

## Article

# Gated Recurrent Graph Convolutional Attention Network for Traffic Flow Prediction

Xiaoyuan Feng <sup>1</sup>, Yue Chen <sup>1</sup>, Hongbo Li <sup>1</sup>, Tian Ma <sup>2</sup> and Yilong Ren <sup>1,3,4,\*</sup> <sup>1</sup> School of Transportation Science and Engineering, Beihang University, Beijing 102206, China<sup>2</sup> School of Automation Science and Engineering, Beihang University, Beijing 100191, China<sup>3</sup> Beihang Hangzhou Innovation Institute Yuhang, Hangzhou 310023, China<sup>4</sup> Zhongguancun Laboratory, Beijing 100094, China

\* Correspondence: yilongren@buaa.edu.cn

**Abstract:** Traffic flow prediction is an important function of intelligent transportation systems. Accurate prediction results facilitate traffic management to issue early congestion warnings so that drivers can avoid congested roads, thus directly reducing the average driving time of vehicles, which means less greenhouse gas emissions. However, traffic flow data has complex spatial and temporal correlations, which makes it challenging to predict traffic flow accurately. A Gated Recurrent Graph Convolutional Attention Network (GRGCAN) for traffic flow prediction is proposed to solve this problem. The model consists of three components with the same structure, each of which contains one temporal feature extractor and one spatial feature extractor. The temporal feature extractor first introduces a gated recurrent unit (GRU) and uses the hidden states of the GRU combined with an attention mechanism to adaptively assign weights to each time step. In the spatial feature extractor, a node attention mechanism is constructed to dynamically assign weights to each sensor node, and it is fused with the graph convolution operation. In addition, a residual connection is introduced into the network to reduce the loss of features in the deep network. Experimental results of 1-h traffic flow prediction on two real-world datasets (PeMSD4 and PeMSD8) show that the mean absolute percentage error (MAPE) of the GRGCAN model is as low as 15.97% and 12.13%, and the prediction accuracy and computational efficiency are better than the baselines.



**Citation:** Feng, X.; Chen, Y.; Li, H.; Ma, T.; Ren, Y. Gated Recurrent Graph Convolutional Attention Network for Traffic Flow Prediction. *Sustainability* **2023**, *15*, 7696. <https://doi.org/10.3390/su15097696>

Academic Editors: Wei Liu and Chia-Huei Wu

Received: 21 March 2023

Revised: 28 April 2023

Accepted: 4 May 2023

Published: 8 May 2023



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

**Keywords:** traffic flow prediction; graph convolutional networks; attentional mechanisms

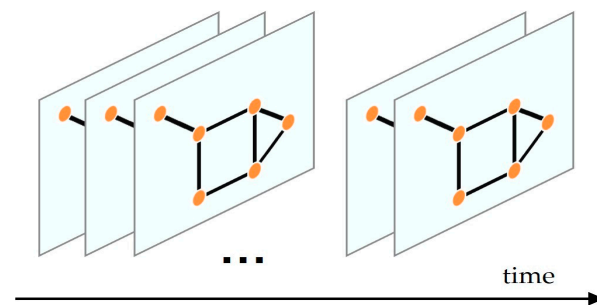
## 1. Introduction

In the urbanization process of countries all over the world, the holdings of cars have been rising [1]. While private cars have brought convenience to the lives of residents, they have also created serious traffic congestion problems and contributed to higher greenhouse gas emissions [2]. To solve this problem, many countries began to promote the construction of intelligent transportation systems (ITS) [3].

ITS is an integrated system that applies advanced communication, control, sensing, and computer technologies to solve traffic management and control problems [4]. The primary goal of ITS is to provide a safe, efficient, and reliable transportation environment for traffic participants [5]. In addition, ITS also has important positive effects on the natural environment by promoting transportation technology innovation and reducing greenhouse gas emissions [6,7]. Take traffic flow prediction as an example; as one of the main tasks of ITS [8], accurate traffic flow prediction facilitates traffic management to release congestion warning early so that drivers can avoid congested roads, thus directly reducing the average driving time of vehicles, which means less greenhouse gas emissions [9].

The essence of traffic prediction is to extract the embedded characteristics of the region through the geographical information of the road network and historical traffic data and to predict the traffic flow in the future period accordingly [8]. With the emphasis on traffic data, many sensors are deployed on the roads. The dataset consisting of time series data

collected by the sensors and the geographic location of the sensors provides a solid data basis for the field of traffic prediction [10]. Traffic flow data is a kind of data with complex spatial and temporal characteristics. First, traffic flow data is an obvious time series data, but the main difference from other time series data is that it is influenced by the spatial structure of the road network [11]. As shown in Figure 1, The traffic flow measured by a sensor at a particular time is related not only to the historical flow here but also to the relative spatial location of that sensor in the road network. For example, the traffic flow on a highway depends on the traffic flow on the merging ramps as well as the traffic flow on the exiting ramps. Therefore, accurate traffic flow prediction is a challenging problem. It is necessary to model and analyze both the temporal characteristics of traffic flow and the spatial characteristics of the road network in order to effectively improve the prediction accuracy.



**Figure 1.** Spatial and temporal correlation of transportation networks.

Existing traffic flow prediction methods have yielded promising results, yet several challenges remain. Statistical methods [12–14], traditional machine learning methods [15–19], and early deep learning methods [20–25] tend to consider traffic flow data as time-series data and ignore the influence of the spatial structure of the road network [26]. The methods using convolutional neural networks (CNN) are able to capture spatial features but are only effective for grid structures [27–29]. The methods using advanced techniques such as graph neural networks (GNN) or attention mechanisms can effectively capture the spatial features of the road network, but they often apply only one or two separate techniques and thus have a slight lack of ability to extract spatio-temporal features [30–40].

In this article, a gated recurrent graph convolutional attention network (GRGCAN) for traffic flow prediction is proposed, which overcomes the above drawbacks. To capture the spatial and temporal features in traffic flow data, a gated recurrent unit (GRU) [23] combined with an attention mechanism is first used to learn temporal features in the data. An attention mechanism and a graph convolution [31] module are fused to extract spatial features among sensor nodes, and finally, feature loss in the network is reduced by a residual connection.

In brief, our main work is as follows:

- A temporal feature extractor is constructed, which introduces a GRU and uses its hidden states of it combined with an attention mechanism to adaptively assign weights to each time step.
- A node attention mechanism fused with graph convolution operation is constructed, which can dynamically assign weights to each sensor node. A spatial feature extractor based on this method is used to synthetically extract spatial features of traffic flow data from a graph-based road network structure. In addition, a residual connection is introduced into the network to reduce the loss of features in the deep network.
- To test the effectiveness of the proposed GRGCAN, the model and several other baselines are applied to several real-world traffic flow datasets. The results show that the GRGCAN can make accurate predictions of traffic flows with higher prediction accuracy than baselines. In addition, the GRGCAN does not require module reuse and thus has high training efficiency.

The rest of the article is organized as follows. Section 2 reviews the existing traffic flow prediction methods and their shortcomings. Section 3 introduces the definition of the traffic flow prediction problem, followed by a detailed description of the structural details of the model by introducing three feature extractors. Section 4 gives the details of the experiment, including the datasets we used and how they were preprocessed, the experimental settings, the evaluation metrics, and the baselines for comparison, followed by an analysis of the experimental results. Section 5 is the conclusion of the article and future works.

## 2. Related Works

Traffic flow data is a kind of spatio-temporal data that exhibits strong dynamic correlation in both spatial and temporal dimensions, so the prediction of traffic flow has been a challenging and meaningful task [8,10,11]. After years of continuous research, researchers have achieved rich results in the field of traffic flow prediction, mainly including statistical methods [12–14], traditional machine learning methods [15–19], and deep learning methods [20,24,25,28,29,32–36,39,40].

Early traffic flow prediction works generally use statistical methods. Hamed et al. [12] used the autoregressive integrated moving average (ARIMA) method to develop a time series model to predict the short-term traffic flow on urban arterials. Williams et al. [13] modeled univariate traffic flow data as a seasonal ARIMA process. Zivot et al. [14] used vector autoregressive (VAR) models for the prediction of multivariate time series. These statistical methods consider traffic flow data as mere time series data and make a large number of assumptions about the traffic flow system, and therefore have major limitations and poor prediction accuracy.

With the rise of machine learning, these algorithms have been applied to traffic flow prediction. Ding et al. [15] first applied a support vector machine (SVM) to the traffic flow time series prediction work and made the prediction of short-term traffic flow more effective. Sun et al. [16] proposed a Bayesian network-based traffic flow prediction method in which the traffic flow between adjacent roads in a traffic network is modeled as a Bayesian network. The joint probability distribution between the cause node (the data used for prediction) and the effect node (the data to be predicted) is described as a Gaussian mixture model (GMM), with its parameters estimated by the competitive expectation maximization (CEM) algorithm. Jeong et al. [17] proposed an online learning weighted support-vector regression (OLWSVR) model based on support-vector regression, which can make effective predictions of short-term traffic flow. Johansson et al. [18] used a random forest as a base model for time series prediction, which allows for determining the size of the prediction intervals by using out-of-bag estimates instead of requiring a separate calibration set. Zheng et al. [19] proposed a method based on the k-nearest neighbor (KNN) algorithm to predict short-term traffic flow, which has the advantage of being insensitive to extreme values. However, these methods have difficulty capturing non-linear features in the data.

Due to the significant development of computer performance in recent years, deep learning methods with the ability to process large-scale data and extract non-linear features are widely used in traffic flow prediction. Hua et al. [20] used a feedforward neural network for the first time to predict traffic flow, showing the great potential of deep learning methods in this field. Recurrent neural networks (RNN) [21] are a class of neural networks that process serial data inputs, and RNN and their variants, long short-term memory (LSTM) [22] networks and GRU [23] networks are commonly used to process time series data. For example, Fu et al. [24] used LSTM and GRU to predict short-term traffic flow and showed that both LSTM and GRU achieved better accuracy compared to statistical methods.

The above machine learning and deep learning methods have improved the prediction accuracy of traffic flow compared with statistical methods, but they are still based on the analysis of temporal features of traffic flow data and ignore spatial features [25].

With the gradual understanding of traffic flow, the complex spatial characteristics it contains are recognized, which are derived from the spatial structure of the road network. CNN [26] are models that are commonly used to extract local features of images. Ma et al. [27] converted traffic flow data into images and then applied CNN to them to extract features of the traffic flow for prediction. Yang et al. [28] combined CNN and LSTM to construct the ConvLSTM model, which can predict future traffic flow in the absence of data.

However, CNN can only extract features from grid-structured data, which is difficult to handle road network structures with non-Euclidean properties. This problem is solved by the advent of GNN, which can represent arbitrary graph structures by adjacency matrices to extract features of non-Euclidean data and are, therefore, more suitable for application to traffic networks. Graph convolutional neural networks (GCN) [29] apply convolutional operations to graph structures and can effectively extract features of graphs. Defferrard et al. [30] proposed ChebNet, which uses Chebyshev polynomial approximation to compute the graph convolution and substantially optimizes the computational efficiency of GCN. Within the field of traffic flow prediction, GCN is often fused with other deep learning methods to extract spatio-temporal features of the data simultaneously. Zhao et al. [31] combined GCN and GRU and proposed the temporal graph convolutional network (T-GCN), which can obtain the spatio-temporal correlation from traffic data. Yu et al. [32] proposed the spatio-temporal graph convolutional network (STGCN) consisting of ST-Conv blocks, which captures spatio-temporal correlations through GCN and CNN in each ST-Conv block. Geng et al. [33] proposed the spatio-temporal multigraph convolution network (ST-MGCN), which uses multigraph convolution to capture different types of correlations between regions. Ge et al. [34] designed the global spatial-temporal graph convolutional network (GSTGCN) for urban traffic prediction, in which temporal features are extracted using 1D CNN, and residual connectivity and spatial features are extracted using GCN, considering the influence of external factors. Wei et al. [35] proposed the novel spatial-temporal graph synchronous aggregation model (STGSA), which constructs the time dependency in time series as a graph with reference to the spatial graph and aggregates it with the spatial graph to extract spatio-temporal features. However, features may be lost in the process of graph construction and aggregation.

The attention mechanism is a method for extracting key information from data, which is widely used in the fields of image processing [36] and natural language processing [37] and has been used in recent years in the field of traffic flow prediction. The ST-MetaNet proposed by Liang et al. [38] has a meta-graph attention network to capture diverse spatial correlations and a meta-recurrent neural network to consider diverse temporal correlations. Attention-based spatial-temporal graph convolutional networks (ASTGCN) proposed by Guo et al. [39] used a spatio-temporal attention mechanism combined with spatio-temporal convolution, which allows dynamic learning of correlations between space and time. The spatial-temporal attention wavenet (STAWnet) proposed by Tian et al. [40] applies temporal convolution and self-attention networks to capture the spatio-temporal features of the data without prior knowledge of the graph.

Inspired by the above studies and considering the complex spatio-temporal characteristics of traffic flow data, we construct the model using GRU, attention mechanism, GCN, and CNN concurrently.

### 3. Method

#### 3.1. Problem Definition

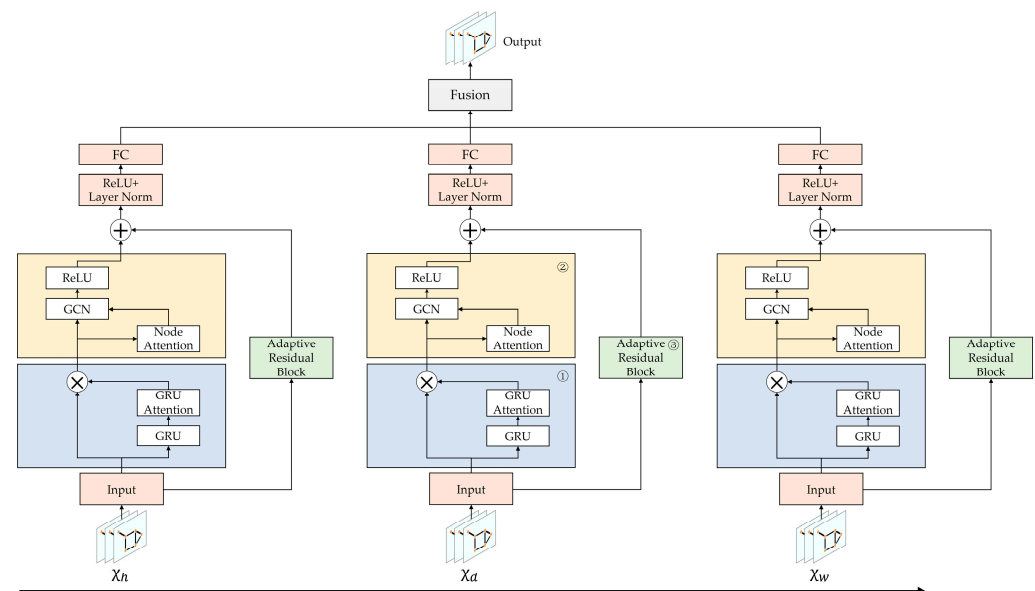
In this study, the road network is defined by the graph  $G = (V, E, A)$ , where  $V$  is a finite set denoting  $|V| = N$  traffic flow sensor nodes;  $E$  is a set consisting of edges between nodes in graph  $G$ , representing the connectivity between nodes;  $A \in \mathbb{R}^{N \times N}$  is the normalized adjacency matrix of graph  $G$ , representing the direction and distance between nodes. In graph  $G$ , the graph signal of time step  $t$  is  $X_t = \{x_t^1, \dots, x_t^N\} \in \mathbb{R}^{N \times F}$ , where  $x_t^n$  ( $n \in \{1, \dots, N\}$ ) are all the features collected by the  $n$ -th sensor at time step  $t$ ;  $F$  is the number of features observed at each node.

The goal of traffic flow prediction is to find a model  $f_{\theta}(\cdot)$ , where  $\theta$  are learnable parameters. The model takes the historical traffic flow sequence with a length of  $T$  and the adjacency matrix  $A$  as inputs to give predictions for the next  $T'$  time steps. The input sequence is denoted as  $\chi = \{X_{t-T+1}, \dots, X_t\} \in \mathbb{R}^{N \times F \times T}$  and the output sequence is denoted as  $\{X_{t+1}, \dots, X_{t+T'}\} \in \mathbb{R}^{N \times F \times T'}$ .

$$\{X_{t+1}, \dots, X_{t+T'}\} = f_{\theta}(X_{t-T+1}, \dots, X_t; A) = f_{\theta}(\chi; A) \quad (1)$$

### 3.2. The Architecture of GRGCAN

Figure 2 demonstrates the structure of the GRGCAN model. The GRGCAN model consists of three independent components with the same structure, and their inputs are historical time series, day-period time series, and week-period time series, respectively. Each component consists of three main parts: (1) Temporal feature extractor: for extracting temporal features of traffic flow data, (2) spatial feature extractor: for extracting spatial features of traffic flow data, (3) adaptive residual block: for reducing feature loss in deep networks adaptively.



**Figure 2.** Structure of the GRGCAN model. (1) Temporal feature extractor: consists of GRU and attention module applied to the hidden state of GRU. (2) Spatial feature extractor: consists of a GCN module fused with a node attention mechanism. (3) Adaptive residual block: consists of a  $1 \times 1$  convolutional network and a fully connected layer.

#### 3.2.1. Time Feature Extractor

Recurrent neural networks are the most used models for extracting features from time series data, but traditional RNNs have problems of gradient disappearance or gradient explosion when the sequence is too long. The advent of LSTM has solved these problems to some extent, but its structure is complex and requires a long computation time. GRU streamlines the unit structure while inheriting the ideas of LSTM, and the accuracy is also improved. Therefore, we choose GRU as the component of the temporal feature extractor<sup>1</sup> in the model. Instead of using GRU directly to predict the time series, the hidden states of GRU are used to obtain the temporal features indirectly. The calculation process is as follows:

$$z_t = \sigma(W_z X_t + U_z h_{t-1} + b_z) \quad (2)$$

$$r_t = \sigma(W_r X_t + U_r h_{t-1} + b_r) \quad (3)$$



$$\tilde{h}_t = \tanh(W_h X_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (4)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (5)$$

where  $h_t$  is the output state at time step  $t$ ;  $\tilde{h}_t$  is the candidate hidden state at time step  $t$ ;  $z_t$  is the update gate which determines how much information needs to be retained in the current state  $h_t$  from the historical state  $h_{t-1}$ ;  $r_t$  is the reset gate which determines how much information needs to be retained in the candidate hidden state  $\tilde{h}_t$  from the historical state  $h_{t-1}$ ;  $W_z, U_z, b_z, W_r, U_r, b_r, W_h, U_h, b_h$  are learnable parameters;  $\sigma(\cdot)$  denotes the sigmoid function;  $\odot$  denotes the Hadamard product.

For the traffic flow prediction, the impact of each historical time step on the future is not equal. To better capture the temporal features in traffic flow data, an attention mechanism is used to learn the output states of GRU to adaptively assign weights to each historical time step. The calculation process is as follows:

$$A_{GRU} = \text{softmax} \left( \frac{HW_{q1}(HW_{k1})^T}{\sqrt{T}} \right) \quad (6)$$

$$\hat{H}_G = A_{GRU}X \quad (7)$$

where  $A_{GRU}$  is the weighting matrix for historical time steps;  $H = \{h_1, \dots, h_t\} \in \mathbb{R}^{N \times F \times T}$  is the output state of GRU at  $T$  historical time steps;  $W_{q1}$  and  $W_{k1}$  are learnable parameters. As shown in Equation (7), the output  $\hat{H}_G = \{\hat{h}_{G1}, \dots, \hat{h}_{GT}\} \in \mathbb{R}^{N \times F \times T}$  is obtained by weighted summation, which will be used as the input of the spatial feature extractor.

### 3.2.2. Spatial Feature Extractor

The extraction of spatial features of road networks has been the key to traffic flow prediction. In general, the spatial structure of the road network is represented by the adjacency matrix, which reflects the location of the sensor nodes, so extracting the spatial features of the road network is to extract the location features of the sensor nodes. A node attention graph convolution operation is proposed to extract spatial features.

The attention mechanism is able to dynamically capture important information in the data. An attention mechanism is applied to learn the input to adaptively assign weights to each sensor node and capture the correlation between nodes. The calculation process is as follows:

$$A_{Node} = \text{softmax} \left( V^T \sigma \left( \frac{(\hat{H}_G W_{q2}) W_{k2}^T}{\sqrt{N}} + b \right) \right) \quad (8)$$

where  $A_{Node}$  is the weighting matrix for sensor nodes;  $W_{q2}, W_{k2}, W, V, b$  are learnable parameters;  $\sigma(\cdot)$  denotes the sigmoid function.

After that, the spatial features of the road network need to be extracted. The graph convolution based on spectral methods [30] is suitable for traffic flow data with non-Euclidean spatial structures. First, we calculate the normalized Laplacian matrix  $L$  of the graph  $G$  and make an eigendecomposition of it:

$$L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T \quad (9)$$

where  $I_N$  is an identity matrix;  $D$  is the degree matrix of the graph  $G$ ;  $U$  is the eigenvector matrix of  $L$ ;  $\Lambda$  is the diagonal matrix consisting of the eigenvalues of  $L$ .

Based on that, the graph convolution operation  $*_G$  of the graph signal  $x$  with  $C$  filters  $g_\theta$  is defined as:

$$g_\theta *_G x = U g_\theta U^T x = U \sum_{k=0}^{K-1} \beta_k T_k(\tilde{\Lambda}) U^T x = \sum_{k=0}^{K-1} \beta_k T_k(\tilde{L}) x \quad (10)$$

$$\tilde{\Lambda} = \frac{2}{\lambda_{max}} \Lambda - I_N \tag{11}$$

$$\tilde{L} = U \tilde{\Lambda} U^T = \frac{2}{\lambda_{max}} L - I_N \tag{12}$$

where  $K$  is the order of the Chebyshev polynomial;  $T_k(\cdot)$  is the Chebyshev polynomial of order  $k$ ;  $\beta_k$  denotes the polynomial coefficients and also the learnable parameters;  $\tilde{\Lambda}$  is the diagonal matrix consisting of adjusted eigenvalues, which ensures the inputs of the Chebyshev polynomial satisfy the range  $[-1,1]$ ;  $\lambda_{max}$  is the maximum eigenvalue of the Laplacian matrix  $L$ ;  $\tilde{L}$  is the Laplacian matrix with adjusted eigenvalues.

In the model,  $C$  and  $K$  are hyperparameters that need to be set. Similar to convolutional neural networks, the number of filters  $C$  is currently set mainly by experience. The value of  $C$  is set to 64 by references [39,41].  $K$  is the order of the Chebyshev polynomial, which means that the range of information extraction in the graph convolution is from 1st to  $K$ -th order neighbors around each node [42]. As  $K$  increases, the performance of the model improves slightly, but the computational cost also increases. Considering that extracting information from the 1st–3rd order neighbors of each node will provide good performance, and it is difficult to significantly improve performance by further increasing  $K$ , the value of  $K$  is set to 3.

In the above process, we replace the input signal  $x$  with  $\hat{H}_G$ , multiply it with the weight matrix  $A_{Node}$  of sensor nodes, and use the rectified linear unit (ReLU) as the activation function, then the node attention graph convolution is calculated as:

$$\hat{H}_N = ReLU \left( \sum_{k=0}^{K-1} \beta_k T_k(\tilde{L}) A_{Node} \hat{H}_G \right) \tag{13}$$

where  $\hat{H}_N = \{\hat{h}_{N1}, \dots, \hat{h}_{NT}\} \in \mathbb{R}^{N \times C \times T}$  is the output of this module.

### 3.2.3. Adaptive Residual Block

To reduce the loss of spatio-temporal features in the deep network, a residual connection is constructed, which can project the input into the feature space of the output of the spatial feature extractor by  $1 \times 1$  convolution. After summing with the adaptive residual output, the output  $\hat{H} = \{\hat{h}_1, \dots, \hat{h}_N\} \in \mathbb{R}^{N \times C \times T}$  is obtained by the ReLU function. The calculation process is as follows:

$$\hat{H} = ReLU(\hat{H}_N + W_r \odot \Gamma_{\theta_r}(X)) \tag{14}$$

where  $\Gamma_{\theta_r}(\cdot)$  denotes the  $1 \times 1$  convolution operation with  $\theta_r$  as the parameter;  $W_r$  is a learnable parameter.

Finally,  $\hat{H}$  is normalized, and an output that matches the predicted target shape is subsequently obtained through the fully connected layer.

### 3.2.4. Multi-Component Fusion

GRGCAN model contains three structurally identical components, each with the outputs  $\hat{H}_h$ ,  $\hat{H}_d$ , and  $\hat{H}_w$ . These three outputs are of different importance to the prediction results [39]. For example, the importance of the day-period component and the week-period component will be higher when predicting traffic flow on weekday morning peaks compared to predicting traffic flow on suburban roads. Therefore, a learnable weight is assigned to each output to learn the fusion method from the historical traffic flow data. The calculation process is as follows:

$$Y = W_h \odot \hat{H}_h + W_d \odot \hat{H}_d + W_w \odot \hat{H}_w \tag{15}$$

where  $W_h$ ,  $W_d$  and  $W_w$  are learnable parameters.

## 4. Experiment

### 4.1. Datasets and Preprocessing

To test the performance of the GRGCAN model, we conducted experiments on two real-world traffic flow datasets, PeMSD4 and PeMSD8. These PeMS datasets [43] are collected by the Caltrans Performance Measurement System; they record traffic data for major freeways in California over a period, updated every 5 min, i.e., the time step is 5 min. The data collected by redundant sensors were removed according to the method of [39] to ensure that the distance between any adjacent sensors is larger than 3.5 miles. Processed PeMSD4 records traffic flow data from 307 sensors in the California Bay Area from 1 January 2018 to 28 February 2018. Processed PeMSD8 records traffic flow data from 170 sensors in San Bernardino, California, from 1 July 2016 to 31 August 2016.

The dataset is divided into the training set, validation set, and test set in the ratio of 6:2:2 according to the time order. In addition, to accelerate the convergence of the model during training, the data were transformed by using zero-mean normalization to make them average zero. The calculation process is as follows:

$$x = \tilde{x} - \text{mean}(\tilde{x}) \quad (16)$$

where  $x$  is the processed traffic flow data;  $\tilde{x}$  is the raw traffic flow data;  $\text{mean}(\cdot)$  denotes the mean value operation.

### 4.2. Experiment Settings

We built the GRGCAN model using the deep learning framework PyTorch and conducted experiments on a computer with a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz CPU, NVIDIA GeForce RTX3070 Laptop GPU, and 16G-DDR5 RAM.

We use 1 h of historical traffic flow data as input, i.e., the input sequence length  $T$  is 12, to predict the traffic flow in the next 1 h, i.e., the output sequence lengths  $T'$  are 3, 6, and 12, respectively. In the training process, we used the mean absolute error (MAE) as the loss function (L1 loss function) and adaptive moment estimation (Adam) optimizer. With a balance of training efficiency and equipment limitations, the learning rate was set to 0.001, the batch size was set to 32, and the model was trained 100 times.

### 4.3. Evaluation Metrics

We used three common metrics for evaluating deep learning models to assess the performance of the GRGCAN model: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). At time step  $t$ , they are calculated as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (17)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (18)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} \quad (19)$$

where  $N$  denotes the number of nodes in graph  $G$ ;  $\hat{y}_i$  is the predicted value of traffic flow;  $y_i$  is the true value of traffic flow.

### 4.4. Baselines

To test the performance of the proposed GRGCAN model, the following seven models were used as baselines.

- GRU [23]: Gated recurrent unit network: treating traffic flow data as simple time series.



- T-GCN [31]: Temporal graph convolutional network: a model that uses two-layer GCN to extract spatial features and GRU to extract temporal features.
- MSTGCN [39]: Multi-component spatial-temporal graph convolutional networks: a model that uses GCN and CNN to extract spatial and temporal features of the data, respectively.
- ASTGCN [39]: Attention-based spatial-temporal graph convolutional networks: an MSTGCN-based model that uses spatio-temporal attention mechanism and spatio-temporal convolution to extract features.
- STSGCN [41]: Spatial-temporal synchronous graph convolutional networks: a model that constructs localized spatio-temporal graphs and applies GCN.
- STAWnet [40]: Spatial-temporal attention wavenet: A model that applies temporal convolution to capture temporal features and a self-attention network to capture dynamic spatial features without requiring prior knowledge of the graph.

#### 4.5. Result Analysis

Table 1 shows the performance of the GRGCAN model with the other baseline models on the two datasets. Considering that the warning period of traffic congestion is roughly 30 min [44], the time steps to be predicted are set to 3 (15 min), 6 (30 min), and 12 (1 h).

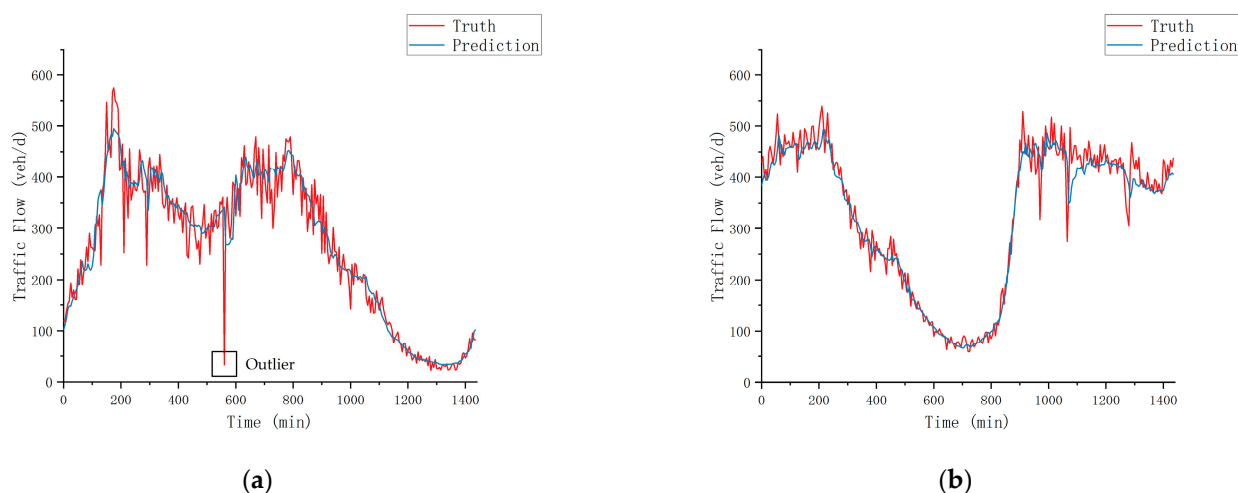
**Table 1.** Performance of each model at a given prediction time step.

Dataset	Model	15 min ( $T' = 3$ )			30 min ( $T' = 6$ )			60 min ( $T' = 12$ )		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PeMSD4	GRU	27.43	44.76	17.95%	25.28	43.07	16.29%	31.17	49.19	20.73%
	T-GCN	26.55	41.87	19.01%	24.27	39.53	16.87%	29.64	45.50	20.82%
	MSTGCN	20.74	32.60	14.13%	24.00	37.15	16.40%	30.35	44.92	21.26%
	ASTGCN	19.97	31.59	13.54%	21.51	33.86	14.64%	25.06	38.79	17.34%
	STSGCN	19.72	31.81	13.12%	21.63	<b>33.81</b>	14.22%	24.78	38.55	16.17%
	STAWnet	20.87	32.09	13.57%	23.02	35.09	14.82%	<b>24.70</b>	<b>37.81</b>	16.61%
	GRGCAN	<b>19.54</b>	<b>31.49</b>	<b>12.71%</b>	<b>21.32</b>	34.37	<b>13.77%</b>	24.95	39.12	<b>15.97%</b>
PeMSD8	GRU	24.51	38.64	14.66%	23.16	37.37	13.53%	27.23	41.70	16.32%
	T-GCN	23.15	37.37	13.77%	22.24	36.60	12.97%	25.63	39.26	15.51%
	MSTGCN	16.61	25.61	10.44%	19.11	29.61	11.82%	24.87	37.54	15.35%
	ASTGCN	16.14	24.90	10.24%	18.05	27.81	11.21%	21.90	33.12	13.31%
	STSGCN	15.97	24.76	10.55%	17.29	27.19	11.26%	<b>19.45</b>	30.74	12.49%
	STAWnet	15.95	24.45	10.92%	17.73	27.14	12.01%	20.10	<b>30.44</b>	13.41%
	GRGCAN	<b>15.56</b>	<b>24.26</b>	<b>9.65%</b>	<b>17.23</b>	<b>26.91</b>	<b>10.45%</b>	20.28	31.47	<b>12.13%</b>

Bold represents the best performance.

Figure 3 shows the prediction results of the GRGCAN model for the traffic flow during 24 h on both datasets.

Based on the experimental results, the following were observed. (1) GRGCAN achieves excellent accuracy on both datasets and performs best on most metrics, especially when the number of time steps to be predicted is small. (2) All models that consider the spatial characteristics of the data outperform the GRU net, which implies that the spatial information of the traffic data is important for prediction. (3) The trends of traffic flow predicted by GRGCAN are generally consistent with the trends of the actual values. (4) As shown in Figure 3a, the predicted value at the outlier in the dataset is not disturbed by the outlier, which indicates the model's good robustness.



**Figure 3.** Visualization of traffic flow prediction results of GRGCAN on PeMSD4 and PeMSD8 datasets. (a) 24-h prediction results on the PeMSD4 dataset; (b) 24-h prediction results on the PeMSD8 dataset.

Due to the immediacy of traffic flow, the computational efficiency of traffic flow prediction models is important. As shown in Table 2, we compared the training efficiency of GRGCAN and baselines on the PeMSD4 dataset (except GRU, which does not consider spatial features), indicated by the average training time of 1 epoch for each model.

**Table 2.** Average training time of each model.

Model	Average Training Time (s/epoch)
T-GCN	25.22
MSTGCN	32.80
ASTGCN	43.36
STSGCN	103.39
STAWnet	44.63
<b>GRGCAN</b>	<b>19.71</b>

Bold represents the best performance.

It is observed that GRGCAN achieves the best training efficiency, which shows its streamlined and effective structure. In T-GCN, two repeated graph convolution operations are performed, which leads to a rise in computational effort. In MSTGCN and ASTGCN, the ST block needs to be reused twice to achieve better results, which leads to a rise in the number of parameters. STSGCN needs to construct the localized spatial-temporal graph first, and STAWnet uses a self-learning adjacency matrix, both of which lead to the generation of additional computations.

Due to GRGCAN's excellent computational efficiency and short-time prediction accuracy, it is well suited to be used for real-time traffic regulation and other tasks.

#### 4.6. Ablation Experiment

To verify the validity of each module in the GRGCAN model, the temporal feature extractor, the spatial feature extractor, and the adaptive residual block were removed from the model, respectively. Then the prediction experiments were conducted on the PeMSD4 dataset for the future 1-h traffic flow ( $T' = 12$ ). We name the three degenerate models GRGCAN-1, GRGCAN-2, and GRGCAN-3, respectively. The experimental results are shown in Table 3.

**Table 3.** Results of ablation experiments.

Model	MAE	RMSE	MAPE
GRGCAN	<b>24.95</b>	<b>39.12</b>	<b>15.97%</b>
GRGCAN-1	26.61	41.34	18.51%
GRGCAN-2	28.35	43.26	23.88%
GRGCAN-3	25.81	39.65	18.39%

Bold represents the best performance.

The experimental results show that the original model outperforms the three degenerated models. Thus, the temporal feature extractor, spatial feature extractor, and adaptive residual block all positively impact the model's performance. Among them, the most significant impacts on the model performance are the spatial feature extractor, which indicates that the application of the node attention mechanism helps to effectively extract the spatial features of the traffic flow data.

## 5. Conclusions

To support the construction of intelligent transportation systems, relieve traffic pressure, and reduce greenhouse gas emissions, a GRGCAN model for traffic flow prediction is proposed. In this model, GRU and GCN are combined with an attention mechanism to adaptively extract spatio-temporal features of traffic flow and reduce the loss of features in the deep network by adaptive residual connection. The experimental findings of one-hour traffic flow prediction using two real-world datasets, namely PeMSD4 and PeMSD8, indicate that the GRGCAN model has a significantly lower MAPE of 15.97% and 12.13%, respectively. Moreover, it outperforms the baseline models in terms of accuracy. Notably, the streamlined model does not reuse structures, which results in an efficient computational performance. The average training time per epoch is as low as 19.71 s. In addition, the ablation experiment proves that either temporal feature extractor, spatial feature extractor, or adaptive residual connection has a positive effect on the performance of the model. In conclusion, the GRGCAN is a novel traffic prediction model that can effectively capture the spatio-temporal features in graph-structured traffic data and provide accurate prediction results.

In future research, we hope to construct more accurate models by considering factors that have an impact on traffic flow, such as weather [45], epidemic [46], or driver's driving style [47]. In addition, we will further consider the impact of cyclical vacations on traffic and try to research using techniques such as continual learning [48]. It is possible to contribute to a more environmentally friendly intelligent transportation system by predicting the greenhouse gas emissions generated by road traffic accordingly.

**Author Contributions:** Conceptualization, X.F. and Y.C.; Funding acquisition, T.M. and Y.R.; Methodology, Y.C.; Project administration, Y.R.; Software, H.L.; Supervision, Y.R.; Validation, X.F. and Y.R.; Visualization, T.M.; Writing—original draft, Y.C.; Writing—review and editing, Y.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (Grant No. U1964206) and the Beijing Municipal Science and Technology Plan (Grant No. Z211100004221008).

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

**Informed Consent Statement:** Not applicable.

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

**Acknowledgments:** We are thankful for all the funding support for the research and to the authors listed in the references for the important research content and conclusions.

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

## References

1. Kober, T.; Schiffer, H.W.; Densing, M.; Panos, E. Global energy perspectives to 2060—WEC’s World Energy Scenarios 2019. *Energy Strategy Rev.* **2020**, *31*, 100523. [[CrossRef](#)]
2. Ali, L.; Nawaz, A.; Iqbal, S.; Aamir Basheer, M.; Hameed, J.; Albasher, G.; Shah, S.A.R.; Bai, Y. Dynamics of Transit Oriented Development, Role of Greenhouse Gases and Urban Environment: A Study for Management and Policy. *Sustainability* **2021**, *13*, 2536. [[CrossRef](#)]
3. Telang, S.; Chel, A.; Nemade, A.; Kaushik, G. Intelligent Transport System for a Smart City. In *Security and Privacy Applications for Smart City Development*; Tamane, S.C., Dey, N., Hassanien, A.E., Eds.; Studies in Systems, Decision and Control; Springer: Cham, Switzerland, 2021; Volume 308, pp. 171–187.
4. Singh, B.; Gupta, A. Recent Trends in Intelligent Transportation Systems: A Review. *J. Transp. Lit.* **2015**, *9*, 30–34. [[CrossRef](#)]
5. Haydari, A.; Yilmaz, Y. Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 11–32. [[CrossRef](#)]
6. Chen, B.; Ji, X.; Ji, X. Dynamic and Static Analysis of Carbon Emission Efficiency in China’s Transportation Sector. *Sustainability* **2023**, *15*, 1508. [[CrossRef](#)]
7. Yu, L.; Zheng, J.; Ma, G.; Jiao, Y. Analyzing the evolution trend of energy conservation and carbon reduction in transportation with promoting electrification in China. *Energy* **2023**, *263*, 126024. [[CrossRef](#)]
8. Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; Yin, B. Deep learning on traffic prediction: Methods, analysis and future directions. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4927–4943. [[CrossRef](#)]
9. Zhao, C.; Wang, K.; Dong, X.; Dong, K. Is smart transportation associated with reduced carbon emissions? The case of China. *Energy Econ.* **2022**, *105*, 105715. [[CrossRef](#)]
10. Cao, P.; Dai, F.; Liu, G.; Yang, J.; Huang, B. A survey of traffic prediction based on deep neural network: Data, methods and challenges. In Proceedings of the 11th EAI International Conference, CloudComp 2021, Melbourne, Australia, 9–10 December 2021.
11. Yuan, H.; Li, G. A Survey of Traffic Prediction: From Spatio-Temporal Data to Intelligent Transportation. *Data Sci. Eng.* **2021**, *6*, 63–85. [[CrossRef](#)]
12. Hamed, M.M.; Al-Masaeid, H.R.; Said, Z.M.B. Short-term prediction of traffic volume in urban arterials. *J. Transp. Eng.* **1995**, *121*, 249–254. [[CrossRef](#)]
13. 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](#)]
14. Zivot, E.; Wang, J. *Modeling Financial Time Series with S-PLUS*<sup>®</sup>, 2nd ed.; Springer: New York, NY, USA, 2006; pp. 369–413.
15. Ding, A.; Zhao, X.; Jiao, L. Traffic flow time series prediction based on statistics learning theory. In Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, Singapore, 6 September 2002. [[CrossRef](#)]
16. Sun, S.; Zhang, C.; Yu, G. A Bayesian network approach to traffic flow forecasting. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 124–132. [[CrossRef](#)]
17. Jeong, Y.S.; Byon, Y.J.; Castro-Neto, M.M.; Easa, S.M. Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1700–1707. [[CrossRef](#)]
18. Johansson, U.; Boström, H.; Löfström, T.; Linusson, H. Regression conformal prediction with random forests. *Mach. Learn.* **2014**, *97*, 155–176. [[CrossRef](#)]
19. Zheng, Z.; Su, D. Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. *Transp. Res. Part Emerg. Technol.* **2014**, *43*, 143–157. [[CrossRef](#)]
20. Hua, J.; Faghri, A. Applications of artificial neural networks to intelligent vehicle-highway systems. *Transp. Res. Rec.* **1994**, *1453*, 83–90.
21. Elman, J.L. Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* **1991**, *7*, 195–225. [[CrossRef](#)]
22. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
23. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 26–28 October 2014. [[CrossRef](#)]
24. Fu, R.; Zhang, 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, Wuhan, China, 11–13 November 2016. [[CrossRef](#)]
25. 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 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
26. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; Arbib, M.A., Ed.; The MIT Press: Cambridge, MA, USA, 1998; pp. 255–258.
27. 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](#)]
28. Yang, G.; Wang, Y.; Yu, H.; Ren, Y.; Xie, J. Short-Term Traffic State Prediction Based on the Spatiotemporal Features of Critical Road Sections. *Sensors* **2018**, *18*, 2287. [[CrossRef](#)]

29. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and deep locally connected networks on graphs. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
30. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
31. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [[CrossRef](#)]
32. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018. [[CrossRef](#)]
33. Geng, X.; Li, Y.; Wang, L.; Zhang, L.; Yang, Q.; Ye, J.; Liu, Y. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019. [[CrossRef](#)]
34. Ge, L.; Li, S.; Wang, Y.; Chang, F.; Wu, K. Global spatial-temporal graph convolutional network for urban traffic speed prediction. *Appl. Sci.* **2020**, *10*, 1509. [[CrossRef](#)]
35. Wei, Z.; Zhao, H.; Li, Z.; Bu, X.; Chen, Y.; Zhang, X.; Lv, Y.; Wang, F. STGSA: A Novel Spatial-Temporal Graph Synchronous Aggregation Model for Traffic Prediction. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 226–238. [[CrossRef](#)]
36. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. In Proceedings of the NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 8–13 December 2014.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
38. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019.
39. 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 33rd AAAI Conference on Artificial Intelligence, Hilton Hawaiian, HI, USA, 27 January–1 February 2019. [[CrossRef](#)]
40. Tian, C.; Chan, W. Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intell. Transp. Syst.* **2021**, *15*, 549–561. [[CrossRef](#)]
41. 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. [[CrossRef](#)]
42. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
43. Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; Jia, Z. Freeway performance measurement system: Mining loop detector data. *Transp. Res. Rec.* **2001**, *1748*, 96–102. [[CrossRef](#)]
44. Jia, R.; Jiang, P.; Liu, L.; Cui, L.; Shi, Y. Data Driven Congestion Trends Prediction of Urban Transportation. *IEEE Internet Things J.* **2018**, *5*, 581–591. [[CrossRef](#)]
45. Hou, G. Evaluating Efficiency and Safety of Mixed Traffic with Connected and Autonomous Vehicles in Adverse Weather. *Sustainability* **2023**, *15*, 3138. [[CrossRef](#)]
46. Shaik, M.E.; Ahmed, S. An overview of the impact of COVID-19 on road traffic safety and travel behavior. *Transp. Eng.* **2022**, *9*, 100119. [[CrossRef](#)]
47. Feng, T.; Liu, K.; Liang, C. An Improved Cellular Automata Traffic Flow Model Considering Driving Styles. *Sustainability* **2023**, *15*, 952. [[CrossRef](#)]
48. Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; Wayne, G. Experience replay for continual learning. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2019; pp. 350–360.

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