International Journal of
Geo-Information

MDPI

Article

STSGAN: Spatial-Temporal Global Semantic Graph Attention Convolution Networks for Urban Flow Prediction

Junwei Zhou, Xizhong Qin *, Kun Yu , Zhenhong Jia and Yan Du

College of Information Science and Engineering, Xinjiang University, Urumqi 830000, China;
zjw6390@stu.xju.edu.cn (J.Z.); ykun@stu.xju.edu.cn (K.Y.); jzhh@xju.edu.cn (Z.J.);
15299182353dy@stu.xju.edu.cn (Y.D.)
* Correspondence: qmqqxz@163.com

Abstract: Accurate urban traffic flow prediction plays a vital role in Intelligent Transportation System (ITS). The complex long-term and long-range spatiotemporal correlations of traffic flow pose a significant challenge to the prediction task. Most current research methods focus only on spatial correlations in local areas, ignoring global geographic contextual information. It is challenging to capture spatial information from distant nodes using shallow graph neural networks (GNNs) to model long-range spatial correlations. To handle this problem, we design a novel spatiotemporal global semantic graph-attentive convolutional network model (STSGAN), which is a deep-level network to achieve the simultaneous modelling of spatiotemporal correlations. First, we propose a graph-attentive convolutional network (GACN) to extract the importance of different spatial features and learn the spatial correlation of local regions and the global spatial semantic information. The temporal causal convolution structure (TCN) is utilized to capture the causal relationships between long-short times, thus enabling an integrated consideration of local and overall spatiotemporal correlations. Several experiments are conducted on two real-world traffic flow datasets, and the results show that our approach outperforms several state-of-the-art baselines.

Keywords: traffic flow prediction; spatial–temporal modeling; graph convolutional network; attention mechanism

check for updates

Citation: Zhou, J.; Qin, X.; Yu, K.; Jia, Z.; Du, Y. STSGAN: Spatial-Temporal Global Semantic Graph Attention Convolution Networks for Urban Flow Prediction. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 381. https://doi.org/10.3390/ijgi11070381

Academic Editors: Wolfgang Kainz and Maria Antonia Brovelli

Received: 23 May 2022
Accepted: 4 July 2022
Published: 8 July 2022

1. Introduction

With the accelerated urbanization, the rapid increase in the number of private cars in cities has brought tremendous pressure on the existing traffic system. Traffic flow prediction, as an indispensable component of intelligent transportation systems [1], has been committed to the development of intelligent transportation systems (ITS) in many countries, aiming to predict traffic conditions accurately in real-time and provide a scientific basis for the transportation department to optimize the scheduling of traffic resources to alleviate urban traffic congestion effectively. Because of the importance of traffic flow prediction, it has attracted the attention of many researchers in recent years to the task of accurate and long-term traffic prediction.

There are complex and long-term spatiotemporal dependencies in urban transportation networks. The flow of each area in the city is influenced by its neighboring regions and the spatial semantic correlation between distant roads. Due to public transportation such as buses and cabs, vehicles have cross-regional mobility. This situation indicates that as the observations of a node evolve, they are often related not only to its neighboring nodes but also to the historical information of more distant nodes [2]. Thus, the simultaneous existence of spatial interdependencies between local and distant nodes, i.e., the presence of more semantic pairwise correlations between more distant roads, is also crucial for traffic flow prediction [3].

Two areas in a city with geographically functionally similar attributes (e.g., shopping or residential areas) strongly depend on their traffic distribution. However, they are not spatially adjacent or even distant from each other [4–6]. As an example, shown in Figure 1, to predict the traffic flow T_2 of $S2$, we can refer to the historical traffic flow T_1 of its geographical neighbor $S1$ due to the cross-regional flow of traffic. In addition, the traffic flow of $S2$ is correlated with $S5$, which is also a residential area, as shown in Figure 2, because they have similar regional functions and temporal patterns. In addition, the historical traffic flow of $S7$ should also be considered due to the highly similar temporal pattern of traffic flow with $S2$. Similar previous traffic flow patterns can also be observed at distant nodes, reflecting the interaction between multiple spatial relationships. Therefore, Zhang et al. [7] used a multi-scale attention network to learn spatial semantics correlations from a global perspective to achieve more accurate prediction results. Thus, long-range spatial correlations between regions play an increasingly important role in traffic flow prediction.

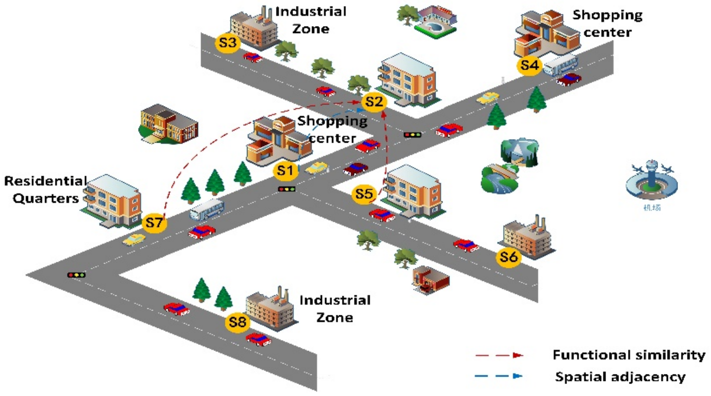


Figure 1. Illustration of multiple types of node correlations. For node $S2$, $S1$ is its geographic neighbor while having similar local functionality to $S5$.

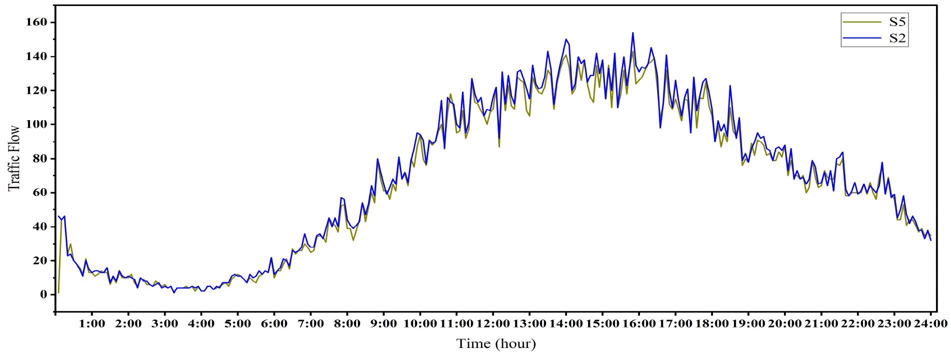


Figure 2. Example illustrating traffic flow correlation in functionally similar areas. Traffic flows at nodes $S2$ and $S5$ during the same period.

Graph neural networks (GNNs) have been applied in deep learning research [8]. They have also become a popular approach to problems of traffic flow prediction due to their powerful ability to capture spatial correlations from non-Euclidean data [9]. For example, Spatio-Temporal Graph Convolutional Networks (STGCN) [10], GraphWaveNet [11], Attention-based Spatiotemporal Graph Convolutional Network (ASTGCN) [12], etc. Since Graph Convolutional Network (GCN) models combined with Gated Recurrent Unit (GRU), long short-term memory networks (LSTM) [13], and other methods are more capable of modeling temporal or spatial dependencies, e.g., Zhao L et al. [14] and others have achieved good prediction results by combining GCN and GRU.

However, two problems are neglected in the above approach: on the one hand, when GNNs capture information about the distant spatial topology, the number of nodes in the receptive field of each node grows exponentially as the number of network layers increases. In addition, GNNs themselves have worse scalability, and the over-squeezing problem occurs when aggregating information from different nodes using multi-layer GNNs [15–17], making it difficult to propagate information among distant nodes in the graph. The features of all nodes in the deeper layers converge to the exact representation. These drawbacks limit the depth of GNNs networks, making it challenging to obtain more profound and more comprehensive spatial features. On the other hand, adaptively modelling long-term temporal correlation is vital for capturing long-range spatial correlation. However, when recurrent neural networks (RNN), LSTM, and GRU capture long-time correlation, it will gradually introduce error accumulation as the number of network layers increases. It will suffer from gradient explosion or disappearance [18], inevitably losing some essential information.

Moreover, theory and research have shown that it is difficult for RNNs to learn to store very long time series [18–20]. ASTGCN [13] and Diffusion Convolutional Recurrent Neural Network (DCRNN) [21] use an iterative prediction mechanism in which all predicted values for multiple time steps are obtained by a single uniform evaluation rather than various iterations. Therefore, combining long-term and short-term prediction tasks is challenging.

To capture local and global spatial dependencies, we propose a novel graph attention convolutional network model to effectively solve the over-squeezing problem of multi-layer graph neural networks and achieve the construction of depth nets. Inspired by the Dynamic Time Warping (DTW) algorithm [22], the regional functional similarity is used by us to construct a semantic adjacency matrix than can capture long-range spatial correlations. Finally, spatial contextual information and global traffic correlations across different regions enhance the spatiotemporal pattern representation. More extended practical historical observations are obtained to learn nonlinear long efficient, and short-term temporal correlations to capture global temporal correlations. We designed a TCN network structure based on time series modelling, containing dilation convolution and residual modules to obtain longer practical historical nonlinear temporal features. Finally, the TCN module is combined with the graph attention network cascade to get more profound and richer long-range temporal correlations.

In summary, the contributions of this paper are as follows:

- We design a data-driven approach to construct semantic similarity graphs, which preserve hidden global spatiotemporal dependencies. This data-driven adjacency matrix can extract semantic correlations that may not be present in the spatial graph.
- We propose a novel STSGAN model based on graph attention convolutional networks to effectively addresses the multi-layer graph neural network over-squeezing problem. We successfully model depth networks to simultaneously capture the long-range spatial dependencies of different importance. The cascaded temporal causal convolution module is then utilized to analyze the causal relationships between long-term and short-term time in parallel to simultaneously capture the local and global temporal correlations of traffic data.
- Extensive experiments were conducted on two real-world traffic datasets to evaluate the STSGAN reasonably in this paper. Compared with the state-of-the-art baseline, the model in this paper has better prediction performance within the 1 h forecast.

The remainder of the paper is organized as follows. We provide a comprehensive overview of the work related to traffic forecasting in Section 2. Section 3 details the problem definition of traffic forecasting and indicates the study's goals. The available framework of our STSGAN and the characteristic solutions are detailed in Section 4. In Section 5, we design several experiments to evaluate our model. Our work and the direction of future work are summarized in Section 6.

2. Related Work

2.1. Traffic Flow Prediction

As a typical spatiotemporal prediction problem, traffic flow prediction has been studied in recent decades. with before studies using significant modes such as AutoRegressive Integrated Moving Average (ARIMA) [22], Vector Auto-Regression (VAR) [23] and other machine learning models such as K-Nearest Neighbor (KNN) [24]. However, these models only consider temporal correlation and ignore spatial correlation. In addition, machine learning methods rely heavily on feature engineering, making it challenging to believe in high-dimensional spatial correlations. In recent years, to characterize spatial correlation, deep-learning-based prediction methods usually divide the whole city into grid regions, and convolutional neural network (CNN)-based methods such as [25–27] learn the traffic network as an image to effectively extract spatial features of grid data. However, these methods are designed for grid data and cannot effectively capture non-Euclidean sensor data from real-world road topology [28,29]. Thus, most of them use graph convolutional networks to capture spatial correlation, such as [16]. Zhao [14] used Temporal Graph Convolutional Networks (T-GCN) to obtain the topology of road networks for spatial correlation modelling. Guo [13] proposed an ASTGCN model. The attention mechanism is integrated with temporal convolution to capture the dynamic spatiotemporal features of traffic data. Therefore, most current studies adopt graph convolutional networks to capture spatial correlations. For example, STGCN [10] integrates graph convolution and gated temporal convolution into spatiotemporal convolution blocks to learn spatiotemporal correlations. Spatial–Temporal Synchronous Graph Convolutional Networks (STSGCN) [29] captured spatial correlations by temporal convolution blocks integrating graph and gated temporal convolution to model spatiotemporal correlations. However, they can only learn between adjacent regions and cannot capture long-range spatial dependencies. Spatial–Temporal Adaptive Fusion Graph Network (STFAGN) [30] combines fused convolutional layers with novel adaptive dependency matrices through end-to-end training to capture hidden spatio-temporal dependencies on the data and obtain hidden spatio-temporal dependencies through fusion operations in parallel. The above methods assume that spatial correlation exists only on corresponding or near nodes. Local approaches do not consider the non-local spatial correlation between nodes on the transportation network. Most of them also think only of the proximity relationships between different regions, thus ignoring the semantic relationships between other areas globally.

2.2. Spatio-Temporal Graph Convolutional Network

Brunal [31] proposed initially graph neural networks through a natural extension of convolutional neural networks (CNNs) on structured graph data. In traffic flow prediction tasks, many researchers have devoted GCNs to capturing complex spatial topologies [6–9,12]. For example, DeepSTN+ [7] proposed a deep convolutional model to model remote spatial dependencies between crowd flows, using multiple fusion mechanisms to capture complex relationships between features at different levels. Temporal Multi-Graph Convolutional Network (TMGCN) [3] proposed temporal multigraph convolutional networks to jointly extract potential semantic relevance and global spatial features.

Using cyclic units of RNN and LSTM to learn long-term correlations gradually introduces error accumulation and costs additional training time. Limited by problems such as gradient disappearance and gradient explosion [18], making it challenging to learn to store more comprehensive time-series information [26]. Therefore, to achieve accurate prediction under long-term and complex spatial conditions. DCRNN [21] proposed an approach to model traffic flow in diffusion form on directed graphs, using code-and-decode architecture and scheduled sampling techniques to improve long-term prediction performance. Spatial–Temporal Graph Attention network (STGAT) [32] proposed a dual-path architecture model based on graph attention networks to process long time series by stacking gated temporal convolutional layers. Spatial–Temporal Dynamic Network (STDN) [27] further proposed a periodic shift of attention mechanism to integrate long-term regular information. Zhang et al. [33] propose a deep spatial and temporal convolutional graph attention network that employs a multi-resolution transformer network to capture the traffic dependence between different regions. The above approaches are mainly based on local spatial correlation and rarely capture global features. In contrast, our model STSGAN combines spatiotemporal correlation into a graph-attentive convolutional neural network.

We employ multiple adjacency matrices to capture multi-resolution relationships between global and local jointly, exploring the hidden relationships across regions and between long and short periods.

2.3. Graph Attention Network

Petar Velickovi [34] initially proposed graph attention networks (GATs), a novel convolutional neural network based on graph network data. GATs focus on generating new node representations by aggregating neighbouring nodes and attention coefficients to distinguish the significance of each node in the graph relative to its neighbours. Many attempts have been incorporated in graph convolution have been made to incorporate attention mechanisms [3,27,32,34,35]. GAT [32] has achieved state-of-the-art results in several benchmark tests for graph-related tasks. Recent studies [36] introduced a multi-interval attention mechanism to aggregate information from different areas to automatically learn the importance of different intervals. Long Short-Term Traffic Prediction with Graph Convolutional Networks (LSGCN) [37] proposed a cosAtt graph attention network to capture complex spatial features while satisfying long- and short-term prediction tasks. Subsequently, Graph Multi-Attention Network (GMAN) [35] proposed an encoder-decoder network architecture with multiple spatial attention modules to capture complex spatial correlations. As an attention-based method, GMAN simply calculates the sum of spatial attention scores of all vertices. Adaptive Spatial–Temporal Graph Attention Network (ASTGAT) [38] proposes an adaptive graph attention network for capturing spatial dependencies and designs a gate-time convolutional layer to handle long time series data.

The above methods show superior performance in traffic prediction but fail to capture both global and local spatial correlations in the traffic road network. Thus, attention aggregation mechanisms are utilized by us to capture local and global traffic dependencies across regions.

3. Preliminaries

In this section, we describe some of the essential elements of urban flow and define the problem of urban flow prediction.

Definition 1. (Spatial network graph):

We denote the road as a network graph and use the directed graph $\mathcal{G} = (V, E, A)$ to define the spatial road network topological structure. $|V| = N$ denotes the set of all nodes (representing the sensors in the road network), where N denotes the number of nodes in the graph. E denotes the set of connected edges between two nodes. $A \in \mathbb{R}^{N \times N}$ represents the adjacency matrix, where each element represents the spatial road graph $\mathcal{G}$ degree of connectivity between different nodes.

The spatial graph $\mathcal{G}$ represents the correlation between different nodes in the spatial dimension. Spatial network relationships do not change with time, and in our work, we depict the spatial network as an undirected graph.

Definition 2. (Traffic Flow):

We defined $\_x_{t,v} \in \mathbb{R}^F$ to denote all the feature vectors of node v at time t with the number of input features F. $X_{\mathcal{G}}^{(t)} = (x_{t,1}, x_{t,2}, \cdots, x_{t,N}) \in \mathbb{R}^{N \times F}$, denotes all the feature values of all nodes at time t, the length of the time step denoted by t, $\chi = \left(X_{\mathcal{G}}^{(1)}, X_{\mathcal{G}}^{(2)} \cdots, X_{\mathcal{G}}^{(p)}\right) \in \mathbb{R}^{N \times F \times p}$ defines all the eigenvalues of all nodes within time slice p. The graph signal matrix χ represents the traffic condition of the spatial network $\mathcal{G}$ of the spatial graph $\mathcal{G}$ in time slice p.

Definition 3. (Traffic Flow Prediction Problem):

The spatiotemporal traffic flow prediction target can be described as follows: given the entire traffic network P historical time segments, the observations of N vertices, defined as

$\chi = \left(X_{\mathcal{G}}^{(t-P+1)}, X_{\mathcal{G}}^{(t-P+2)} \cdots, X_{\mathcal{G}}^{(t)}\right) \in \mathbb{R}^{N\times F\times P}$. The function f is learned from the traffic observation data of P historical time steps to predict the traffic flow conditions of all vertices on the road network for the next Q time steps, denoted as $\mathcal{Y} = \left(X_{\mathcal{G}}^{(t)}, X_{\mathcal{G}}^{(t+1)} \cdots, X_{\mathcal{G}}^{(t+Q)}\right) \in \mathbb{R}^{N\times F\times Q}$.

$$\left(X_{\mathcal{G}}^{(t-P+1)}, X_{\mathcal{G}}^{(t-P+2)} \cdots, X_{\mathcal{G}}^{(t)}\right) \xrightarrow{\mathcal{Y}=f(\chi\ ;\mathcal{G})} \left(\mathcal{Y}_{\mathcal{G}}^{(t)}, \mathcal{Y}_{\mathcal{G}}^{(t+1)} \cdots, \mathcal{Y}_{\mathcal{G}}^{(t+Q)}\right), \tag{1}$$

4. Methodology

In a natural traffic environment, traffic conditions in each city region are influenced by nearby and distant regions [4–7]. Therefore, we propose a spatiotemporal global semantic graph attention convolutional network framework to mine the global spatiotemporal evolution relationships of different regions. STSGAN has multiple blocks framework of the model proposed in this paper is depicted in Figure 3.

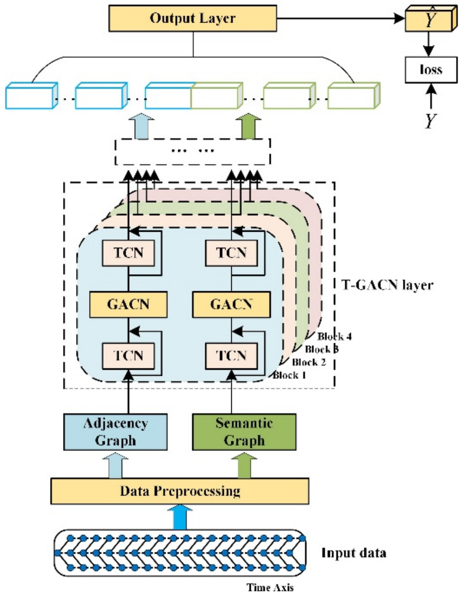


Figure 3. The framework of STSGAN. The input is $\chi \in \mathbb{R}^{N\times F\times P}$, where N is the number of nodes, P is the time step length, and F is the feature of each node. The output is $Y \in \mathbb{R}^{N\times Q}$, representing N node prediction speed in Q time steps. Four parallel T-GACN blocks form a T-GACN layer in the model, and three T-GACN layers are connected to extract high-dimensional features. Two of these layers parallel two graph attention blocks per layer. The graph attention convolutional network is sandwiched between two temporal causal convolutions.

In particular, we explored the hidden spatial features through a dual-path architecture, where each path shares the same structure to capture spatiotemporal dependencies. One path uses an adaptive adjacency matrix as input to model the spatial dependencies between neighbouring regions at a fine-grained level. The other path uses a global semantic adjacency matrix to model the spatial correlations between different areas from a global perspective. The two types of adjacency matrices are input into the T-GACN layer simultaneously. The interactions between local and global long-term and short-term time are captured by the temporal causal convolution module (TCN): Next, the input is fed to the graph attention convolution module to simultaneously capture the long-range spatial dependencies. Four-layer modules were employed in a stacked structure with residual connections [39] and layer normalization [40] within each module to guarantee the model's deep and efficient training. All layers in the T-GACN block have the same number of dimensions and obtain comprehensive spatio-temporal features.

4.1. Traffic Map Construction

4.1.1. Traffic Adjacency Spatial Graph

The traffic adjacency matrix is constructed according to STGCN, as shown in Figure 4a, to measure the proximity between nodes according to the geometric distance between different nodes. The distance function defines the values between other nodes. The distance function is defined as follows.

$$A^{sp}_{v_i,v_j} = \begin{cases} exp\left(-\frac{d^2_{v_i,v_j}}{\sigma^2}\right) & , \ if \ \ exp\left(-\frac{d^2_{v_i,v_j}}{\sigma^2}\right) \geq \epsilon; \\ 0 & , \quad otherwise. \end{cases}, \tag{2}$$

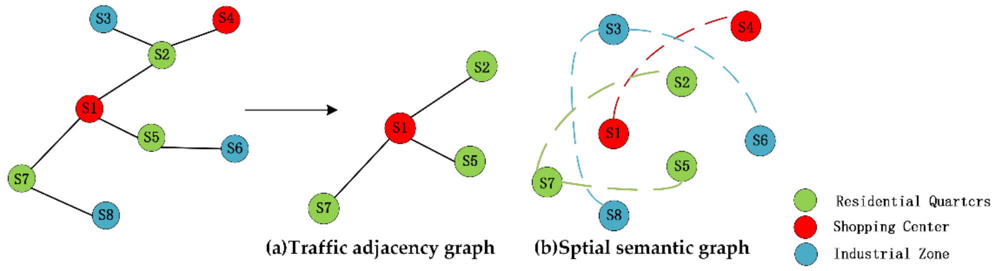


Figure 4. Traffic map construction: (**a**) Traffic spatial adjacency graph is structured according to traffic proximity. Regions geographically connected to region *S*1 (shopping center) are connected. (**b**) Global semantic adjacency graphs, where regions with similar functions have similar traffic patterns, are constructed to connect edges between regions with similar functions. For example, regions *S*2, *S*6 and *S*8 are residential quarters, so they are connected.

The distance between node v_i and node v_j in Equation (2) is expressed as d_{v_i,v_j}: hyper-parameters σ^2 and ϵ to control the sparsity of the spatial adjacency matrix.

4.1.2. Global Semantic Spatial Graph

As shown in Figure 5, the dynamic time warping (DTW) algorithm [22,41] can better reflect the similarity of two-time series than the Euclidean distance. The DTW algorithm has an excellent sensitivity to shape similarity and is superior to other metrics [39]. Because two regions with similar functional properties have similar temporal traffic patterns, we applied the DTW algorithm to evaluate whether they are similar in terms of function (e.g., they both belong to a business zone). Then, we constructed a global spatial semantic graph based on semantic similarity.

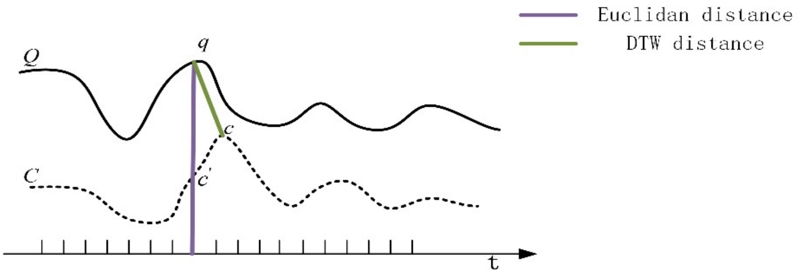


Figure 5. Example of the difference between Euclidean distance and DTW distance.

The sum of the Euclidean distance $d(q_i, c_j)$ between grid locations q_i and c_j is separated into two-time series. The cumulative distance of the smallest neighbouring element that can reach the point is denoted as the cumulative distance $\gamma(i, j)$. As in Equation (3).

$$\gamma(i, j) = d(q_i, c_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}, \tag{3}$$

Here, $\gamma(i,j)$ denotes the shortest distance between two nodes $Q = (q_1, q_2, \cdots, q_i)$ and $C = \left(C_1, C_2, \cdots, C_j\right)$, so DTW$(Q, C)$= $\gamma(m,n)$ is taken as the final distance between nodes Q and node C, which can better reflect the similarity of two-time series compared with the Euclidean distance.

Therefore, we define the semantic adjacency matrix by DTW distance as follows.

$$A^{se}_{ij} = \begin{cases} 1 & ,if\ \ DTW\left(Q^i, C^j\right) \le \epsilon; \\ 0 & , \qquad otherwise. \end{cases}, \tag{4}$$

Here, Q^i and C^j denote the time series of node i and node j, respectively, and set ϵ assigned to 0.6 for controlling the sparsity of the adjacency matrix.

4.2. Attention-Based Graph Neural Network

4.2.1. Spatial Graph Convolution Layer

Most work employs the GCN approach to capture spatial correlations across regional dimensions, extending traditional convolutional operations from structured data to graphs, enabling them to capture unstructured pattern information hidden in the graph [35–37]. As shown in Figure 6, the idea of GNN is to aggregate information about neighbouring nodes to produce an up-to-date representation of the nodes, and then aggregate the representation with a linear projection transformation and finally activate it by nonlinearity [42]. However, there are challenges in modelling long-term dependencies between high-dimensional data using graph convolutional networks. Uri Alon et al. [17] demonstrated that GNNs suffer from over-compression when fitting remote signals to training data, cannot spread messages from long-distance nodes when performing remote interactions and perform poorly in feature aggregation.

$$GCN\left(H^{(l+1)}\right) = \sigma\left(\hat{A}H^{(l)}W^{(l)}\right), \tag{5}$$

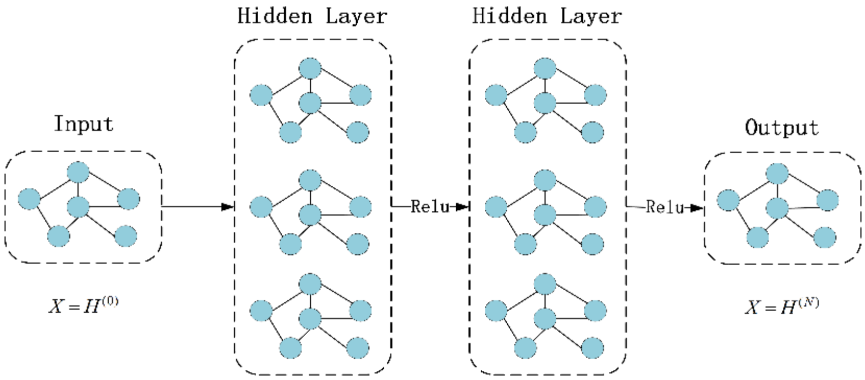


Figure 6. Graphical convolutional network to extract spatial features.

$H^{(l)}$ is the node input feature representation, $\hat{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, representing the interaction relationship between nodes, and σ represents the non-linear activation function.

For interactive propagation of long-range information, the graph neural network has to increase the number of layers. However, the effect is worse. Figure 7 shows that the over-squeezing phenomenon occurs [18]. Therefore, we introduce a self-attention mechanism in the graph convolutional network to solve the over-squeezing problem of graph neural networks in capturing long-range spatial information [42].

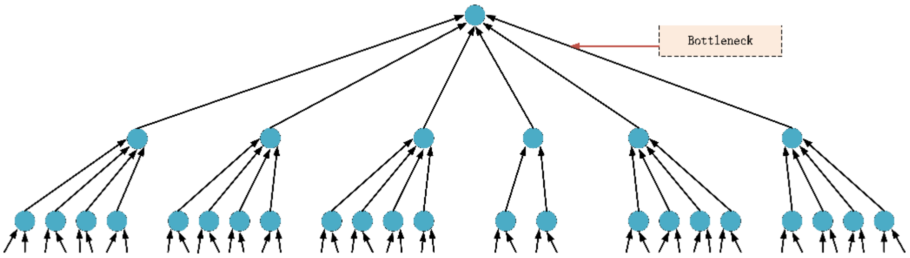


Figure 7. The bottleneck of graph neural networks.

4.2.2. Graph Attention Layer

Velickovic [34] processed graph structure data by adding a self-attention layer to graph neural networks and obtained state-of-the-art results. Inspired by this, we adopt the graph attention convolution neural network with a self-attention layer. Furthermore, self-attention allows the representation of each symbol to be directly informed by the representation of all other symbols in the sequence so that self-attention can capture the context information of the local space. Moreover, the self-attention mechanism effectively captures global dependencies and can produce a practical global receiving domain [43]. Considering that the spatial correlation is challenging to capture over long distances, multilayer parallel T-GAT blocks are utilized to capture better spatio-temporal features. Each layer contains two TCN blocks and one GACN block. Sharing the parameters of GAT in the block can help reduce the excessive extrusion of GNN so that our proposed model can make an accurate long-range prediction.

The Graph Attention Network (GAT) adds a hidden self-attention layer to GCN. By overlaying the self-attention layer, different importance is assigned to different nodes in the neighbourhood in the convolution process. Different sizes of neighbours are dealt with simultaneously. Its structure is shown in Figure 7.

In the case of limited computing power, the attention mechanism allocates computing resources to more critical nodes (for example, we focus on the nodes with similar functions in the two regions) to reduce the attention to other nodes and filter out the information we do not care about. This method can effectively alleviate the over-squeezing of information and pay attention to the general information while dealing with local information.

The input to the graph attention network (GAT) is a set of node features, $X = \left\{ \vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N \right\} \in \mathbb{R}^{B \times N \times T \times F}$, N is the number of nodes and F is the number of features in each node. We first transpose and reshape it to $X^T \in \mathbb{R}^{BT \times N \times F}$ and then generate a new set of node features $X' = \left\{ \vec{x'_1}, \vec{x'_2}, \cdots, \vec{x'_N} \right\}$, which will $\vec{x'_i} \in \mathbb{R}^{BT \times N \times F'}$ as the output.

We adopted a learnable linear transformation function to convert the input features into a higher-level feature representation. For this reason, the shared linear transformation $W \in \mathbb{R}^{F \times F'}$ parameterized by the weight matrix is applied to each node. Then, we use a shared attention mechanism A to perform self-attention on any pair of $\vec{x_i}$ attention and $\vec{x_j}$ attention nodes and calculate the attention scores as follows:

$$\text{score} = A\left(\vec{x_i} w_i, \vec{x_j} w_j\right), \quad w_i, w_j \in F \times F', \tag{6}$$

where w_i and w_j are the weight matrix components and $A(\cdots)$ is the attention mechanism operation function. In order to make the attention coefficients between different nodes easy to compare and distinguish, the softmax function is utilized to normalize the attention coefficient scores between node neighbors to obtain the following output.

$$a_{ij} = \text{softmx}(\text{score}) = \frac{exp(\text{score})}{\sum_{k \in \mathcal{N}_i} exp(\text{score}_{ik})}, \tag{7}$$

$$\vec{x}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} a_{ij}\vec{x}_j w_j\right), \tag{8}$$

where $\mathcal{N}_i$ is the set of neighbors of node i in the graph, and σ is the activation function.

Based on road networks, roads neighboring each other usually have similar road conditions, as shown in Figure 8. We designed a new graph attention property to extract similar road situations in traffic networks. In GAT, we used the global graph attention network to learn the similarity of any two roads in the traffic network.

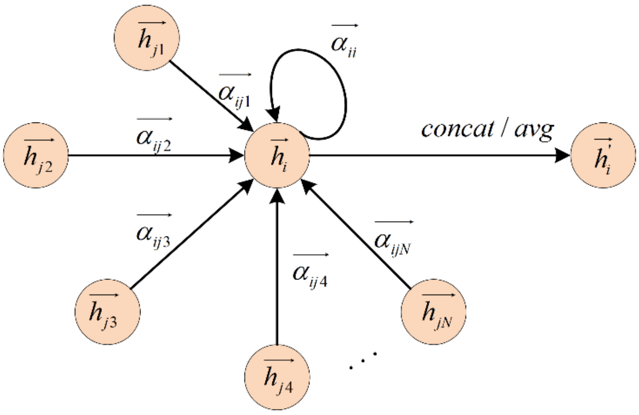


Figure 8. The example of using graph attention layer operation for neighbors around node 1, where $\vec{h}_i \in \mathbb{R}^{N\times F}$ represents the features of node i, N is the number of nodes, and F is the number of features. $\vec{h}_i'$ is the update of the hidden feature.

We parallelize the temporal traffic data using TCN to obtain global long-term and short-term temporal correlations. Specifically, for the fusion of spatiotemporal relationships in the traffic data, we used the node spatial feature set output from GAT as the input to the next TCN block. After the TCN layer, we obtained the global long-term spatiotemporal correlations of the road network.

4.3. Temporal Causal Convolution Module

In addition to the multi-scale spatial correlation between different nodes that we considered above, temporal causality also exists between traffic events, so capturing the long-term temporal dependence of nodes is also very significant. RNN-based models (e.g., LSTM and GRU) have achieved outstanding results in time-series analysis. However, they have drawbacks such as model difficulty in training and great computational effort in extracting long-time series features. Temporal convolutional networks were evaluated by Bai S et al. [44] to prove more efficient and superior long-range capture capability in modelling time-series data [45]. Thus, the TCN architecture shown in Figure 9 is composed of many causal convolutional layers with exponential scaling factors that expand the perceptual domain. Moreover, the dilation convolution has good scalability and obtains a larger receptive field to better capture long-range spatio-temporal dependencies.

Briefly, the TCN consists of a one-dimensional FCN and a causal convolution [43,46], and the dilated causal convolution of the TCN is formulated as follows.

$$y_t = \Theta * \mathcal{T}_d x_t = \sum_{K=0}^{K-1} w_k x_t - dk, \tag{9}$$

Here, $*\mathcal{T}_d$ is the dilation causal operator with dilation d to control the jump distance, and $\Theta = [w_0, \cdots, w_{k-1}] \in \mathbb{R}^K$ is the convolution kernel.

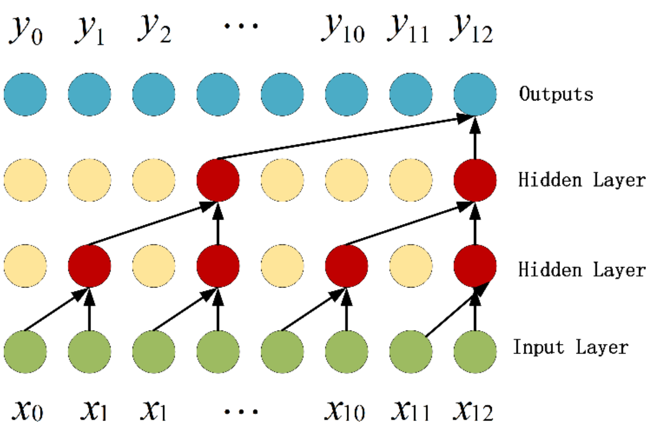

y_0 y_1 y_2 ··· y_{10} y_{11} y_{12}
Outputs
Hidden Layer
Hidden Layer
Input Layer
x_0 x_1 x_1 ··· x_{10} x_{11} x_{12}

Figure 9. The causal convolution layer of TCN multiple extension: $x_1, x_2, \cdots, x_{12}$ is the input sequence, and $y_1, y_2, \cdots, y_{12}$ is the output sequence. It is admitted that the input history length of the network is the same as the future prediction length.

4.3.1. Dilated Convolution

Since the spatial distances of our observation points are far from each other, the spatial correlation between nodes may have an hour or even longer temporal correlation [37]. We superimpose a 4-layer TCN convolution for extracting local and global long-term and short-term temporal correlation features. The model input is a 3D traffic graph data signal with $\chi \in \mathbb{R}^{P \times N \times F_1}$, and an extended causal convolution is utilized to generate the j^{th} output feature of node i at time t, as shown in Equation (10).

$$y^m_{t,n} = \rho(\Theta_n * \mathcal{T}_d \chi^m_t) = \rho\left(\sum_{i=1}^{F_I} \sum_{k=0}^{K-1} \mathcal{W}_{n,i,k} \chi^m_{t-dk,i}\right), \tag{10}$$

where $1 \leq n \leq F_O$, $y^m_{t,n} \in \mathbb{R}$ is the n^{th} output feature of node m at time t, and $\chi^m_{t-dk,n} \in \mathbb{R}$ is the i^{th} input feature of node m at time $t - dk$. The kernel $\Theta_n \in \mathbb{R}^{K \times F_I}$ is trainable. F_O is the number of output features.

All nodes in the traffic road network graph are calculated using the same convolution kernel to generate new features for each node. The convolutional layer is calculated as in Equation (11).

$$y = \rho(\Theta * \mathcal{T}_d \chi), \tag{11}$$

where $\chi \in \mathbb{R}^{P \times N \times F_I}$ denotes the historical observations of the whole traffic network over the past P time slices as the input to the TCN, $\Theta \in \mathbb{R}^{K \times F_I \times F_0}$ denotes the causal convolution kernel, and $y \in \mathbb{R}^{P \times N \times F_0}$ is the output of the TCN layer.

$$y^{(l+1)} = \sigma\left(\Theta^l * \mathcal{T}_d y^{(l)}\right), \tag{12}$$

where $y^{(l)}$ is the input of the l layer, $y^{(l+1)}$ is the output, $y^{(0)} = \chi$, and $d = 2^{(l)}$ is the dilation rate of the l^{th} layer.

4.3.2. Residual Connection

Inspired by residual networks [39], the over-smoothing problem can be alleviated by adding residual connections between layers. Furthermore, residual networks have been proven to be an effective method for training deep networks [47], enabling the network to transfer information in a cross-layer manner. Using a convolution-based residual network to perform the spatial relationship between two regions in a city in terms of distance and proximity can ensure that the model's prediction accuracy is not affected by the in-depth structure of the neural network [48]. In this paper, we construct a residual module as shown in Figure 10, which replaces one layer of convolution, effectively relieving the problem of gradient disappearance and enhancing the propagation of features.

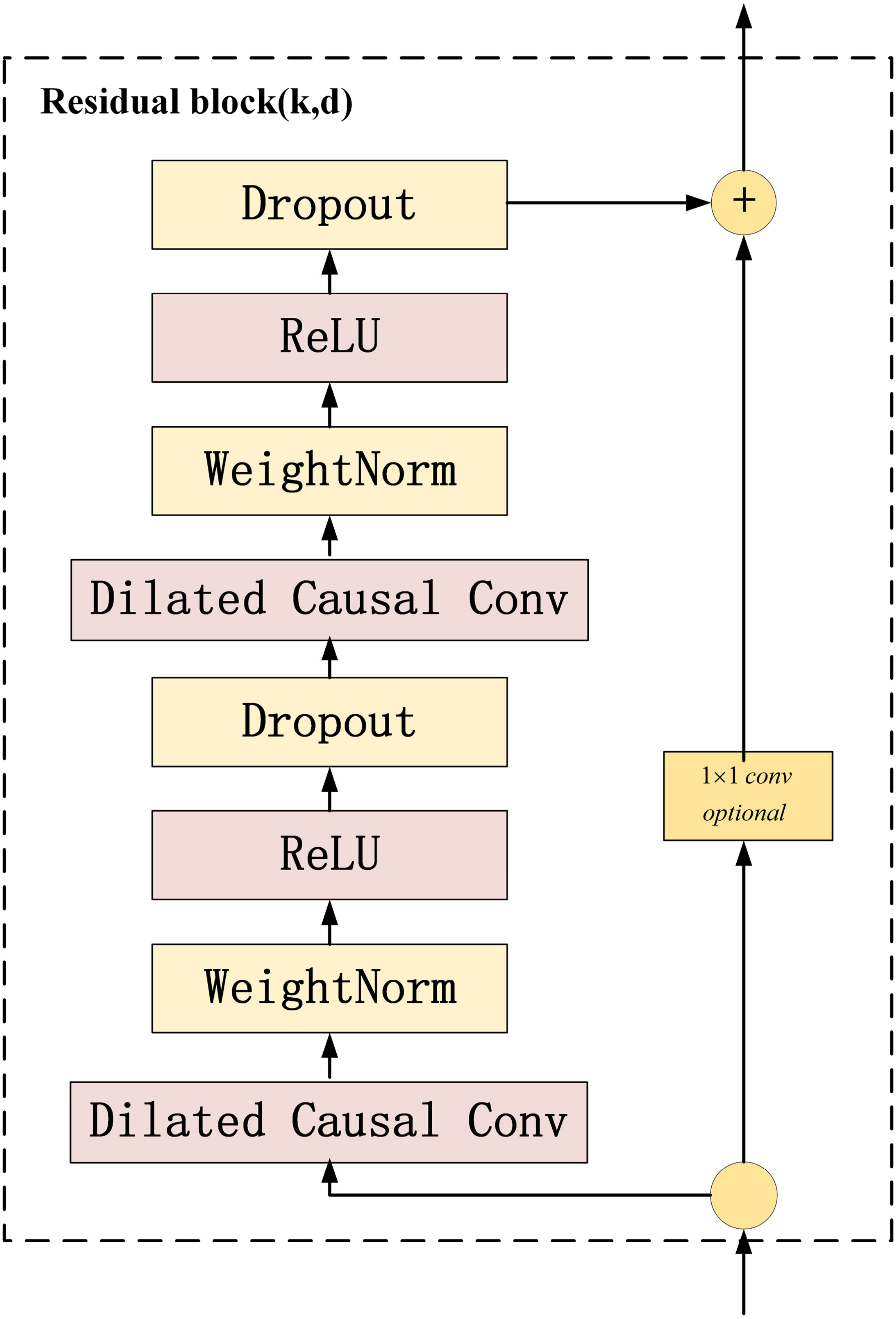


Figure 10. A residual block contains two layers of extended convolution and non-linear graphs, and WeightNorm and Dropout are also integrated with each layer to normalize the network.

To guarantee the stability of TCN, we use residual modules instead of convolutional layers to avoid problems such as gradient disappearance.

4.4. Multi-Module Fusion Output

The outputs of different components are fused, and the influence weights of the adjacency and semantic matrix are different for each node learned from the historical data. Thus, after the T-GACN layer, the maximum pooling operation is first executed to aggregate the information according to the influence weight size selectively and then converted into the final prediction result through the max-pooling layer. The calculation process is as follows.

$$\hat{Y} = W_a \odot \hat{Y}_a + W_s \odot \hat{Y}_s, \tag{13}$$

where $\odot$ is the Hadamard multiplicity, and W_a and W_s are learning parameters reflecting the effect of the two matrices on the prediction target.

Because Huber loss is less sensitive to discrete points than squared error loss, we choose Huber loss (1992, STSGCN) [29].

$$\mathrm{L}(Y, \hat{Y}) = \begin{cases} \frac{1}{2}\left(Y - \hat{Y}\right)^2 & , \left|Y - \hat{Y}\right| \leq \delta; \\ \delta\left|Y - \hat{Y}\right| - \frac{1}{2}\delta^2 & , otherwise. \end{cases}, \tag{14}$$

Y and $\hat{Y}$ are the real and predicted value of the model, respectively, and δ is used as a threshold hyperparameter to control the range of variance loss.

5. Experiment

To evaluate the performance of our model, we conducted comparative experiments on two real-world highway traffic datasets.

5.1. Datasets

The PEMS04 and PEMS08 [13,29] data used in the experiments are collected in real-time every 30 s by the Caltrans Performance Measurement System (PeMS). This system has more than 39,000 detectors deployed on freeways in major metropolitan areas of California. The raw traffic flow data is aggregated into 5 min intervals, which means there are 288 time steps in a day's traffic flow. More detailed information is shown in Table 1.

Table 1. Datasets description.

Datasets	Node	Edges	Time Steps	Time Range	Missing Ratio
PEMS04	307	340	16,992	1 January 2018–28 February 2018	3.182%
PEMS08	170	295	17,856	1 July 2016–31 August 2016	0.696%

The experiment uses two metrics, including MAE (Mean Absolute Error) and RMSE (Root-Mean-Square Error), to evaluate all methods, which are widely used to assess the accuracy of regression problems. For all metrics, the lower the value, the better. They are defined as

$$\mathrm{MAE}(\hat{y}_i, y_i) = \frac{1}{n}\sum\nolimits_{i=1}^{n} |y_i - \hat{y}_i|, \tag{15}$$

$$\mathrm{RMSE}(\hat{y}_i, y_i) = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n} (y_i - \hat{y}_i)^2}, \tag{16}$$

5.2. Baseline Methods

To evaluate the performance of our model, we compare it with 8 baselines.

- ARIMA [23]: the autoregressive integrated moving average model, a well-known statistical model for time-series analysis, uses past data to predict future trends.
- FC-LSTM [49]: Short-term memory (LSTM) networks with fully connected hidden units are a well-known network framework robust in capturing sequential dependencies.

- DCRNN [21] (Li et al., 2017): Gated recurrent units with integrated graph convolution capture temporal dynamics using bi-directional graph random wandering to simulate spatial dependencies.
- STGCN [10]: A deep learning framework for traffic prediction uses graph convolution and one-dimensional gated temporal convolution to capture spatial correlation and temporal correlation.
- GraphWaveNet (Wu et al., 2019) [11]: GraphWaveNet combines adaptive graph convolution with extended casual convolution to automatically capture hidden spatial dependencies.
- STSGCN [29]: a model that directly captures local spatiotemporal correlations synchronously using multiple local subgraph modules while considering spatial data heterogeneity.
- LSGCN [37]: Long Short-Term Graph Convolutional Network (LSGCN), which uses spatially gated convolutional blocks with an attention mechanism to capture spatiotemporal features.

5.3. Experimental Parameter Settings

We refer to ASTGCN [13] and STSGCN [29], and all datasets are decomposed into training, validation, and test sets in the ratio of 6:2:2. Specifically, the total time length of the PEMS04 dataset is 59 days, with 16,992 time steps. Therefore, the first 10,195 time steps are used as the training set, 3398 time steps are used as the validation set, and 3398 time steps are used as the test set. The total time length of the PEMS08 dataset is 62 days, of which there are 17,856 time steps. The first 10,714 time steps are used as the training set, 3572 time steps are used as the validation set, and 3572 time steps are used as the test set. In addition, we normalize the data samples of each road segment by the following Equation (17) and input the normalized data into the model, which is optimized by inverse mode automatic differentiation and Adam [50].

$$x' = \frac{x - mean(x)}{std(x)}, \tag{17}$$

We used one hour of historical data to predict the next hour of the data, and in order to make a fair comparison, all of our experimental results were attempted ten times on each data set.

We utilize the PyTorch framework to implement our STSGAN model, and all experiments are compiled on a Linux server with a test environment of (CPU: Intel(R) Core(TM) i7-9900k CPU @ 3.60 GHz, GPU: NVIDIA TITAN RTX 24G). the hidden size of the TCN blocks is set to 64, 32, and 64, and each layer contains three standard blocks. The normalization hyperparameter is set to 0.8, the thresholds of the semantic adjacency matrix and spatial adjacency matrix are set to 0.6 and 0.5, respectively, and the σ values of the semantic and spatial adjacency matrices are set to 0.1 and 10. the batch size is set to 32, the learning rate is set to 0.001, and the model is optimized using the Adam optimizer with 200 iterations.

5.4. Experimental Results and Comparative Analysis

Our STSGAN model is compared with the proposed eight-method baseline on two real-world datasets. The experiment results are shown in Table 2.

Table 2. Performance comparison of T-GACN and its variants in traffic flow prediction.

Dataset	Model Elements	MAE	RMSE
PEMS04	T	23.12	35.98
	T+G	23.18	36.12
	T+A+G	21.38	34.03
	T+S+G	21.25	33.60
	STSGAN	20.80	32.81

These experiments will be summarized to answer the following research questions:

- RQ1: What is the performance of STSGAN for overall traffic prediction compared to various baselines?
- RQ2: How do the different sub-modules designed to improve the model performance?
- RQ3: What is the performance of the designed modules on long-term prediction problems?
- RQ4: How do the parameter settings of the model affect the experimental results?

The results in Figure 11 show that STSGAN has the highest prediction accuracy compared with other baselines. Mainly ARIMA and FC-LSTM only consider temporal correlation, whereas spatial correlation is essential for prediction performance, so their prediction performance is the worst. DCNN is a spatiotemporal data prediction method based on RNN. Due to the superior capability to capture long-term temporal correlation will be dissipated by gradient and other problems limitations, its prediction accuracy is much lower than our model.

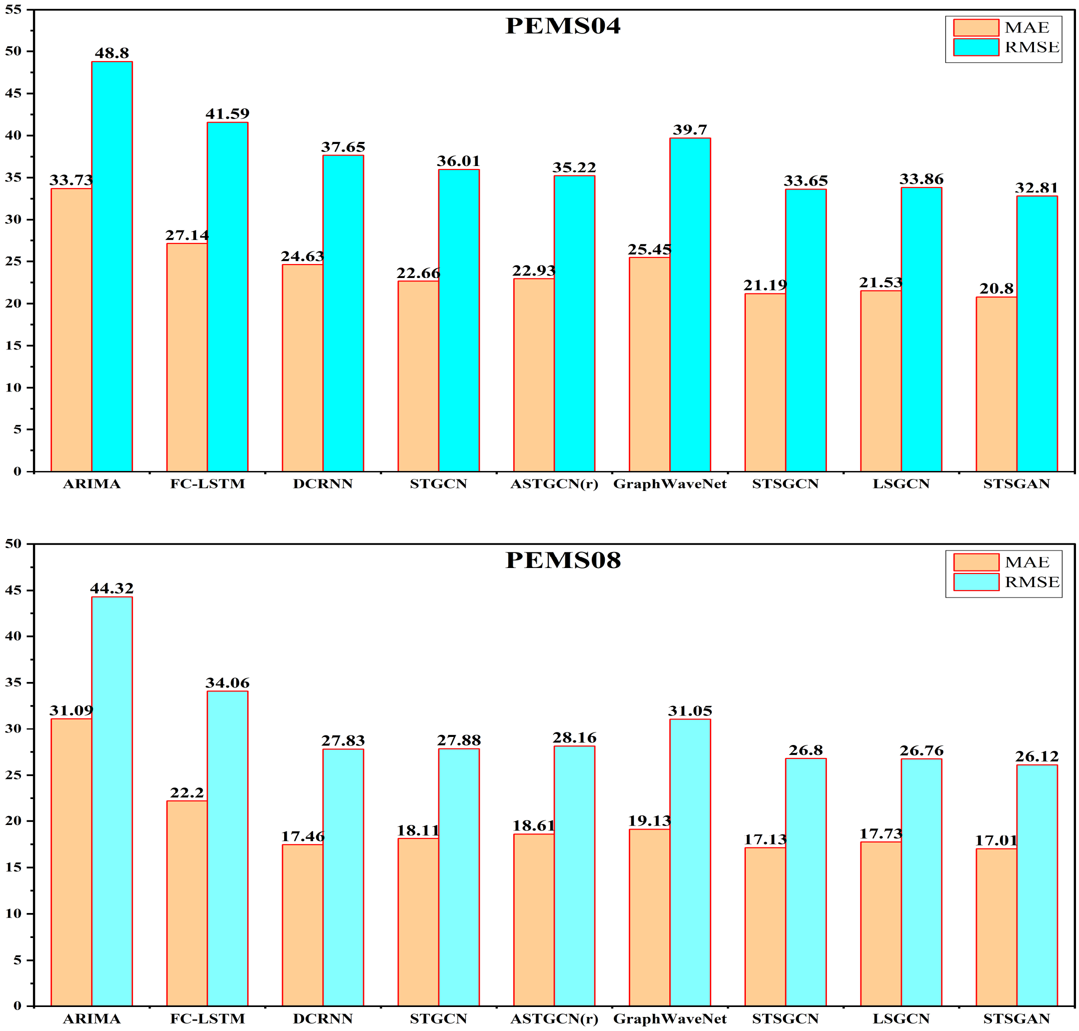
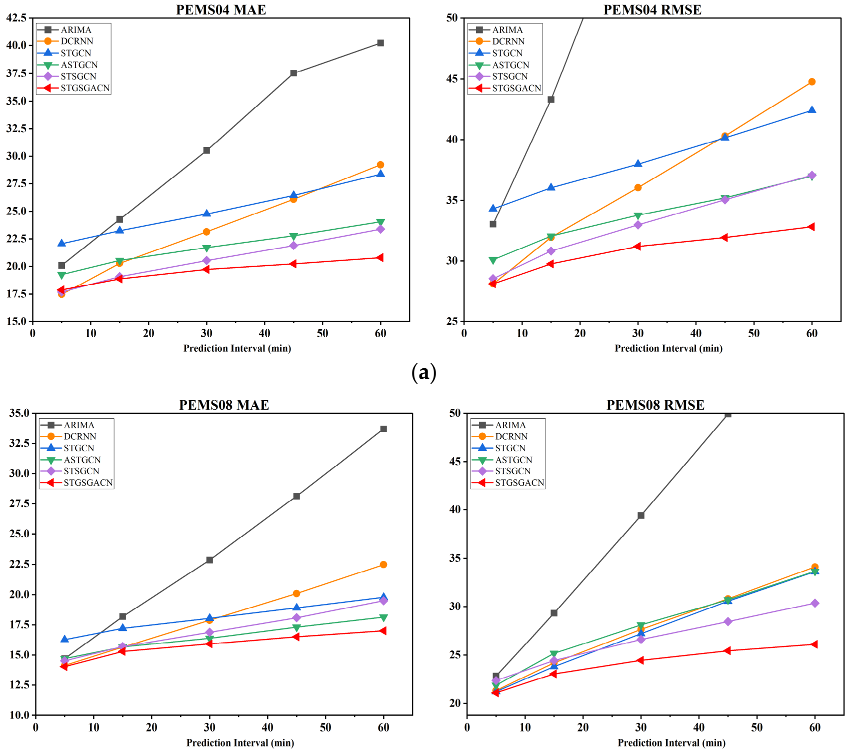


Figure 11. The subfigures PEMS04 and PEMS08 show the prediction performance of our STSGAN model on the two datasets, respectively. It is demonstrated that our method outperforms the other eight baseline models compared.

Although STGCN, ASTGCN, Graph Wave Net, and STSGCN all consider temporal correlation and spatial correlation features, generally, they research the local spatial problem and ignore the interaction relationship between long-range Spatial. Our method considers the local spatial correlation and captures the long-range spatial interactions to extract the global features of the space. Thus, our method has better performance than these methods.

Figure 12 illustrates the effect of increasing the prediction time interval on the prediction performance of different methods. As the prediction time interval increases, the difficulty of prediction will also increase, and the model's prediction performance will decline. However, in most cases, our model outperforms all the baseline methods for long-term prediction. Furthermore, it can be viewed from the figure that the decline in the prediction performance of our model is relatively small as the time interval increases, making the advantage of the model for long-term prediction more obvious.



(**a**)

(**b**)

Figure 12. Prediction the performance of different methods with increasing prediction time intervals. (**a**) The prediction results of different methods on the dataset PEMS04. (**b**) The prediction results of different methods on the dataset PEMS08.

5.5. Ablation Experiments

To further evaluate the effects of different components on STSGAN, we constructed four simple variation modules based on T-GACN. For easy expression, some symbols are utilized to denote the module names, including adjacency graph with (L), global semantic graph (S), graph attention layer (A), and graph convolutional neural network (G). All these modules have experimented on the PEMS04 dataset, and Table 2 shows the prediction performance of T-GACN and its different variants, and the experimental results are analyzed.

Except for the above differences, all the adaptations have the same experimental parameter settings as T-GACN.

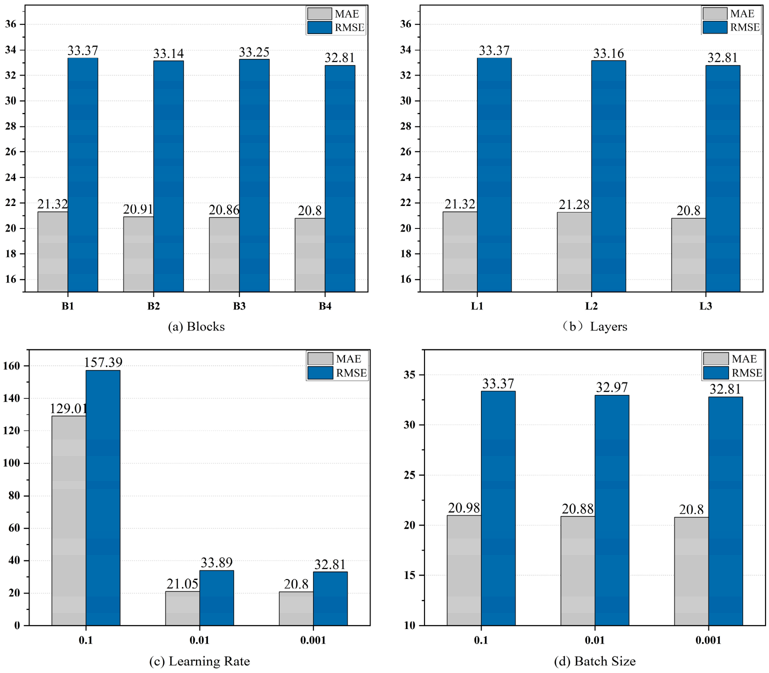
- T: The temporal causal convolution module was used to predict future traffic flow using historical traffic flow as input.
- T+G: utilized T as the base module and added a graph convolutional neural network (GCN) for capturing local and global spatial correlations. It explores whether over smoothing occurs when GCN captures long-range spatial correlations.
- T+A+G: Introduce graph attention convolutional neural network for capturing long-range spatial correlations.
- T+S+G: jointly a global semantic adjacency matrix to capture spatial features with a global perspective. Compared to the lack of graph self-attentive mechanism.

Except for the above differences, all the modifications have the same settings as T-GACN.

From Table 2, we can observe that STSGAN has better MAE and RMSE than other variants. The result analysis indicates that the model prediction performance does not rise and fall when GCN is integrated to capture spatial features. The information over-squeezing phenomenon occurs when GCN is used to capture long-distance spatial correlation, which leads to performance degradation. Therefore, adding the graph attention convolution module can effectively alleviate the over-smoothing phenomenon to capture distant spatial features and improve model performance. Furthermore, the introduction of a semantic adjacency matrix to improve the model performance indicates that it can improve the model's local and global spatial perception, which in turn improves the model prediction performance. Therefore, the prediction performance of the STSGAN model can be effectively improved by using these components.

5.6. The Influence of Network Configuration

To further research how hyperparameters and the number of network layers influence model performance. We conducted experiments on different network configurations. All of the following experiments follow the same setup described in Section 5.3 except for the variations in the network configuration under study. Figure 13a,b show that B1 represents that there is one T-GACN module and B4 represents that there are four T-GACN modules. As the number of B (T-GACN modules) increases, we can observe that the model's performance is gradually improving, which illustrates the effectiveness of our use of spatio-temporal feature parallel processing modules. Four T-GACN modules form a T-GACN network layer. L1 represents a network configured with one layer of T-GCAT layers, and L3 represents a network configured with three layers of T-GCAT layers. As the L (T-GACN network layer) number increases, the model's prediction performance also improves. We find that the model performance continues to improve as the number of network layers increases beyond two layers, indicating that our use of attentional convolutional networks can effectively alleviate the GNN overcrowding problem.



(a) Blocks

（b）Layers

(c) Learning Rate

(d) Batch Size

Figure 13. Network configuration analysis. In (**a**), B1 represents the network configuration with one T-GACN module, and the number of other modules increases sequentially. In (**b**), L1 represents the network configuration with one T-GACN layer, and L3 represents the network configuration with three T-GACN layers. (**c**,**d**), respectively, analyze the effects of learning rate and batch size on the network.

As can be seen from Figure 13c,d, the best performance is achieved when the model's learning rate is set to 0.001. Adjusting the model's batch size makes the network performance more stable.

6. Conclusions and Future Work

In this paper, we propose STSGAN, a novel graph-attentive convolutional neural network model, which successfully overcomes the limitations of graph neural network over-squeezing and thus captures long-range spatiotemporal correlations. Our model introduces a semantic adjacency matrix and utilizes a graph convolutional neural network with self-attentive layers to extract spatial correlation features at various levels. A temporal causal convolutional network learns the long-term and short-term temporal dependencies. Finally, the fusion module aggregates local and global spatial features to enhance the spatiotemporal pattern representation of the model. The comprehensive experiments indicate that the method in this paper performs significantly better than eight state-of-the-art baselines on two real-world datasets. The ablation experiments demonstrate the effectiveness of our proposed GAT. However, external factors such as weather and road emergencies can also affect urban traffic flow prediction. In future work, we will consider this external information to improve the model's prediction performance.

Author Contributions: Conceptualization, Junwei Zhou and Xizhong Qin; methodology, Junwei Zhou; software, Junwei Zhou and Xizhong Qin; validation, Junwei Zhou, Xizhong Qin, Kun Yu and Yan Du; formal analysis, Xizhong Qin; investigation, Junwei Zhou, Kun Yu and Yan Du; resources, Junwei Zhou; writing—review and editing, Junwei Zhou and Xizhong Qin; visualization, Junwei Zhou; supervision, Xizhong Qin; project administration, Xizhong Qin, Zhenhong Jia; funding acquisition, Xizhong Qin. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region and Major Science and Technology Special Projects in Xinjiang Uygur Autonomous Region. The funded project number is: 2019D01C058 and 2020A03001-2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhang, J.; Wang, F.Y.; Wang, K.; Lin, W.H.; Xu, X.; Chen, C. Data-driven intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1624–1639. [CrossRef]
2. Zhang, X.; Huang, C.; Xu, Y.; Xia, L. Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19 October 2020; pp. 1853–1862.
3. Lv, M.; Hong, Z.; Chen, L.; Zhu, T.; Ji, S. Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3337–3348. [CrossRef]
4. Du, B.; Hu, X.; Sun, L.; Liu, J.; Qiao, Y.; Lv, W. Traffic demand prediction based on dynamic transition convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1237–1247. [CrossRef]
5. Wang, C.; Zhu, Y.; Zang, T.; Liu, H.; Yu, J. Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual, 8–12 March 2021; pp. 616–634.
6. Zhang, X.; Huang, C.; Xu, Y.; Xia, L.; Dai, P.; Bo, L.; Zheng, Y. Traffic flow forecasting with spatial-temporal graph diffusion network. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 15008–15015.
7. Lin, Z.; Feng, J.; Lu, Z.; Li, Y.; Jin, D. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 1020–1027.
8. 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] [PubMed]
9. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

10. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
11. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
12. 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; Volume 33, pp. 922–929.
13. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
14. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
15. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
16. Li, Q.; Han, Z.; Wu, X.M. Deeper insights into graph convolutional networks for semi-supervised learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
17. Alon, U.; Yahav, E. On the bottleneck of graph neural networks and its practical implications. *arXiv* **2020**, arXiv:2006.05205.
18. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Atlanta, GA, USA, 17–19 June 2013; pp. 1310–1318.
19. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
20. Roy, A.; Roy, K.K.; Ahsan Ali, A.; Amin, M.A.; Rahman, A.K.M. SST-GNN: Simplified Spatio-Temporal Traffic Forecasting Model Using Graph Neural Network. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Virtual, 11–14 May 2021; Springer: Cham, Switzerland, 2021; pp. 90–102.
21. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
22. Giorgino, T. Computing and visualizing dynamic time warping alignments in R: The dtw package. *J. Stat. Softw.* **2009**, *31*, 1–24. [CrossRef]
23. Williams, B.M.; Hoel, L.A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *J. Transp. Eng.* **2003**, *129*, 664–672. [CrossRef]
24. Lu, Z.; Zhou, C.; Wu, J.; Jiang, H.; Cui, S. Integrating granger causality and vector auto-regression for traffic prediction of large-scale WLANs. *KSII Trans. Internet Inf. Syst. (TIIS)* **2016**, *10*, 136–151.
25. 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.
26. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [CrossRef]
27. 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.
28. Guo, S.; Lin, Y.; Li, S.; Chen, Z.; Wan, H. Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3913–3926. [CrossRef]
29. 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; Volume 34, pp. 914–921.
30. Kong, X.; Zhang, J.; Wei, X.; Xing, W.; Lu, W. Adaptive spatial-temporal graph attention networks for traffic flow forecasting. *Appl. Intell.* **2022**, *52*, 4300–4316. [CrossRef]
31. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2013**, arXiv:1312.6203.
32. Huang, Y.; Bi, H.; Li, Z.; Mao, T.; Wang, Z. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6272–6281.
33. Zhang, X.; Xu, Y.; Shao, Y. Forecasting traffic flow with spatial–temporal convolutional graph attention networks. *Neural Comput. Appl.* **2022**, 1–23. [CrossRef]
34. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
35. Zheng, C.; Fan, X.; Wang, C.; Qi, J. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1234–1241.
36. Chen, W.; Chen, L.; Xie, Y.; Cao, W.; Gao, Y.; Feng, X. Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3529–3536.

37. Huang, R.; Huang, C.; Liu, Y.; Dai, G.; Kong, W. LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2020; pp. 2355–2361.
38. Yang, S.; Li, H.; Luo, Y.; Li, J.; Song, Y.; Zhou, T. Spatiotemporal Adaptive Fusion Graph Network for Short-Term Traffic Flow Forecasting. *Mathematics* **2022**, *10*, 1594. [CrossRef]
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
40. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
41. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the KDD Workshop, Seattle, WA, USA, 30–31 July 1994; Volume 10, pp. 359–370.
42. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
43. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
44. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
45. Lea, C.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks: A unified approach to action segmentation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 47–54.
46. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
47. Guo, G.; Zhang, T. A residual spatio-temporal architecture for travel demand forecasting. *Transp. Res. Part C Emerg. Technol.* **2020**, *115*, 102639. [CrossRef]
48. Zhang, J.; Zheng, Y.; Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–10 February 2017.
49. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
50. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. ICLR. 2015. *arXiv* **2015**, arXiv:1412.6980.