound and intricate spatial relationships, leading to improved long-term forecasting performance, particularly in the Beijing inner city roads with more complex traffic conditions. Additionally, ADSTGCN incorporates an adaptive matrix for parameter sharing, enhancing the flexibility of the graph convolutional network model and facilitating the capture of evolving traffic states, resulting in improved performance.

On the PEMS_BAY dataset, the ADSTGCN model exhibits superior forecast performance for both short-term (15 min) and long-term (60 min) forecasts. GMAN model uses RNN to achieve better long-term forecast results, and ADSTGCN outperforms it in short-term forecasts by 4.48% in MAE. For long-term forecasting results, both models exhibit a similar performance. GWnet achieves superior short-term forecasting results using a purely convolutional model, and ADSTGCN outperforms it by 1.54% in MAE for short-term forecasts and by 4.62% in MAE for long-term forecasts. MTGNN improves the extraction of spatio-temporal dependencies using hybrid jump propagation and achieves a better comprehensive result in both short-term and long-term forecasts. ADSTGCN improves short-term and long-term forecasts compared to it, where short-term forecasts outperform it by 3.03% in MAE, and long-term forecasts outperform it by 4.12% in MAE.

ADSTGCN shows better forecast results in both short-term and long-term forecasts of NE_BJ datasets under more complex traffic situations, with better long-term forecast results. GMAN uses RNN to achieve better long-term forecast results, and ADSTGCN outperforms it by 1.74% in MAE for long-term forecasts and by 7.35% in MAE for short-term forecasts. ADSTGCN's short-term forecast is worse than GWnet in MAE, and its MAE is 1.07% behind GWnet's, but its long-term forecast is 5.21% better than GWnet in MAE. ADSTGCN is significantly affected by external factors in more complex traffic situations in the short term,

and the forecast effect is insufficient. Still, ADSTGCN has a more stable performance in medium- and long-term forecasts.

DCRNN and AGCRN use GCN and RNN to model spatio-temporal relationships, as RNNs are good at sequence data and have better long-term forecast performance than short-term. STGCN, GWnet, and MTGNN use GCN and CNN to model spatio-temporal relationships, are more concise, and achieve better short-term forecast results than long-term. The GMAN model adopts the multi-attention model and an encoding–decoding mechanism to achieve better long-term forecasts than other baseline models. On the basis of GCN, ADSTGCN acquires deeper spatial neighborhood dependencies, extracts richer shared features, and uses adaptive matrices to make the network more flexible. This enables the extraction of richer traffic graph features and learning of more flexible traffic graph structures, and therefore the model improves the forecasting performance. Deepening the graph network makes it easier to discover deeper and more complex spatial relationships between neighboring nodes, thus achieving better performance in long-term forecasting. Figure 10 compares the forecasting performance of ADSTGCN and the nine baseline models on the PEMS_BAY and NE_BJ datasets, respectively.

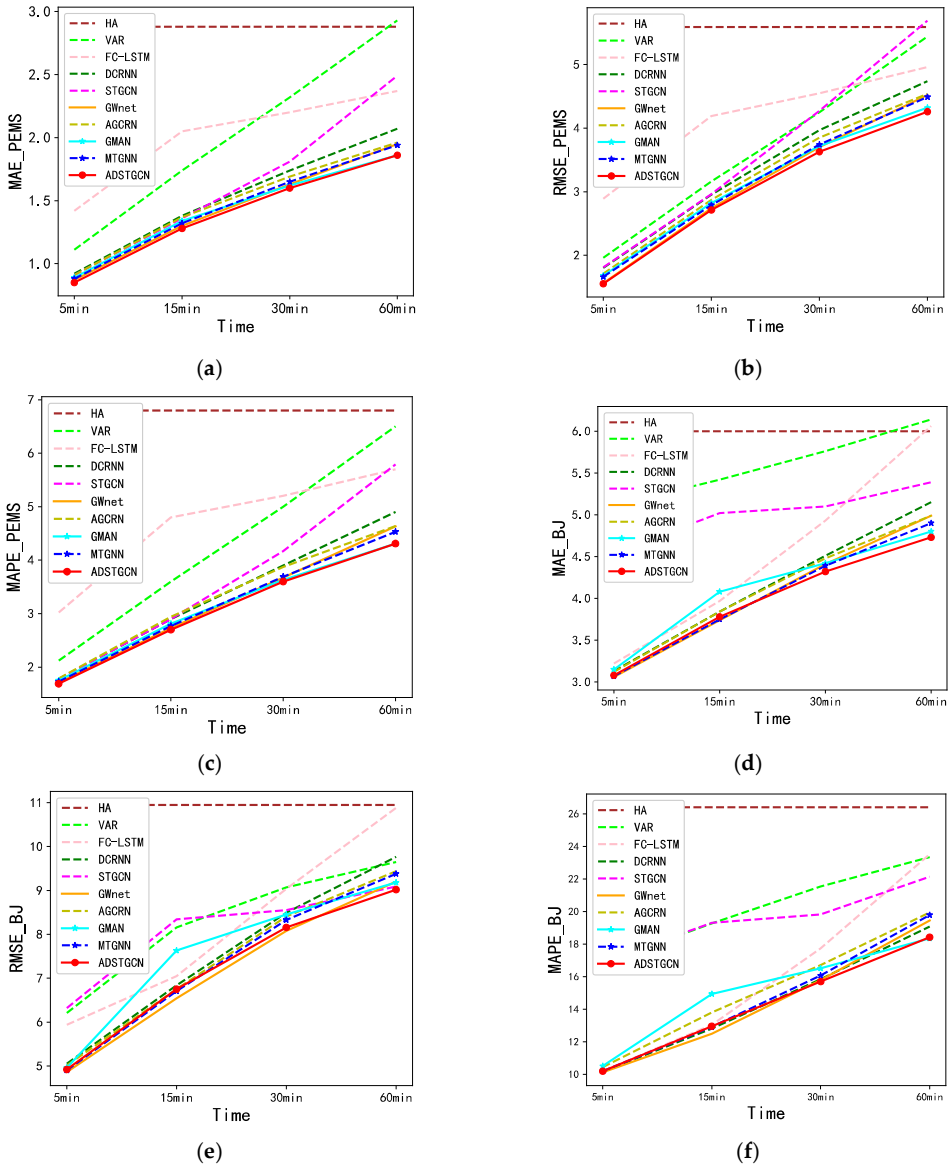
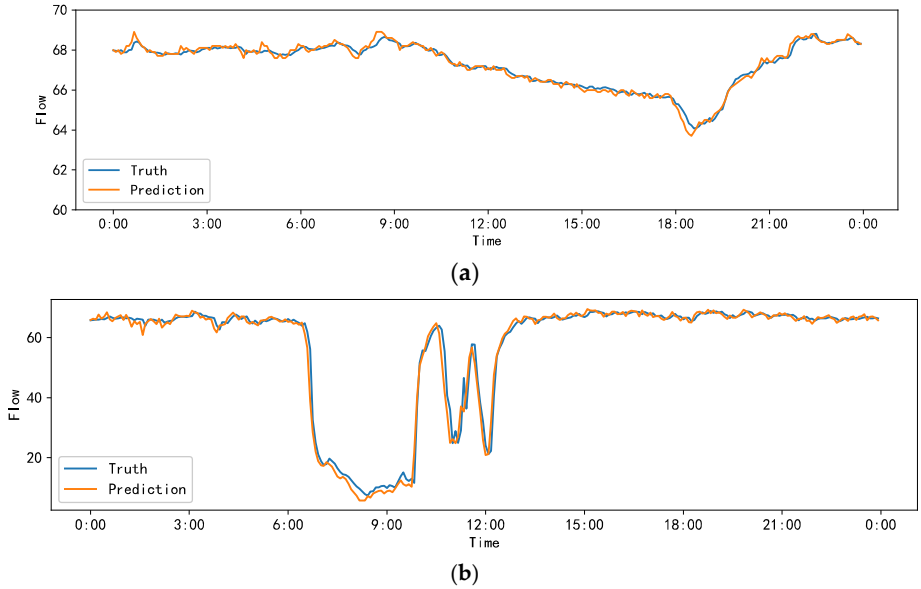


Figure 10. Performance comparison of ADSTGCN with each baseline model. (**a**) MAE(PEMS_BAY); (**b**) RMSE(PEMS_BAY); (**c**) MAPE(PEMS_BAY); (**d**) MAE(NE_BJ); (**e**) RMSE(NE_BJ); (**f**) MAPE(NE_BJ).

Figure 11 compares the actual and predicted traffic forecasting of the ADSTGCN on the PEMS_BAY and NE_BJ datasets on a specific day.



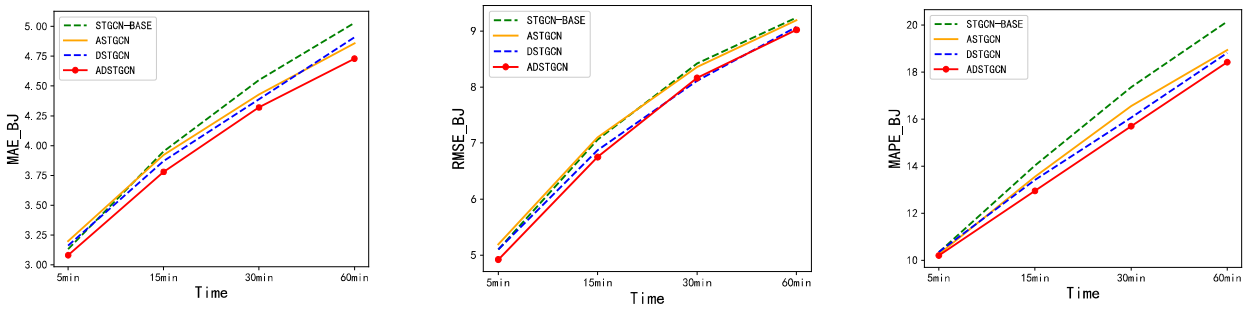
(**a**)

(**b**)

Figure 11. Comparison of the truth and predicted values of the ADSTGCN on the PEMS_BAY and NE_BJ datasets. (**a**) PEMS_BAY; (**b**) NE_BJ.

4.5. Ablation Study

In this section, we conduct experimental ablation research on key model components to verify the method's effectiveness and help us to improve the model further. We study the following ablation models: STGCN: a base model that only includes a two-order neighborhood GCN; DSTGCN: an STGCN-based model that deepens GCN neighborhoods; ASTGCN: a model that adds a parameter-sharing adaptive adjacency matrix to the STGCN. Our proposed ADSTGCN deepens the GCN neighborhood based on the STGCN and adds a parameter-sharing adaptive adjacency matrix model. Taking the NE_BJ dataset as an example, we compare the MAE, RMSE, and MAPE values of the ablation and ADSTGCN model forecast results at 15, 30, and 60 min, respectively, as shown in Figure 12.



(**a**) (**b**) (**c**)

Figure 12. Performance comparison of ADSTGCN with each ablation model in the NE_BJ dataset. (**a**) MAE(NE_BJ); (**b**) RMSE(NE_BJ); (**c**) MAPE(NE_BJ).

The figure shows that the NE_BJ dataset, which has more complex traffic situations, exhibits favorable short-term and long-term traffic forecasting performance when using the ADSTGCN model with the parameter-sharing adaptive adjacency matrix and the adaptive hidden layer connection method. The overall performance of the ASTGCN model using the parameter-sharing adaptive adjacency matrix is better than the basic STGCN model, and its long-term forecast effect is better than its short-term forecast. The comprehensive

performance of the DSTGCN using the adaptive hidden layer connection method is better than that of the basic STGCN model. Because this method can deepen the model and restrain the over-smoothing problem, the short-term and long-term forecast performance is relatively stable.

We compare the ASTGCN with the ASTGCN-NOC adaptive matrix with the parameter sharing removed on the PEMS-BAY and NE_BJ datasets to verify the superior effect of parameter sharing on adaptive matrix adjacency. Their contrasting results on MAE values are shown in Figure 13. It can be seen from the figure that using the parameter-sharing method to extract the adjacent composite and random-feature-space common features further influence the model forecast effect. Adjacent composite spatial convolution is based on composite spatial matrices with neighbors and similarities, while random eigenspace convolution is based on adaptive and eigenspace matrices. In addition to their different parameters, they also have something in common. By extracting the common features of feature and space, the fusion of feature and space is further strengthened to improve the forecast effect.

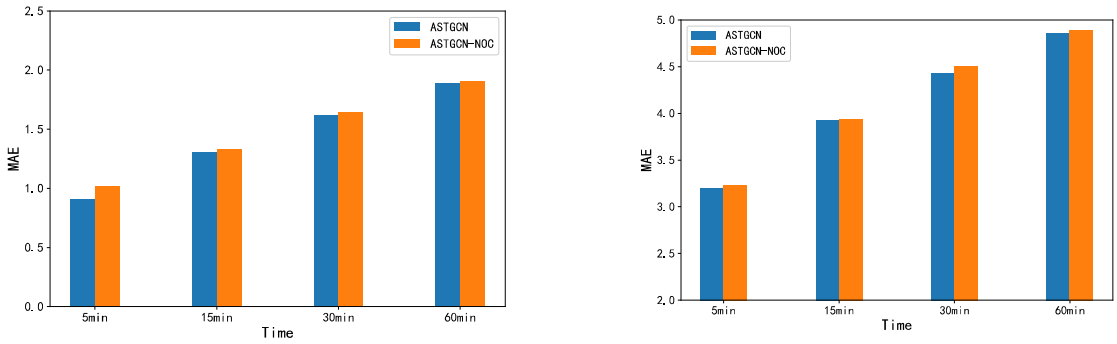


Figure 13. Comparison of the impact of parameter sharing on the forecast performance of the adaptive adjacency matrix on the PEMS_BAY and NE_BJ datasets. (**a**) MAE(PEMS_BAY); (**b**) MAE(NE_BJ).

5. Conclusions

This paper mainly studies the traffic flow forecasting problem using deep Graph Convolutional Networks, as well as traffic road network graph adaptability, and the use of multi-strategy information extraction in traffic forecasting models. We introduce a novel traffic forecasting model, Dynamic Adaptive Deeper Spatio-Temporal Graph Convolutional Networks for Multi-Step Traffic Forecasting (ADSTGCN), using GCN and TCN to obtain spatio-temporal relationships, respectively. The model deepens the neighborhood convolution of the graph while mitigating the network over-smoothing problem using hidden layer connectivity, allowing the model to extract deeper and richer features. The flexibility of node structures in traffic graphs is enhanced using a parameter-sharing adaptive approach. The ADSTGCN performs well when evaluated on two real datasets, highways and urban roads. In our future research, we aim to optimize the model further, validate the model on more comprehensive experimental environments and datasets, and improve the model's efficiency.

Author Contributions: Conceptualization, Z.C. and H.J.P.; methodology, Z.C. and H.J.P.; software, Z.C. and J.Z.; analysis, G.N. and H.J.P.; resources, G.N. and H.J.P.; data curation, G.N. and H.J.P. visualization, Z.C. and J.Z.; supervision, G.N. and H.J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are public datasets that can be downloaded from the public data provider https://pems.dot.ca.gov (accessed on 2 August 2023).

Conflicts of Interest: The authors declare no conflict of interest regarding the publication of this paper.

References

1. Bui, K.-H.N.; Cho, J.; Yi, H. Spatial-Temporal Graph Neural Network for Traffic Forecasting: An Overview and Open Research Issues. *Appl. Intell.* **2022**, *52*, 2763–2774. [CrossRef]
2. Xu, Y.; Cai, X.; Wang, E.; Liu, W.; Yang, Y.; Yang, F. Dynamic Traffic Correlations Based Spatio-Temporal Graph Convolutional Network for Urban Traffic Prediction. *Inf. Sci.* **2022**, *621*, 580–595. [CrossRef]
3. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 914–921. [CrossRef]
4. Zhang, X.; Huang, C.; Xu, Y.; Xia, L.; Dai, P.; Bo, L.; Zhang, J.; Zheng, Y. Traffic Flow Forecasting with Spatial-Temporal Graph Diffusion Network. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 15008–15015. [CrossRef]
5. Chen, W.; Chen, L.; Xie, Y.; Cao, W.; Gao, Y.; Feng, X. Multi-Range Attentive Bicomponent Graph Convolutional Network for Traffic Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 3529–3536. [CrossRef]
6. Zhou, Z.; Yang, Z.; Zhang, Y.; Huang, Y.; Chen, H.; Yu, Z. A Comprehensive Study of Speed Prediction in Transportation System: From Vehicle to Traffic. *iScience* **2022**, *25*, 103909. [CrossRef]
7. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In Proceedings of the International Conference on Learning Representations, ICLR, Vancouver, BC, Canada, 30 April–3 May 2018.
8. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 3634–3640.
9. Huang, R.; Huang, C.; Liu, Y.; Dai, G.; Kong, W. LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; International Joint Conferences on Artificial Intelligence Organization: Yokohama, Japan, 2020; pp. 2355–2361.
10. Guo, K.; Hu, Y.; Sun, Y.; Qian, S.; Gao, J.; Yin, B. Hierarchical Graph Convolution Network for Traffic Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 151–159. [CrossRef]
11. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. *arXiv* **2019**, arXiv:1906.00121.
12. Chen, T.; Zhou, K.; Duan, K.; Zheng, W.; Wang, P.; Hu, X.; Wang, Z. Bag of Tricks for Training Deeper Graph Neural Networks: A Comprehensive Benchmark Study. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2769–2781. [CrossRef] [PubMed]
13. Rong, Y.; Huang, W.; Xu, T.; Huang, J. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. *arXiv* **2019**, arXiv:1907.10903.
14. Oono, K.; Suzuki, T. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. *arXiv* **2019**, arXiv:1905.10947.
15. Park, C.; Lee, C.; Bahng, H.; Tae, Y.; Jin, S.; Kim, K.; Ko, S.; Choo, J. ST-GRAT: A Novel Spatio-Temporal Graph Attention Networks for Accurately Forecasting Dynamically Changing Road Speed. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Conference, 19–23 October 2020; pp. 1215–1224.
16. Han, L.; Du, B.; Sun, L.; Fu, Y.; Lv, Y.; Xiong, H. Dynamic and Multi-Faceted Spatio-Temporal Deep Learning for Traffic Speed Forecasting. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 547–555.
17. Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. Spectral Temporal Graph Neural Network for Multivariate Time-Series Forecasting. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 17766–17778.
18. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17804–17815. [CrossRef]
19. Ye, J.; Xue, S.; Jiang, A. Attention-Based Spatio-Temporal Graph Convolutional Network Considering External Factors for Multi-Step Traffic Flow Prediction. *Digit. Commun. Netw.* **2021**, *8*, 343–350. [CrossRef]
20. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 922–929. [CrossRef]
21. Zheng, C.; Fan, X.; Wang, C.; Qi, J. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *arXiv* **2019**, arXiv:1911.08415. [CrossRef]
22. Cui, Z.; Henrickson, K.; Ke, R.; Wang, Y. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4883–4894. [CrossRef]
23. Kipf, T.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
24. Atwood, J.; Towsley, D. Diffusion-Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29, pp. 2001–2009.

25. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, ICLR, Vancouver, BC, Canada, 30 April–3 May 2018.
26. Niepert, M.; Ahmed, M.; Kutzkov, K. Learning Convolutional Neural Networks for Graphs. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, JMLR.org, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 2014–2023.
27. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852.
28. Huang, J.; Luo, K.; Cao, L.; Wen, Y.; Zhong, S. Learning Multiaspect Traffic Couplings by Multirelational Graph Attention Networks for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 20681–20695. [CrossRef]
29. Zhang, K.; He, F.; Zhang, Z.; Lin, X.; Li, M. Graph Attention Temporal Convolutional Network for Traffic Speed Forecasting on Road Networks. *Transp. B Transp. Dyn.* **2021**, *9*, 153–171. [CrossRef]
30. Zhu, J.; Wang, Q.; Tao, C.; Deng, H.; Zhao, L.; Li, H. AST-GCN: Attribute-Augmented Spatiotemporal Graph Convolutional Network for Traffic Forecasting. *IEEE Access* **2021**, *9*, 35973–35983. [CrossRef]
31. 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. Transport. Syst.* **2021**, *23*, 7743–7758. [CrossRef]
32. Huang, X.; Tang, J.; Yang, X.; Xiong, L. A Time-Dependent Attention Convolutional LSTM Method for Traffic Flow Prediction. *Appl. Intell.* **2022**, *52*, 17371–17386. [CrossRef]
33. Sserwadda, A.; Ozcan, A.; Yaslan, Y. Structural and Topological Guided GCN for Link Prediction in Temporal Networks. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 9667–9675. [CrossRef]
34. Ni, Q.; Zhang, M. STGMN: A Gated Multi-Graph Convolutional Network Framework for Traffic Flow Prediction. *Appl. Intell.* **2022**, *52*, 15026–15039. [CrossRef]
35. Chen, Y.; Xie, Z. Multi-Channel Fusion Graph Neural Network for Multivariate Time Series Forecasting. *J. Comput. Sci.* **2022**, *64*, 101862. [CrossRef]
36. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.
37. Cui, Z.; Zhang, J.; Noh, G.; Park, H.J. MFDGCN: Multi-Stage Spatio-Temporal Fusion Diffusion Graph Convolutional Network for Traffic Prediction. *Appl. Sci.* **2022**, *12*, 2688. [CrossRef]
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
40. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 855–864.
41. Zhou, K.; Dong, Y.; Wang, K.; Lee, W.S.; Hooi, B.; Xu, H.; Feng, J. Understanding and Resolving Performance Degradation in Deep Graph Convolutional Networks. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 2728–2737.
42. Liu, M.; Gao, H.; Ji, S. Towards Deeper Graph Neural Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; Association for Computing Machinery, San Francisco, CA, USA, 6–10 July 2020; pp. 338–348.
43. Smith, B.L.; Demetsky, M.J. Traffic Flow Forecasting: Comparison of Modeling Approaches. *J. Transp. Eng.* **1997**, *123*, 261–266. [CrossRef]
44. Ang, A.; Piazzesi, M. A No-Arbitrage Vector Autoregression of Term Structure Dynamics with Macroeconomic and Latent Variables. *J. Monet. Econ.* **2003**, *50*, 745–787. [CrossRef]
45. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014*; MIT Press: Cambridge, MA, USA, 2014; Volume 2, pp. 3104–3112.
46. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), Virtual Event, CA, USA, 23–27 August 2020*; ACM: New York, NY, USA, 2020. [CrossRef]