

## Article

# Dynamic Spatio-Temporal Hypergraph Convolutional Network for Traffic Flow Forecasting

Zhiwei Ye <sup>1</sup>, Hairu Wang <sup>1</sup>, Krzysztof Przystupa <sup>2,3,\*</sup> , Jacek Majewski <sup>4</sup>, Nataliya Hots <sup>5</sup> and Jun Su <sup>1,\*</sup>

<sup>1</sup> School of Computer Science, Hubei University of Technology, Wuhan 430068, China; hgcsyzw@mail.hbut.edu.cn (Z.Y.); wanghr1086@hbut.edu.cn (H.W.)

<sup>2</sup> Department of Automation, Lublin University of Technology, Nadbystrzycka 38D, 20-618 Lublin, Poland

<sup>3</sup> Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius, Lithuania

<sup>4</sup> Department of Automatics and Metrology, Lublin University of Technology, Nadbystrzycka 38D, 20-618 Lublin, Poland; j.majewski@pollub.pl

<sup>5</sup> Department of Measuring-Information Technologies, Lviv Polytechnic National University, 79013 Lviv, Ukraine; nataliia.y.hots@lpnu.ua

\* Correspondence: k.przystupa@pollub.pl (K.P.); sujuncs@hbut.edu.cn (J.S.)

**Abstract:** Graph convolutional networks (GCN) are an important research method for intelligent transportation systems (ITS), but they also face the challenge of how to describe the complex spatio-temporal relationships between traffic objects (nodes) more effectively. Although most predictive models are designed based on graph convolutional structures and have achieved effective results, they have certain limitations in describing the high-order relationships between real data. The emergence of hypergraphs breaks this limitation. A dynamic spatio-temporal hypergraph convolutional network (DSTHGCN) model is proposed in this paper. It models the dynamic characteristics of traffic flow graph nodes and the hyperedge features of hypergraphs simultaneously, achieving collaborative convolution between graph convolution and hypergraph convolution (HGCN). On this basis, a hyperedge outlier removal mechanism (HOR) is introduced during the process of node information propagation to hyper-edges, effectively removing outliers and optimizing the hypergraph structure while reducing complexity. Through in-depth experimental analysis on real-world datasets, this method has better performance compared to other methods.

**Keywords:** hypergraph convolutional networks; spatio-temporal data; neural networks; traffic flow forecasting



**Citation:** Ye, Z.; Wang, H.; Przystupa, K.; Majewski, J.; Hots, N.; Su, J.

Dynamic Spatio-Temporal Hypergraph Convolutional Network for Traffic Flow Forecasting. *Electronics* **2024**, *13*, 4435. <https://doi.org/10.3390/electronics13224435>

Received: 7 October 2024

Revised: 31 October 2024

Accepted: 4 November 2024

Published: 12 November 2024



**Copyright:** © 2024 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

Driven by continuous economic growth, self-driving has become the primary mode of transportation, with traffic congestion becoming increasingly severe. Consequently, there is an urgent need for an efficient intelligent transportation system to enhance road usage efficiency, optimize traffic flow, and meet the rising demand for mobility. In real traffic networks, excessive traffic flow may lead to congestion or accidental traffic accidents; therefore, accurate traffic flow forecasting can not only save travel time but also reduce the probability of occurring dangerous accidents.

Domestic and foreign scholars have conducted in-depth research on traffic flow data, fully considering external factors such as time, period, and weather, and have achieved a series of prominent research results in traffic flow prediction accuracy. We can categorize these methods into statistical prediction methods, machine learning-based prediction methods, neural network-based prediction methods, and graph convolution-based prediction methods. Among them, prediction methods based on statistics rely on the statistical analysis of historical traffic data, which makes it difficult to capture the complex dynamic characteristics of traffic flow data. Machine learning-based prediction methods predict future traffic flow by learning from historical traffic flow data using machine learning algorithms. As the

volume of traffic flow data samples continues to increase, the process of finding the optimal parameters may significantly reduce the efficiency and accuracy of the model learning. In recent years, with the continuous development of deep learning, prediction methods based on neural networks and graph convolution have overcome the shortcomings of the first two methods and can better handle the complexity and nonlinearity of traffic flow data.

Among neural network-based prediction methods, the most commonly used are convolutional neural networks (CNN) [1,2], recurrent neural networks (RNN) [3], and long short-term memory (LSTM) [4–6]. Although these methods have significant advantages in feature extraction and prediction accuracy, the process of training neural networks requires a significant amount of computational resources and storage space, presenting numerous issues; hence, neural network-based prediction methods face certain difficulties.

Urban traffic networks are highly complex. Although we can convert data into grid image data, we can only capture local spatial features within the network. Capturing the global spatial and temporal features of the network is still a significant challenge [7–10]. To effectively capture the spatio-temporal features within the transportation network, researchers have combined graph convolutional networks with recurrent network models in their predictive methods to capture spatio-temporal features. Li et al. [11] considered the relationship of spatio-temporal features and proposed the STGCN model, which captures the road network's spatio-temporal features by stacking spatio-temporal convolutional blocks using graph convolution and gated linear units. Based on the STGCN model, Guo et al. [12] introduced the ASTGCN model incorporating spatio-temporal attention mechanisms to capture features at different levels. The adaptive graph convolutional recurrent network (AGCR) is different from the models previously discussed [13]. It employs an innovative data-adaptive graph construction mechanism that dynamically generates graphs based on the inherent structure and characteristics of the data. Li et al. [14] proposed the DGCRN model, which integrates dynamic features of (velocity and timestamps) to construct a dynamic adjacency matrix. There are also many researchers [15–19] who combine graph neural networks with other models to construct spatio-temporal models for traffic flow forecasting.

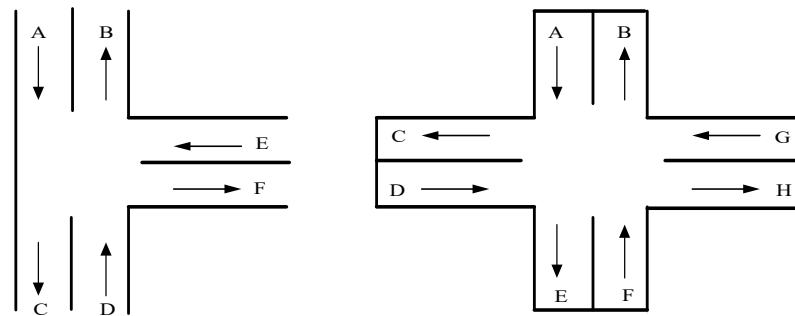
Although these methods have achieved commendable research outcomes, the complexity and long-term spatio-temporal dependencies within traffic networks make spatio-temporal modeling of traffic flow more challenging. Based on the predictive outcomes of previous researchers, models based on graph convolutional networks have significantly enhanced forecasting accuracy. However, models based on graph structures still have some issues: (1) most graph-structure-based models are highly dependent on the node-to-node graph structure, and the static graph structure generated by the physical connections of the road network cannot explore the potential dynamic spatial correlations in traffic flow data or adaptively establish nonlinear temporal correlations; (2) simple graph structures only describe the pairwise interactions between entities and are incapable of depicting higher-order relationships among multiple entities. In real traffic networks, the traffic flow on a road segment is not only affected by the directly connected road segments but may also be influenced by other indirectly connected road segments.

For instance, Figure 1 illustrates two traffic network structures, each reflecting the complex relationships between multiple nodes of traffic flow input and output. Simple graph structures are incapable of effectively representing these complex relationships.

The major contributions are as follows:

- (1) Proposed a new dynamic spatio-temporal hypergraph convolutional network (DSTHGCN) framework for traffic flow prediction, which effectively captures more accurate traffic flow information by combining traffic flow graphs and hypergraphs through collaborative convolution.

- (2) Unlike most traffic flow prediction models, this paper proposes a feature extraction module that can extract dynamic features of nodes and edges separately. These features are used to update the hyperedges and graph node information in the hypergraph, thereby revealing more complex underlying relationships in the dynamic traffic system.
- (3) Introduced a hyperedge outlier removal mechanism to identify and remove outliers in the hyperedges, thereby optimizing the hypergraph structure and better capturing the higher-order relationships within the data.



**Figure 1.** Schematic diagram of two types of intersection structures in a traffic network.

## 2. Related Work

In recent research, traffic flow prediction models developed using graph neural network technology have shown a rapid pace of development. These models have transitioned from traditional methods such as time series analysis and regression models to more sophisticated deep learning approaches, and their predictive accuracy has greatly improved. For instance, [12] simulates the complex spatio-temporal correlations between different locations through spatial attention. Ref. [20] proposes a novel learning component for graph structures that optimally obtains the graph's adjacency matrix from a micro and macro viewpoint and enhances the model's predictive performance through a two-stage training process. Ref. [21] has developed an innovative graph learning module to capture the hidden relationships between variables. Furthermore, the study introduces a comprehensive framework that not only models multivariate time series data effectively but also learns and infers the graphical structure of the data. Ref. [22] proposed the TSGDC model that combines the graph neural networks and transformers into the spatio-temporal graph diffusion convolutional network, which can better understand and utilize the complex spatio-temporal relationships in traffic data.

Although these graph-based models have achieved good performance, they have neglected the interactions between two or more nodes in practical situations and cannot model the higher-order relationships among nodes. To break through the limitations of traditional graph structures, more and more scholars have begun to consider hypergraph structures [23].

In addition, hypergraph convolutional networks are increasingly becoming a research hotspot due to their unique advantages in handling complex higher-order interaction relationships. These networks can more naturally express and learn multi-element relationships within data, demonstrating strong application potential across various fields. As research progresses, the design of models based on hypergraph convolutional networks continues to innovate and develop. For instance, [24] proposes a spatio-temporal hyperrelation to fully model complex local spatio-temporal relationships. Ref. [25] incorporates hypergraph convolution and attention mechanisms as end-to-end trainable operators in graph neural networks to acquire in-depth embeddings of higher-order graph-structured data. Ref. [26] explores hidden hyperedges by employing a local hypergraph attention mechanism across different time spans and then optimizes the hypergraph using a global attention mechanism. In [27], a novel high-order multi-modal and multi-type data associa-

tion modeling framework called HGNN+ is introduced. This framework is designed to capture and learn the optimal representation of data through a single hypergraph structure, thereby enhancing the efficiency and accuracy of data processing and analysis. Ref. [28] proposes a multi-level graph neural network capable of handling hypergraphs and paired graphs. Ref. [29] proposed a new multi-task hypergraph learning framework, which is composed of a primary task and an associated task. The latent features of hidden layers are shared between the tasks through a feature compression unit during training. In recent years, research on hypergraph convolutional networks has revealed some new trends. For instance, [30] proposed a unified method to capture local correlations and cross-network isomorphism by using the K-means clustering algorithm and the connectivity characteristics of the physical road network. Ref. [31] proposed a new network structure that can handle both temporal and spatial information, introducing a temporal hypergraph to capture action dependencies in time series. Ref. [32] proposed a new network structure that can effectively model dynamic higher-order relationships between nodes across multiple time scales, thereby enhancing the model's predictive capabilities. Ref. [33] proposed a new multi-task spatio-temporal network model MT-STNet, which adopts an encoder-decoder structure, builds the encoder and decoder through spatio-temporal blocks, and integrates information about the physical structure into model of the spatio-temporal dependencies of the highway network. Ref. [34] proposed the Res-HGCN model, which combines residual blocks with hypergraph convolutional networks for a multi-sensor data-driven fault diagnosis method.

In the existing hypergraph model structure, this paper proposes a new hypergraph network structure that fully considers the spatio-temporal features of transportation networks. We propose to extract more comprehensive spatio-temporal feature information using co-convolution of traffic flow graphs and hypergraphs. Additionally, we introduce a hyperedge outlier removal mechanism to optimize the hypergraph structure, thereby enhancing the model's predictive performance.

### 3. Methodology

#### 3.1. Preliminaries

In the traffic flow prediction task, historical data are analyzed, and a prediction model is constructed to predict the traffic flow data of a specific road section for a future period of time, including time series analysis and spatial distribution characteristics of the traffic flow. In this section, we define the graph, hypergraph, GCN, and HGCN related to the predictive model.

1. Graph: The traffic network  $G$  is defined as  $G = (V, E, A)$ , where  $V$  ( $|V| = N$ ) and  $E$  correspond to the number of nodes (the number of sensors) and the set of edges in the traffic network, respectively.  $A \in R^{N \times N}$  denotes the adjacency matrix of the graph, indicating the proximity between two nodes. The left hand side half of Figure 2 shows a simple graph converted into the adjacency matrix, where the elements of the matrix indicate whether there is an edge between the nodes in the graph. If there is an edge between node  $i$  and  $j$ , then the element in the  $i$ -th row and  $j$ -th column of the matrix is 1; otherwise, it is 0. Given the adjacency matrix and historical information over  $T'$  time steps, learning a function  $f$  that uses historical data from  $T'$  time steps to predict traffic information for the next  $T$  time steps:

$$[X_{t+1}, X_{t+2}, \dots, X_{t+T}] = f[A, (X_{t-T'+1}, X_{t-T'+2}, \dots, X)]. \quad (1)$$

2. GCN: This is a graph-based convolution that captures the interrelationships between nodes through graph convolution operations, thereby updating node features. We can describe the convolution process as follows:

$$GCN(X) = \sigma(\hat{A}X\Theta). \tag{2}$$

where  $\hat{A} \in R^{N \times N}$  refers to the adjacency matrix after normalization,  $\Theta$  represents the learnable parameters, and  $\sigma(\cdot)$  denotes the nonlinear activation function.

- Hypergraph: The hypergraph is defined as  $G_h = (V_h, E_h, H)$ , where  $V_h$  and  $E_h$  denote the node set and the set of nodes within hyperedges, respectively;  $H \in R^{E \times N}$  is the incidence matrix of  $\epsilon \in E_h$  and  $v_i \in V_h$ . When  $v_i$  is associated with  $\epsilon$ ,  $H_{i\epsilon} = 1$ ; otherwise, it is 0. Then, the degrees of nodes and hyperedges are represented by  $D_v \in R^{N \times N}$  and  $D_e \in R^{E \times E}$ , respectively. The diagonal matrix  $W$  corresponds to hyperedge weights. The right hand side half of Figure 2 shows a hypergraph converted into a matrix. The rows of the matrix represent nodes, and the columns represent edges. If a node is connected to an edge, the corresponding element is 1; otherwise, it is 0. The specific representation is shown as follows:

$$D_v = \sum_{\epsilon=1}^E W_{\epsilon\epsilon} H_{i\epsilon}, \tag{3}$$

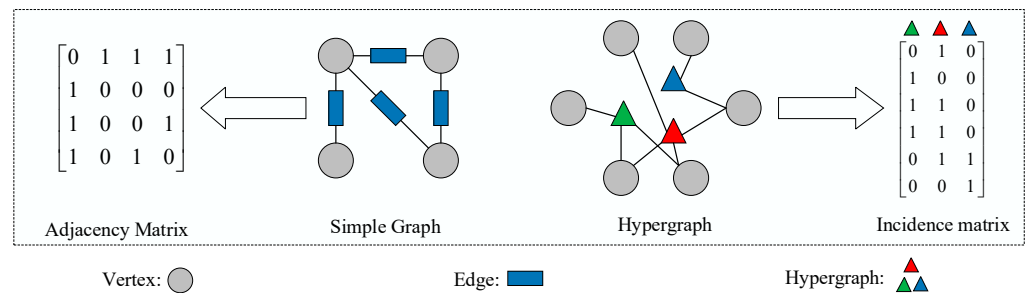
$$D_e = \sum_{i=1}^N H_{i\epsilon}. \tag{4}$$

- HGCN: The convolution process of a hypergraph is defined as follows:

$$HGCN(X_h) = \sigma(HWH^T X_h\Theta). \tag{5}$$

As it lacks a constrained spectral radius, we have normalized it, and the reformulation is as follows:

$$HGCN(X_h) = \sigma(D_v^{-1}HWD_E^{-1}H^T X_h\Theta). \tag{6}$$



**Figure 2.** Graph structure and hypergraph structure.

### 3.2. Framework of DSTHGCN

The structure of the model is shown in Figure 3. It consists of three main modules: the input layer, output layer, and dynamic spatio-temporal convolutional layer. The input part transforms traffic data into a high-dimensional representation. The output part performs skip connection operations for each output result. The most important dynamic spatio-temporal convolutional layer consists of three parts: dynamic feature extraction, dynamic hypergraph convolution (DHGCN), and dynamic graph convolution (DGCN). The following sections will provide a comprehensive discussion of the content related to these three parts.

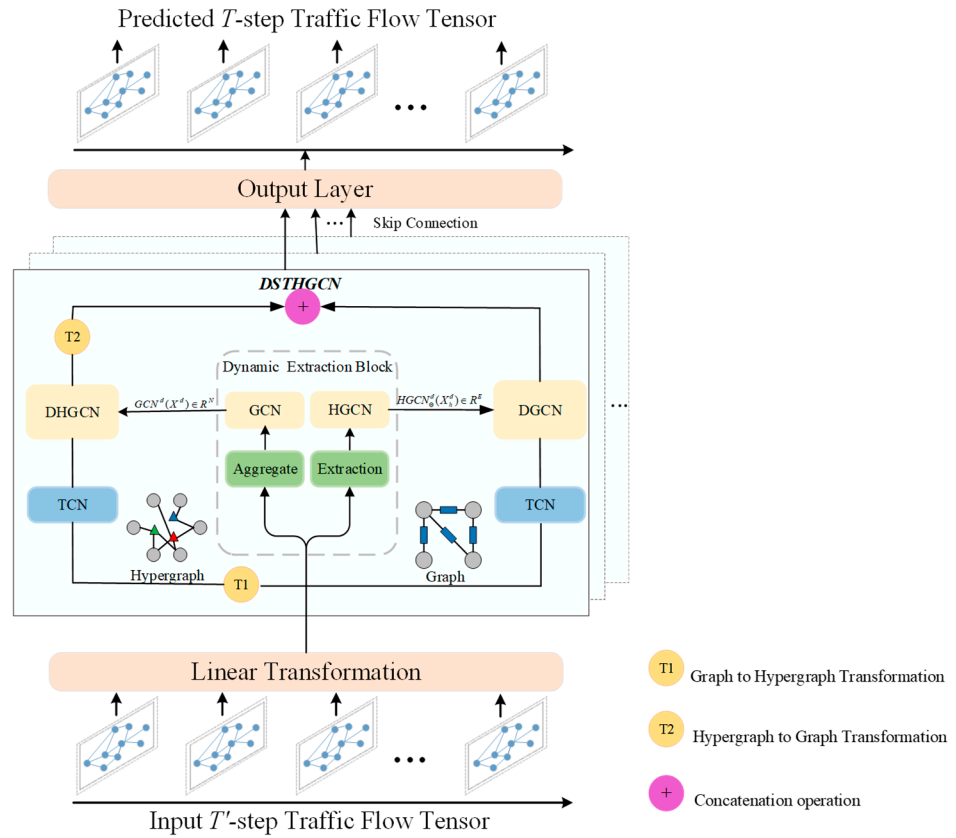


Figure 3. The overall framework of the model.

### 3.2.1. Dynamic Feature Extraction

In order to capture more complex and dynamic correlations of traffic data, this paper designs a dynamic feature extraction module to extract the dynamic features of the graph nodes of traffic flow as inputs for the DHGCN, thereby updating the corresponding edges of the hypergraph. Similarly, by extracting the edge features in the hypergraph as inputs for the DGCN, the node features of the traffic flow graph are updated. Currently, modules similar to feature extraction rarely appear in other models. It has the function of fusing information between the traffic flow graph and the hypergraph, thereby enabling the model to achieve more effective prediction results.

#### (1) Extraction of dynamic features of traffic flow graph nodes

First, the aggregate operation is used to aggregate node features along the temporal dimension, followed by the use of diffusion GCN to capture spatial correlations, with specific operations as shown below:

$$X^d = \text{Aggregate}(\mathcal{X}) \in R^{N \times F}, \tag{7}$$

$$GCN^d(X^d) = \sigma\left(\sum_{n=0}^N P_f^n X^d \Theta_{n,f} + P_b^n X^d \Theta_{n,b}\right), \tag{8}$$

where  $\mathcal{X}$  represents the node features of the traffic flow graph;  $\sigma(\cdot)$  denotes a nonlinear activation function like ReLU;  $P_f = A / \text{rowsum}(A)$  and  $P_b = A^T / \text{rowsum}(A^T)$  represent the forward and backward transition matrices of the directed graph, respectively;  $P$  and  $N$  represent the state transition matrix and diffusion coefficient, respectively; and  $\Theta$  signifies learnable parameters. The output of  $GCN^d(X^d)$  will be used as the input for DHGCN.

(2) Extraction of dynamic features of the graph edges of traffic flow

First, the extraction operation is used to obtain node-related information of the directed edges in the traffic flow graph as initial edge features; then, convolutional operations are used to update the edge features.

$$X_h = [(W'_1 \mathcal{X})[ind_{src}, :]; (W'_2 \mathcal{X})[ind_{dst}, :]] \in R^{E \times 2F''}, \tag{9}$$

$$X_h^d = Conv_{1 \times 1}(X_h), \tag{10}$$

where  $W'_1, W'_2$  represent trainable parameters;  $(\cdot)[ind, :]$  is a tensor composed of slices of  $ind_{src}$  and  $ind_{dst}$ , selected from the source and target nodes of the directed edges; and  $[\cdot, \cdot]$  performs the concatenation operation.

The process of implementation of dynamic edge features is as follows:

$$W_h = D_v^{-1} H W_{adp} D_e^{-1} H^T \in R^{E \times E}, \tag{11}$$

$$HGNCN_{\Theta}^d(X_h^d) = \sigma\left(\sum_{n=0}^{N-1} W_h^n X_h^d \Theta_n\right) \in R^E, \tag{12}$$

where  $W_{adp}$  represents the diagonal matrix of the adaptively learned hyperedge weight vectors. The output of  $HGNCN_{\Theta}^d(X_h^d)$  will be used as the input for DGCN.

### 3.2.2. Dynamic Graph Convolution

First, the temporal convolutional network (TCN) combines one-dimensional causal convolution and dilated convolution to extract temporal correlations within traffic data along the time dimension.

$$F(\mathcal{X}) = \tanh(\Theta_1 * \mathcal{X} + b) \odot \text{sigmoid}(\Theta_1 * \mathcal{X} + c) \tag{13}$$

where  $\Theta_1, \Theta_2, b$  and  $c$  are trainable parameters;  $\odot$  denotes the product of elements;  $\tanh(\cdot)$  is an activation function acting as a filter; and  $\text{sigmoid}(\cdot)$  is an activation function that controls the ratio of information transfer.

Subsequently, the proposed DGCN is used to capture the spatio-temporal features of the traffic data, with the output of  $HGNCN_{\Theta}^d(X_h^d)$  serving as the input. The specific implementation process is shown in Equation (14):

$$DGCN(\mathcal{X}) = \sigma\left(\sum_{n=0}^{N-1} (P_f \odot D_f)^n \mathcal{X} \Theta'_{n,f} + (P_b \odot D_b)^n \mathcal{X} \Theta'_{n,b} + A_{adp}^n \mathcal{X} \Theta'_{n,adp}\right) \tag{14}$$

where  $D_f = \text{reshape}(HGNCN_{\Theta_f}^d(X_h^d))$  and  $D_b = \text{reshape}(HGNCN_{\Theta_b}^d(X_h^d))^T$  utilize  $\text{reshape}(\cdot)$  to reshape  $HGNCN_{\Theta}^d(X_h^d)$  into a sparse matrix;  $\Theta'_f, \Theta'_b$ , and  $\Theta'_{adp}$  are all trainable parameters.

### 3.2.3. Dynamic Hypergraph Convolution

Dynamic hypergraph convolution utilizes the TCN from (13) to extract the features along the temporal dimension of traffic flow data and then employs DHGCN to capture the spatio-temporal features.

$$W_{hd} = D_v^{-1} H D_w D_e^{-1} H^T \in R^{E \times E}, \tag{15}$$

$$DHGCN(\mathcal{X}_h) = \sigma\left(\sum_{n=0}^{N-1} W_{hd}^n \mathcal{X}_h \Theta'_n\right), \tag{16}$$

where  $D_w$  represents the diagonalization of output results of  $GNCN^d(X^d)$ , and  $\Theta'$  is a trainable parameter.



Finally, the node features of the traffic flow graph from (14) are connected with the hypernode features of the hypergraph from (16) to obtain the output of the spatio-temporal convolutional layer. DSTHGCN can be summarized as Algorithm 1.

---

**Algorithm 1:** Training Algorithm for DSTHGCN.

---

**Input:** Node features of the traffic flow graph  $\mathcal{X}$ ;

**Output:** New node features  $\mathcal{X}_{new}$  generated by the collaborative convolutions on the traffic flow graph and its dual hypergraph;

1: The dynamic features  $X^d \leftarrow GCN^d(X^d)$  of the nodes are obtained through Equation (8);

2: The dynamic edge features  $X_h^d \leftarrow HGCN_\Theta^d(X_h^d)$  are obtained through Equation (12);

3: Update  $\mathcal{X} \leftarrow F(\mathcal{X})$  by Equation (13);

4: Update  $\mathcal{X} \leftarrow DGCN(\mathcal{X})$  by Equation (14);

5: Transform the node features of the traffic flow graph into the hyperedge features of the hypergraph;

6: Update  $\mathcal{X}_h \leftarrow F(\mathcal{X})$  by Equation (13);

7: Update  $\mathcal{X}_h \leftarrow DHGCN(\mathcal{X}_h)$  by Equation (16);

8: Map the hyperedge features from Equation (16) back to the node features of the traffic flow graph to obtain  $\mathcal{X}'$ ;

9: Finally, a concatenation operation is performed to obtain  $\mathcal{X}_{new} = [\mathcal{X}; \mathcal{X}']$ ;

---

### 3.2.4. Hyperedge Outlier Removal Mechanism

Hypergraph theory offers an innovative approach to modeling traffic flow, effectively capturing complex higher-order relationships in real datasets by transferring information from nodes to hyperedges and then feedback from hyperedges to nodes. In this model, each hyperedge can connect multiple nodes, thereby containing richer traffic flow information, which allows the model to more accurately reflect the dynamic characteristics of the traffic network. However, in practical applications, outlier nodes in traffic networks are inevitable. These outlier nodes may have a negative impact on the model's training process, thereby causing deviations in the prediction results. To address this issue, we propose a mechanism that removes outliers on hyperedges during the process of information propagation between nodes and hyperedges. The core of this mechanism lies in dynamically adjusting the structure of the hypergraph to reduce the impact of outlier nodes on model performance. Specifically, the mechanism identifies and removes outlier nodes that have a negative impact on model training and prediction performance by learning embeddings. This dynamic adjustment strategy effectively removes outliers and enhances the model's ability to handle noise and irrelevant information.

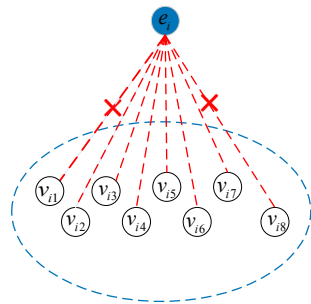
As shown in Figure 4,  $e_i$  and  $v_{ij}$  represent hyperedges and nodes, respectively. We believe that important nodes should have a high degree of similarity with their associated hyperedges. Therefore, this paper uses cosine similarity (Equation (17)) to calculate the similarity between each node  $v_{ij}$  and its corresponding hyperedge  $e_i$ . Then, based on the value of similarity to identify outliers, we sort by the magnitude of the similarity value and consider the bottom 10% of nodes as outliers, removing these outlier nodes from the hypergraph. As shown in Figure 4, if nodes  $v_{i1}$  and  $v_{i8}$  have low similarity with hyperedge  $e_i$ , and their similarity ranking is in the bottom 10% of all nodes, then these two nodes will be recognized as outliers. Subsequently, these outlier nodes will be removed from hyperedge  $e_i$ . This mechanism helps to optimize the structure of the hypergraph because it reduces the complexity of the model by removing outlier nodes, thereby improving the model's efficiency and accuracy.

$$\cos_{ij} = \frac{x_i \cdot y_i}{|x_i| \cdot |y_i|}, \quad (17)$$

where  $x_i$  is the representation of node  $v_{ij}$ ;  $y_i$  is the representation of hyperedge  $e_i$ ; and  $|\cdot|$  refers to the norm operation on vectors. Typically, we assess the influence of a node on a hyperedge based on the similarity between the node and hyperedge features; the



higher the similarity, the greater the influence; and vice versa: the lower the similarity, the less influence.



**Figure 4.** Hyperedge outlier mechanism.

### 3.2.5. Evaluation Metrics

In this study, we chose the Huber loss function as the primary loss function, with the aim to enhance the accuracy and reliability of model predictions. The Huber loss function is a very effective statistical tool. In comparison to the commonly used the mean squared error (MSE) loss function, the Huber loss function can more robustly handle outliers in the data. We assume that  $y$  and  $\hat{y}$  represent the true values and predicted values, respectively. The Huber loss function is defined as follows:

$$\mathcal{L}_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{for } |y - \hat{y}| \leq \delta \\ \delta |y - \hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (18)$$

The mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) will be used for the overall evaluation of DSTHGCN and the baselines. The optimal values for these three metrics are obtained after running 200 epochs on the validation data.

$$MAE = \frac{1}{|\Omega|} \sum_{i \in \Omega} |y_i - \hat{y}_i| \quad (19)$$

$$RMSE = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} (y_i - \hat{y}_i)^2} \quad (20)$$

$$MAPE = \frac{1}{|\Omega|} \sum_{i \in \Omega} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (21)$$

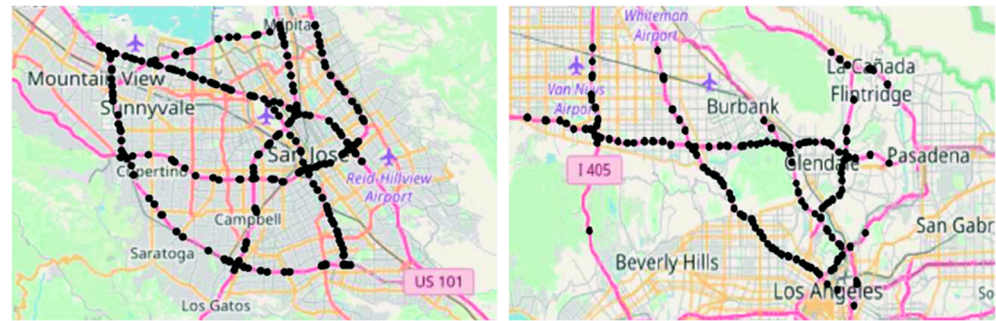
## 4. Experiments

### 4.1. Datasets

To comprehensively validate the performance and accuracy of the model, we employed a multidimensional evaluation approach. We selected two widely recognized real-world traffic datasets, PeMSD4 [4] and PeMSD8 [35], to conduct experiments on the model proposed in this paper and the baseline models. These datasets include information on traffic flow, velocity, and occupancy rates. This paper mainly focuses on the study of traffic flow [13,36]. The PeMSD4 dataset, commencing on 1 January 2018, consists of traffic flow data collected continuously over 59 days by 307 sensors, with data gathered every five minutes. The PeMSD8 dataset, starting from 1 July 2016, comprises traffic flow data collected continuously over 62 days by 170 sensors, with data being gathered every five minutes. More detailed information about these two datasets is shown in Table 1. To gain a deeper understanding of the geographical context of the data and the interrelationships among the data, Figure 5 illustrates the sensor distribution maps of the PeMSD4 and PeMSD8 datasets. These maps not only intuitively display the specific geographic locations where the data are collected but also delineate the layout of the traffic network in detail.

**Table 1.** Summary statistics for PeMSD4 and PeMSD8.

Datasets	#Nodes	Edges	#TimeSteps	Time Interval	Time Range
PeMSD4	307	340	16,992	5 min	1/1/2018–2/28/2018
PeMSD8	170	295	17,856	5 min	7/1/2016–8/31/2016

**Figure 5.** PeMSD4 and PeMSD8 datasets sensor distribution maps.

#### 4.2. Experimental Setups

This paper divides the entire dataset into three parts based on chronological order: training set, validation set, and test set. The ratio of this division is 6:2:2, with the training set accounting for 60% and the validation set and test set each accounting for 20%. The traffic flow prediction task takes historical traffic flow information from 12 time steps as input to forecast the traffic flow for the next 12 time steps, with each time step being five minutes long. This experiment was conducted on NVIDIA GeForce RTX 3090 GPU using the Adam optimizer for training. The initial learning rate is set to 0.001, the batch size is 16, the number of stacked layers in the model is  $B = 2$ , and the number of filters is  $F = 40$ .

#### 4.3. Baselines

We compared the method proposed in this paper with the following six traffic flow prediction methods. These baseline models are either widely used or are new models proposed in the last two years. A brief description of each baseline model is provided as follows:

- (1) HA: The historical average method uses the mean traffic flow from the same time point over the past few days or weeks as the predicted value.
- (2) ARIMA: Autoregressive integrated moving average model, a classic method for time series forecasting.
- (3) FC-LSTM: This model is a combination of fully connected layers and Long Short-Term Memory networks, often used for time series problems.
- (4) ASTGCN (r) [12]: This model captures the dynamic spatio-temporal features of traffic flow data using a spatio-temporal attention mechanism while considering the periodicity of the spatio-temporal network.
- (5) STGODE [37]: This model utilizes ordinary differential equations on graphs to model the dynamics of spatio-temporal data.
- (6) Graph WaveNet [38]: It combines graph convolution with causal convolutional networks to capture the spatio-temporal dependencies in the data.
- (7) DSTAGNN [39]: This model employs an improved multi-head attention mechanism to capture the dynamic spatial dependencies between nodes.

#### 4.4. Experimental Results and Analysis

We compared our model with the seven abovementioned baseline models on two real-life datasets. To ensure the fairness of the experiment, the parameters of each baseline were adjusted.

As can be seen from Table 2, the model we proposed significantly outperforms the existing baseline models in terms of predictive accuracy. DSTHGCN outperforms all seven baseline models in all three-evaluation metrics. On the PeMSD4 dataset, DSTHGCN improved the evaluation metrics of MAE, RMSE, and MAPE by 10.6%, 6.8%, and 9.74% respectively, with respect to the most comparable baseline model STGODE. On the PeMSD8 dataset, the evaluation metrics of MAE, RMSE, and MAPE improved by 11.26%, 7.98%, and 14.19%, respectively, with respect to the baseline model STGODE.

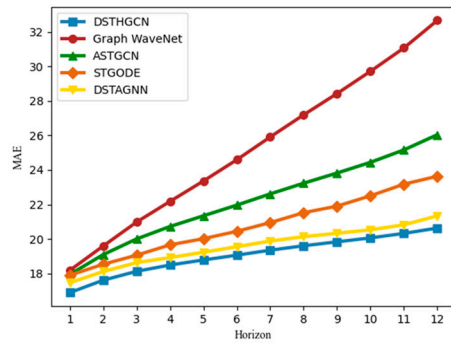
**Table 2.** Analysis of prediction performance of traffic forecasting models on different datasets.

Models	PeMSD4			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	38.03	59.24	27.88	34.86	59.24	27.88
ARIMA	33.73	48.80	24.18	31.09	44.32	22.73
FC-LSTM	26.77	40.65	18.23	23.09	35.17	14.99
ASTGCN(r)	22.20	34.69	15.0	17.31	26.93	11.0
Graph WaveNet	25.32	39.04	17.51	19.16	30.59	12.24
STGODE	20.77	32.50	14.06	16.78	25.80	11.27
DSTAGNN	19.58	31.91	13.28	15.75	25.08	10.54
DSTHGCN	18.95	30.29	12.69	14.89	23.74	9.67

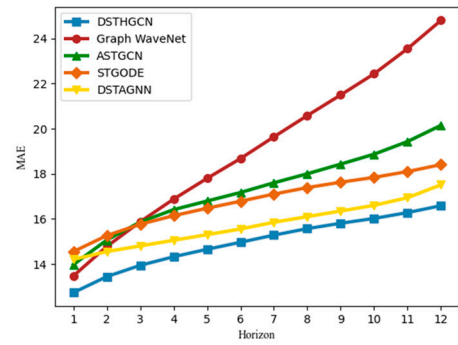
Table 3 provides an in-depth analysis of the predictive performance of the five models, such as ASTGCN, Graph WaveNet, STGODE, DSTAGNN, and DSTHGCN at different time steps (3, 6, and 12 time steps). Through this analysis, we can gain a more comprehensive understanding of the capabilities and limitations of these models in handling prediction tasks across various time spans. It can be observed that the longer time of prediction, the greater the forecasting error. We can see from Table 3 that the DSTHGCN model outperforms the other five baseline models at every forecasting horizon. To more intuitively see the outstanding forecasting level of the models, a line chart is used to display the forecasting level of each time step for these five models. As shown in Figure 6, the DSTHGCN model performs better on all metrics (MAE, RMSE, MAPE), and its error growth trend is slower with the increase in time steps. The DSTAGNN and STGODE models perform second only to DSTHGCN, while ASTGCN and Graph WaveNet models perform poorly, especially the Graph WaveNet model, mainly because as time progresses, the model needs to consider more variables and uncertainties, leading to increased complexity in forecasting.

**Table 3.** Comparison predictive performance of traffic forecasting models at 15 min, 30 min, and 60 min horizons (3, 6, and 12 time steps).

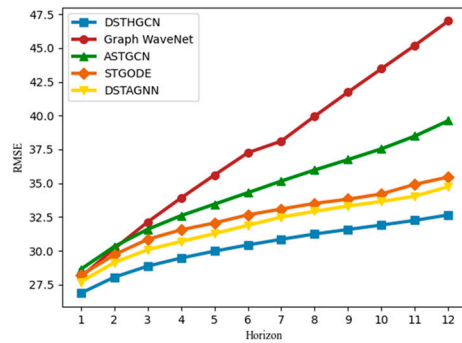
Datasets	Models	15 Min			30 Min			60 Min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PeMSD4	ASTGCN (r)	20.00	31.59	13.0	21.96	34.3	15.0	26.04	39.63	18.0
	Graph WaveNet	20.97	32.92	14.67	24.58	38.11	16.67	32.66	49.16	23.09
	STGODE	18.89	30.85	13.31	20.44	32.62	14.04	23.63	35.42	15.32
	DSTAGNN	18.63	30.08	12.64	19.55	31.89	13.21	21.34	34.75	14.41
	DSTHGCN (ours)	18.12	29.11	12.12	19.05	30.53	12.68	20.63	32.72	13.69
PeMSD8	ASTGCN (r)	15.86	24.69	10.0	17.17	26.75	11.0	20.15	30.84	12.0
	Graph WaveNet	15.87	25.34	9.98	18.67	30.10	11.73	24.80	39.09	16.21
	STGODE	15.75	23.94	10.24	16.78	25.87	11.19	18.4	28.64	12.64
	DSTAGNN	14.8	23.73	9.77	15.55	24.98	10.46	17.75	27.38	11.61
	DSTHGCN (ours)	14.07	22.09	9.04	15.02	23.87	9.63	16.70	26.42	10.69



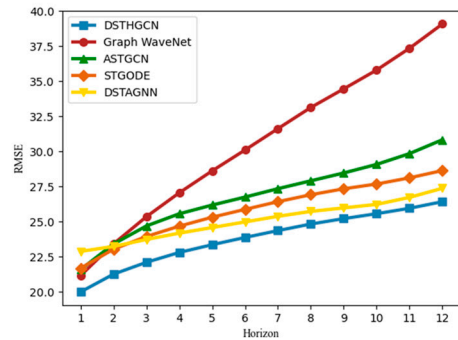
(a) MAE on PeMSD4



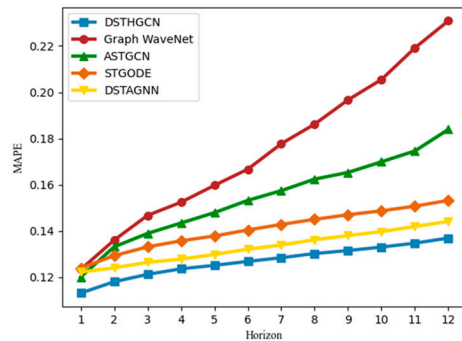
(b) MAE on PeMSD8



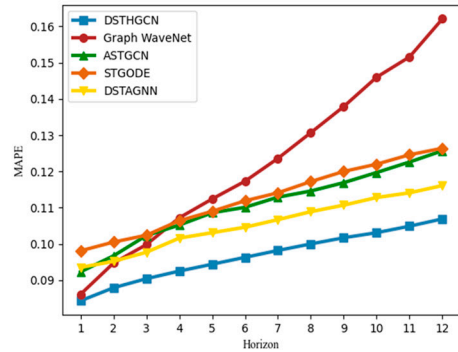
(c) RMSE on PeMSD4



(d) RMSE on PeMSD8



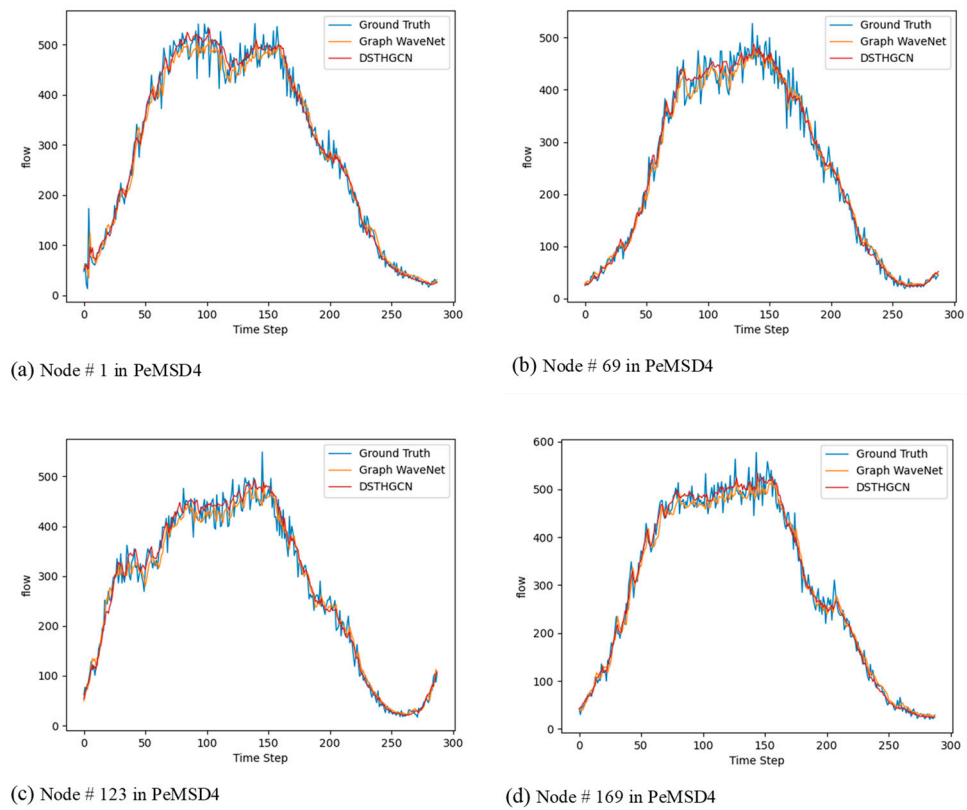
(e) MAPE on PeMSD4



(f) MAPE on PeMSD8

**Figure 6.** Performance comparison of different forecasting models across multiple datasets at various time steps.

To conduct an in-depth analysis of the model’s performance, we selected four nodes from the PeMSD4 dataset and visualized the traffic flow data for these nodes over a day. We also conducted a comparative analysis with ground truth and Graph WaveNet, and the comparative curve chart is shown in Figure 7. The forecast curves for the DSTHGCN model on all four nodes are very close to the actual data, demonstrating its high accuracy in traffic flow prediction. The forecast curves of the Graph WaveNet model are close to the actual data for most time steps, but there are some deviations at key points (such as traffic flow peaks). Overall, the DSTHGCN model outperforms the Graph WaveNet model in terms of prediction performance on these nodes.



**Figure 7.** Comparison of predictive curves for DSTHGCN with ground truth and Graph WaveNet on PeMSD4 at Nodes #1, #69, #123, and #196.

4.5. Ablation Study

The experimental results mentioned above demonstrate that the DSTHGCN model outperforms most baseline models across various forecasting levels, with this significant advantage attributed to the innovative integration of hypergraph convolution technology and hyperedge outlier processing mechanisms. Therefore, this section focuses on verifying whether these two modules have a positive effect on the final prediction results. We have developed two variants, DSTHGCN-1 and DSTHGCN-2, and conducted ablation studies on these variants to explore the contributions of their different components to the overall forecasting capability. Under the same datasets and evaluation metrics, the ablation study results of the DSTHGCN model are shown in Table 4:

**Table 4.** Results of ablation studies on PeMSD4 and PeMSD8.

Models	PeMSD4			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DSTHGCN-1	19.26	30.70	13.28	15.40	24.26	10.53
DSTHGCN-2	19.08	30.60	13.23	15.26	24.16	10.45
DSTHGCN	18.95	30.29	12.69	14.89	23.74	9.67

In DSTHGCN-1, the dynamic hypergraph convolution module is removed. Then, TCN is combined with DGCN to capture the complex relationships within the traffic network. It is evident that the accuracy of DSTHGCN-1, which lacks the hypergraph convolutional module, significantly decreased. This is because hypergraphs are an extension of traditional graphs, capable of representing non-pairwise relationships between nodes, thereby simulating the intrinsic connections between high-order data and capturing more complex traffic flow information, which, in turn, enhances the prediction accuracy. In DSTHGCN-2,

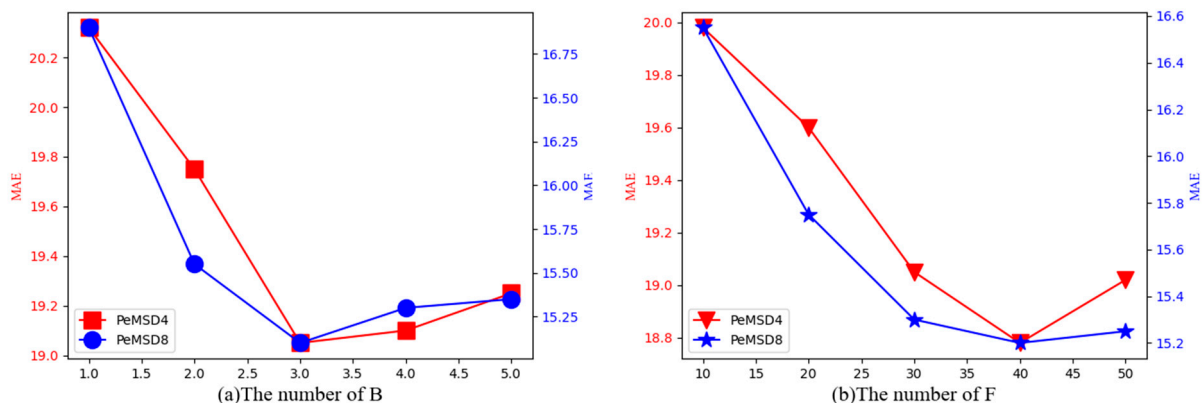


the hyperedge outlier removal mechanism is removed, and the dynamic graph convolution and dynamic hypergraph convolution modules work in tandem to capture node features through a cooperative convolution approach. As shown in Table 4, the accuracy of DSTHGCN-2, which lacks the hyperedge outlier removal mechanism, also decreased. This is because although hypergraphs contain abundant traffic flow information, there may be outliers in these data that could negatively impact the model's predictive performance. The mechanism of hyperedge outlier removal is capable of identifying and removing these outliers, optimizing the hypergraph structure, and enhancing prediction accuracy.

#### 4.6. Hyperparameter Sensitivity Analysis

Parameters are very important for the performance of the model. By analyzing the model under different hyperparameter settings, it can help us better understand the model's sensitivity to different parameters and how these hyperparameters affect the model's learning process and final performance. Therefore, this section analyzes different values of two important hyperparameters, B and F.

**B:** In graph convolutional networks, the number of stacked layers of the model is crucial for capturing complex relationships between nodes. In this paper, we explored the range of values for B, from 1 to 5, and incrementally increased the number of layers in experiments to determine the value of B that optimizes model performance. As shown in Figure 8a, when  $B = 3$ , the model's MAE metric achieved the best performance on both datasets. When B is greater than 3, the excessive number of stacked layers causes a waste of computational resources, so the performance decreases. Therefore, the optimal number of stacked layers is  $B = 3$ .



**Figure 8.** Hyperparameter study on PeMSD4 and PeMSD8.

**F:** As for the number of filters, it has a significant impact on the effectiveness of feature extraction. In experiments, we start with an initial value of F at 8 and incrementally increase its value, doubling the previous value each time, to find the optimal number of filters F for enhancing the model's feature extraction capability. As shown in Figure 8b, when  $F = 40$ , the model's MAE metric achieved the best performance on both datasets. When F is greater than 40, the performance decreases due to the excessive number of filters leading to overfitting. Therefore, the optimal number of filters is  $F = 40$ .

#### 4.7. Computational Time Analysis

To better assess the complexity of the DSTHGCN model, Table 5 compares the training and inference times of the baseline ASTGCN, Graph WaveNet, STGODE, DSTAGNN, and DSTHGCN models.

Table 5 clearly shows that the STGODE and DSTACNN models have the shortest training and inference times on the PeMSD4 dataset. The DSTHGCN model has the second shortest training time after STGODE. On the PeMSD4 dataset, Graph WaveNet and DSTHGCN have the shortest training and inference times, respectively, while the inference



time of the DSTHGCN model is second only to Graph WaveNet. Due to the complexity of the DSTHGCN model, although it is not optimal overall, it is within an acceptable range.

**Table 5.** Comparison of training and inference times of various models on PeMSD4 and PeMSD8 datasets.

Datasets	Computation Time (Training (s/epoch)/Inference (s))				
	ASTGCN	Graph WaveNet	STGODE	DSTAGNN	DSTHGCN
PeMSD4	84.395/9.455	80.31/7.52	32.04/6.01	45.58/4.26	37.66/7.44
PeMSD8	45.58/4.26	24.61/1.67	27.69/5.13	28.51/3.19	21.58/2.43

From the table, we can also observe that an increase in dataset complexity leads to an extension of training and inference time, which implies that processing large-scale datasets in real-world applications may consume more computational resources, potentially limiting the feasibility of the model in practical applications. To overcome this challenge, we plan to introduce advanced attention mechanisms and combine them with optimized dynamic graph structures in future research. Attention mechanisms can help the model focus on key features in the data, thereby improving processing efficiency. At the same time, the optimization of dynamic graph structures can further reduce the computational burden on the model during training and inference. By employing these strategies, our goal is to achieve a reduction in the model's demand for computational resources while maintaining existing predictive accuracy.

## 5. Conclusions

This paper proposes a dynamic spatio-temporal hypergraph attention convolutional network that combines graph convolutional networks and hypergraph convolutional networks to form a predictive model capable of capturing more comprehensive spatio-temporal features in the task of traffic flow prediction. Furthermore, we have introduced a hyperedge outlier removal mechanism that can effectively eliminate abnormal nodes in the hypergraph, thereby optimizing the hypergraph structure and reducing the complexity of the model. To thoroughly assess the model's performance, we meticulously designed an extensive series of experiments that covered two representative real-world datasets, and the results strongly demonstrated the model's excellent capabilities.

At the same time, we also consider that the paper may have the following limitations: there may be data compatibility and integration issues in the process of integrating with existing traffic management systems, insufficient computational resources, and issues with the model's generalization ability. When dealing with large-scale data and needing to expand the prediction range, the model's response time and resource consumption requirements are higher. Therefore, to reduce the impact of these limitations on the model, our future research directions will include the following aspects:

(1) Develop advanced data fusion and preprocessing techniques to solve problems in data integration, ensuring data consistency and accuracy. (2) Incorporate attention mechanisms to help the model better focus on key time steps and spatial locations, thereby improving the accuracy of long-term predictions. (3) Explore methods to decompose long-term forecasting tasks into multiple short-term predictions to reduce the burden on individual forecasts. (4) Optimize the model structure, such as using shallower network layers or optimized graph convolution methods, which can not only reduce the demand for computational resources but also reduce the model's latency.

**Author Contributions:** Conceptualization, Z.Y. and K.P.; methodology, J.M.; software, H.W.; validation, Z.Y., K.P., and N.H.; formal analysis, H.W.; investigation, N.H.; resources, J.S.; data curation, J.S.; writing—original draft preparation, Z.Y., J.S., and K.P.; writing—review and editing, H.W. and J.M.; visualization, H.W.; supervision, N.H.; funding acquisition, J.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was financed as part of the Lublin University of Technology projects FD-24/IM-5/087 and FD-24/EE-2/801. This research was supported by the National Natural Science Foundation of China (Grant Nos. 62376089, 62302153, 62302154), the Key Research and Development Program of Hubei Province, China (Grant No. 2023BEB024), the Young and Middle-aged Scientific and Technological Innovation Team Plan in Higher Education Institutions in Hubei Province, China (Grant No. T2023007), and the National Natural Science Foundation of China (Grant No. U23A20318).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors would like to thank Orest Kochan for his assistance in preparing this paper.

**Conflicts of Interest:** Authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
2. Tian, H.; Su, J.; Kochan, O. Research on Traffic Flow Prediction Based on ISMA-CNN-GRU Model. In Proceedings of the COLINS, Kharkiv, Ukraine, 20–21 April 2023; Volume 1, pp. 40–50.
3. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
4. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
5. Yao, Y.; Ye, Z.; Bai, W.; Kochan, O.; Mokhun, S. Time Series Prediction Based on LSTM and Modified Hybrid Breeding Optimization Algorithm. In Proceedings of the 2023 13th International Conference on Advanced Computer Information Technologies (ACIT), Wrocław, Poland, 21–23 September 2023; pp. 584–590. [[CrossRef](#)]
6. Sun, L.; Qin, H.; Przystupa, K.; Majka, M.; Kochan, O. Individualized Short-Term Electric Load Forecasting Using Data-Driven Meta-Heuristic Method Based on LSTM Network. *Sensors* **2022**, *22*, 7900. [[CrossRef](#)] [[PubMed](#)]
7. 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](#)]
8. Zheng, C.; Fan, X.; Pan, S.; Jin, H.; Peng, Z.; Wu, Z.; Wang, C.; Yu, P.S. Spatio-temporal joint graph convolutional networks for traffic forecasting. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 372–385. [[CrossRef](#)]
9. Jiang, R.; Cai, Z.; Wang, Z.; Yang, C.; Fan, Z.; Chen, Q.; Tsubouchi, K.; Song, X.; Shibasaki, R. DeepCrowd: A deep model for large-scale citywide crowd density and flow prediction. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 276–290. [[CrossRef](#)]
10. Gu, J.; Zhou, Q.; Yang, J.; Liu, Y.; Zhuang, F.; Zhao, Y.; Xiong, H. Exploiting interpretable patterns for flow prediction in dockless bike sharing systems. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 640–652. [[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. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January 2019.
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, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; Li, Y. Dynamic graph convolutional recurrent network for traffic forecasting: Benchmark and solution. *ACM Trans. Knowl. Discov. Data* **2021**, *17*, 1–21.
15. Sun, X.; Cheng, H.; Liu, B.; Li, J.; Chen, H.; Xu, G.; Yin, H. Self-supervised hypergraph representation learning for sociological analysis. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 11860–11871. [[CrossRef](#)]
16. Liu, S.; Chen, H.; Chen, X.Y.; He, J.J. A dual-branch spatio-temporal graph convolutional neural network for traffic flow prediction. *Inf. Control.* **2022**, 1–14. [[CrossRef](#)]
17. Zeng, Y.C.; Shao, M.H.; Sun, L.J.; Lu, C. Traffic prediction and congestion control based on directed graph convolutional neural network. *China J. Highw. Transp.* **2021**, *34*, 239–248.
18. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3558–3565.
19. Pedronette, D.C.G.; Valem, L.P.; Almeida, J.; Torres, R.D.S. Multimedia retrieval through unsupervised hypergraph-based manifold ranking. *IEEE Trans. Image Process.* **2019**, *28*, 5824–5838. [[CrossRef](#)]
20. 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](#)]

21. 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.
22. Wei, S.; Yang, Y.; Liu, D.; Deng, K.; Wang, C. Transformer-Based Spatiotemporal Graph Diffusion Convolution Network for Traffic Flow Forecasting. *Electronics* **2024**, *13*, 3151. [[CrossRef](#)]
23. Naganand, Y.; Madhav, N.; Prateek, Y.; Vikram, N.; Anand, L.; Partha, T. HyperGCN: A new method of training graph convolutional networks on hypergraphs. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 1511–1522.
24. Zhang, Y.K.; Wu, Z.H.; Lin, Y.F.; Zhao, Y.J. Spatio-temporal hypergraph convolutional network for traffic flow prediction. *J. Comput. Appl.* **2021**, *41*, 3578–3584.
25. Bai, S.; Zhang, F.; Torr, P.H.S. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* **2021**, *110*, 107637. [[CrossRef](#)]
26. Wang, J.; Zhang, Y.; Hu, Y.; Yin, B. Metro flow prediction with hierarchical hypergraph attention networks. *IEEE Trans. Artif. Intell.* **2021**, *5*, 3012–3021. [[CrossRef](#)]
27. Gao, Y.; Feng, Y.; Ji, J.R. HGNN+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3181–3199. [[CrossRef](#)]
28. Chen, H.; Yin, H.; Sun, X.; Chen, T.; Gabrys, B.; Musial, K. Multi-level graph convolutional networks for cross-platform anchor link prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020.
29. Wang, J.; Zhang, Y.; Wang, L.; Hu, Y.; Piao, X.; Yin, B. Multitask Hypergraph Convolutional Networks: A Heterogeneous Traffic Prediction Framework. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18557–18567. [[CrossRef](#)]
30. Zhao, Z.Z.; Shen, G.J.; Zhou, J.J.; Jin, J.C.; Kong, X.J. Spatial-temporal hypergraph convolutional network for traffic forecasting. *PeerJ Comput. Sci.* **2023**, *9*, 341–345. [[CrossRef](#)] [[PubMed](#)]
31. Wang, S.; Zhang, Y.; Qi, H.; Zhao, M.; Jiang, Y. Dynamic spatial-temporal hypergraph convolutional network for skeleton-based action recognition. In Proceedings of the 2023 IEEE International Conference on Multimedia and Expo (ICME), Brisbane, Australia, 10–14 July 2023; pp. 2147–2152.
32. Dong, Z.; Yu, S.; Shen, Y. Multi-scale dynamic hypergraph convolutional network for traffic flow forecasting. *J. Shanghai Jiaotong Univ. (Sci.)* **2024**. [[CrossRef](#)]
33. Zou, G.; Lai, Z.; Wang, T.; Liu, Z.; Li, Y. MT-STNet: A novel multi-task spatio-temporal network for highway traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8221–8236. [[CrossRef](#)]
34. Xia, L.; Liang, Y.; Zheng, P.; Huang, X. Residual-hypergraph convolution network: A model-based and data-driven integrated approach for fault diagnosis in complex equipment. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–11. [[CrossRef](#)]
35. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 914–921.
36. Chen, Y.; Segovia-Dominguez, I.; Gel, Y.R. Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021.
37. Fang, Z.; Long, Q.; Song, G.; Xie, K. Spatial-temporal graph ODE networks for traffic flow forecasting. *arXiv* **2021**, arXiv:2106.12931.
38. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph WaveNet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
39. Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; Li, P. DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 11906–11917.

**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.