

Article

Dynamic Spatio-Temporal Graph Fusion Convolutional Network for Urban Traffic Prediction

Haodong Ma ^{1,2}, Xizhong Qin ^{1,2,*}, Yuan Jia ³ and Junwei Zhou ^{1,2}¹ College of Information Science and Engineering, Xinjiang University, Urumqi 830049, China² Xinjiang Key Laboratory of Signal Detection and Processing, Urumqi 830049, China³ Ming Li College, Renmin University of China, Beijing 100872, China

* Correspondence: qinxz@xju.edu.cn

Abstract: Urban traffic prediction is essential for intelligent transportation systems. However, traffic data often exhibit highly complex spatio-temporal correlations, posing challenges for accurate forecasting. Graph neural networks have demonstrated an outstanding ability in capturing spatial correlations and are now extensively applied to traffic prediction. However, many graph-based methods neglect the dynamic spatial features between road segments and the continuity of spatial features across adjacent time steps, leading to subpar predictive performance. This paper proposes a Dynamic Spatio-Temporal Graph Fusion Convolutional Network (DSTGFCN) to enhance the accuracy of traffic prediction. Specifically, we designed a dynamic graph fusion module without prior road spatial information, which extracts dynamic spatial information among roads from observed data. Subsequently, we fused the dynamic spatial features of the current time step and adjacent time steps to generate a dynamic graph for each time step. The graph convolutional gated recurrent network was employed to model the spatio-temporal correlations jointly. Additionally, residual connections were added to the model to enhance the ability to extract long-term temporal relationships. Finally, we conducted experiments on six publicly available traffic datasets, and the results demonstrated that DSTGFCN outperforms the baseline models with state-of-the-art predictive performance.

Keywords: traffic prediction; dynamic graph structure; graph convolutional network; spatio-temporal modeling



Citation: Ma, H.; Qin, X.; Jia, Y.; Zhou, J. Dynamic Spatio-Temporal Graph Fusion Convolutional Network for Urban Traffic Prediction. *Appl. Sci.* **2023**, *13*, 9304. <https://doi.org/10.3390/app13169304>

Academic Editors: Jin Guo, Wei Su and Wendong Xiao

Received: 19 July 2023

Revised: 11 August 2023

Accepted: 15 August 2023

Published: 16 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid growth of the number of vehicles in urban areas, cities face numerous challenges, such as traffic congestion and environmental pollution resulting from vehicle emissions. Intelligent Transportation Systems (ITS) can effectively alleviate the above problems and provide many conveniences to people's lives [1]. As a primary foundation in advancing ITS, precise traffic prediction plays a critical role in facilitating vehicle allocation, mitigating road congestion, reducing traffic accidents, and optimizing the operational capacity of urban road networks [2].

Traffic data often exhibit complex and dynamic spatio-temporal relationships [3], making accurate predictions challenging. In early studies [4,5], traffic data were considered as linear time series for analysis, neglecting non-linear temporal relationships. With the advancement of deep learning, Recurrent Neural Networks (RNNs) have proven effective in capturing non-linear dependencies within time series, leading to their widespread application in traffic prediction [6]. Nevertheless, these methods focus solely on capturing traffic data's temporal aspects, failing to encompass the intrinsic spatial characteristics. A road's traffic state is influenced by its historical traffic state and its neighbors. That is to say, the traffic state of a road is significantly correlated with the spatial structure of the road network. Thus, it is necessary to model the spatial correlation among various road nodes in the traffic network [7]. Some early studies modeled road networks as grids and

used Convolutional Neural Networks (CNNs) to capture spatial features [8]. However, due to the irregularity of traffic roads and the fact that CNN-based methods typically deal with Euclidean structured data, they cannot effectively capture the complex spatial features of urban road networks [9]. Recently, Graph Neural Networks (GNNs) have gained adequate traction in traffic prediction [10], providing a more suitable framework for modeling the spatial characteristics of road networks. GNN-based methods consider each road segment in the road network as a node in the graph [11,12], while the relationships between different road segments are treated as edges. This way, the road network is constructed as a structured graph.

While most GNN-based methods have shown promising results, they often rely on predefined static adjacency matrices that cannot effectively capture the complex and dynamically changing spatial dependencies in traffic data. Some methods also extract spatial features by constructing adaptive graph structures [12,13], where the model generates the adjacency matrix through learning. However, the adaptive and predefined adjacency matrices remain static, limiting their ability to capture the dynamic spatial dependencies. In real-life scenarios, each road node in a road network can have varying effects on the traffic state of its neighboring road nodes over time. The correlations between road segments are dynamically changing. Figure 1a illustrates a traffic road network instance where sensors are strategically placed on the primary roads to record traffic speed data. Sensor 1 and Sensor 4 record the vehicle speeds on roads within residential areas, while Sensor 2 and Sensor 5 record the vehicle speeds on roads within the office area. Sensor 3 is located on a road that lies between these two areas. These sensors are abstracted as nodes in the graph, and the strength of spatial correlation between them is abstracted as edges. As time progresses, the traffic states on various road segments change, and the spatial correlation between road nodes also varies. Figure 1b illustrates the dynamic changes in correlation between nodes. During the morning peak hours, the traffic states in the residential area significantly affect Sensor 3, but the influence gradually diminishes over time.

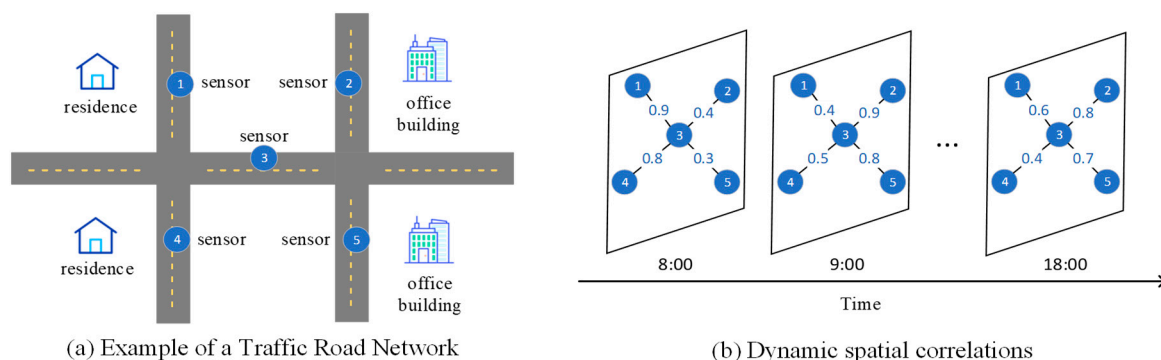


Figure 1. Transportation systems.

Furthermore, existing models commonly employ RNN-based methods or CNN-based methods to model temporal correlations [14,15]. RNNs capture temporal dependencies in time series effectively. However, due to the typical sequential structure of RNNs, RNN-based methods require multiple iterations to model long-term temporal correlations, which can lead to error accumulation and gradient explosion issues [16]. Unlike RNNs, CNN-based methods have advantages such as parallel operations and gradient stabilization. However, CNNs perform implicit temporal modeling. The time steps are not visible, which leads to a lack of flexibility [17]. Several research studies have employed Transformer-based architectures to extract the temporal dependencies [18,19]. These approaches have demonstrated promising capabilities in modeling long-term temporal correlations. However, Transformers rely on positional encoding to capture the order information within a sequence, which leads to limited effectiveness in capturing local temporal correlations [20].

Considering the multifaceted aspects and complexities of the matter at hand, we propose a method named DSTGFCN based on an encoder–decoder framework to achieve traffic prediction. In particular, the DSTGFCN captures dynamic spatio-temporal features from observed data to construct dynamic adjacency matrix. With the graph convolution gated recurrent network and the dynamic adjacency matrix, the model achieves simultaneous modeling of dynamic spatio-temporal correlations and adds residual connections between the graph convolution gated recurrent layers to address problems such as error accumulation and gradient explosion. The main contributions of this paper can be summarized as follows:

- A multi-step ahead prediction model is proposed to achieve accurate traffic prediction. A dynamic graph fusion module can extract spatial information from observed data without prior knowledge and fuse dynamic spatial features from adjacent time steps to generate a dynamic adjacency matrix.
- We effectively modeled the dynamic spatio-temporal correlations by combining the Graph Convolutional Gated Recurrent Unit (GC-GRU) with the dynamic adjacency matrix. Residual connections were added between the GC-GRU layers to propagate gradients and extract long-term temporal dependencies efficiently.
- The proposed model was tested against multiple baselines on six real-world traffic datasets and showed superior predictions. In addition, ablation experiments validated the effectiveness of each component.

This paper is organized as follows: Section 2 presents a comprehensive review of related works. Section 3 introduces the preliminary content and formulates the research problem. Next, in Section 4, we provide a detailed description of our proposed approach. The experiments, including comparative experiments, ablation experiments, and visualization of predictions, are presented in Section 5. Section 6 discusses the advantages of the proposed method. In the end, Section 7 summarizes the paper and presents plans.

2. Related Work

In the past decades, traffic prediction has been an essential component of ITS and has been extensively researched. Earlier research efforts were usually based on statistical methods for traffic prediction, ignoring the nonlinear characteristics and complex variations in traffic data [4]. Machine learning methods can capture the nonlinear dependencies in traffic data compared to statistical methods [5]. However, they rely on high-quality manual features, which can be time-consuming to extract.

With the continuous development of traffic big data and artificial intelligence technologies [21], a growing body of research has proposed various spatio-temporal modeling methods to capture the spatio-temporal features within traffic data [7]. Existing approaches usually model traffic data's temporal and spatial dimensions separately [22]. Sequence models are typically used to extract temporal relationships, such as Long Short-Term Memory (LSTM) [23], Gated Recurrent Units (GRUs) [24], and Temporal Convolutional Networks (TCNs) [25]. As traffic road networks naturally possess a non-Euclidean structure, GNN-based approaches are well suited for capturing the non-Euclidean relationships between multiple traffic time series to model spatial dependencies [26]. For instance, the Temporal Graph Convolutional Network (T-GCN) leverages graph convolutional networks (GCNs) to capture spatial features and GRUs to capture temporal features, which can effectively model spatio-temporal correlations [27].

Recent research has highlighted the limitations of predefined adjacency matrices in adequately capturing the spatial relationships and latent information between nodes when designing spatio-temporal GNNs. The Adaptive Graph Convolutional Recurrent Network (AGCRN) addresses this issue by learning specific node attributes and constructing adaptive graphs to explore latent spatial relationships further [13]. Similarly, Ta et al. [28] adopted a macroscopic and microscopic perspective to learn global and local spatial structures, aiming to acquire an optimal graph structure. Jiang et al. [29] proposed a Meta-Graph Learner that relies solely on observed data to construct an adaptive adjacency matrix. The

abovementioned methods have further improved prediction accuracy in traffic forecasting tasks, indicating that adaptive adjacency matrices can compensate for the limitations of predefined adjacency matrices in modeling spatial correlations. However, in traffic data, the spatial relationships among road segments are vary with time. Li et al. [17] designed a dynamic graph generator that extracts static, dynamic, and temporal information from traffic data to generate dynamic graphs. Zhao et al. [30] employed a channel attention mechanism to allocate dynamic weights to historical traffic sequences at different time steps to achieve dynamic adjustment of spatio-temporal correlation. In the work of Hu et al. [31], dynamic graphs were generated by combining spatial heterogeneity information and geospatial proximity information at each time step. Zhang et al. [32] modeled dynamic spatial correlations by exploring fine-grained features between nodes. Zheng et al. [33] concatenated spatial information from recent time steps and each past time step to generate dynamic spatio-temporal graphs. Despite achieving promising predictive performance, the above methods rely on prior road spatial knowledge and fail to effectively extract dynamic features from traffic data and model dynamic spatial correlations.

Motivated by the abovementioned research, we propose a novel traffic prediction model called DSTGFCN. This model addresses the challenges of complex and dynamic road networks by extracting dynamic spatial information from observed data and generating a dynamic adjacency matrix at each time step, all without relying on prior knowledge of the road spatial relationships. Therefore, DSTGFCN is not limited to a fixed spatial structure and applies to large-scale traffic road networks.

3. Preliminaries

Definition 1 (Traffic Network). We used a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ to represent the spatial topological structure of the traffic road network. \mathcal{V} is the set of $|\mathcal{V}| = N$ nodes, and each node relates to each traffic sensor that records traffic information. \mathcal{E} is the set of $|\mathcal{E}| = M$ edges. $\mathcal{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix, where each element signifies the connection strength between nodes.

Definition 2 (Traffic State). A traffic state vector $X_t \in \mathbb{R}^{N \times c}$ represents the observed values of all traffic sensors in the traffic network \mathcal{G} at time step t , such as traffic speed or flow. Here, c represents the number of features.

Problem (Traffic Prediction). Given a road network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ and its observed P step traffic states $\mathcal{X} = (X_{t-P+1}, X_{t-P+2}, \dots, X_{t-P}) \in \mathbb{R}^{P \times N \times c}$, traffic prediction aims to predict the subsequent Q step traffic states $\mathcal{Y} = (\hat{X}_{t+1}, \hat{X}_{t+2}, \dots, \hat{X}_{t+Q}) \in \mathbb{R}^{Q \times N \times c}$ by learning the function \mathcal{F} , represented as follows:

$$(X_{t-P+1}, X_{t-P+2}, \dots, X_{t-P}; \mathcal{G}) \xrightarrow{\mathcal{F}} (\hat{X}_{t+1}, \hat{X}_{t+2}, \dots, \hat{X}_{t+Q}) \quad (1)$$

4. Methodology

The overall framework of the proposed DSTGFCN is shown in Figure 2. This framework employs an encoder–decoder structure to facilitate multi-step prediction. Inspired by the research in [13], we substituted all the linear layers in the GRU with graph convolutions to construct GC-GRU as the fundamental unit for spatio-temporal modeling. During the encoding stage, the dynamic graph fusion module extracts spatial information for each time step based on the traffic state, time information, and learnable spatial node embeddings to model dynamic spatial correlations. Then, it fuses the dynamic spatial features of adjacent time slots to generate a dynamic graph. The GC-GRU receives the dynamic adjacency matrix to model dynamic spatio-temporal correlations. Simultaneously, the spatial node embeddings adequately learn the dynamic and latent spatial information from historical traffic data to construct an adaptive adjacency matrix. Since future traffic states cannot be observed during the decoding phase, the decoder utilizes an adaptive adjacency matrix to model spatio-temporal correlations and achieve multi-step traffic prediction. Furthermore,

residual connections are added between layers of the GC-GRU to harness the capacity of the multi-layer network in a stable training process.

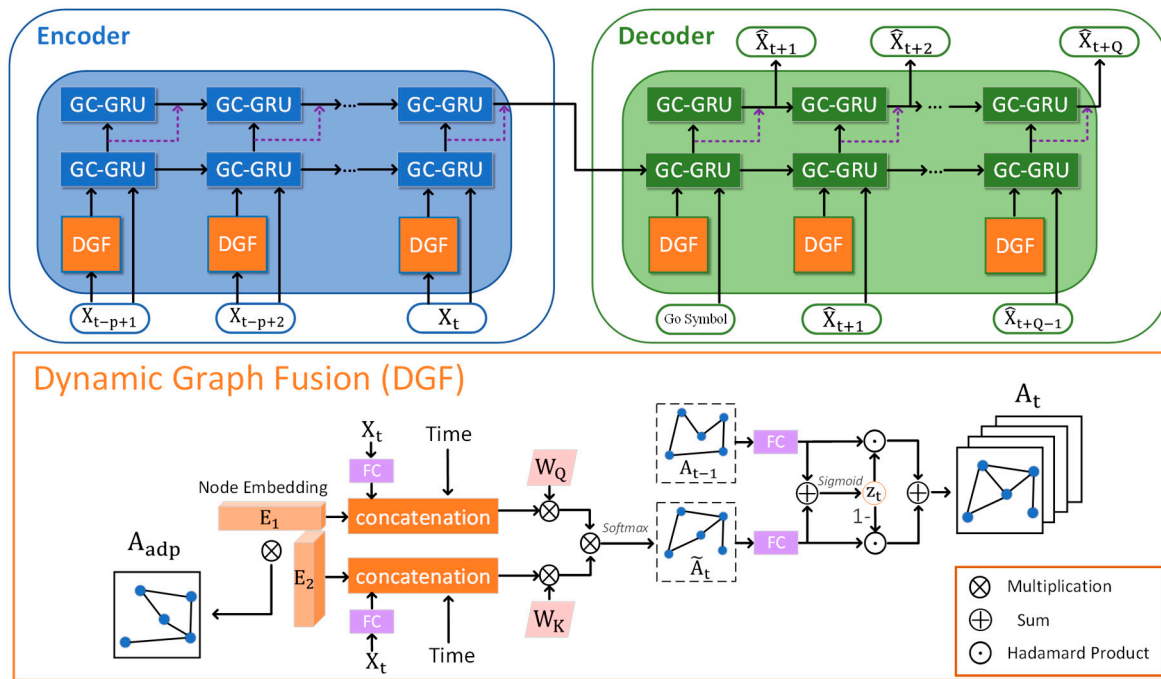


Figure 2. The overall framework of DSTGFCN. The encoder and decoder consist of two GC-GRU layers, with residual connections between the layers. The GC-GRU receives dynamic or adaptive graphs and simultaneously models the spatio-temporal correlations.

4.1. Dynamic Graph Fusion Module

In this section, we will design a dynamic graph fusion module. This module aims to generate an adjacency matrix that represents the dynamic spatial correlations in the traffic road network by fusing the dynamic features extracted from the road attributes. As the dynamic spatio-temporal correlations heavily depend on real-time traffic states, it is essential to model the dynamic spatial correlations by inputting real-time traffic states.

The core of constructing the dynamic feature matrix ensures a comprehensive encoding of the input’s dynamic, latent spatial, and temporal information. To achieve this, we incorporate the following components at each time step: the current traffic state $X_t \in R^{N \times c}$, and the time-related embeddings including time of day $T_t^D \in R^{N \times d}$ and day of the week $T_t^W \in R^{N \times d}$. To further efficiently extract the hidden space features between nodes, we use two spatial node embeddings $E_1 \in R^{N \times e}$ and $E_2 \in R^{N \times e}$. Additionally, we extract features for the traffic state X_t using two non-linear fully connected layers and convert the dimensionality from $N \times c$ to $N \times h$. From this, at time step t , we create two dynamic feature matrices by fusing the above features in a concatenated manner as follows:

$$DF_t^1 = FC(X_t) \parallel E_1 \parallel T_t^D \parallel T_t^W \tag{2}$$

$$DF_t^2 = FC(X_t) \parallel E_2 \parallel T_t^D \parallel T_t^W \tag{3}$$

where $DF_t^1, DF_t^2 \in R^{N \times (h+e+2d)}$, N is the number of nodes, h is the feature dimension, e is the node embedding dimension, and d is the temporal embedding dimension. $FC(\cdot)$

denotes the network of two non-linear fully connected layers. We then compute the dynamic feature matrix at the current time step using the self-attention mechanism [34]:

$$\tilde{A}_t = \text{Softmax} \left(\frac{(DF_t^1 W_Q)(DF_t^2 W_K)^T}{\sqrt{h}} \right) \tag{4}$$

where $W_Q, W_K \in R^{(h+e+2d) \times h}$ are the parameters of the self-attention mechanism. $\tilde{A}_t \in R^{N \times N}$ denotes the spatial correlation between road nodes at time step t . In this way, each dynamic feature matrix can learn unique adjacency relationships at each input time step through Equation (4), which reflects the time-varying traffic topology.

Although the traffic conditions are dynamic, these changes occur gradually. For instance, the relationships between neighboring road segments exhibit variations during peak and off-peak periods. However, within consecutive time intervals, the local spatial dependencies between neighboring road segments change slowly. Hence, we employ a gating mechanism to extract and fuse crucial spatial topological information from the current time step’s dynamic feature matrix \tilde{A}_t and the previous time step’s dynamic adjacency matrix A_{t-1} , as follows:

$$z_t = \text{Sigmoid} \left(\tilde{A}_t W_{A_t} + A_{t-1} W_{A_{t-1}} \right) \tag{5}$$

where $W_{A_t}, W_{A_{t-1}} \in R^{N \times N}$ are two learnable linear transformation matrices. Finally, we can obtain the dynamic adjacency matrix $A_t \in R^{N \times N}$ at time step t :

$$A_t = \begin{cases} \tilde{A}_0, & t = 0 \\ z_t \odot \tilde{A}_t + (1 - z_t) \odot A_{t-1}, & t > 0 \end{cases} \tag{6}$$

The dynamic graph fusion module combines the dynamic spatial information of the road network at each time step to generate the dynamic adjacency matrix. However, during the decoding phase, future traffic states cannot be observed, and the input to each GC-GRU in the decoder is the previous time step’s predicted output. Using predicted outputs to construct the dynamic adjacency matrix may introduce errors and inaccurately represent the road network structure. In the encoding stage, E_1 and E_2 implicitly learn the dynamic and latent features from historical information through the dynamic graph fusion module. Therefore, we utilize E_1 and E_2 to construct an adaptive adjacency matrix in the decoder to represent the spatial structure:

$$A_{adp} = \text{Softmax} \left(\text{ReLU} \left(E_1 E_2^T \right) \right) \tag{7}$$

During the decoding and prediction phase, the adaptive adjacency matrix $A_{adp} \in R^{N \times N}$ can effectively extract spatial dependencies within the road network through graph convolution.

4.2. Graph Convolutional Gated Recurrent Layer

The spectral-based GCN has shown great potential in capturing spatial correlations among traffic sequences [35,36]. Given the traffic road nodes, the GCN is a fundamental operation for extracting features from these nodes. The graph convolution operation is approximated using a first-order Chebyshev polynomial expansion as follows:

$$Z = X *_G \Theta = \left(I_N + D^{\frac{1}{2}} A D^{\frac{1}{2}} \right) XW + b \tag{8}$$

Here, $X \in R^{N \times c}$ and $Z \in R^{N \times h}$ are the input and output of the graph convolution operation ($*_G$). $A \in R^{N \times N}$ is the adjacency matrix, $D \in R^{N \times N}$ is the degree matrix,

and $W \in R^{c \times h}$ and $b \in R^h$ denote the learnable weight and bias, respectively. However, Equation (8) only considers the effect of first-order neighboring nodes. According to the summary and analysis of Yin et al. [37], we employ a diffusion convolution layer to model the graph signal's diffusion process within K finite steps. Thus, for Equation (8), we utilize the diffusion convolution in the following manner:

$$Z = X *_{\mathcal{G}} \Theta = \sum_{k=0}^K \left(I_N + D^{\frac{1}{2}} A D^{\frac{1}{2}} \right)^k X W_k + b \tag{9}$$

Besides spatial correlation, traffic prediction is also influenced by complex temporal correlation. The GRU has gained popularity and has been extensively applied in time series prediction. Similar to previous works [18], we combined the diffusion graph convolution and GRU modules and refer to them as the GC-GRU. As illustrated in Figure 3, the GC-GRU replaces the linear layers responsible for the gating and update gates in the GRU with the graph convolution. As a result, the GC-GRU can effectively model both temporal and spatial correlations in the input graph signal, as shown in the following equation:

$$\begin{cases} u_t = \text{Sigmoid}([X_t, H_{t-1}] *_{\mathcal{G}} \Theta_u) \\ r_t = \text{Sigmoid}([X_t, H_{t-1}] *_{\mathcal{G}} \Theta_r) \\ \hat{h}_t = \text{Tanh}([X_t, (r_t \odot H_{t-1})] *_{\mathcal{G}} \Theta_{\hat{h}_t}) \\ H_t = u_t \odot H_{t-1} + (1 - u_t) \odot \hat{h}_t \end{cases} \tag{10}$$

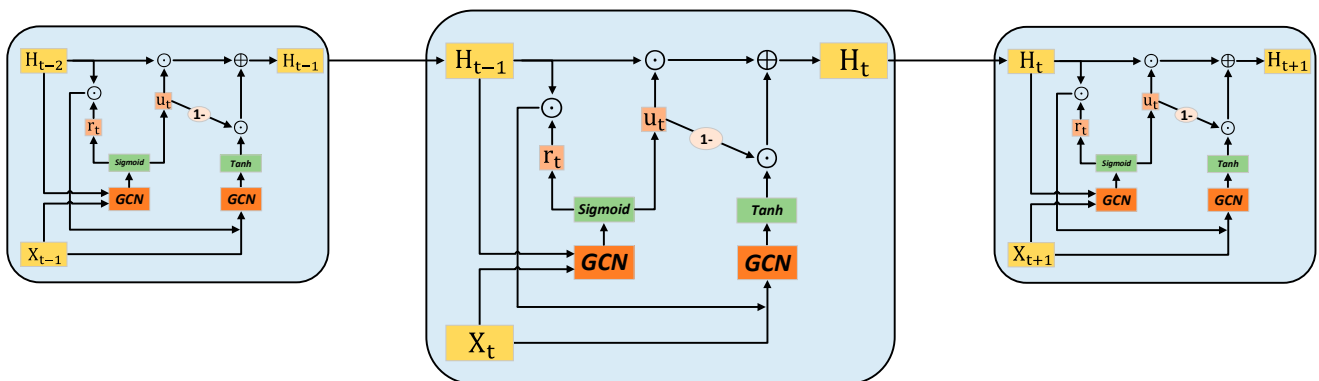


Figure 3. Structure of GC-GRU.

At time step t , X_t represents the input and H_t represents the output hidden state of GC-GRU. u_t and r_t denote the update gate and reset gate. The notation $*_{\mathcal{G}}$ denotes the diffusion graph convolution operation defined by Equation (10), and $\Theta_u, \Theta_r, \Theta_{\hat{h}_t}$ are the learnable parameters corresponding to the diffusion graph convolution. \odot denotes the Hadamard product.

Although the GRU addresses the issue of vanishing gradients in RNNs during back-propagation, it cannot retain all the information for long durations. In a multi-layer GRU, the lower layers can capture local temporal dependencies, while higher layers can capture longer-range temporal dependencies. However, using multiple layers of GRU during training can lead to problems like vanishing or exploding gradients. Residual connections mitigate the decay of gradients during the propagation between layers. Adding residual connections in multi-layer GRUs can alleviate the vanishing or exploding gradient issues, making the training process more stable. As shown in Figure 2, DSTGFCN adopts two layers of GC-GRU in both the encoder and decoder to enhance the model's ability to extract spatial and temporal features in long-term prediction scenarios. The P units (corresponding to P historical time steps) form a graph convolution gated recurrent layer in a cascading

manner, and residual connections are added between the layers to enhance the model's prediction capability and stability.

4.3. Multi-Step Traffic Prediction

In Figure 2, the decoder module is employed for multi-step traffic prediction. It utilizes the hidden states from the encoder and the adaptive graph adjacency matrix obtained from the dynamic graph structure learning module to recursively generate multi-step predictions, i.e., the future traffic state. \mathcal{L}_1 loss is selected as the loss function:

$$\mathcal{L}_1(\Theta) = \frac{1}{Q} \frac{1}{N} \sum_{t=1}^{t=Q} \sum_{i=1}^{i=N} |\hat{X}_{i,t}(\Theta) - X_{i,t}| \quad (11)$$

Here, Θ denotes all trainable parameters in the model, Q is the count of prediction steps, and N is the quantity of road nodes. $\hat{X}_{i,t}(\Theta)$ and $X_{i,t}$ represent the prediction and ground truth of node i at time t .

5. Experiments

Next, we conducted experiments on six real-world datasets to demonstrate the effectiveness of DSTGFCN in traffic speed or flow prediction tasks. In this section, we will first introduce the datasets, experimental settings, evaluation metrics, and representative baselines. Next, we will discuss the experiments comparing the performance of DSTGFCN against other baselines. Furthermore, we conducted ablation experiments to assess the impact of individual components in the model on predictive performance. Finally, we will visualize the predicted values and dynamic adjacency matrix for a more intuitive understanding and evaluation of the model.

5.1. Datasets

We evaluated the performance of our model using six real-world traffic datasets, which encompass two types of traffic data: traffic speed and traffic flow.

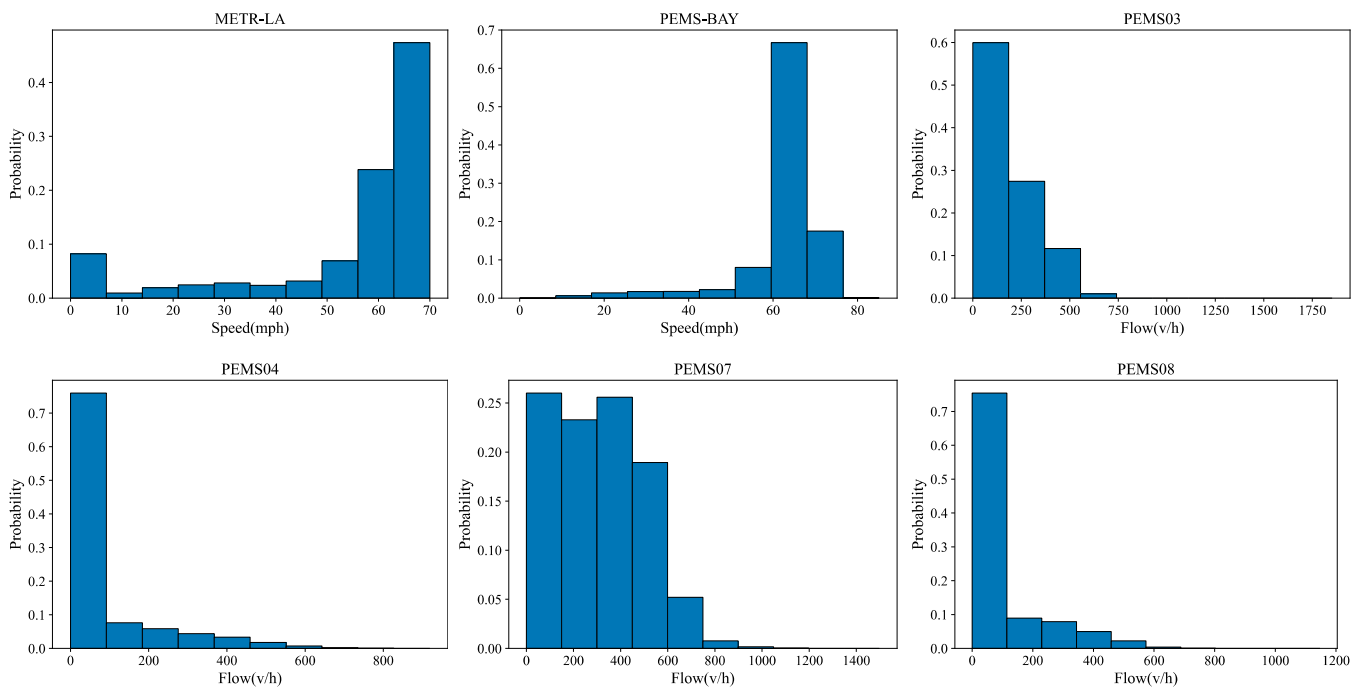
- METR-LA is a dataset of traffic speed collected from 207 sensors on the highways in Los Angeles.
- PEMS-BAY is a dataset comprising traffic speed data from 325 traffic road sensors in the Bay Area.
- PEMS03 is a dataset of traffic flow collected from 358 sensors in the California Third District.
- PEMS04 is a dataset of traffic flow collected from 307 San Francisco Bay Area sensors.
- PEMS07 is a dataset of traffic flow collected from 883 sensors in the California Seventh District.
- PEMS08 is a dataset composed of traffic flow data collected from 170 sensors in the San Bernardino area.

Table 1 presents the detailed information on these six datasets. Following previous research works [12,38], we divided the first two traffic speed datasets into training, validation, and testing sets in a ratio of 7:1:2. The division ratio was 6:2:2 for the other traffic flow datasets. All data points were collected every 5 min. Z-score normalization was used to standardize all the datasets.

Table 1. Details of datasets.

Dataset	Nodes	Time Steps	Time Range	Data Type
METR-LA	207	34,272	2012.03–2012.07	speed
PEMS-BAY	325	52,116	2017.01–2017.06	speed
PEMS03	358	26,208	2018.09–2018.11	flow
PEMS04	307	16,992	2018.01–2018.02	flow
PEMS07	883	28,224	2017.05–2017.08	flow
PEMS08	170	17,856	2016.07–2016.08	flow

We analyzed the six datasets mentioned above, as shown in Figure 4. For the traffic speed datasets, we display the distribution of speed values. The METR-LA dataset exhibits some extreme values, which can be attributed to missing data. In contrast, the speed distribution in the PEMS-BAY dataset was concentrated between 50 mph and 80 mph, indicating a relatively simple traffic pattern with less congestion. We illustrated the distribution of flow values of these traffic flow datasets. PEMS03, PEMS04, and PEMS08 displayed similar flow distributions, with traffic flow concentrated between 0 and 300 vehicles per hour. In contrast, the flow distribution in the PEMS07 dataset was more uniform, lacking clear traffic patterns.

**Figure 4.** Distribution of traffic speed or flow.

5.2. Experiment Settings

All experiments were performed on a computer with an Intel Core i9 13900K/F CPU@5.8 GHz and a GeForce RTX 3090 GPU card with 24 G of video memory, and the model was implemented based on the PyTorch 1.12.0 framework. The number of hidden states was 32. The time and node embedding dimension were 15 and 20, respectively. Both the historical observation and prediction data steps were set to 12. We used the Adam optimizer to optimize the model, where the learning rate was set to 0.01, and batch size was set to 32. The optimization time was 100 epochs, and early stopping was employed to avoid overfitting.

5.3. Evaluation Metrics

The experiment used three metrics that are widely used to assess the accuracy of traffic prediction, which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The details of the equations are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_{true(i)} - y_{pred(i)}| \quad (12)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{true(i)} - y_{pred(i)})^2} \quad (13)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_{true(i)} - y_{pred(i)}}{y_{true(i)}} \right| \times 100\% \quad (14)$$

Among them, MAE is the average of the absolute errors between the predicted values and the ground truth, reflecting the prediction accuracy. RMSE measures the concentration of prediction results around the line of best fit. MAPE reflects the relative magnitude of the deviations between the predicted values and the ground truth. n is the number of samples. $y_{true(i)}$ and $y_{pred(i)}$ denote the ground truth and the predicted values of the i^{th} sample. Smaller values indicate better prediction performance for the above three metrics.

5.4. Baseline Methods

We compared DSTGFCN with eight baselines, including a traditional time series analysis method (ARIMA), deep learning-based methods (FC-LSTM, STGCN, and DCRNN), and excellent existing GNN-based methods (GW-net, AGCRN, DSTAGCN, and STG-NCDE).

- ARIMA [14]. A statistical model commonly used for analyzing and predicting time series data.
- FC-LSTM [14]. LSTM network with the fully connected network to generate traffic series predictions.
- STGCN [11]. This model employs graph convolution and 1D convolution to capture spatial and temporal features, respectively.
- DCRNN [14]. DCRNN combines dual directional diffusion convolution and GRUs for traffic prediction.
- GW-Net [12]. Graph WaveNet combines diffusion causal convolution with GCNs based on an adaptive adjacency matrix to capture potential spatial correlations.
- AGCRN [13]. The model captures spatial correlations between roads through the two proposed adaptive learning modules.
- DSTAGCN [33]. The model connects multiple time frames to construct a dynamic spatio-temporal graph, capturing global spatio-temporal correlations.
- STG-NCDE [39]. The model employs two neural control differential equations to forecast traffic states.

5.5. Experimental Results and Comparative Analysis

Table 2 presents the performance comparison between DSTGFCN and the baseline models for 15 min (short-term), 30 min (mid-term), and 60 min (long-term) predictions on the METR-LA and PEMS-BAY datasets. Table 3 displays the performance comparison between DSTGFCN and the baseline models for average one-hour predictions on the four traffic flow datasets. Our proposed model demonstrated superior predictive performance in traffic speed and traffic flow prediction tasks. The traditional statistical method ARIMA performed the worst, failing to capture the nonlinear relationships in the traffic sequences. FC-LSTM, being a classical recurrent neural network, effectively extracted nonlinear features from sequences. However, it only modeled temporal correlations and overlooked the spatial correlations in the traffic road network, resulting in lower accuracy than the graph-

based models. This finding highlights the importance of modeling spatial correlations to achieve accurate predictions.

Table 2. Performance comparison of DSTGFCN and baseline models on traffic speed prediction tasks.

Dataset	Model	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	ARIMA	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
	FC-LSTM	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
	STGCN	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	DCRNN	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.59	10.50%
	GW-Net	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
	AGCRN	2.87	5.58	7.70%	3.23	6.58	9.00%	3.62	7.51	10.38%
	DSTAGCN	2.74	5.24	7.12%	3.14	6.27	8.65%	3.59	7.33	10.26%
	DSTGFCN	2.53	4.97	6.52%	2.88	6.03	7.79%	3.28	7.13	9.59%
PEMS-BAY	ARIMA	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	FC-LSTM	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	STGCN	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	DCRNN	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	GW-Net	1.30	2.74	2.70%	1.63	3.70	3.7%	1.95	4.52	4.6%
	AGCRN	1.37	2.87	2.94%	1.69	3.85	3.87%	1.96	4.54	4.64%
	DSTAGCN	1.36	2.85	2.88%	1.70	3.84	3.83%	2.01	4.60	4.71%
	DSTGFCN	1.29	2.73	2.69%	1.59	3.63	3.52%	1.85	4.30	4.25%

Bold in the table indicates optimal results.

Table 3. Performance comparison of DSTGFCN and baseline models for traffic flow prediction tasks.

Model	PEMS03			PEMS04			PEMS07			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	35.41	47.59	33.78%	33.73	48.80	24.18%	38.17	59.27	19.46%	31.09	44.32	22.73%
FC-LSTM	21.33	35.11	23.33%	27.14	41.59	18.20%	29.98	45.84	13.20%	22.20	34.06	14.20%
STGCN	17.55	30.42	17.34%	21.16	34.89	13.83%	25.33	39.34	11.21%	17.50	27.09	11.29%
DCRNN	17.99	30.31	18.34%	21.22	33.44	14.17%	25.22	38.61	11.82%	16.82	26.36	10.92%
GW-net	19.12	32.77	18.89%	24.89	39.66	17.29%	26.39	41.50	11.97%	18.28	30.05	12.15%
AGCRN	15.98	28.25	15.23%	19.83	32.26	12.97%	22.37	36.55	9.12%	15.95	25.22	10.09%
DSTAGCN	15.31	25.30	14.91%	19.48	30.98	12.93%	22.07	35.80	9.21%	15.83	24.70	10.03%
STG-NCDE	15.57	27.09	15.06%	19.21	31.09	12.76%	20.53	33.84	8.80%	15.45	24.81	9.92%
DSTGFCN	14.60	25.45	15.66%	18.53	30.51	12.37%	19.70	32.94	8.39%	15.25	24.56	9.88%

Bold in the table indicates optimal results.

STGCN and DCRNN are typical spatio-temporal data prediction models. Both models consider spatial factors, which lead to improved prediction accuracy. However, these methods only utilize adjacency matrices defined by spatial distances for graph convolution operations, which may only partially capture spatial relationships. GW-Net, AGCRN, and STG-NCDE employ adaptive adjacency matrices to explore further hidden spatial features, which can be understood as learning the optimal graph topology of the traffic road network. Although they demonstrated excellent performance, these models still employed static graphs when modeling spatial correlations and did not consider the dynamic variations of spatial relationships. DSTAGCN connects the graphs of the recent and past time frames to construct a dynamic adjacency matrix, resulting in improved prediction capabilities compared to static graphs. Compared to the models above, DSTGFCN fully extracted the dynamic features from the traffic data and combined dynamic spatial features from multiple time frames to generate the dynamic adjacency matrix more efficiently and appropriately. As a result, DSTGFCN exhibited excellent predictive capabilities.

Figure 5 visualizes the prediction errors of DSTGFCN and two other baselines at each time step on the PEMS04 and PEMS08 datasets. The error growth rates of AGCRN and

STG-NCDE were similar, but STG-NCDE performed better than AGCRN in short-term predictions. AGCRN exhibited good performance in long-term predictions on the PEMS08 dataset, where DSTGFCN slightly lagged behind AGCRN regarding MAPE. However, overall, DSTGFCN demonstrated lower errors across the entire time range, showcasing the superior performance of our model.

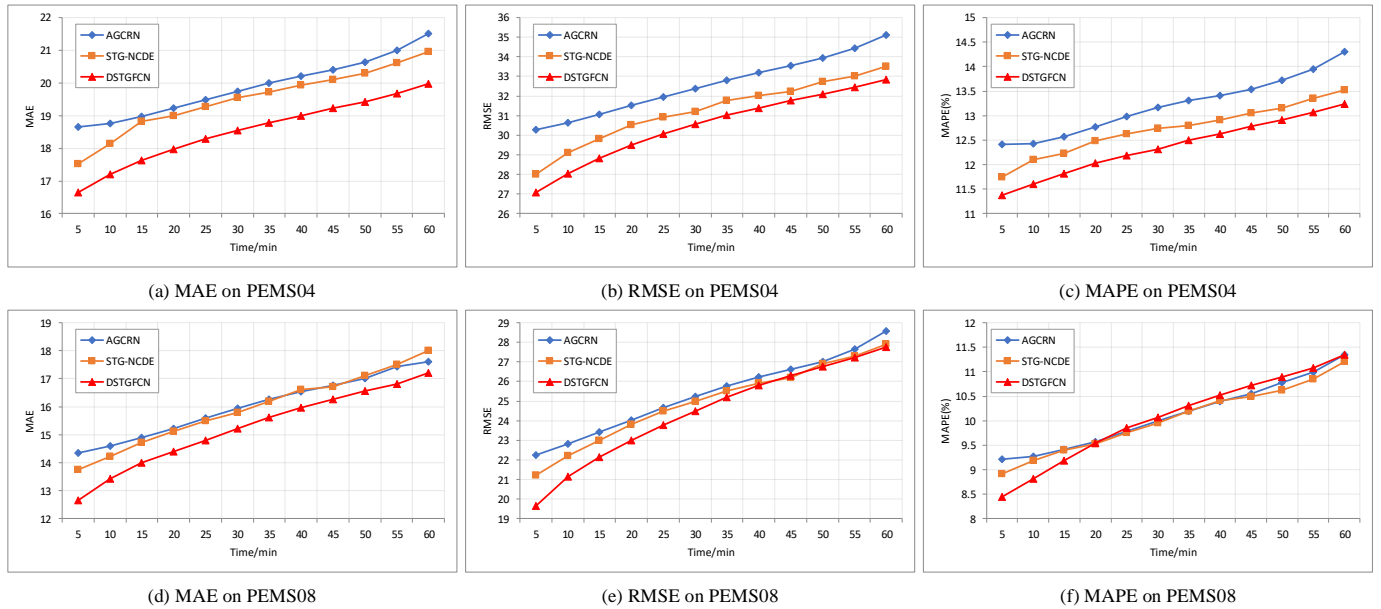


Figure 5. Prediction errors for each time step on PEMS04 and PEMS08.

Lastly, we found that the difficulty of prediction varied across different datasets. Traffic flow data exhibited a more comprehensive range of variations than traffic speed data, as traffic speed is usually constrained within a specific speed range. Therefore, traffic flow data are more complex, leading to more significant errors for all models in traffic flow prediction tasks. In Figure 4, the PEMS-BAY dataset shows a relatively simple traffic pattern, resulting in significantly better prediction results. However, in contrast, the PEMS07 dataset exhibits a more complex traffic pattern, leading to larger MAE and RMSE values for all models on that dataset. Consistently achieving better prediction results across all the datasets demonstrated that DSTGFCN effectively captures the dynamic spatio-temporal dependencies in the traffic road network. This allows DSTGFCN to demonstrate outstanding performance in both traffic speed and traffic flow prediction tasks.

5.6. Ablation Experiments

We conducted ablation experiments to validate the effectiveness of each component in the proposed DSTGFCN model. All ablation experiments were performed on the METR-LA dataset. We named the variants of DSTGFCN as follows:

- w/o Dg: We replaced the dynamic adjacency matrix in DSTGFCN with a predefined adjacency matrix. That is, dynamic graph fusion was also removed. The predefined adjacency matrix was constructed in reference to the way defined by Li et al. [14]. The calculation formula is as follows:

$$A_{v_i, v_j} = \begin{cases} \exp\left(-\frac{d_{v_i, v_j}^2}{\sigma^2}\right), & \text{if } d_{v_i, v_j} \leq k \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where d_{v_i, v_j} represents the road network distance from sensor node v_i to v_j . σ is the standard deviation of the distance and k is the threshold value, which was assigned a value of 0.1.

- w/o Fus: We removed dynamic graph fusion when building the dynamic graphs.
- w/o Res: We removed the residual connections in the graph convolution gated recurrent layer.
- w/o Fus & Res: We removed both dynamic graph fusion and the residual connections.
- Dg2Sg: DSTGFCN replaced the dynamic adjacency matrix with a predefined adjacency matrix while removing dynamic graph fusion and the residual connections.
- Dg w/o X: Traffic states are not considered as input when constructing the dynamic graph.
- Dg w/o T: Time embedding was not considered as input when constructing the dynamic graph.
- Dg w/o E: Node embedding was not considered as input when constructing the dynamic graph.

As shown in Table 4, we can observe that DSTGFCN outperformed other variants in terms of prediction accuracy for 15 min, 30 min, and 60 min. Analysis of the results indicates a significant decline in predictive performance when the dynamic adjacency matrix was removed (w/o Dg, Dg2Sg). Therefore, it is necessary to construct a dynamic adjacency matrix to capture dynamic spatio-temporal features effectively. Additionally, dynamic graph fusion further enhanced the prediction performance (w/o Fus), validating that the adjacency relationships of road nodes exhibit certain similarities in adjacent time steps. The residual connections improved the model's ability to capture long-term dependencies. Specifically, in 60 min predictions, DSTGFCN achieved an MAE reduction from 3.45 to 3.28 compared to DSTGFCN w/o Res, resulting in a 4.92% improvement in prediction accuracy. Meanwhile, DSTGFCN w/o Fus & Res validated that only considering the dynamic adjacency matrix is inadequate for accurate prediction. Ablation experiments on the input of dynamic feature extraction indicate that the deficiencies in the current traffic state (Dg w/o X), time embedding (Dg w/o T), and node embedding (Dg w/o E) will lower the prediction performance. In summary, these components are all crucial for the prediction performance of DSTGFCN.

Table 4. Ablation experiments on the METR-LA dataset.

	15 min			30 min			60 min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DSTGFCN	2.53	4.97	6.52%	2.88	6.03	7.79%	3.28	7.13	9.59%
w/o Dg	2.70	5.37	7.07%	3.15	6.59	8.91%	3.68	7.87	11.16%
w/o Fus	2.56	5.08	6.55%	2.92	6.07	8.05%	3.33	7.16	9.95%
w/o Res	2.62	5.27	6.82%	3.03	6.35	8.43%	3.45	7.45	10.32%
w/o Fus & Res	2.63	5.39	6.95%	3.01	6.41	8.35%	3.41	7.47	10.07%
Dg2Sg	2.71	5.40	7.03%	3.16	6.53	8.84%	3.70	7.81	11.16%
Dg w/o X	2.54	5.07	6.67%	2.91	6.13	8.24%	3.32	7.21	10.08%
Dg w/o T	2.55	5.10	6.70%	2.93	6.17	8.21%	3.36	7.34	10.07%
Dg w/o E	2.66	5.38	7.07%	3.01	6.36	8.52%	3.39	7.35	10.17%

Bold in the table indicates optimal results.

5.7. Visualization

To further visually understand and evaluate the proposed model in this paper, we visualized the ground truth and the model's predictions. As shown in Figure 6, we selected two nodes from the METR-LA dataset, Node 89 and Node 101, and displayed their data for an entire day on 11 June 2012 (from the test set). These two nodes displayed distinct traffic patterns. For example, Sensor 89 showed traffic congestion only during the morning peak hours, while Sensor 101 experienced traffic congestion not only during the morning peak hours but also during the evening peak hours. The results indicate that DSTGFCN can capture different traffic patterns of different nodes. Additionally, we can observe that the ground truth curves were highly irregular with significant fluctuations. Our model could effectively adapt to these abrupt trend changes and make predictions that closely

approximate the ground truth as much as possible. However, specific local details in the predictions may be less accurate due to random solid noise, such as the sudden and significant fluctuations during the morning peak hours in Figure 6a.

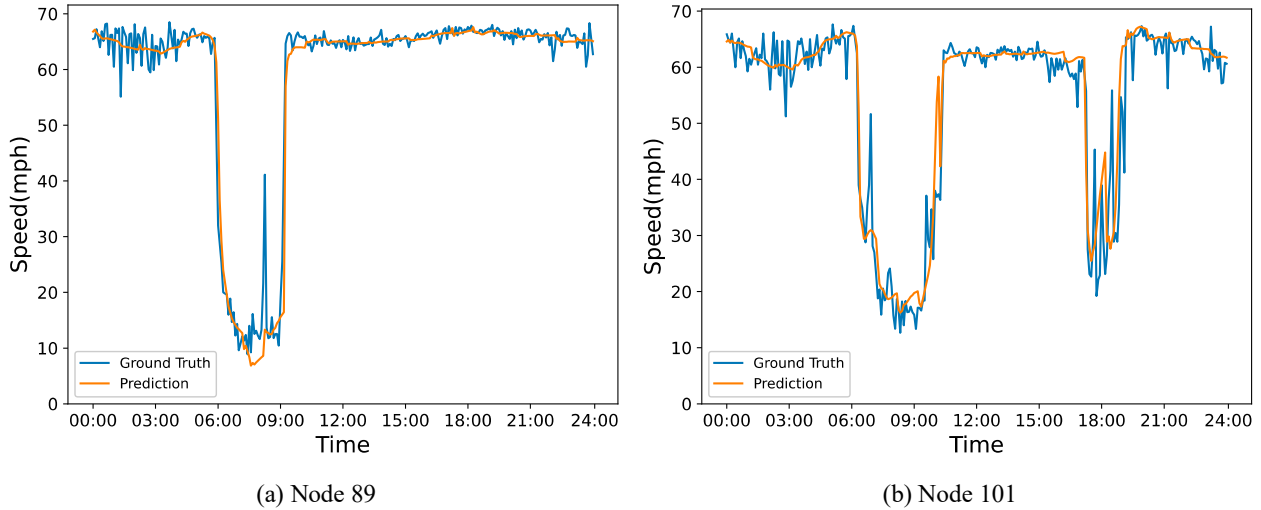


Figure 6. Visualization of the predicted and true values.

To visually observe the dynamic spatial correlations in the traffic road network, we selected 25 nodes from the METR-LA dataset. We visualize their dynamic adjacency matrices for two time periods in Figure 7. It can be observed that the dynamic adjacency matrices changed over time. For instance, Node 20 and Node 5 exhibited a strong correlation at 9:00, but their correlation weakened at 18:00. Additionally, Node 18 showed a similar correlation with other nodes in both periods. This indicates that specific road segments have similar traffic patterns during peak hours. These findings demonstrate that DSTGFCN effectively constructs dynamic adjacency matrices to capture the dynamic topological relationships in the traffic road network.

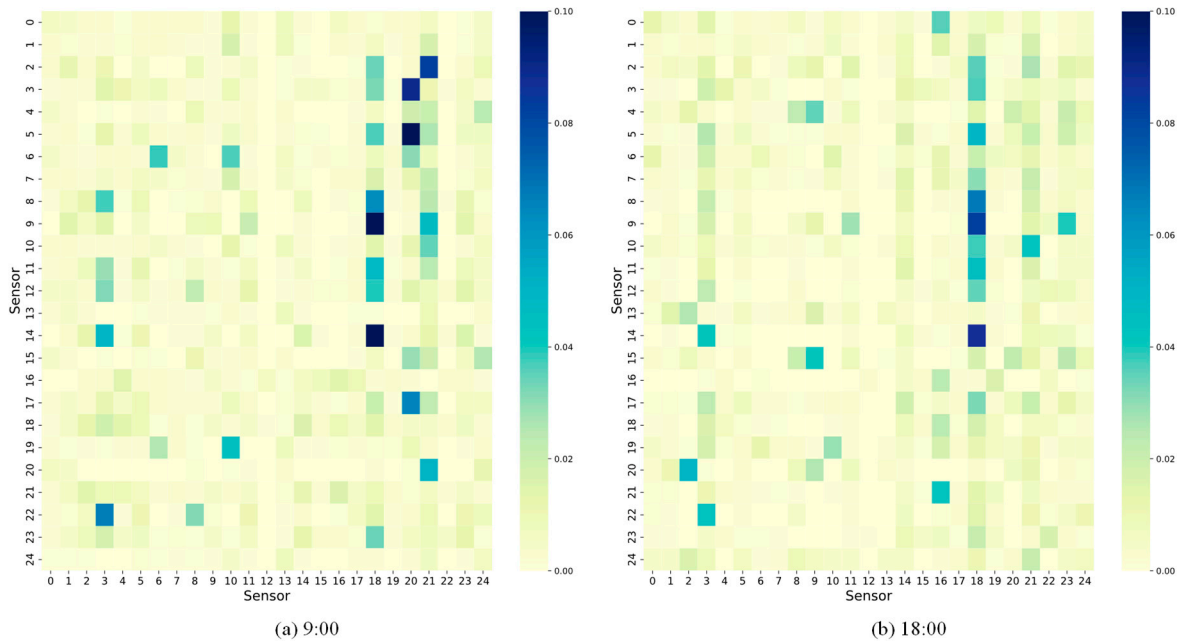


Figure 7. Dynamic adjacency matrices for two time slots.

6. Discussion

This section outlines the primary advantages of the proposed DSTGFCN in traffic speed and flow prediction. DSTGFCN relies solely on observed data to capture dynamic spatial features compared to other GNN-based methods. Subsequently, the dynamic features from adjacent time steps are fused to construct the dynamic adjacency matrix. The experiments demonstrated a significant enhancement in predictive performance by incorporating the dynamic adjacency matrix. Furthermore, the model does not require prior road spatial knowledge, making it more suitable for traffic prediction tasks on large-scale road networks. Specifically, DSTGFCN can infer the spatial relationships between roads in the large-scale road network from the PEMS07 dataset, which includes up to 883 road nodes, and exhibits superior predictive performance. Adding residual connections between GC-GRU layers captures long-term temporal dependencies and mitigates error accumulation, further enhancing the model's predictive capacity. Our model achieved state-of-the-art predictive performance in traffic speed and flow prediction tasks, underscoring its strong generalization capability.

7. Conclusions

This paper introduced a novel approach called DSTGFCN for traffic prediction. Considering the intricate dynamic spatial dependencies among roads in traffic road networks, we first extract current-time dynamic features from observed data. Subsequently, we fuse these features with dynamic features from the previous time step to construct the dynamic adjacency matrix. This dynamic adjacency matrix is utilized in the GC-GRU to model dynamic spatio-temporal correlations simultaneously. Additionally, to capture long-term temporal dependencies and alleviate error accumulation, we introduce residual connections between layers of the GC-GRU. Ultimately, extensive experiments on traffic speed and traffic flow datasets consistently demonstrated the superiority of our DSTGFCN over the baselines, showcasing its robust generalization capability.

In future work, we will focus on generating optimal graph structures and enhancing the model's resilience to noise interference to improve its predictive capacity.

Author Contributions: Conceptualization, H.M. and X.Q.; methodology, H.M.; software, H.M. and X.Q.; validation, H.M., X.Q., Y.J. and J.Z.; formal analysis, X.Q.; investigation, H.M., Y.J. and J.Z.; resources, H.M.; writing—original draft preparation, H.M.; writing—review and editing, H.M. and X.Q.; visualization, H.M.; supervision, X.Q.; project administration, X.Q.; funding acquisition, X.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Major Science and Technology Special Projects of Xinjiang Uygur Autonomous Region, grant number 2020A03001.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Tedjopurnomo, D.A.; Bao, Z.; Zheng, B.; Choudhury, F.M.; Qin, A.K. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 1544–1561. [[CrossRef](#)]
2. Xia, Z.; Wu, J.; Wu, L.; Chen, Y.; Yang, J.; Yu, P.S. A comprehensive survey of the key technologies and challenges surrounding vehicular ad hoc networks. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 37. [[CrossRef](#)]
3. Peng, H.; Wang, H.; Du, B.; Alam Bhuiyan, Z.; Ma, H.; Liu, J.; Wang, L.; Yang, Z.; Du, L.; Wang, S.; et al. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Inf. Sci.* **2020**, *521*, 277–290. [[CrossRef](#)]
4. Ahmed, M.S.; Cook, A.R. *Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques*; Transportation Research Board: Washington, DC, USA, 1979; Volume 722, pp. 1–9.
5. Chen, P.; Ding, C.; Lu, G.; Wang, Y. Short-term traffic states forecasting considering spatial–Temporal impact on an urban expressway. *Transp. Res. Rec.* **2016**, *2594*, 61–72. [[CrossRef](#)]

6. Yu, R.; Li, Y.; Shahabi, C.; Demiryurek, U.; Liu, Y. Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2017; pp. 777–785.
7. Jiang, W.; Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [[CrossRef](#)]
8. Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Li, Z. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5668–5675.
9. Lu, B.; Gan, X.; Jin, H.; Fu, L.; Zhang, H. Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Galway, Ireland, 19–23 October 2020; pp. 1025–1034.
10. 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.* **2020**, *32*, 4–24. [[CrossRef](#)]
11. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
12. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
13. 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.
14. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
15. 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 International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020; pp. 753–763.
16. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. *arXiv* **2015**, arXiv:1506.00019.
17. Li, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; Li, Y. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 1–21. [[CrossRef](#)]
18. Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.J.; Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv* **2020**, arXiv:2001.02908.
19. Song, X.; Wu, Y.; Zhang, C. TSTNet: A sequence to sequence transformer network for spatial-temporal traffic prediction. In *Artificial Neural Networks and Machine Learning—ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, 14–17 September 2021, Proceedings, Part I 30*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 343–354.
20. Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A survey of transformers. *AI Open* **2022**, *3*, 111–132. [[CrossRef](#)]
21. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.-Y. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 865–873. [[CrossRef](#)]
22. Wang, Y.; Ren, Q.; Lv, X.; Sun, J. CPNet: Conditionally parameterized graph convolutional network for traffic forecasting. *Phys. A Stat. Mech. Its Appl.* **2023**, *617*, 128667. [[CrossRef](#)]
23. Van Lint, J.W.C.; Van Hinsbergen, C. Short-term traffic and travel time prediction models. *Artif. Intell. Appl. Crit. Transp. Issues* **2012**, *22*, 22–41.
24. Tian, Y.; Pan, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; IEEE: Piscataway Township, NJ, USA, 2015; pp. 153–158.
25. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
26. Shao, Z.; Zhang, Z.; Wei, W.; Wang, F.; Xu, Y.; Cao, X.; Jensen, C.S. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *arXiv* **2022**, arXiv:2206.09112. [[CrossRef](#)]
27. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [[CrossRef](#)]
28. Ta, X.; Liu, Z.; Hu, X.; Yu, L.; Sun, L.; Du, B. Adaptive spatio-temporal graph neural network for traffic forecasting. *Knowl. Based Syst.* **2022**, *242*, 108199. [[CrossRef](#)]
29. Jiang, R.; Wang, Z.; Yong, J.; Jeph, P.; Chen, Q.; Kobayashi, Y.; Song, X.; Fukushima, S.; Suzumura, T. Spatio-Temporal Meta-Graph Learning for Traffic Forecasting. *arXiv* **2022**, arXiv:2211.14701. [[CrossRef](#)]
30. Zhao, J.; Liu, Z.; Sun, Q.; Li, Q.; Jia, X.; Zhang, R. Attention-based dynamic spatial-temporal graph convolutional networks for traffic speed forecasting. *Expert Syst. Appl.* **2022**, *204*, 117511. [[CrossRef](#)]
31. Hu, J.; Lin, X.; Wang, C. DSTGFCN: Dynamic Spatial-Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Sens. J.* **2022**, *22*, 13116–13124. [[CrossRef](#)]
32. 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.* **2022**, *250*, 109028. [[CrossRef](#)]

33. Zheng, Q.; Zhang, Y. DSTAGCN: Dynamic spatial-temporal adjacent graph convolutional network for traffic forecasting. *IEEE Trans. Big Data* **2022**, *9*, 241–253. [[CrossRef](#)]
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
35. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
36. 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.
37. Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; Yin, B. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4927–4943. [[CrossRef](#)]
38. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5415–5428. [[CrossRef](#)]
39. Choi, J.; Choi, H.; Hwang, J.; Park, N. Graph neural controlled differential equations for traffic forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; AAAI Press: Palo Alto, CA, USA, 2022; Volume 36, pp. 6367–6374.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.