

Article

# Attention-Based Spatial–Temporal Convolution Gated Recurrent Unit for Traffic Flow Forecasting

Qingyong Zhang <sup>†</sup>, Wanfeng Chang <sup>†</sup> , Conghui Yin, Peng Xiao <sup>\*</sup> , Kelei Li and Meifang Tan

School of Automation, Wuhan University of Technology, 122 Luoshi Road, Wuhan 430070, China; wfchang@whut.edu.cn (W.C.)

<sup>\*</sup> Correspondence: xp@whut.edu.cn

<sup>†</sup> These authors contributed equally to this work.

**Abstract:** Accurate traffic flow forecasting is very important for urban planning and traffic management. However, this is a huge challenge due to the complex spatial–temporal relationships. Although the existing methods have researched spatial–temporal relationships, they neglect the long periodic aspects of traffic flow data, and thus cannot attain a satisfactory result. In this paper, we propose a novel model Attention-Based Spatial–Temporal Convolution Gated Recurrent Unit (ASTCG) to solve the traffic flow forecasting problem. ASTCG has two core components: the multi-input module and the STA-ConvGru module. Based on the cyclical nature of traffic flow data, the data input to the multi-input module are divided into three parts, near-neighbor data, daily-periodic data, and weekly-periodic data, thus enabling the model to better capture the time dependence. The STA-ConvGru module, formed by CNN, GRU, and attention mechanism, can capture both temporal and spatial dependencies of traffic flow. We evaluate our proposed model using real-world datasets and experiments show that the ASTCG model outperforms the state-of-the-art model.

**Keywords:** traffic flow forecasting; attention mechanism; multi-input; spatial–temporal data



**Citation:** Zhang, Q.; Chang, W.; Yin, C.; Xiao, P.; Li, K.; Tan, M.

Attention-Based Spatial–Temporal Convolution Gated Recurrent Unit for Traffic Flow Forecasting. *Entropy* **2023**, *25*, 938. <https://doi.org/10.3390/e25060938>

Academic Editors: Zhanlong Zhang and Yihua Dan

Received: 17 May 2023

Revised: 10 June 2023

Accepted: 12 June 2023

Published: 14 June 2023



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

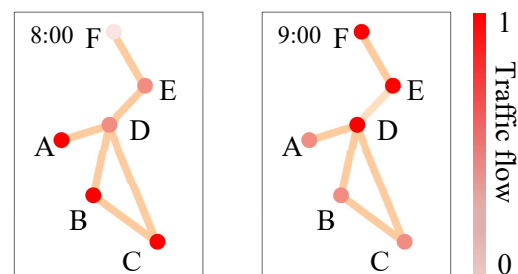
## 1. Introduction

In the process of urbanization, traffic congestion poses an urgent issue that needs to be addressed. Many countries are implementing intelligent transportation systems [1], and real-time and accurate traffic flow prediction is a critical requirement for the establishment of such systems. With accurate traffic flow prediction, traffic management can anticipate future traffic conditions based on historical data, allowing people to plan their trips in advance and providing help for traffic guidance and route planning. However, traffic flow is influenced not only by the passage of time, but also by the interconnectedness of roads, forming a complex mesh structure [2,3]. Accurate traffic flow forecasting is a challenging task.

Fortunately, with the development of industry, many sensors [4] and other information-collecting devices are installed on traffic road networks. These devices can collect a large amount of data for research. Early methods based on statistical analysis, such as historical average (HA) [5], autoregressive integrated moving average (ARIMA) [6], Kalman filter (KF) [7], and exponential smoothing, can be used for traffic flow forecasting. However, they are limited in capturing the nonlinear dependence of time series and are unable to cope well with sudden changes in traffic flow. With the advancement of deep learning, deep learning models are used in many places, such as image processing, natural language processing, power prediction [8–10], etc. They have also gained attention in traffic flow prediction. Recurrent neural network (RNN) and its variants, such as long short-term memory (LSTM) [11], gated recurrent unit (GRU) [12], are common methods for time series prediction. While these models can handle nonlinear problems and perform well on single time series, they often overlook the spatial structure characteristics of the traffic

road network and fail to utilize spatial correlation, resulting in suboptimal prediction performance. Some researchers have explored the use of convolutional neural networks (CNNs) [13] to model traffic flow data spatially, but CNNs struggle to capture temporal correlation, leading to limited results. To address both temporal and spatial correlation, many researchers have combined RNN and CNN to formulate integrated models for traffic flow prediction [14,15]. In recent years, attention mechanism has been proposed and applied to traffic flow forecasting [16–18], showing improved prediction accuracy compared to traditional methods, but there is still room for further improvement.

Figure 1 shows the traffic flow network map at 8:00 and 9:00, respectively. The darker the color of the node, the higher the flow at that node. The traffic flow of nodes D, E, and F will be affected by nodes A, B, and C at previous moments. In other words, the traffic flow of each node is interrelated with other neighboring nodes [19]. When predicting the traffic flow of one node, the traffic flows of other nodes can also be properly input into the model. In addition, the traffic flow is also highly nonlinear and periodic [20], which makes it more difficult to predict. The traffic flow of the traffic road network is very dynamic in the temporal and spatial aspects, so it is a very challenging task to predict the traffic flow data accurately.



**Figure 1.** The spatial–temporal correlation of traffic flow.

In order to address the above challenges, we propose the Attention-Based Spatial–Temporal Convolution Gated Recurrent Unit (ASTCG), which is employed to predict the traffic flow. This model combines CNN, GRU, and attention mechanism to accurately process traffic flow data. Our contributions of this paper are summarized as follows:

- (1) Our proposed ASTCG model integrates CNN, GRU, and attention mechanism to capture both temporal and spatial correlations. GRU is utilized for capturing temporal correlation, while CNN is employed for capturing spatial correlation. ASTCG is also able to effectively utilize long history data due to the inclusion of the attention mechanism.
- (2) Utilizing the periodic characteristics of traffic flow data, the data input is partitioned into three components in the model: near-neighbor data, daily-periodic data, and weekly-periodic data.
- (3) By employing the real dataset PEMS for evaluation, we showcase that our proposed model outperforms the existing baseline models in terms of prediction accuracy.

The structure of this paper is as follows. Section 2 provides an overview of the research and development of traffic flow prediction. Section 3 presents the definition of the traffic flow prediction problem. In Section 4, we present the general framework and detailed architecture of our proposed model. Section 5 presents the experimental results of our model. Finally, Section 6 summarizes the entire paper.

## 2. Related Work

**Statistical learning:** Common statistical learning methods include KF, ARIMA, and Bayesian methods [21], which can be applied to traffic flow forecasting. The KF model assumes that the observed data are noisy and predicts future traffic flow based only on the state of the previous time step. However, KF is a linear prediction model and may have limitations in handling nonlinear and uncertain characteristics of traffic flow data. Shahriari et al. [22] combined bootstrap and ARIMA to improve the prediction

accuracy while maintaining the ARIMA theory, but the prediction accuracy is poor when the flow changes suddenly. Thus, traditional statistical learning methods are limited by the assumption of stationary process and linear combinations, and may be less effective in predicting uncertain and complex traffic flow sequences, which may not meet the current practical engineering needs.

**Machine learning:** More scholars have studied how machine learning methods can be applied to the field of traffic flow forecasting than statistical learning methods. The traditional K-nearest neighbor (KNN) [23] and support vector machines [24] can model complex data, but they require detailed feature engineering and do not achieve the ideal results, so some scholars have improved them. Wang et al. [25] designed a KNN prediction algorithm with asymmetric loss and an asymmetric loss index, and the experimental results showed that when the asymmetric loss index decreased by more than 10%, the predicted value was closer to the upper edge of the actual traffic volume. Luo et al. [26] proposed a hybrid prediction method that combines discrete Fourier transform and support vector regression. The experimental results demonstrated that this algorithm achieves higher accuracy compared to traditional methods, making it an effective approach for holiday traffic flow prediction. Castro-Neto et al. [27] proposed an online supported support vector regression supervised statistical learning technique that can effectively and accurately predict short-term highway traffic flows for typical and atypical scenarios. Although machine learning methods are effective in capturing nonlinear features in traffic flow time series, they often require prior assumptions and extensive feature engineering to achieve excellent experimental results.

**Deep learning:** Since deep learning has powerful autonomous learning ability and nonlinear extraction capability, it has become an inevitable trend to apply deep learning in traffic flow forecasting problems [28–31]. The backpropagation neural network (BP) is one of the simplest neural network models. Chang et al. [32] utilized BP to forecast the traffic flow of a road section in Beijing during peak hours. RNN and its variants, LSTM and GRU [33], taking into account the correlation between multiple output data, so that the information at the previous time can be passed to the following cells, giving the neural network the function of memory, which is often used in the prediction of time series. CNN can extract spatial dependencies by convolutional operations, thus making full use of road network structure information for traffic flow forecasting [34,35]. Zheng et al. [36] combined CNN and LSTM to extract the spatial–temporal features of traffic flow. Zhai et al. [37] designed a novel self-supervised spatial–temporal holistic convolutional neural network to extract the temporal and spatial characteristics of traffic sequences, and the model has fewer parameters and faster inference speed. Since the spatial connection between multiple cross-sections in a traffic road network is an irregular data structure, the graph construction in GCN makes it more suitable for the representation of non-Euclidean spatial structure data, so some prediction methods construct a fixed graph structure based on the relationship on the actual geographic location of multiple cross-sections and construct prediction models on the fixed spatial structure graph to accomplish the task of multisection traffic flow [38]. Zhao et al. [39] designed a traffic speed prediction method based on temporal graphical convolutional networks, which unifies GCN and GRU in the special spatial–temporal component of the model, thus enabling the model to learn both non-Euclidean spatial features of the road network and temporal features of the traffic flow. Chang et al. [40] developed a novel framework called structure-learning convolution, which explicitly models structural information as convolution operations and thus designs local and global modules to learn static and dynamically changing structural information. Xu et al. [41] designed a novel hybrid adjacency matrix and combined it with a temporal attention mechanism for travel time prediction. Wang et al. [42] designed a trend space attention module whose main idea is to pass information between nodes with similar attributes to solve the spatial heterogeneity problem. Zhang et al. [43] extracted the spatial–temporal dependence of traffic flow by taking advantage of the graph attention mechanism for modeling non-Euclidean structured data and the LSTM cell for modeling time series.

Guo et al. [44] introduced a latent network for spatial–temporal feature extraction in the prediction model to construct the dynamic road network graph adjacency matrix adaptively, and the experimental results showed that the adaptively learned dynamic Laplacian matrix has good ability to extract the spatial–temporal correlation of traffic data.

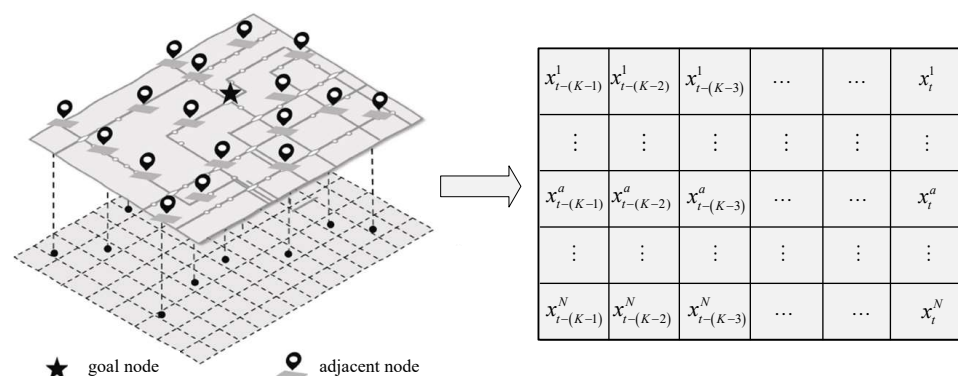
However, few scholars have considered how to make full use of the periodic features of traffic flow to improve the prediction accuracy, and even though Song et al. [45] used the periodic features of traffic flow data, its prediction effect is not made obvious by only adding the module of processing time inside the model. In the past studies, researchers have focused more on how to improve the internal structure of the model and ignored the influence of data input [46,47], but the prediction results are highly related to the input data of the model.

Motivated by the above research, we model traffic data using convolutional neural networks, gated recurrent units, attention mechanisms, and multiple input strategies considering the spatial–temporal dynamic correlation and periodicity of traffic flow data.

### 3. Preliminaries

The task of one-node traffic flow prediction involves predicting the number of vehicles passing through a specific section at future time intervals using historical traffic flow data from multiple sampling intervals. Since the one-node traffic flow is not only near-neighborly in the time dimension, but also exhibits the characteristics of daily and weekly cyclicity, as well as strong spatial correlations with neighboring sections, all of these nodes can impact the traffic flow values of the node in question at future time intervals. Therefore, incorporating traffic flow data from multiple neighboring nodes can lead to relatively accurate predictions of traffic flow at a particular node.

As CNNs are commonly used for image data processing, feature extraction is achieved by scanning the gridded data in the image. Image data typically consist of multiple channels, with each channel containing small indivisible squares, each with its own unique location and pixel information. As illustrated in Figure 2, the spatial–temporal image of the traffic flow is constructed from the following three steps, based on the geographic distribution relationship between the goal node and its associated nodes.



**Figure 2.** Spatial–temporal information structure of traffic flow.

- (1) The plan is divided into a grid based on the relative position of each traffic sensor on the map, so that all the goal nodes and their associated nodes are divided into corresponding small squares, with each small square containing a traffic node.
- (2) The traffic flow data recorded by the sensors at these nodes are filled in as pixel values in the small cells.
- (3) This city sensor map is converted into a spatial–temporal image of traffic flow with length  $N$  squares and width  $K$  squares, where the coordinates of the goal node are  $(a, b)$ , where  $1 < a < N, 1 < b < K$ , and the coordinates of the adjacent node are  $(n, k)$ , where  $n = 1, 2, \dots, N; k = 1, 2, \dots, K; n \neq a; k \neq b$ .

Before introducing the one-node traffic flow prediction model, the input data, output data, and prediction tasks of the prediction model are mathematically defined. Assuming that there are  $N$  traffic nodes in the road network where the predicted goal node is located, the traffic flow collected by the sensor at the node of  $n$  can be defined as follows:

$$X_t^n = \{x_{t-(K-1)}^n, x_{t-(K-2)}^n, \dots, x_t^n\} \quad (1)$$

where we define the traffic flow at time  $t$  and node  $n$  as  $x_t^n$ ,  $n = 1, 2, \dots, N$ .  $K$  represents the length of the input sequence.

In order to more accurately capture the dynamic correlation of traffic flows, three different temporal components, namely, near-neighbor data, daily-periodic data, and weekly-periodic data, denoted as  $\{I^r, I^d, I^w\}$ , are used as inputs to the model for feature extraction. Therefore, the historical traffic flow data of the goal node and its neighboring nodes form a spatial-temporal matrix  $I^r$ , which is mathematically defined as

$$I^r = \begin{bmatrix} X_t^1 \\ X_t^2 \\ \vdots \\ X_t^N \end{bmatrix} = \begin{bmatrix} x_{t-(K-1)}^1 & x_{t-(K-2)}^1 & \cdots & x_t^1 \\ x_{t-(K-1)}^2 & x_{t-(K-2)}^2 & \cdots & x_t^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{t-(K-1)}^N & x_{t-(K-2)}^N & \cdots & x_t^N \end{bmatrix} \quad (2)$$

The travel patterns of people exhibit regularity, and traffic flow often show periodic fluctuations, such as morning and evening peaks on weekdays, that may exhibit similar traffic patterns. Additionally, traffic flow on weekdays may show similarities with the traffic flow of the previous weekday, and can be distinguished from nonworking days. Hence, in order to capture the daily and weekly cycle of the cross-sectional traffic flow data, two spatial-temporal matrices,  $I^d$  and  $I^w$ , are constructed. The definitions of  $I^d$  and  $I^w$  are as follows:

$$I^d = \begin{bmatrix} X_{t^d}^1 \\ X_{t^d}^2 \\ \vdots \\ X_{t^d}^N \end{bmatrix} = \begin{bmatrix} x_{t^d-(K-1)}^1 & x_{t^d-(K-2)}^1 & \cdots & x_{t^d}^1 \\ x_{t^d-(K-1)}^2 & x_{t^d-(K-2)}^2 & \cdots & x_{t^d}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{t^d-(K-1)}^N & x_{t^d-(K-2)}^N & \cdots & x_{t^d}^N \end{bmatrix} \quad (3)$$

$$I^w = \begin{bmatrix} X_{t^w}^1 \\ X_{t^w}^2 \\ \vdots \\ X_{t^w}^N \end{bmatrix} = \begin{bmatrix} x_{t^w-(K-1)}^1 & x_{t^w-(K-2)}^1 & \cdots & x_{t^w}^1 \\ x_{t^w-(K-1)}^2 & x_{t^w-(K-2)}^2 & \cdots & x_{t^w}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{t^w-(K-1)}^N & x_{t^w-(K-2)}^N & \cdots & x_{t^w}^N \end{bmatrix} \quad (4)$$

where  $t^d$  represents the moment at time  $t$  corresponding to the previous day, that is,  $t^d = t - 288$ .  $t^w$  represents the moment in the previous week that corresponds to time  $t$ , that is,  $t^w = t - 2016$ . This is because in the traffic flow dataset, the time interval of recorded traffic flow data is 5 min, resulting in 288 traffic flow data points collected in one day, and 2016 traffic flow data points collected in one week.

The output of the one-node traffic flow is defined as follows:

$$Out = X_t^m = \{x_{t+1}^m, x_{t+2}^m, \dots, x_{t+P}^m\} \quad (5)$$

where  $m$  represents the target cross-section in the road network,  $m = 1, 2, \dots, N$ ;  $x$  represents the traffic flow of the node  $m$ .

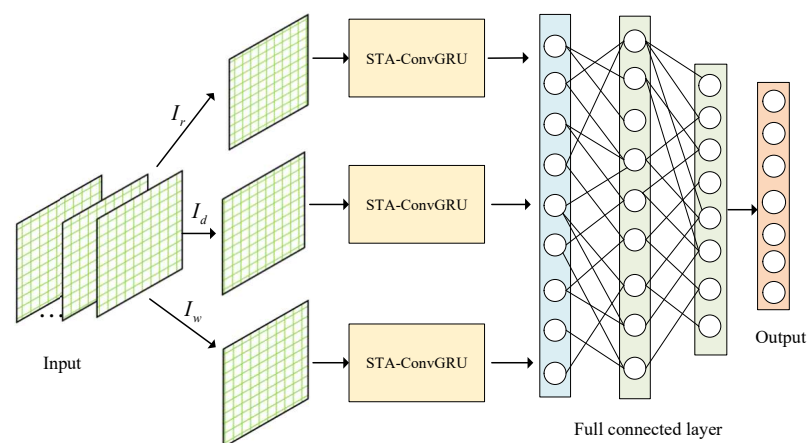
Therefore, the one-node spatial-temporal traffic flow prediction task can be considered as learning a mapping function  $F$  from a large amount of traffic flow data  $I = \{I^r, I^d, I^w\}$ . Using this mapping function and the traffic flow data of the previous  $K$  moments, the traffic

flow values of the future  $P$  moments are predicted, and its mathematical expression can be defined as follows:

$$\{x_{t+1}^m, x_{t+2}^m, \dots, x_{t+P}^m\} = F\left(\{I^r, I^d, I^w\}\right) \quad (6)$$

#### 4. Model Structure

Figure 3 illustrates the overall structure of the ASTCG model. The STA-ConvGRU module integrates the fine-grained feature extraction capability of CNN, the efficient temporal relationship modeling of GRU, and the attention mechanism for focus capturing. The input sequence is that the spatial information of the goal node is processed through CNN convolutional and pooling layers, which is passed to GRU for further processing. The attention module quantifies the historical information in the traffic flow sequence, addressing the limitation of GRU in distinguishing important and unimportant information in the sequence. Subsequently, near-neighbor data, daily-periodic data, and weekly-periodic data are used as input to the STA-ConvGRU module, enabling finer-grained extraction of spatial-temporal characteristics of the traffic flow at the goal node and reducing the random influence of uncertainty on the overall traffic flow distribution. Finally, the outputs of the three components are concatenated and transformed into feature vector data, and the prediction results are obtained through two fully connected layers.



**Figure 3.** ASTCG architecture.

##### 4.1. ConvGRU Module

The ConvGRU module consists of CNN and GRU, where the convolutional kernel sliding operation of CNN captures the spatial correlation of traffic flow at a fine granularity, and the special gating unit mechanism in GRU efficiently extracts the temporal dependence of traffic flow. The combination of CNN and GRU in the ConvGRU module is shown in Figure 4. The CNN contains two convolutional layers and one pooling layer, which is due to the complex spatial characteristics of the traffic road network and the limited expression capability of the single-layer convolutional kernel, so two layers of convolutional layer 1 and convolutional layer 2 are used to extract more comprehensive spatial correlation, followed by filtering unnecessary information as well as reducing the dimensionality of the input data through the pooling layer; finally, the output of the CNN pooling layer is used as the input of GRU, and the output value of the module is obtained after two layers of GRU.

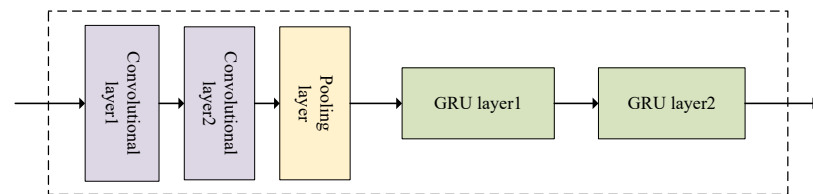
The input of the ConvGRU module is represented as  $I = [X_t^1, X_t^2, \dots, X_t^N]^T$ . In the convolution layer 1 and convolution layer 2, a 1D convolution operation is selected to process the input spatial-temporal traffic flow data, and the spatial influence of adjacent

nodes on the traffic flow of the goal node is extracted by sliding the convolution kernel over the input data. The convolution layer 1 and convolution layer 2 are calculated as follows:

$$Y_1 = \sigma(W_{c1} * I + b_{c1}) \quad (7)$$

$$Y_2 = \sigma(W_{c2} * Y_1 + b_{c2}) \quad (8)$$

where  $W_{c1}, W_{c2}$  are the weight parameters of the convolution kernel;  $b_{c1}, b_{c2}$  are the deviation parameters of the convolution kernel;  $*$  represents the convolution operation;  $\sigma(\cdot)$  is the activation function;  $Y_1, Y_2$  are the outputs of convolution layer 1 and convolution layer 2.



**Figure 4.** ConvGRU module structure.

Pooling layers are useful to speed up the computation and prevent overfitting. This is because the pooling layer can effectively reduce the size of the parameter matrix, thus reducing the number of parameters in the final connection layer. During the pooling process, a large amount of useless data are filtered out, thus ensuring better extraction capability of the model when processing traffic flow data. After the pooling layer is processed, the multidimensional data are converted to a 1D sequence by using the *flatten()* operation.

GRU is an improved model based on RNN, which is a type of self-mapping neural network with strong computational power and long-term memory. GRU has two gating structures, namely, the update gate  $z_t$  and the reset gate  $r_t$ . The update gate determines how much information from the previous time step is incorporated to update the information of the unit at the current time step. On the other hand, the reset gate determines the degree of ignoring information from the previous time step. The calculation formulas for  $z_t$  and  $r_t$  are as follows:

$$z_t = \sigma(w_z * [h_{t-1}, x_t] + b_z) \quad (9)$$

$$r_t = \sigma(w_r * [h_{t-1}, x_t] + b_r) \quad (10)$$

where  $w_z, w_r$  are the weight matrices of the update gate and reset gate;  $x_t$  is the input of the current cell, and  $h_{t-1}$  is the state information of the cell at the previous moment.

The cell state information at each time step in GRU is passed on to the next time step.  $h_t$  represents the output value of the cell at the current time step, and its expression is as follows:

$$h_t = (1 - z_t) * h_{t-1} + z_t * h'_t \quad (11)$$

$$h'_t = \tanh(w_h * [r_t * h_{t-1}, x_t]) \quad (12)$$

Compared to traditional RNN, GRU is capable of better learning time-dependent features due to the presence of update and reset gates, which mitigate the issues of gradient explosion and gradient disappearance that may arise when dealing with long sequences of data. In this model, two layers of GRU are stacked to extract the time-dependent features of traffic flow. The GRU units take the temporal data input for each time window and the hidden state as input.

In our model, the output  $C_{out} = [C_{t-(K-1)}, C_{t-(K-2)}, \dots, C_t]$  of the CNN serves as the input for the first layer of GRU units, and the hidden state value of the first layer of GRU units is used as the input for the subsequent layer of GRU. The formula for each layer of GRU units is as follows:

$$Z_t = \sigma(W_z * [H_{t-1}, F_t] + b_z) \quad (13)$$

$$R_t = \sigma(W_r * [H_{t-1}, F_t] + b_r) \quad (14)$$

$$H'_t = \tanh(W_h * [R_t * H_{t-1}], F_t) \quad (15)$$

$$H_t = (1 - Z_t) * H_{t-1} + Z_t * H'_t \quad (16)$$

where  $F_t$  represents the input of the GRU at time  $t$ ;  $\sigma(\cdot)$  and  $\tanh(\cdot)$  are the activation function;  $W_z, W_h, W_r$  are the weight parameters;  $b_z, b_r$  are the deviation parameters;  $Z_t, R_t$  represent the output of the update gate and the reset gate;  $H_t$  is the output of the GRU unit.

#### 4.2. STA-ConvGRU Module

The historical traffic flow information at different moments has varying effects on the prediction results. However, the GRU is unable to identify the key sequence information in the traffic flow sequence, leading to all the information in the input sequence being equally calculated. This can result in decreased model prediction accuracy and increased computation time. To address this issue, we designed the STA-ConvGRU module, which incorporates the attention mechanism to reduce attention to unimportant information. This allows us to obtain potential information during traffic flow changes and quantify the importance of historical traffic flow data at different locations and moments. Figure 5 depicts the structure of the STA-ConvGRU module, where the outputs of the CNN and the ConvGRU module are combined as input to the temporal attention mechanism module. The attention coefficients calculated by the module are then combined with the output of the ConvGRU module to obtain the final output of the STA-ConvGRU module.

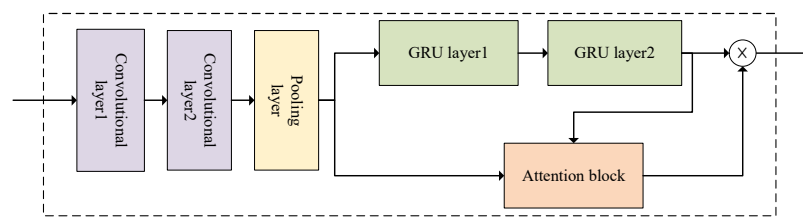


Figure 5. STA-ConvGRU module structure.

In the attention module, each element of the input traffic flow sequence is assigned a corresponding attention allocation probability, which is calculated internally as

$$S_t = W_{a3} \tanh(W_{a1} * C_{out} + W_{a2} * H_{2,t}) \quad (17)$$

where  $W_{a3}, W_{a2}, W_{a1}$  are the weight parameters;  $C_{out}$  is the output of the CNN;  $S_t = (s_{t-(K-1)}, s_{t-(K-2)}, \dots, s_t)$  represents the importance of each historical moment of the traffic flow sequence.

The attention coefficient is defined as

$$a_{t-n} = \frac{\exp(s_{t-n})}{\sum_{n=0}^{n=K} \exp(s_{t-n})} \quad (18)$$

where  $a_{t-n}$  is the attention factor, which represents the degree of influence of each historical traffic flow time step on future traffic flow,  $n = 0, 1, \dots, K$ . Therefore, the output of the ConvGRU module at each time step is multiplied by the attention coefficient and summed to obtain the output of the STA-ConvGRU module. The calculation formula for the output of the STA-ConvGRU module is as follows:

$$H_{a,t} = \sum_{n=0}^{n=K} a_{t-n} h_{2,t-n} \quad (19)$$

## 5. Experiments

### 5.1. Experimental Setup

#### 5.1.1. Experimental Data

The datasets used in this paper are PEMS04 and PEMS08, which are real-time traffic flow datasets collected by Caltrans Performance Measurement System. The data are collected every 30 s and then aggregated every 5 min. A brief description of these datasets is listed in Table 1.

The PEMS04 dataset consists of feature data in three dimensions: traffic flow, average speed, and average lane occupancy. It includes data from 307 nodes in the San Francisco Bay Area and spans from 1 January 2018 to 28 February 2018.

The PEMS08 dataset comprises feature data for 170 nodes in Los Angeles County, including traffic flow, average speed, and average lane occupancy. The dataset spans from 1 July 2016 to 31 August 2016.

In the experimental part, nodes 104 and 307 are chosen as the goal nodes from the PEMS04 dataset. The dataset is divided into 6:2:2, with 35 days (1 January 2018 to 4 February 2018) for the training set, 12 days (5 February 2018 to 16 February 2018) for the validation set, and 12 days (17 February 2018 to 28 February 2018) for the test set. For the PEMS08 dataset, nodes 58 and 100 are selected as the goal nodes. The dataset is divided into 38 days (1 July 2016 to 7 August 2016) for the training set, 12 days (8 August 2016 to 19 August 2016) for the validation set, and 12 days (20 August 2016 to 31 August 2016) for the test set.

**Table 1.** Datasets description.

| Datasets              | PEMS04                            | PEMS08                        |
|-----------------------|-----------------------------------|-------------------------------|
| Goal node number      | 104, 307                          | 58, 100                       |
| Number of node        | 307                               | 170                           |
| Train time range      | 1 January 2018–4 February 2018    | 1 July 2016–7 August 2016     |
| Validation time range | 5 February 2018–16 February 2018  | 8 August 2016–19 August 2016  |
| Test time range       | 17 February 2018–28 February 2018 | 20 August 2016–31 August 2016 |

Data normalization is necessary due to the significant variability of traffic flow data at different moments, which can influence model training and testing.

$$X = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (20)$$

where  $X_{\max}$ ,  $X_{\min}$  represent the maximum and minimum values in the traffic flow sequence.

#### 5.1.2. Hyperparameter Settings

Our ASTCG model is implemented using TensorFlow 1.14 and is run on an Nvidia GeForce RTX 2080Ti GPU. In the experiments, the convolution layer is configured with 15 convolutional kernel channels, each with a size of 7. The sliding window step size for input traffic flow data is set to 1, and computational padding is applied based on the size of the convolution kernels to ensure that the convolution output size matches the input size. The GRU is configured with 24 output units, fully connected layer 1 and fully connected layer 2 have 20 and 10 output units, respectively, and the output layer has 12 output units. During model training, the model is trained in 70 batches with a data batch size of 128, using the Adam optimizer for optimization. The Adam algorithm is an effective stochastic optimization algorithm that combines first-order moment estimation of the gradient and second-order moment estimates to update the parameters. The time interval of the dataset is 5 min, and the historical time length  $K$  is set to 12 (representing one hour in the past), while the prediction length  $P$  is set to 12 (representing one hour in the future).

### 5.1.3. Evaluation Metrics

For all prediction models, we use mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) as our evaluation metrics to assess the performance of the model. The three are calculated as follows.

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N} \left( \sum_{i=1}^N |y_i - \hat{y}_i| \right) \quad (21)$$

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N} \left( \sum_{i=1}^N (y_i - \hat{y}_i)^2 \right)} \quad (22)$$

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (23)$$

### 5.1.4. Compared Methods

In the experimental part, HA, BP, GRU, LSTM, and Spatial–Temporal Graph Convolution Network (STGCN) and Temporal Graph Convolutional Network (T-GCN), are used as the baseline models. The dataset division and hyperparameter settings for the baseline models are kept consistent with the proposed model to ensure fair evaluation of their performance. The details of the baseline models are shown below:

HA [5]: The HA model treats the traffic flow sequence as a seasonal process and generates predictions by taking the weighted average of previous seasons.

BP [32]: BP is a multilayer feedforward network trained using the error backpropagation algorithm. It captures the nonlinear mapping relationship within the traffic flow sequence and dynamically adjusts the weights and thresholds of the network through backpropagation, ensuring that the predicted values are consistently close to the true values.

LSTM [11]: Similar to the GRU, the LSTM also utilizes internal gating units to control the flow of historical information, enabling effective management of historical traffic flow data and achieving high prediction performance.

GRU [12]: The gating unit in GRU effectively captures the time dependence of traffic flow while addressing the issues of gradient explosion and gradient disappearance that can arise from long sequences.

STGCN [48]: It utilizes the ChebNet and 2D convolutional network to model spatial–temporal graph data, offering fast training speed and low model complexity.

T-GCN [39]: T-GCN is a spatial–temporal data mining model that leverages the combination of GCN and GRU to extract spatial–temporal features for accurate traffic flow prediction.

## 5.2. Experimental Results

Table 2 presents the comparison of our proposed method with other baselines on PEMS04 dataset, while Table 3 displays the comparison of our proposed method with other baselines on PEMS08 dataset. The results clearly demonstrate that our proposed ASTCG model surpasses all baseline models in terms of all evaluation metrics.

The prediction results of traditional time series forecasting methods are not satisfactory, indicating their limited ability in dealing with complex spatial–temporal traffic flow data. The poor performance of the HA model can be attributed to its simplistic approach of taking the average value of traffic flow data from previous moments as the prediction for the next moment, without considering the nonlinear temporal variations in traffic flow.

While the BP model accounts for the nonlinearity and instability of traffic flow, it lacks consideration for the time dependence of traffic flow. This limitation results in inferior prediction performance compared to GRU and LSTM models, which have gating units that effectively capture both short-term and long-term dependencies in time series data. The STGCN and T-GCN models not only incorporate the temporal dependence of traffic flow, but also incorporate a spatial extraction component to capture spatial features of all nodes. This design effectively enhances the prediction performance of the models, surpassing the prediction models LSTM and GRU, which only consider temporal features. The MAE, RMSE, and MAPE of the STGCN model are 4.13%, 2.86%, and 2.98% lower than that of the GRU model at node 307 of the PEMS04 dataset. The MAE, RMSE, and MAPE of the STGCN model are decreased by 6.89%, 5.81%, and 6.18% compared to the GRU model at node 100 of the PEMS08 dataset.

Our proposed ASTCG model addresses long time dependencies and complex spatial structures by combining GRU, CNN, and self-attention mechanism. The data input is divided into three parts, including near-neighbor data, daily-periodic data, and weekly-periodic data, which are incorporated into the model. The ASTCG model achieved the best prediction results among all baseline models, with MAE values reduced by 10.08% relative to the T-GCN model, and RMSE values reduced by 9.83% relative to the STGCN model on the prediction task at node 307 of the PEMS04 dataset. On the prediction task at node 100 of the PEMS08 dataset, the MAE value is reduced by 7.60% and the RMSE value is reduced by 3.40% relative to the T-GCN model, which indicates that the ASTCG model can effectively enhance the extraction of spatial-temporal features of traffic flow.

**Table 2.** Comparison with different baselines on PEMS04.

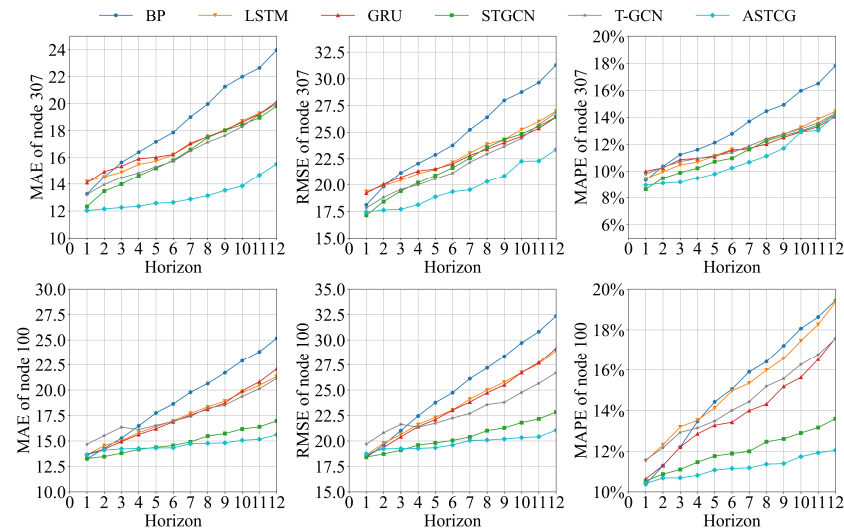
| Model | Node 307 of the PEMS04 |       |        | Node 104 of the PEMS04 |       |        |
|-------|------------------------|-------|--------|------------------------|-------|--------|
|       | MAE                    | RMSE  | MAPE   | MAE                    | RMSE  | MAPE   |
| HA    | 19.73                  | 26.32 | 18.04% | 20.32                  | 26.18 | 18.41% |
| BP    | 18.64                  | 25.07 | 13.39% | 19.14                  | 25.73 | 15.23% |
| LSTM  | 16.81                  | 22.94 | 11.78% | 17.40                  | 23.87 | 15.21% |
| GRU   | 16.94                  | 22.72 | 11.76% | 17.39                  | 23.75 | 13.91% |
| STGCN | 16.24                  | 22.07 | 11.41% | 14.94                  | 20.43 | 12.02% |
| T-GCN | 16.36                  | 21.96 | 11.34% | 16.48                  | 22.45 | 12.50% |
| ASTCG | 14.71                  | 19.90 | 10.84% | 14.58                  | 19.80 | 11.20% |

**Table 3.** Comparison with different baselines on PEMS08.

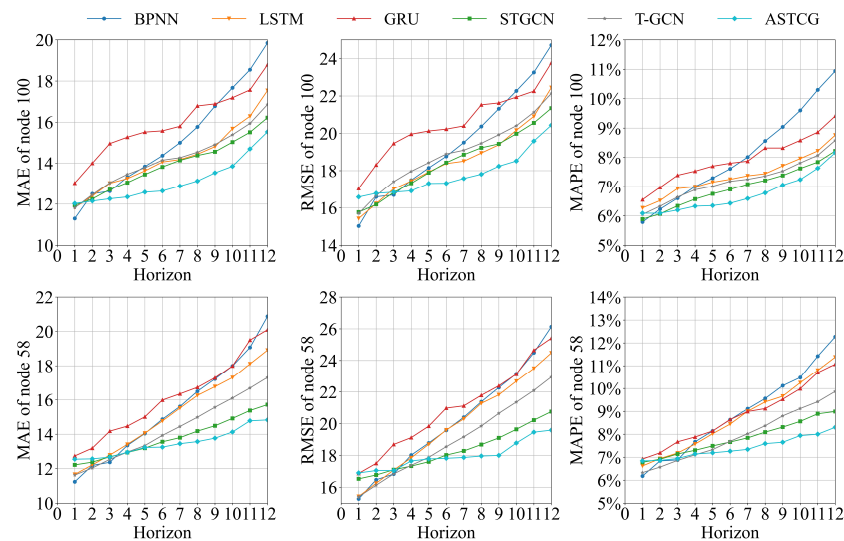
| Model | Node 100 of the PEMS08 |       |       | Node 58 of the PEMS08 |       |       |
|-------|------------------------|-------|-------|-----------------------|-------|-------|
|       | MAE                    | RMSE  | MAPE  | MAE                   | RMSE  | MAPE  |
| HA    | 16.7                   | 21.45 | 8.45% | 16.21                 | 21.32 | 9.12% |
| BP    | 15.13                  | 19.71 | 8.08% | 15.44                 | 20.48 | 8.94% |
| LSTM  | 14.26                  | 18.66 | 7.37% | 15.14                 | 20.10 | 8.77% |
| GRU   | 14.95                  | 19.62 | 7.44% | 15.12                 | 20.11 | 8.81% |
| STGCN | 13.92                  | 18.48 | 6.98% | 13.78                 | 18.33 | 7.83% |
| T-GCN | 14.21                  | 18.93 | 7.22% | 14.29                 | 19.02 | 7.95% |
| ASTCG | 13.13                  | 17.85 | 6.47% | 13.47                 | 18.01 | 7.42% |

Figures 6 and 7 present the MAE, RMSE, and MAPE evaluation metric values for different models on the PEMS04 dataset and the PEMS08 dataset for various time-step prediction tasks. The ASTCG model consistently achieves the best prediction performance across all prediction time steps, as indicated by the lower values of the evaluation metrics. This demonstrates the advantage of the ASTCG model in long-term traffic flow prediction. It is observed that the prediction performance of all baseline models deteriorates with increasing prediction intervals, which is expected as longer prediction time steps provide less useful data for the prediction model to learn from. The ASTCG model exhibits a

similar decay rate compared to the STGCN and T-GCN models in the first four time steps. However, as the time step increases, the decay rate of the ASTCG model is lower than the other two models, indicating its superior ability in extracting temporal features. This implies that the ASTCG model can still extract useful information from historical data even as the prediction time step increases, highlighting the effectiveness of the attention mechanism in quantifying temporal correlations in traffic flow sequences.

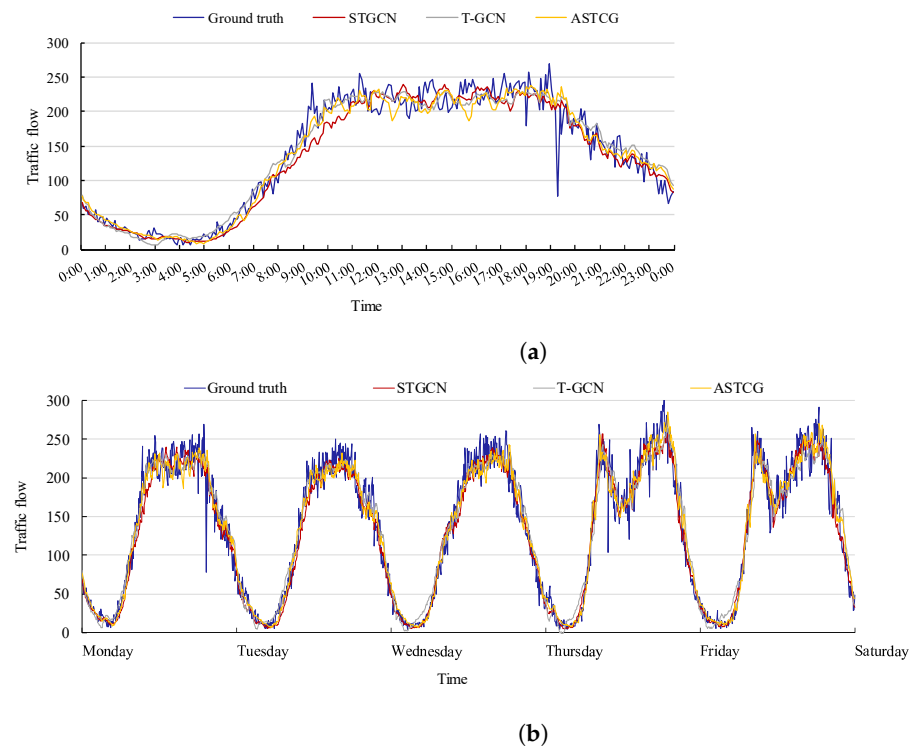


**Figure 6.** Evaluation metrics of different models on the PEMS04 dataset.

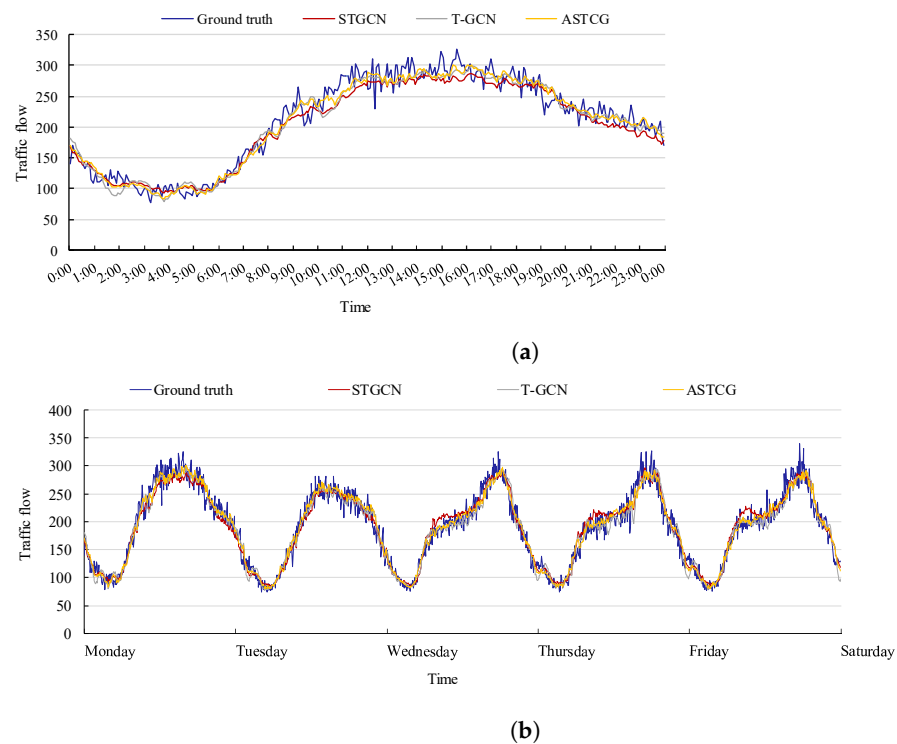


**Figure 7.** Evaluation metrics of different models on the PEMS08 dataset.

To effectively showcase the prediction performance of the ASTCG model, we visualize the traffic flows of STGCN, T-GCN, and ASTCG for one day and one week at node 104 on the PEMS04 dataset and node 58 on the PEMS08 dataset. Figures 8 and 9 depict that the real traffic flow is more accurately followed and the prediction accuracy is higher with the ASTCG model compared to the STGCN and T-GCN.



**Figure 8.** Traffic flow visualization of three spatial-temporal prediction models on the PEMS04 dataset. (a) One-day traffic flow visualization at node 104; (b) One-week traffic flow visualization at node 104.



**Figure 9.** Traffic flow visualization of three spatial-temporal prediction models on the PEMS08 dataset. (a) One-day traffic flow visualization at node 58; (b) One-week traffic flow visualization at node 58.

### 5.3. Ablation Experiments

The ASTCG model comprises three main components, including a convolutional recurrent network component, an attention mechanism, and multiple input modules for

near-neighbor data, daily-periodic data, and weekly-periodic data. The model is evaluated for traffic flow prediction tasks at various time intervals, such as 15 min, 30 min, and 60 min, at node 307 of the PEMS04 dataset, and node 100 of the PEMS08 dataset. The experimental results are presented in Table 4, and the specific details of the three variants of the model are as follows:

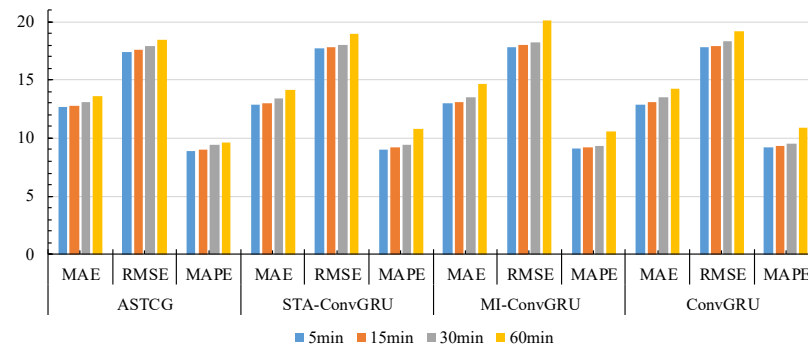
- (a) ConvGRU: This model seamlessly combines a convolutional neural network with a recurrent neural network to capture the spatial-temporal dependencies of traffic flow data.
- (b) STA-ConvGRU: This model enhances ConvGRU by incorporating an attention module that quantifies the importance of historical time steps for improved prediction accuracy.
- (c) MI-ConvGRU: This model extends ConvGRU by incorporating a multi-input component for temporal data, which captures temporal dependence of traffic flow from multiple aspects by incorporating near-neighbor data, daily-periodic data, and weekly-periodic data as inputs.

The experimental results show that the prediction performance of all four models declines as the length of the prediction time step increases on both datasets. This is mainly attributed to the reduced knowledge that the models can glean from historical traffic flow data when predicting larger time steps, resulting in decreased prediction accuracy. Additionally, it can be observed that the STA-ConvGRU and MI-ConvGRU models outperform the ConvGRU model in terms of prediction performance. This is reasonable, as the STA-ConvGRU and MI-ConvGRU models incorporate different temporal dependency extraction components, which enable them to capture richer time-related information. This finding further validates the effectiveness of the proposed multi-input component and attention mechanism component. Among the four models, the ASTCG model exhibits the best prediction performance. For instance, in the 60 min prediction task at node 100 of the PEMS08 dataset, the MAE value of ASTCG is reduced by 2.72% compared to MI-ConvGRU, and the MAPE value is reduced by 2.80% compared to STA-ConvGRU. This indicates that the inclusion of the feature extraction component in the model effectively enhances the prediction accuracy. Moreover, the ASTCG model demonstrates accurate predictions of traffic flow at four different nodes in the two datasets, showcasing its excellent generalization ability.

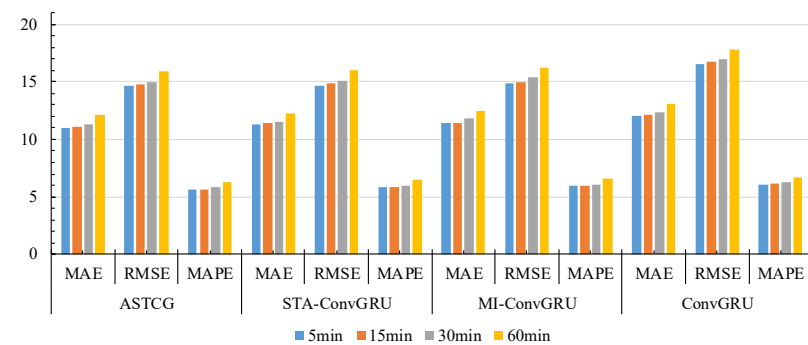
**Table 4.** Evaluation metric values of ASTCG and three variants of the model at different time steps.

| Model       | Horizon | Node 307 of the PEMS04 |       |        | Node 100 of the PEMS08 |       |       |
|-------------|---------|------------------------|-------|--------|------------------------|-------|-------|
|             |         | MAE                    | RMSE  | MAPE   | MAE                    | RMSE  | MAPE  |
| ConvGRU     | 5 min   | 12.89                  | 17.81 | 9.23%  | 12.02                  | 16.57 | 6.09% |
|             | 15 min  | 13.12                  | 17.91 | 9.36%  | 12.14                  | 16.75 | 6.12% |
|             | 30 min  | 13.46                  | 18.33 | 9.48%  | 12.33                  | 16.95 | 6.25% |
|             | 60 min  | 14.22                  | 19.18 | 10.84% | 13.13                  | 17.86 | 6.74% |
| STA-ConvGRU | 5 min   | 12.91                  | 17.68 | 8.99%  | 11.28                  | 14.71 | 5.87% |
|             | 15 min  | 12.97                  | 17.78 | 9.21%  | 11.42                  | 14.92 | 5.89% |
|             | 30 min  | 13.36                  | 18.02 | 9.43%  | 11.55                  | 15.12 | 5.98% |
|             | 60 min  | 14.11                  | 19.01 | 10.8%  | 12.28                  | 16.01 | 6.43% |
| MI-ConvGRU  | 5 min   | 12.98                  | 17.77 | 9.09%  | 11.38                  | 14.86 | 5.92% |
|             | 15 min  | 13.14                  | 18.02 | 9.21%  | 11.45                  | 14.98 | 5.95% |
|             | 30 min  | 13.54                  | 18.23 | 9.35%  | 11.80                  | 15.43 | 6.11% |
|             | 60 min  | 14.69                  | 20.10 | 10.54% | 12.47                  | 16.23 | 6.55% |
| ASTCG       | 5 min   | 12.72                  | 17.44 | 8.91%  | 11.03                  | 14.71 | 5.63% |
|             | 15 min  | 12.82                  | 17.58 | 9.04%  | 11.09                  | 14.75 | 5.68% |
|             | 30 min  | 13.12                  | 17.91 | 9.46%  | 11.31                  | 14.98 | 5.85% |
|             | 60 min  | 13.59                  | 18.47 | 9.66%  | 12.13                  | 15.89 | 6.25% |

To visually compare the prediction performance of the four models, Figures 10 and 11 display the prediction evaluation metrics at node 307 of the PEMS04 dataset and node 100 of the PEMS08 dataset. It is evident from the visualizations that ASTCG consistently achieves the best prediction results for all four evaluation metrics at different prediction time steps, showcasing the superior performance of this model.



**Figure 10.** Prediction performance of the four models at node 307 of the PEMS04 dataset.



**Figure 11.** Prediction performance of the four models at node 100 of the PEMS08 dataset.

## 6. Conclusions

In this paper, we propose a new spatial–temporal attention-based model, Attention-Based Spatial–Temporal Convolution Gated Recurrent Unit (ASTCG), applied to traffic flow forecasting. The model combines a convolution neural network, a gated recurrent unit, and a spatial–temporal attention mechanism to capture the spatial–temporal correlation of traffic flow. Furthermore, our model leverages the cyclic nature of traffic flow data by incorporating near-neighbor data, daily-periodic data, and weekly-periodic data, which enhances the prediction accuracy. Our proposed model is tested on two real datasets and outperforms all baseline methods. However, it should be noted that traffic flow prediction is influenced by various factors, such as weather, holidays, and social events. In future research, it would be beneficial to consider these factors to further improve the effectiveness of the model. We want the research work in this paper to reach cooperation with related enterprises or traffic management departments so that it can provide data support for route planning and traffic guidance, and help people to travel on a daily basis.

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

**Funding:** This research was funded by Natural Science Foundation of Hubei Province (2019CFB571).

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

**Data Availability Statement:** A publicly available dataset was analyzed in this study. It can be found here: <https://github.com/wanhuaayu/ASTGCN/tree/master/data> (accessed on 1 May 2023).

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

## References

1. Zhang, J.; Wang, F.Y.; Wang, K.; Lin, W.H.; Xu, X.; Chen, C. Data-Driven Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1624–1639. [\[CrossRef\]](#)
2. Yuan, J.; Fan, B. Synthesis of short-term traffic flow forecasting research progress. *Urban Transp. China* **2012**, *10*, 73–79.
3. Ishak, S.; Al-Deek, H. Performance evaluation of short-term time-series traffic prediction model. *J. Transp. Eng.* **2002**, *128*, 490–498. [\[CrossRef\]](#)
4. Dan, Y.; Zhang, Z.; Gan, P.; Ye, H.; Li, Q.; Deng, J. Performance Analysis of Corroded Grounding Devices with an Accurate Corrosion Model. *CSEE J. Power Energy Syst.* **2023**, *9*, 1235–1247.
5. Liu, J.; Wei, G. A Summary of Traffic Flow Forecasting Methods. *J. Highw. Transp. Res. Dev.* **2004**, *21*, 82–85.
6. Wang, H.; Liu, L.; Dong, S.; Qian, Z.; Wei, H. A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework. *Transp. B Transp. Dyn.* **2015**, *4*, 159–186. [\[CrossRef\]](#)
7. Okutani, I.; Stephanedes, Y.J. Dynamic prediction of traffic volume through Kalman filtering theory. *Transp. Res. Part B Methodol.* **1984**, *18*, 1–11. [\[CrossRef\]](#)
8. Hossain, M.Z.; Sohel, F.; Shiratuddin, M.F.; Laga, H. A Comprehensive Survey of Deep Learning for Image Captioning. *ACM Comput. Surv.* **2019**, *51*, 1–36. [\[CrossRef\]](#)
9. Zhang, Z.; Yang, Y.; Zhao, H.; Xiao, R. Prediction method of line loss rate in low-voltage distribution network based on multi-dimensional information matrix and dimensional attention mechanism-long-and short-term time-series network. *IET Gener. Transm. Distrib.* **2022**, *16*, 4187–4203. [\[CrossRef\]](#)
10. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.G.; Le, Q.; Salakhutdinov, R. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. *arXiv* **2019**, arXiv:1901.02860
11. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
12. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
13. Guo, S.; Lin, Y.; Li, S.; Chen, Z.; Wan, H. Deep Spatial-Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1–14. [\[CrossRef\]](#)
14. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv* **2017**, arXiv:1707.01926.
15. 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*; MIT Press: Cambridge, MA, USA, 2015; Volume 28, pp. 55–71.
16. Liang, Y.; Ke, S.; Zhang, J.; Yi, X.; Yu, Z. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; Volume 2018, pp. 1–16.
17. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929.
18. Wang, Y.; Zheng, J.; Du, Y.; Huang, C.; Li, P. Traffic-GGNN: Predicting Traffic Flow via Attentional Spatial-Temporal Gated Graph Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18423–18432. [\[CrossRef\]](#)
19. Xiong, J.; Xiong, Z.; Zhuang, Y.; Cheong, J.W.; Dempster, A.G. Fault-Tolerant Cooperative Positioning Based on Hybrid Robust Gaussian Belief Propagation. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6425–6431. [\[CrossRef\]](#)
20. Fang, M.; Tang, L.; Yang, X.; Chen, Y.; Li, C.; Li, Q. FTPG: A Fine-Grained Traffic Prediction Method With Graph Attention Network Using Big Trace Data. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5163–5175. [\[CrossRef\]](#)
21. Jian, W.; Wei, D.; Guo, Y. New Bayesian combination method for short-term traffic flow forecasting. *Transp. Res. Part C* **2014**, *43*, 79–94.
22. Shahriari, S.; Ghasri, M.; Sisson, S.A.; Rashidi, T. Ensemble of ARIMA: Combining Parametric and Bootstrapping Technique for Traffic Flow Prediction. *Transp. A Transp. Sci.* **2020**, *16*, 1552–1573. [\[CrossRef\]](#)
23. Lint, H.V.; Hinsbergen, C.V. Short-Term Traffic and Travel Time Prediction Models. *Transp. Res. E-Circ.* **2012**, *22*, 22–41.
24. 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. [\[CrossRef\]](#)
25. Wang, Z.; Ji, S.; Yu, B. Short-Term Traffic Volume Forecasting with Asymmetric Loss Based on Enhanced KNN Method. *Math. Probl. Eng.* **2019**, *2019*, 1–11. [\[CrossRef\]](#)
26. Luo, X.; Li, D.; Zhang, S. Traffic Flow Prediction during the Holidays Based on DFT and SVR. *J. Sens.* **2019**, *2019*, 1–10. [\[CrossRef\]](#)
27. Castro-Neto, M.; Jeong, Y.S.; Jeong, M.K.; Han, L.D. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **2009**, *36*, 6164–6173. [\[CrossRef\]](#)

28. 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 Emerg. Technol.* **2015**, *54*, 187–197. [\[CrossRef\]](#)
29. Wang, L.; Chai, D.; Liu, X.; Chen, L.; Chen, K. Exploring the Generalizability of Spatio-Temporal Traffic Prediction: Meta-Modeling and an Analytic Framework. *IEEE Trans. Knowl. Data Eng.* **2021**, *4*, 1–16. [\[CrossRef\]](#)
30. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting. *IEEE Trans. Knowl. Data Eng.* **2021**, *5*, 1–10. [\[CrossRef\]](#)
31. Zhang, Q.; Yu, K.; Guo, Z. Graph Neural Network-Driven Traffic Forecasting for the Connected Internet of Vehicles. *IEEE Trans. Netw. Sci. Eng.* **2022**, *4*, 3015–3027. [\[CrossRef\]](#)
32. Zhang, Q.; Liu, S. Urban traffic flow prediction model based on BP artificial neural network in Beijing area. *J. Discret. Math. Sci. Cryptogr.* **2018**, *21*, 849–858. [\[CrossRef\]](#)
33. Yang, B.; Sun, S.; Li, J.; Lin, X.; Tian, Y. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing* **2019**, *332*, 320–327. [\[CrossRef\]](#)
34. Ma, X.; Zhuang, D.; He, Z.; Ma, J.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors* **2017**, *17*, 818–833. [\[CrossRef\]](#)
35. Ke, R.; Li, W.; Cui, Z.; Wang, Y. Two-Stream Multi-Channel Convolutional Neural Network for Multi-Lane Traffic Speed Prediction Considering Traffic Volume Impact. *Transp. Res. Rec.* **2020**, *2674*, 459–470. [\[CrossRef\]](#)
36. Zheng, H.; Lin, F.; Feng, X.; Chen, Y. A Hybrid Deep Learning Model With Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1–11. [\[CrossRef\]](#)
37. Zhai, L.; Yang, Y.; Song, S.; Ma, S.; Yang, F. Self-supervision Spatiotemporal Part-Whole Convolutional Neural Network for Traffic Prediction. *Phys. A Stat. Mech. Its Appl.* **2021**, *579*, 126141. [\[CrossRef\]](#)
38. Gu, Z.; Chen, C.; Zheng, J.; Sun, L. Traffic flow prediction based on STG-CRNN. *Control Decis.* **2022**, *37*, 645–653.
39. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1–11. [\[CrossRef\]](#)
40. Zhang, Q.; Chang, J.; Meng, G.; Xiang, S.; Pan, C. Spatio-Temporal Graph Structure Learning for Traffic Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1177–1185.
41. Xu, M.; Liu, H. A flexible deep learning-aware framework for travel time prediction considering traffic event. *Eng. Appl. Artif. Intell.* **2021**, *106*, 104491–104505. [\[CrossRef\]](#)
42. Wang, C.; Tian, R.; HU, J.; Ma, Z. A trend graph attention network for traffic prediction. *Inf. Sci.* **2023**, *623*, 275–292. [\[CrossRef\]](#)
43. Zhang, T.; Guo, G. Graph Attention LSTM: A Spatio-Temporal Approach for Traffic Flow Forecasting. *IEEE Intell. Transp. Syst. Mag.* **2020**, *14*, 190–196. [\[CrossRef\]](#)
44. Guo, K.; Hu, Y.; Qian, Z.; Sun, Y.; Yin, B. Dynamic Graph Convolution Network for Traffic Forecasting Based on Latent Network of Laplace Matrix Estimation. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1–10. [\[CrossRef\]](#)
45. 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.
46. Wei, C.; Sheng, J. Spatial-temporal graph attention networks for traffic flow forecasting. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*; IOP: Bristol, UK, 2020; Volume 587, pp. 65–78.
47. Shi, X.; Qi, H.; Shen, Y.; Wu, G.; Yin, B. A Spatial-Temporal Attention Approach for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4909–4918. [\[CrossRef\]](#)
48. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *Statistics* **2017**, *22*, 1–13.

**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.