Augmented Multi-Component Recurrent Graph Convolutional Network for Traffic Flow Forecasting

Chi Zhang, Hong-Yu Zhou, Qiang Qiu, Zhichun Jian, Daoye Zhu, Chengqi Cheng, Liesong He, Guoping Liu, Xiang Wen and Runbo Hu

1 Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, China; chiizhang@pku.edu.cn (C.Z.); zhudaoye@pku.edu.cn (D.Z.)
2 Department of Computer Science, The University of Hong Kong, Hong Kong 999077, China; u3007989@connect.hku.hk
3 Institute of Computing Technology, Chinese Academy of Sciences, Beijing 101408, China; qiuxiang@ict.ac.cn
4 Shopee Information Technology Co., Ltd., Shenzhen 518063, China
5 College of Engineering, Peking University, Beijing 100871, China; ccq@pku.edu.cn
6 Xi’an Research Institute of Surveying and Mapping, Xi’an 710000, China; bealiouve@163.com
7 Didi Chuxing, Beijing 100085, China; liuguoping@didiglobal.com (G.L.); wenxiang@didiglobal.com (X.W.); hurunbo@didiglobal.com (R.H.)

* Correspondence: zhichun.jian@shopee.com

Abstract: Due to the periodic and dynamic changes of traffic flow and the spatial–temporal coupling interaction of complex road networks, traffic flow forecasting is highly challenging and rarely yields satisfactory prediction results. In this paper, we propose a novel methodology named the Augmented Multi-component Recurrent Graph Convolutional Network (AM-RGCN) for traffic flow forecasting by addressing the problems above. We first introduce the augmented multi-component module to the traffic forecasting model to tackle the problem of periodic temporal shift emerging in traffic series. Then, we propose an encoder–decoder architecture for spatial–temporal prediction. Specifically, we propose the Temporal Correlation Learner (TCL) which incorporates one-dimensional convolution into LSTM to utilize the intrinsic temporal characteristics of traffic flow. Moreover, we combine TCL with the graph convolutional network to handle the spatial–temporal coupling interaction of the road network. Similarly, the decoder consists of TCL and convolutional neural networks to obtain high-dimensional representations from multi-step predictions based on spatial–temporal sequences. Extensive experiments on two real-world road traffic datasets, PEMSD4 and PEMSD8, demonstrate that our AM-RGCN achieves the best results.

Keywords: traffic forecasting; spatial–temporal prediction; graph convolutional networks; augmented multi-component

1. Introduction

Traffic flow forecasting plays a vital role in Intelligent Transportation Systems (ITSs) [1]. Given a road network, traffic flow forecasting aims to predict the trends of traffic flow in the near future based on historical flow data. As traffic congestion is becoming a serious problem in most cities, how to accurately predict traffic flow is of great significance to transportation management, environmental protection, and public safety.

Traffic flow forecasting is a typical spatial–temporal problem where both spatial features and temporal features should be considered thoroughly. Early forecasting approaches [2,3] mainly made use of traditional statistical models to mine the implicit rules hidden in the data. However, these models are too simple to capture the non-linearity of the traffic series. Contrary to statistical methods, classical machine learning-based approaches [4,5] were able to learn more complex relationships but required careful feature engineering which could be laborious and tedious for human beings. Inspired by advances in deep learning, some attempts [6,7] tried to make predictions based on Convolutional...
Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). However, classical CNNs cannot exploit the non-Euclidean relationship of irregular traffic road networks because of their regular convolutional operations. From this perspective, Graph Convolutional Networks (GCNs) have been introduced to traffic flow forecasting considering their ability to deal with graph data. Moreover, some research works [8–11] combined GCNs with RNNs and CNNs in order to capture spatial and temporal characteristics, respectively.

Although promising advances have been made in traffic flow forecasting, it is still very challenging to achieve highly accurate predictions, mainly due to the two following reasons: (a) the characteristics of periodic temporal shifts in traffic flow are not taken into consideration and (b) the spatial–temporal correlations are not captured effectively.

For the former, most existing approaches [12–17] only paid attention to the periodicity of the traffic flow regardless of periodic temporal shift, which resulted in the non-comprehensive capture of temporal characteristics. Thus, the robustness and accuracy achieved by these models cannot meet expectations. To illustrate, the periodicity of the traffic series is dynamic rather than static because of various factors such as complicated weather or real-time traffic conditions, although daily periodicity and weekly periodicity are commonly recognized as strong contributors to traffic flow forecasting. A typical example of the periodic temporal shift is shown in Figure 1 (data source: Didi’s real-world traffic flow data ranging from 31 October 2019 to 30 November 2019 in Beijing). The daily peak hours in Figure 1a are usually between 6:00 p.m. and 7:00 p.m., but could vary from 5:00 p.m. to 9:00 p.m., depending on whether it is a workday and other factors such as abnormal weather and traffic congestion. Similarly, in Figure 1b, the fluctuation can be observed in weekly numbers. Therefore, it is difficult for current methods to deal with the dynamic and complex situations in actual traffic networks, with only modeling of the static characteristic of periodicity.

![Figure 1. Examples of periodic temporal shift in traffic flow data. The numbers above the circles in the figures denote a time interval (e.g., 6:00 means 6:00–7:00 p.m.).](image)

For the latter, traffic data have tightly coupled spatial–temporal correlations, but recent studies [9,13,18] have not considered the mutual dependence between spatial features and temporal features in traffic flow. They adopted the GCN module to represent the spatial features of the whole traffic network at the same time step, and the CNN module to process temporal features of each road at different time steps. This solution decouples the spatial–temporal correlations and results in the loss of some implicit factors, such as the influence of each road on its surrounding roads at the different time steps. Thus, the accuracy of model prediction is lower than expected.

To address the two above-mentioned challenges, we propose a deep learning-based framework: the Augmented Multi-component Recurrent Graph Convolutional Network (AM-RGCN) for traffic flow forecasting. We first come up with an augmented multi-component module to capture the periodic temporal shift that emerges in the traffic flow series. Then, we present an encoder–decoder architecture where the encoder aims to capture the periodic temporal correlations and the decoder has the ability to obtain high-dimensional representations from multi-step predictions based on spatial–temporal sequences. A fusion module
is finally developed to produce prediction results by incorporating high-dimensional representations. In summary, this paper makes the following contributions:

- We propose an augmented multi-component module to capture the characteristics of the periodic temporal shift in traffic series by adding the temporal shift representations to the periodic representations.
- We propose the Temporal Correlation Learner (TCL) which incorporates one-dimensional convolution into LSTM and combine it with graph convolution in encoder–decoder architecture to handle the spatial–temporal correlations in the road network.
- Extensive experiments on two real-world traffic datasets, PEMSD4 and PEMSD8, verify that our AM-RGCN achieves state-of-the-art results compared with the existing approaches.

The remainder of this article is organized as follows. The related works on traffic flow forecasting are discussed in Section 2. Section 3 introduces our proposed approaches in detail. In Section 4, we conduct comparable experiments with AM-RGCN on real-world traffic datasets and analyze the results. Finally, the conclusions of this study are provided in Section 5.

2. Related Work

2.1. Traffic Flow Forecasting

Traffic flow forecasting has extensively been researched in ITSs. The existing approaches of traffic flow forecasting can be mainly divided into traditional approaches and deep learning approaches. Specifically, the traditional approaches can be further classified into two categories: parametric and non-parametric models [19]. Parametric models, such as Auto-Regressive Integrated Moving Average (ARIMA)-based approaches [2] and Kalman Filtering (KF)-based approaches [3], employ the historical traffic series to statistically mine the implicit rule in time series. However, these approaches are too simple to capture the non-linearity in traffic data. Non-parametric models, such as K-Nearest Neighbor (KNN)-based approaches [5], Support Vector Regression (SVR)-based approaches [4], and Gradient-Boosted Regression Tree (GBRT)-based approaches [20], employ the principle of empirical risk minimization, which may suffer from the overfitting problem.

Deep learning has made great achievements in recent years. Consequently, many researchers apply deep learning approaches to traffic flow forecasting. Some attempts [6,21] applied Long Short-Term Memory (LSTM) [22] and Gated Recurrent Units (GRUs) [23] to traffic flow forecasting and achieved remarkable results. However, these approaches mainly consider the characteristics of time series and neglect the spatial features in traffic networks. In order to handle the spatial features simultaneously, Ma et al. [24] proposed a deep CNN model which exploits a two-dimensional spatial–temporal matrix to capture spatial–temporal information. Yu et al. [25] employed a CNN to obtain the spatial features and then applied LSTM for time series analysis. However, the CNN-based approaches mainly extract the features in Euclidean space rather than the non-Euclidean space. To tackle the problem, GCNs [26] are introduced into traffic flow forecasting. Spatial–Temporal GCNs (STGCNs) [9] and Temporal GCNs (T-GCNs) [27] both adopt GCNs to obtain spatial features and then capture temporal features via one-dimensional CNNs or GRUs, separately. These approaches generally perform better than the approaches without GCNs in prediction. Song et al. [28] proposed the Spatial–Temporal Synchronous GCN (STSGCN) to capture the heterogeneity in spatial–temporal networks and achieved a distinct outcome. Bai et al. [29] integrated GCNs into GRUs to capture spatial–temporal relations simultaneously. Some studies [30–32] attempted to introduce the attention mechanism to solve it.

The GCN-based approaches above mainly use the recent data to represent temporal information while ignoring the periodicity in traffic flow. Roy et al. [14] proposed the Simplified Spatio-temporal Traffic GNN (SST-GNN) to capture the periodic traffic patterns by adopting a novel position encoding scheme. Chen et al. proposed the Temporal Directed GCN (T-DGCN) [15] which utilizes a novel global position encoding strategy to capture temporal dependence such as daily periodicity. Ou et al. [16] proposed Spatial–Temporal
Parallel TrellisNets (STP-TrellisNets) which use the Periodicity TrellisNet (P-TrellisNet) module to capture periodicity in traffic series. These models considered daily periodicity while ignoring the weekly periodicity. Although some scholars proposed the Multi-component STGCN (MSTGCN) [13], Attention-based STGCN (ASTGCN) [18], and Information Geometry and Attention-based GCN (IGAGCN) [17] to represent the daily and weekly periodicity, they left out the impacts of periodic temporal shift. Yao et al. [33] proposed the Spatial-Temporal Dynamic Network (STDN) for periodic temporal shift, which exploited an LSTM network with an attention mechanism to capture the long- and short-term dependencies in traffic series. However, the model is deployed in Euclidean space and focuses on addressing the shifting in daily periodicity. In addition, most of the above-mentioned approaches adopt different models to capture spatial and temporal features separately. Accordingly, they fail to acquire the spatial–temporal correlations effectively.

2.2. Graph Convolution Networks

Traditional approaches [24,25] mainly divided the traffic network into grids and employed CNNs to capture spatial features, which ignored the topological connectivity of traffic networks. Graph convolutional approaches can handle graph-structured data by aggregating the neighbors’ information, which are effective approaches to extract complex spatial topological relationships in traffic networks.

Graph convolutions networks fall into two categories, spectral-based and spatial-based. Spatial-based approaches directly conduct convolution operations on the nodes of the graph. GraphSAGE [34] introduced an aggregation function to define graph convolution. The Graph Attention Network (GAT) [35] exploited attention layers to adjust the importance of each node when applying aggregation functions. Spectral-based approaches [26,36,37] employ a Laplacian matrix to perform convolution operations on graphs in the Fourier domain. According to the selection of convolution kernels, spectral-based approaches mainly include ChebNet [37] and GCNs [26]. GCNs employ ChebNet’s first-order approximation to greatly simplify the parameters of the graph convolution. By stacking multiple GCN layers, the receptive neighborhood range of GCNs can be enlarged.

3. Preliminaries

Given the historical traffic flow data recorded by sensors in the traffic network and the topological graph of the corresponding sensors, the purpose of traffic flow forecasting is to predict the future traffic flow in road networks.

In this study, we define the traffic road network as an undirected graph \( G = (V, E, A) \), where \( V \) is a finite set of \( |V| = N \) nodes denoting the sensors; \( E \) is a set of edges connecting different sensors; \( A \in \mathbb{R}^{N \times N} \) is the adjacency matrix of graph \( G \), representing the connectivity of the whole road network. The adjacency matrix contains only elements of 0 and 1. The element is 1 if there are two sensors directly adjacent on the same road, otherwise the element is 0. The traffic flow observed on \( G \) at time \( t \) is denoted as a graph signal \( X^t \in \mathbb{R}^{N \times F} \), where \( F \) is the feature dimension of each node. The historical traffic flow data at time \( t \) can be defined as \( X = (X^{t-H+1}, X^{t-H+2}, \ldots, X^t) \in \mathbb{R}^{H \times N \times F} \), where \( H \) is the length of historical observation data. The forecasting traffic flow is denoted as \( Y = (X^{t+1}, X^{t+2}, \ldots, X^{t+P}) \in \mathbb{R}^{P \times N \times F}, P \) is the length of forecasting data. Traffic flow forecasting aims to learn a model \( \phi \) that can accurately forecast future \( P \) graph signals given historical \( H \) graph signals of the whole road network: \( (X; G) \xrightarrow{\phi} Y \).

4. Methodology

We propose a general framework named AM-RGCN to address the problem of periodic temporal shift and exploit spatial–temporal correlations. As shown in Figure 2, the proposed AM-RGCN mainly consists of three modules: (1) an augmented multi-component module which intends to capture the characteristics of periodicity and periodic temporal shift synchronously; (2) an encoder module which aims to characterize the spatial–temporal
correlations in traffic flow data; (3) a decoder module which performs multi-step predictions from spatial–temporal sequences.

Figure 2. Overview of AM-RGCN. (1) Augmented multi-component: $X_{h}$, $X_{ds}$, and $X_{ws}$ are the recent component, daily augmented component, and weekly augmented component, separately. Periodicity shift means the dynamic changes in daily and weekly periodicity. (2) Encoder: $X_{t}$ means the $t^{th}$ time slice of augmented multi-component. $A$ denotes adjacency matrix of road network. GCN represents graph convolutional network and TCL denotes temporal correlation learner. (3) Decoder: $\bar{X}_{p}$ denotes $p^{th}$ time slice of forecasting. (4) Fusion module: Conv means convolution with $1 \times 1$ kernel size. $R$ indicates residual information of augmented multi-component module and $\mathcal{F}(X)$ represents the output of decoder. Aggregation denotes the addition operation of $\mathcal{F}(X) + R$.

4.1. Augmented Multi-Component Module for Periodic Temporal Shift

The idea of the multi-component was introduced by Guo et al. [18]. The proposed multi-component module incorporates the recent component, daily periodicity component, and weekly periodicity component of traffic flow data. As shown in Figure 3a, $T_p$ and $t_c$ refer to the predicting window (from 5:00 p.m. to 6:00 p.m. on Thursday) and the current time, respectively. $T_h$, $T_d$, and $T_w$ are numbers of time steps of the above three components describing traffic flow data in different time scales, respectively. Specifically, $T_h = N_h \times T_p$, where $N_h \in \mathbb{N}^+$ represents using the traffic series from the past $N_h$ hour(s). $T_d = N_d \times T_p$, where $N_d \in \mathbb{N}^+$ denotes using the traffic flow of the same period (from 5:00 p.m. to 6:00 p.m.) in the past $N_d$ day(s). $T_w = N_w \times T_p$, where $N_w \in \mathbb{N}^+$ indicates using the traffic records of the same period (from 5:00 p.m. to 6:00 p.m.) in the past $N_w$ Thursday(s).

The augmented multi-component introduces the daily augmented component and weekly augmented component for the daily shift and weekly shift. As shown in Figure 3b, $T_{ds}$ and $T_{ws}$ are the lengths of the daily and weekly augmented component, separately. $S$ denotes periodic offset, indicating that there is a period shift of $S \times T_p$ time steps before and after the daily and weekly periodicity. Specifically, the relationship between the augmented multi-component and multi-component can be expressed as Equation (1):

$$T_{ds} = T_d \times (2 \times S + 1) = N_d \times T_p \times (2 \times S + 1),$$

$$T_{ws} = T_w \times (2 \times S + 1) = N_w \times T_p \times (2 \times S + 1).$$

(1)
The sampling frequency of traffic series is defined as $f$ times a day, then the details of the augmented multi-component are described as follows:

(1) **Recent component**

As shown in Figure 3b, the recent component is the golden part, which represents time series that are the closest to the prediction sequence. Owing to the continuity of traffic flow, we argue that there exist strong correlations within recent moments. The expression of the recent component is described as follows:

$$X_{h} = \left( X^{t_{c}-T_{h}+1}, X^{t_{c}-T_{h}+2}, \ldots, X^{t_{c}} \right) \in \mathbb{R}^{T_{h} \times N \times F},$$

where $N$ represents the number of nodes in the road network, $F$ is the dimension of each node representation, and $X^{t}$ stands for the traffic flow at time $t$.

**Figure 3.** Assume that the time step is 5 min, and the length of forecasting $T_{p}$ is set as 12, which represents the prediction of the traffic flow of the next hour from 5:00 p.m. to 6:00 p.m. on Thursday. (a) The principle of the multi-component in traffic flow: $T_{h} = 24$, $T_{d} = 24$, and $T_{w} = 24$ are all double $T_{p}$, which means using the traffic flow of the past two hours from 3:00 p.m. to 5:00 p.m. and the same period from 5:00 p.m. to 6:00 p.m. of the past two days and past two Thursdays. (b) The principle of the augmented multi-component in traffic flow: $T_{h}$ is set the same as (a). Periodic offset $S$ takes 1, then $S \times T_{p} = 12$, indicating one hour offset in daily and weekly components. According to Equation (1), we can obtain $T_{ds} = 72$ and $T_{ws} = 72$, which means utilizing the traffic data from 4:00 p.m. to 7:00 p.m. of the past two days and past two Thursdays.

(2) **Daily augmented component**

As shown in Figure 3b, the daily component is the green part, which represents data in the same period as the prediction window in the last several days. The impact of the periodic shift is caused by abnormal weather, traffic congestion, and other factors in traffic flow. Consequently, we add offset series, which is the same length as forecasting sequences, before and after daily components, to form the daily augmented component. The component can be expressed as follows and we simplify the result by Equation (1):

$$X_{ds} = \left( X^{t_{c}-N_{d} \times f - S \times T_{p} + 1}, X^{t_{c}-N_{d} \times f + S \times T_{p} + T_{p}}, X^{t_{c}-(N_{d}-1) \times f - S \times T_{p} + 1}, X^{t_{c}-(N_{d}-1) \times f + S \times T_{p} + T_{p}}, \ldots, X^{t_{c}-(N_{d}-1) \times f - S \times T_{p} + T_{p}}, X^{t_{c}} \right) \in \mathbb{R}^{T_{ds} \times N \times F}.$$
Suppose the time step is 5 min and we wish to predict the traffic flow of the next hour \( (T_p = 12) \) from 5:00 p.m. to 6:00 p.m. on Thursday. \( f = 288 \) is the sampling frequency of a day. Let \( S = 1 \) and \( T_d = 24 \), then \( N_d = 2 \). The Equation (3) means we use the traffic flow from 4:00 p.m. to 7:00 p.m. on the most recent Wednesday and Thursday. For instance, \( (X^{1}_t = \text{traffic flow from 4:00 p.m. to 7:00 p.m. on the Thursday two weeks ago while } X^{2}_t = \text{traffic flow from 4:00 p.m. to 7:00 p.m. on the most recent Wednesday}) \) can be taken as the traffic flow from 4:00 p.m. to 7:00 p.m. on the most recent Tuesday while \( (X^{3}_t = \text{traffic flow from 4:00 p.m. to 7:00 p.m. on the most recent Tuesday and Wednesday}) \) is the traffic flow from 4:00 p.m. to 7:00 p.m. on the most recent Wednesday.

(3) Weekly augmented component

As shown in Figure 3b, the weekly component is the red part, which represents data in the same period as the prediction window in the last several weeks. We add an offset series, which is the same length as the forecasting sequence, before and after the weekly component to form the weekly augmented component. Similarly, it can be expressed as:

\[
X^{aw} = (X^{1}_t - N_w \ast f + T_p + 1, \ldots, X^{1}_t - N_w \ast f + S \ast T_p + T_p), X^{2}_t - (N_u - 1) \ast f + T_p + 1, \ldots, X^{2}_t - (N_u - 1) \ast f + S \ast T_p + T_p), \ldots, X^{3}_t - f + S \ast T_p + 1, \ldots, X^{3}_t - f + S \ast T_p + T_p) \in \mathbb{R}^{T_{aw} \ast N \ast F}.
\]

Assume that we have the same setting as described in the daily augmented component. Let \( S = 1 \) and \( T_w = 24 \), then \( N_w = 2 \). Equation (4) indicates that we adopt the traffic records of 4:00 p.m. to 7:00 p.m. from the past two Thursdays. Accordingly, \( (X^{1}_t = \text{traffic flow from 4:00 p.m. to 7:00 p.m. on the Thursday two weeks ago while } X^{2}_t = \text{traffic flow from 4:00 p.m. to 7:00 p.m. on the Thursday two weeks ago}) \) can be taken as the traffic flow from 4:00 p.m. to 7:00 p.m. on the Thursday two weeks ago while \( (X^{3}_t = \text{traffic flow from 4:00 p.m. to 7:00 p.m. on the Thursday two weeks ago}) \) is the traffic flow from 4:00 p.m. to 7:00 p.m. on the last Thursday.

The above three components jointly make up the augmented multi-component module, which takes into account the periodicity and periodic temporal shift in traffic forecasting. Let \( T = T_h + T_{ds} + T_{aw} \) be the length of the augmented multi-component module, and the input data \( X_{am} = (X^{h}_t, X^{ds}_t, X^{aw}_t) \in \mathbb{R}^{T \ast N \ast F} \) are passed to the encoder–decoder architecture.

4.2. Encoder for Spatial–Temporal Correlations

The encoder module is designed to exploit spatial–temporal correlations. As shown in Figure 2, it is composed of a GCN and TCL, both of which are employed to learn the spatial–temporal representations from the augmented multi-component series.

4.2.1. Graph Convolution in Spatial Dimension

As illustrated in Figure 4a, the traffic network is a typical graph structure and the neighbors’ traffic flow of each sensor is essential for forecasting. We choose a GCN to capture spatial topological relationships. As briefly illustrated in Figure 4b, the GCN model can obtain the topological relationship between the central sensor and its first-order surrounding sensors and embed the traffic flow attributes in the network. The two-layer GCN model can be expressed as:

\[
f(X_t, A) = \text{ReLU}(\tilde{A}(\tilde{A}X_t W_0)W_1),
\]

where \( X_t \in \mathbb{R}^{N \ast F} \) denotes the characteristics of the road network at each time slice \( t \in \{1, \ldots, T\} \); \( \tilde{A} = D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} \in \mathbb{R}^{N \ast N} \) indicates the renormalization trick; \( \tilde{A} = A + I \in \mathbb{R}^{N \ast N} \) means adding a self-loop to the adjacency matrix; \( D = \sum_i \tilde{A}_{ij} \in \mathbb{R}^{N \ast N}, W_0 \in \mathbb{R}^{F \ast H} \) and \( W_1 \in \mathbb{R}^{H \ast C} \) represent the parameter matrix from the input feature dimension \( F \) to the output feature dimension \( H \) and \( C \), separately. ReLU is the activation function.

To take full advantage of the topological information, we exploit a two-layer shared weight GCN to capture the spatial features of the traffic network at each time slice.
Figure 4. (a) Assuming that the red sensor with the number 0 stands for the center node of the road network, the green sensors with number 1 are the first-order neighbors of the center node, and the blue sensors with number 2 are the second-order neighbors of the center node. (b) The adjacency relation of sensors in (a) are simplified as nodes. The GCN represents the spatial features through the topological relationship between the center node and first-order neighbors.

4.2.2. Temporal Correlation Learner (TCL) in Temporal Dimension

The traffic flow data are mainly the three-dimensional input of nodes, sequences, and features. In the previous section, we exploited GCNs to represent the mutual spatial correlations among all sensors along the node dimension. In TCL blocks, we first append one-dimensional convolution to integrate the internal characteristic of each sensor in the feature dimension, and then adopt LSTM in the sequence dimension for temporal features. Our proposed TCL is partly inspired by Convolutional LSTM (ConvLSTM) [38]. As illustrated in Figure 5, we denote the spatial representations extracted from GCNs at each moment as $G_t \in \mathbb{R}^{N \times C}$. The TCL first exploits one-dimensional convolution to integrate the spatial characteristic of each sensor based on the previous hidden state $H_{t-1} \in \mathbb{R}^{N \times H}$ and $G_t$, then passes it to the LSTM together with the previous cell memory state $C_{t-1} \in \mathbb{R}^{N \times H}$ to learn the temporal features, where $H$ is the hidden size. Specifically, we initialize all the states of the LSTM to zero before the first input comes. During the training process, we add zero-padding to the hidden states before applying convolutional operations. The kernel size and padding size of the one-dimensional convolution are 3 and 1, separately.

To ensure that spatial and temporal features can be learned simultaneously, we input spatial features $G_t$ as the source of the TCL at each moment of traffic flow series, which helps the model to learn spatial–temporal correlations. Thus, at time step $t$, the computation process of the proposed TCL can be simplified as:

$$H_t, C_t = \text{TCL}(G_t; H_{t-1}; C_{t-1}),$$

where $t \in \{1, \ldots, T\}$. We pass $G_t$ into the TCL and update the cell memory state $C_t$ using the input gate $I_t$, the forget gate $F_t$, and the previous hidden state $H_{t-1}$. Finally, we employ the output gate $O_t$ to update the current hidden state $H_t$. We can summarize the computation process above as follows:

$$I_t = \sigma(W_{gi} * G_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1}),$$
$$F_t = \sigma(W_{gf} * G_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1}),$$
$$C_t = F_t \odot C_{t-1} + I_t \odot \tanh(W_{gc} * G_t + W_{hc} * H_{t-1}),$$
$$O_t = \sigma(W_{go} * G_t + W_{ho} * H_{t-1} + W_{co} \odot C_t),$$
$$H_t = O_t \odot \tanh(C_t),$$

(7)
where $W_{\alpha\beta}(\alpha \in \{g,h,c\}, \beta \in \{i,f,c,o\})$ denotes the learnable parameters in TCL; $\sigma$ and $\tanh$ denote the activation function; $\ast$ represents the one-dimensional convolution operation; $\odot$ denotes the Hadamard product.

The encoder module then passes the final output state $H_T \in \mathbb{R}^{N \times H}, C_T \in \mathbb{R}^{N \times H}$ of TCL including the spatial–temporal features of traffic flow to the decoder.

Figure 5. The architecture of the TCL. At each moment, the spatial feature extracted from the GCN is passed to the learner with the previous hidden state and cell memory state.

### 4.3. Decoder for Multi-Step Prediction

The decoder module is mainly used for multi-step prediction. As shown in Figure 2, it is composed of the TCL and CNN, employing the hidden states obtained from the encoder to produce high-dimensional feature representations from spatial–temporal sequences.

In our decoder, TCL is adopted to unfold the hidden state $H_T$ and the cell memory state $C_T$ from the encoder. Additionally, since there are no input sequences in the TCL, we initialize an all-zero array with the same dimension as the hidden state $H_T$ as the input for simplification. Specifically, at each moment, we employ the hidden state and cell memory state from the previous moment and the all-zero array to forecast the next moment. The purpose is to ensure that the prediction result at each moment is related to the previous moment. Assuming that the size of the prediction window is $P$, the expression of the decoder is:

$$X_{t+1} = \text{TCL} \left( \vec{0}; H_t; C_t \right),$$

where $t \in \{T, \ldots, T + P - 1\}$ and $\vec{0}$ denote the all-zero arrays. We then concatenate $\{X_T, \ldots, X_{T+P}\} \in \mathbb{R}^{P \times N \times H}$ and apply a convolutional operation to convert the multi-step predictions into high-dimensional representations.

The representations obtained from the above augmented multi-component module and encoder–decoder architecture are passed to the fusion module, which consists of a residual connection and CNN, to produce prediction results. Concretely, the fusion module utilizes a convolutional residual connection to integrate the residual information $R$ from the augmented multi-component module with the high-dimensional representations $F(X)$ of the decoder, aiming to speed up the model training and mitigate the overfitting problem. Eventually, a CNN is adopted to guarantee that the predictions $Y \in \mathbb{R}^{P \times N \times F}$ have the same dimensions and shapes as expected.

## 5. Experiment

To evaluate the performance of our model, comparative experiments on two real-world traffic datasets are carried out. Moreover, we carry out ablation studies to demonstrate the effectiveness of different modules.

### 5.1. Datasets

The public traffic datasets PEMSD4 and PEMSD8 are the real highway traffic datasets collected by the California Transportation Agency Performance Measurement System (PeMS). The system is displayed on a map and has more than 39,000 independent sensors deployed on the highway system across all major metropolitan areas of the state of California. The observations of the sensors are aggregated into 5-min windows and the geographic information of the sensors is also included.
We use the popular benchmarks of PEMSD4 and PEMSD8 released by Guo et al. [18] which remove redundant sensors whose distance is less than 3.5 miles and adopt linear interpolation for missing values. The details of the datasets are described in Table 1: (1) PeMSD4 records two months of statistics on traffic flow in the San Francisco Bay Area, ranging from 1 January 2018 to 28 February 2018, including 307 sensors. We choose data from the first 50 days as the training set and validation set, and the remaining 9 days as the test set. (2) PeMSD8 contains two months of statistics on traffic flow in the San Bernardino area, ranging from 1 July 2016 to 31 August 2016, including 170 sensors. We select data from the first 50 days as the training set and validation set, and the remaining 12 days as the test set. In addition, we preprocess the dataset by calculating the maximum value \( \text{Max}(X) \) in the dataset and normalizing the entire dataset by \( X = \frac{X}{\text{Max}(X)} \).

Table 1. Dataset description and statistics.

| Datasets  | Nodes | Edges | Interval | Time Range               | Time Steps |
|-----------|-------|-------|----------|--------------------------|------------|
| PEMSD4    | 307   | 340   | 5 min    | 1 January 2018–28 February 2018 | 16,992     |
| PEMSD8    | 170   | 295   | 5 min    | 1 July 2016–31 August 2016  | 17,856     |

5.2. Model Parameter

All experiments are compiled and tested on a Linux cluster (CPU: 6 Intel Core Processor (Broadwell), GPU: NVIDIA Tesla P40). The model parameters can be divided into three parts: (1) augmented multi-component: in this study, we focus on predicting the traffic flow of the next hour, namely, \( T_p = 12 \). For \( T_p = 6 \) or 3, we use the model parameter of \( T_p = 12 \) for training efficiency. Due to a trade-off between prediction accuracy and computational efficiency in the experiment, we set the three component parameters as \( T_h = 24 \), \( T_d = 12 \), and \( T_w = 12 \) and the periodic offset as \( S = 1 \) for both datasets. Moreover, the augmented intervals cover the range of periodic temporal shifts of peak hours shown in Figure 1. Consequently, the length of the augmented multi-component sequence is \( T_D = 96 \); (2) network structure: the encoder uses two layers of the GCN, whose convolution filters are 128 and 64, separately. The convolution filters of the TCL are consistent with the number of sensors and there are 64 hidden units. In the decoder, there are 64 hidden units of the TCL while the output sequences are \( T_p \) and the convolution filters of the CNN are set as \( T_D \). In the fusion module, the convolution filters are \( T_p \); (3) training hyperparameters: we train our model using the Adam optimizer [39] with a learning rate of 0.001 and weight decay of \( 5 \times 10^{-4} \). We set the dropout [40] as 0.8. In this paper, we apply a mean square error (MSE) between the estimator and the ground truth as the loss function.

5.3. Evaluation Metric

We adopt the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics:

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2},
\]

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_i|
\]

where \( Y_i \) means ground truth, \( \hat{Y}_i \) is the predicted traffic flow, and \( n \) is the number of all predicted values.

5.4. Baseline

We compare our model with the following baselines:

- Historical Average (HA). We use the average of the past 12 time slices in the same period as a week ago to forecast the current time slice.
• ARIMA [2]. A typical traditional forecasting model for time series. We set the auto-regressive coefficient \( p = 0 \), difference coefficient \( d = 0 \), moving average coefficient \( q = 1 \).
• LSTM [22]. A special RNN model for time series prediction. We set historical traffic flow \( T_h = 12 \) and the hidden size \( h = 64 \).
• Gated Recurrent Unit (GRU) network [23]. An improved RNN model for time series prediction. We set historical traffic flow \( T_h = 12 \) and the hidden size as 64.
• STGCN [9]. The model employs one-dimensional convolution and graph convolution to extract spatial–temporal features, which are widely used in traffic flow forecasting. Both the graph convolution kernel size \( K_s \) and temporal convolution kernel size \( K_t \) are set to 3 in the experiments.
• MSTGCN [13]. A multi-component network for traffic flow forecasting. The best combinations adopted in this paper are \( T_h = 36 \), \( T_d = 12 \), and \( T_w = 12 \).
• ASTGCN [18]. A traffic flow forecasting model, which adds spatial–temporal attention to the MSTGCN. The best combinations adopted in this paper are \( T_h = 24 \), \( T_d = 12 \), and \( T_w = 24 \).
• STSGCN [28]. A traffic forecasting model which attempts to capture the complex localized spatial–temporal correlations in spatial–temporal data. The best setting consists of 4 STSGCLs, each STSGCM contains 3 graph convolutional operations with 64, 64, 64 filters, separately.

5.5. Results and Analysis

Overall, in the baseline comparison, our model achieves the best performance in PEMSD4 and PEMSD8, compared to existing traditional methods and deep learning methods. Then, the augmented multi-component method and TCL module are evaluated and proved to be effective in the following aspects: (1) the augmented multi-component method outperforms the multi-component method and is conducive to capturing the periodic temporal shift in traffic flow; (2) our TCL module performs better than its variants and can learn the spatial–temporal correlations effectively.

5.5.1. Baseline Comparison

Table 2 presents the performances of the AM-RGCN and baseline models for 15 min (three time slices), 30 min (six time slices), and 1 h (12 time slices) ahead predictions on two datasets. As shown, our AM-RGCN performs the best on both datasets in terms of all evaluation metrics.

The forecasting performances of traditional baselines (HA and ARIMA) are the worst, limited by their abilities to capture spatial–temporal characteristics from complex time series data. Comparatively, deep learning approaches (GCN- and RNN-based models) outperform them by large margins with their ability to learn from non-linear traffic data. In the one-hour traffic forecasting task on dataset PEMSD8, even the deep learning model with the worst performance (LSTM) works better than the best selected traditional method (HA). Statistically, the former reduces the RMSE and MAE errors by approximately 29.9% and 22.5% compared to the latter. In deep learning approaches, GCN-based models (STGCN, MSTGCN, ASTGCN, STSGCN, and AM-RGCN) generally perform better than RNN-based models (LSTM and GRU). The superior GCN-based models include graph convolution to extract spatial–temporal characteristics, while the latter only considers temporal features.

Among GCN-based models, STGCN is the worst one since it only uses the recent component to capture temporal features and is restricted by its lack of periodicity characteristics. Better methods, such as MSTGCN and ASTGCN, are able to capture daily and weekly periodicity with the multi-component method. Thus, they improve performances of RMSE and MAE by 12.6%, 9.5% and 13.5%, 10.2% in one-hour prediction on PEMSD4. However, these GCN-based models are not efficient in correlation recognition, since they utilize GCN and 1D-CNN modules to model the spatial and temporal characteristics separately. Comparatively, the STSGCN takes the localized spatial–temporal correlations into
account and is superior to the STGCN too, but its neglect of the periodicity prevents its performance from completely surpassing MSTGCN and ASTGCN in all intervals and metrics. The above results of four GCN-based models prove the significance of periodicity and spatial–temporal correlations in traffic flow forecasting. However, their performances are limited by the inability to synchronously handle static periodicity characteristics and spatial–temporal correlations, as well as to introduce the dynamic periodicity shift.

In contrast, our AM-RGCN employs the augmented multi-component to grasp periodic offset characteristics, and it combines the TCL with GCN at each moment to learn the spatial–temporal correlations. It is compared with ASTGCN (the previous state-of-the-art model) in the next hour’s prediction based on the same model parameters (about 1.4M). The results demonstrate that AM-RGCN decreases the RMSE and MAE by 6.3% and 8.0% on PEMSD8, and 8.0% and 9.2% on PEMSD4, although it takes more time (4.5 ms) than ASTGCN (2.4 ms) for one forward iteration at inference due to the employment of a recurrent structure. According to the above comparisons, our model has more advantages in expressing the spatial and temporal characteristics of traffic series.

### Table 2. Performance comparison of various models at different forecasting intervals on PeMSD4 and PeMSD8.

| Data   | Method     | 15 min |        | 30 min |        | 1 h    |        |
|--------|------------|--------|--------|--------|--------|--------|--------|
|        |            | RMSE   | MAE    | RMSE   | MAE    | RMSE   | MAE    |
| PEMSD8 | HA         | 40.14  | 23.15  | 41.49  | 24.64  | 46.37  | 29.20  |
|        | ARIMA [2]  | 28.96  | 27.77  | 30.38  | 29.59  | 48.33  | 44.25  |
|        | LSTM [22]  | 26.02  | 17.95  | 28.35  | 19.68  | 32.56  | 22.61  |
|        | GRU [23]   | 25.92  | 17.97  | 28.35  | 19.71  | 31.80  | 22.18  |
|        | STGCN [9]  | 24.58  | 16.33  | 27.31  | 17.91  | 31.24  | 20.85  |
|        | MSTGCN [13]| 22.38  | 15.15  | 23.90  | 16.09  | 25.46  | 17.11  |
|        | ASTGCN [18]| 21.81  | 14.76  | 23.33  | 15.71  | 24.40  | 16.33  |
|        | STSGCN [28]| 21.93  | 14.20  | 23.71  | 15.28  | 26.05  | 16.67  |
|        | AM-RGCN    | 20.43  | 13.54  | 21.77  | 14.58  | 22.87  | 15.03  |

| PEMSD4 | HA         | 45.40  | 28.88  | 46.96  | 30.40  | 53.20  | 35.59  |
|        | ARIMA [2]  | 36.91  | 33.71  | 46.65  | 41.36  | 52.32  | 47.74  |
|        | LSTM [22]  | 34.00  | 22.02  | 35.81  | 23.34  | 38.81  | 25.58  |
|        | GRU [23]   | 34.17  | 22.05  | 35.88  | 23.45  | 38.84  | 25.83  |
|        | STGCN [9]  | 32.77  | 21.34  | 34.07  | 21.78  | 37.42  | 24.32  |
|        | MSTGCN [13]| 28.97  | 19.40  | 30.61  | 20.49  | 32.71  | 22.01  |
|        | ASTGCN [18]| 29.19  | 19.59  | 30.26  | 20.32  | 32.37  | 21.83  |
|        | STSGCN [28]| 29.74  | 18.52  | 31.52  | 19.73  | 33.63  | 21.06  |
|        | AM-RGCN    | 27.22  | 18.00  | 28.25  | 18.65  | 29.79  | 19.82  |

#### 5.5.2. Effects of Augmented Multi-Component Module

Firstly, to investigate the effectiveness of our proposed augmented multi-component module (AM-RGCN), we conduct comparison experiments with multi-component models (MSTGCN, ASTGCN) which only consider the periodicity in traffic series. To control variables, the ranges of the multi-component and augmented multi-component are set as $T_h = 24$, $T_d = 12$, $T_w = 12$ and $T_{hd} = 24$, $T_{ds} = 36$, $T_{ws} = 36$, respectively, where the periodic offset $S = 1$. The setting indicates that the periodic temporal offset is one hour. The experimental results are shown in Figure 6. We can observe that each approach using the augmented multi-component performs better than that with the multi-component. For example, for the one-hour prediction in PEMSD4 and PEMSD8, MSTGCN, ASTGCN, and AM-RGCN all achieve a better performance when replacing the multi-component with the augmented multi-component. We argue that the factors which actually lead to the worse predictions in the multi-component, such as weather and traffic conditions, are not included in the period interval range. Comparatively, the augmented multi-component
could cover the above factors and capture the characteristics of periodicity and periodic temporal shifts synchronously by augmenting the data range of each period module.

Then, we conduct ablation experiments to further explore the contribution of each component in the augmented multi-component in PEMSD8 (Table 3), and following experimental results are observed: (1) the model equipped with $X_h$ significantly outperforms those with $X_{ds}$ and $X_{ws}$, enhancing the effect by 31.6% and 32.2% and 28.1% and 23.9% for RMSE and MAE, respectively. This indicates that time series forecasting depends primarily on the recent time slices when considering only one component; (2) compared with single $X_h$, the performance can be improved by combining the $X_h$ with $X_{ds}$ or $X_{ws}$. Moreover, the best performance can be achieved when the model is equipped with all components, because increasing the daily augmented multi-component and weekly augmented multi-component can help with modeling the periodicity and periodic temporal shift, compared to only applying the recent component for short-term dependency. Thus, these two experiments support the superiority of our augmented multi-component method in handling the problem of periodic temporal shift.
Table 3. The ablation experiments of each component for predicting the next hour’s traffic flow on PEMSD8.

| Method | $X_h$ | $X_{ds}$ | $X_{ws}$ | RMSE | MAE |
|--------|-------|----------|----------|------|-----|
| AM-RGCN | ✓     | ✓        | ✓        | 36.61 | 25.08 |
|         |       |          |          | 34.80 | 22.34 |
|         | ✓     |          |          | 25.03 | 17.00 |
|         |        | ✓        |          | 32.91 | 22.13 |
|         |       |          | ✓        | 24.56 | 16.85 |
|         | ✓     |          |          | 24.53 | 16.71 |
|         |        |          | ✓        | 22.87 | 15.03 |

5.5.3. Effects of Temporal Correlation Learner

To further verify the advantages of the TCL, we compare AM-RGCN with its variants which replace the TCL with a CNN or LSTM, while all of them are equipped with the same augmented multi-component module. According to the experimental results in Table 4, we can draw several conclusions:

Firstly, AM-LSTM-GCN does not perform as well as AM-CNN-GCN. We suggest the underlying reason is that errors are accumulated when this model generates predictions of multiple steps ahead via a step-by-step approach. To be more specific, if the prediction length is $P$, then we need to loop the LSTM units $P$ times, which results in the accumulation of error in each time. Oppositely, AM-CNN-GCN directly avoids error propagation by employing a CNN to map the final temporal prediction length to $P$.

Secondly, AM-RGCN is superior to AM-CNN-GCN, owing to its combination of a GCN and TCL in the encoder network to capture the spatial–temporal correlations. Concretely, at each predicted moment, it considers the spatial–temporal information of the last time step to achieve a continuous forecast. In contrast, AM-CNN-GCN decouples the correlations between spatial and temporal features and leaves out the influence of each sensor on its surrounding sensors at different time steps in traffic networks.

As for the comparison of AM-RGCN and AM-LSTM-GCN, our model still maintains its significant primacy. This is because the solution to combine one-dimensional convolution and LSTM works effectively when handling spatial topological features from the GCN, while AM-LSTM-GCN flattens it due to the ability restriction. In detail, the flattening operation merges the spatial characteristics and other characteristics into one dimension, which results in the loss of spatial characteristics from the GCN. Accordingly, AM-RGCN is better than the other two variants at processing spatial–temporal correlations.

Table 4. The advantage of TCL in AM-RGCN compared with the variants replacing TCL with CNN or LSTM for predicting next hour traffic flow on PEMSD8.

| Method   | RMSE | MAE | Dataset |
|----------|------|-----|---------|
| AM-CNN-GCN | 24.31 | 16.00 |         |
| AM-LSTM-GCN | 26.85 | 18.19 | PEMSD8  |
| AM-RGCN   | 22.87 | 15.03 |         |

6. Conclusions and Future Work

We propose the Augmented Multi-component Recurrent Graph Convolutional Network (AM-RGCN) to perform traffic flow forecasting. Specifically, we introduce the augmented multi-component module to capture the periodic temporal shift emerging in traffic series. Then, we implement an encoder–decoder architecture where the encoder aims to capture the spatial–temporal correlations and the decoder is designed to obtain high-dimensional representations of multi-step predictions.
In fact, the graph structure adopted in this paper is an undirected graph. In practice, the graph structure of the road network is directed and the conditions in the traffic network are dynamically changed. In the future, we will pay attention to improving the performance of traffic flow forecasting using a directed graph and attention mechanism. However, our AM-RGCN is a general spatial–temporal forecasting framework for the graph structure data, thus it can also be applied to other spatial–temporal prediction tasks, such as traffic speed forecasting.

**Author Contributions:** Conceptualization, Chi Zhang and Zhichun Jian; Formal analysis, Guoping Liu; Funding acquisition, Chengqi Cheng; Investigation, Daoye Zhu and Liesong He; Methodology, Chi Zhang, Hong-Yu Zhou, and Zhichun Jian; Project administration, Chengqi Cheng, Xiang Wen, and Runbo Hu; Software, Chi Zhang, Qiang Qiu, and Daoye Zhu; Supervision, Chengqi Cheng; Validation, Qiang Qiu, Liesong He, and Guoping Liu; Visualization, Chi Zhang; Writing—original draft, Chi Zhang; Writing—review and editing, Hong-Yu Zhou and Zhichun Jian. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Science and Technology Major Special Project of Guangxi, China (GUIKEAA18118025).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data and code have been released at https://github.com/ILoveStudying/AM-RGCN (accessed on 26 January 2022).

**Acknowledgments:** The authors thankfully acknowledge the financial support provided by the Science and Technology Major Special Project of Guangxi, China (GUIKEAA18118025). We also thank Linhao Cai, Meng Chen, and Shaozhe Liu for many helpful comments, discussions, and suggestions.

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

**References**

1. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* 2014, 16, 865–873.
2. Levin, M.; Tsao, Y.D. On Forecasting Freeway Occupancies and Volumes. *Transp. Res. Rec.* 1980, 173, 47–49.
3. Okutani, I.; Stephanedes, Y.J. Dynamic Prediction of Traffic Volume through Kalman Filtering Theory. *Transp. Res. Part B Methodol.* 1984, 18, 1–11.
4. Wu, C.H.; Ho, J.M.; Lee, D.T. Travel-Time Prediction with Support Vector Regression. *IEEE Trans. Intell. Transp. Syst.* 2004, 5, 276–281.
5. Zhang, L.; Liu, Q.; Yang, W.; Wei, N.; Dong, D. An Improved K-nearest Neighbor Model for Short-Term Traffic Flow Prediction. *Procedia-Soc. Behav. Sci.* 2013, 96, 653–662.
6. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data. *Transp. Res. Part C* 2015, 54, 187–197.
7. Rui, F.; Zuo, Z.; Li, L. Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; pp. 324–328. doi:10.1109/YAC.2016.7804912.
8. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
9. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3634–3640.
10. Li, F.; Feng, J.; Yan, H.; Jin, G.; Jin, D.; Li, Y. Dynamic Graph Convolutional Recurrent Network for Traffic Forecasting: Benchmark and Solution. *arXiv* 2021, arXiv:2104.14917.
11. Yang, T.; Tang, X.; Liu, R. Dual Temporal Gated Multi-graph Convolution Network for Taxi Demand Prediction. *Neural Comput. Appl.* 2021, 1–16. https://doi.org/10.1007/s00521-021-06092-6.
12. Zhang, J.; Zheng, Y.; Qi, D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 1655–1661.
13. Feng, N.; Guo, S.; Song, C.; Zhu, Q.; Wan, H. Multi-component Spatial-Temporal Graph Convolution Networks for Traffic Flow Forecasting. *J. Softw.* 2019, 30, 759–769.
14. Roy, A.; Roy, K.K.; Ahsan Ali, A.; Amin, M.A.; Rahman, A. 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, Delhi, India, 11–14 May 2021; pp. 90–102.

15. Chen, K.; Deng, M.; Shi, Y. A Temporal Directed Graph Convolution Network for Traffic Forecasting Using Taxi Trajectory Data. *ISPRS Int. J. Geo-Inf.* 2021, **10**, 624.

16. Ou, J.; Sun, J.; Zhu, Y.; Jin, H.; Liu, Y.; Zhang, F.; Huang, J.; Wang, X. STP-Trellisnets: Spatial-Temporal Parallel Trellisnets for Metro Station Passenger Flow Prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 1185–1194.

17. An, J.; Guo, L.; Liu, W.; Fu, Z.; Ren, P.; Liu, X.; Li, T. IGAGCN: Information Geometry and Attention-based Spatiotemporal Graph Convolutional Networks for Traffic Flow Prediction. *Neural Netw.* 2021, **143**. https://doi.org/10.1016/j.neunet.2021.05.035.

18. 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, Hilton Hawaiian Village, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929.

19. Miglani, A.; Kumar, N. Deep Learning Models for Traffic Flow Prediction in Autonomous Vehicles: A Review, Solutions, and Challenges. *Veh. Commun.* 2019, **20**, 100184.

20. Zheng, L.; Yang, J.; Chen, L.; Sun, D.; Liu, W. Dynamic Spatio-Temporal Feature Optimization with ERI Big Data for Short-Term Traffic Flow Prediction. *Neurocomputing* 2020, **412**, 339–350.

21. Oliveira, D.D.; Rampinelli, M.; Tozatto, G.Z.; Andreão, R.V.; Müller, S.M. Forecasting Vehicular Traffic Flow using MLP and LSTM. *Neural Comput. Appl.* 2021, **33**, 17245–17256.

22. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* 1997, **9**, 1735–1780.

23. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* 2014, arXiv:1412.3555.

24. Ma, X.; Zhuang, D.; Zhengbing, H.; Jihui, M.; Yong, W.; Yumpheng, W. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors* 2017, **17**, 818.

25. Yu, H.; Zhuhai, W.; Shuqin, W.; Yumpheng, W.; Xiaoalei, M. Spatiotemporal Recurrent Traffic Prediction in Transportation Networks. *Sensors* 2017, **17**, 1501.

26. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

27. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* 2019, **21**, 3848–3858. https://doi.org/10.1109/TITS.2019.2935152.

28. 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.

29. Bai, J.; Zhu, J.; Song, Y.; Zhao, L.; Hou, Z.; Du, R.; Li, H. A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting. *ISPRS Int. J. Geo-Inf.* 2021, **10**, 485.

30. 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; pp. 3529–3536. https://doi.org/10.1609/aaai.v34i04.5758.

31. 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. https://doi.org/10.1609/aaai.v34i01.5477.

32. 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, Online, 19–23 October 2020; pp. 1853–1862. https://doi.org/10.1145/3340531.3411941.

33. 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, Hilton Hawaiian Village, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5668–5675.

34. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1025–1035.

35. Velicković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

36. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral Networks and Locally Connected Networks on Graphs. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.

37. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with FastLocalized Spectral Filtering. *Adv. Neural Inf. Process. Syst.* 2016, **29**, 3844–3852.

38. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Adv. Neural Inf. Process. Syst.* 2015, **802–810.**
39. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* 2014, arXiv:1412.6980.

40. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 2014, 15, 1929–1958.