applied sciences

MDPI

Article

Hierarchical Spatial-Temporal Neural Network with Attention Mechanism for Traffic Flow Forecasting

Qingyun Lian [1], Wei Sun [2,*] and Wei Dong [2]

[1] College of Merchant Ship, Shanghai Maritime University, Shanghai 201306, China; qylian@shmtu.edu.cn
[2] College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
* Correspondence: weisun@shmtu.edu.cn

Featured Application: It can help citizens bypass congested roads and avoid rush hours by predicting future traffic flows in advance, thereby reducing their travel time and costs and increasing the operational capacity and efficiency of the road network.

Abstract: Accurate traffic flow forecasting is pivotal for intelligent traffic control and guidance. Manually capturing the intricate dependencies between spatial and temporal dimensions in traffic data presents a significant challenge. Prior methods have primarily employed Recurrent Neural Networks or Graph Convolutional Networks, without fully accounting for the interdependency between spatial and temporal factors. To address this, we introduce a novel Hierarchical Spatial-Temporal Neural Networks with Attention Mechanism model (HSTAN). This model concurrently captures temporal correlations and spatial dependencies using a multi-headed self-attention mechanism in both temporal and spatial terms. It also integrates global spatial-temporal correlations through a hierarchical structure with residuals. Moreover, the analysis of attention weight matrices can depict complex spatial-temporal correlations, thereby enhancing our traffic forecasting capabilities. We conducted experiments on two publicly available traffic datasets, and the results demonstrated that the HSTAN model's prediction accuracy surpassed that of several benchmark methods.

Keywords: traffic flow forecasting; spatial-temporal correlation; multi-headed self-attention mechanism

check for updates

Citation: Lian, Q.; Sun, W.; Dong, W. Hierarchical Spatial-Temporal Neural Network with Attention Mechanism for Traffic Flow Forecasting. *Appl. Sci.* **2023**, *13*, 9729. https://doi.org/10.3390/app13179729

Academic Editors: Wen-Long Shang, Kun Wang, Haoran Zhang, Yanyan Chen and Washington Yotto Ochieng

Received: 14 July 2023
Revised: 17 August 2023
Accepted: 24 August 2023
Published: 28 August 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

The advent of cost-effective traffic sensor technologies in recent years has ushered us into the era of transportation big data. This surge in traffic data has led to the development of the Intelligent Transportation System (ITS) [1], designed to harness this vast data for efficient urban traffic management and planning. A crucial component of ITS is accurate and timely traffic forecasting, which has garnered significant interest. Traffic flows exhibit complex spatial and temporal dependencies, as illustrated in Figure 1, making accurate real-time traffic forecasting a formidable challenge.

Traffic forecasting diverges from conventional time series forecasting due to its susceptibility to spatial and other variables [2]. Temporally, urban traffic flow exhibits a pronounced cyclical pattern, attributable to the commuting habits of city dwellers. However, traffic flows are subject to inevitable fluctuations stemming from external factors such as brief spells of inclement weather and traffic incidents. Spatially, traffic data are propagated through the road network's topology. Traffic conditions at a given observation point are directly impacted by the upstream and downstream traffic on adjacent roads, demonstrating both mobility and transience.

In recent years, traffic models have predominantly utilized recurrent neural networks (RNN) or graph convolutional networks (GCN). However, for accurate traffic forecasting, it is crucial to consider both the temporal correlation and spatial dependency. The main challenge in these models lies in effectively modeling the spatial and temporal correlations,

which exhibit multi-scale characteristics encompassing global and local spatial terms, as well as long and short-term temporal relationships. Zhao proposed T-GCN [3], a model that combines GCN with GRU. Instead of using LSTM, T-GCN employs GRU with fewer parameters for time dependence extraction, along with GCN for capturing spatial dependency and extracting spatial-temporal correlations for traffic forecasting. Building upon T-GCN, Bai introduced A3TGCN [4], which integrates an attention mechanism to dynamically adjust the significance of time points in the time series. Furthermore, Xu proposed STTN [5], comprising stacked spatial-temporal blocks and prediction layers. Each spatial-temporal block consists of a spatial transformer and a temporal transformer, which jointly extract spatial-temporal correlations. Although the attention mechanism proves useful in modeling spatial-temporal correlations, the transformer structure results in an excessive number of parameters. In contrast, ASTTN [6] leverages a self-attention mechanism to model dynamic spatial-temporal correlations in graphs. It employs local spatial-temporal graphs to capture spatial-temporal features, focusing attention on one-hop spatial neighbors. Restricting attention to spatial neighbors and capturing long-range spatial dependencies becomes reasonable as the number of layers increases, given the typically weak long-range spatial-temporal attention.

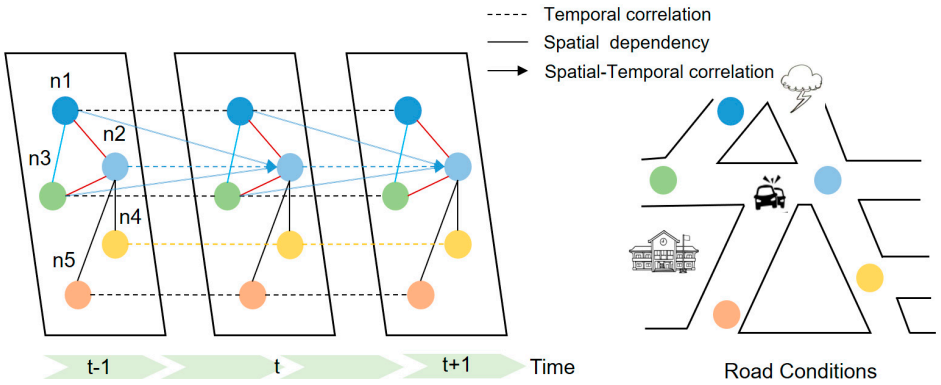


Figure 1. Complex Spatial-Temporal Correlation. Except for the traditional temporal correlations (dotted line) and spatial dependency (solid line), the spatial-temporal correlations (solid line with arrow) will also affect traffic conditions.

Inspired by the recent success of the transformer model in graph domains, this paper presents the HSTAN (Hierarchical Spatial-Temporal Self-Attentive Networks) model. HSTAN employs local multi-headed self-attention within each spatial-temporal block to directly model spatial-temporal correlations on graphs. Moreover, it utilizes multiple layers of spatial-temporal blocks to capture deep spatial-temporal features. By effectively representing spatial-temporal correlation through the combination of local attention and hierarchical structure, HSTAN overcomes the challenges associated with training models with a large number of parameters. The main contributions of this paper can be summarized as follows:

(1) We propose the STA-Blocks structure, designed to encapsulate local spatial-temporal correlations. This structure utilizes a temporal attention layer to capture temporal correlations, while local spatial correlations within a one-hop radius are captured through a spatial attention layer. Both of these layers operate on the basis of a multi-headed self-attention mechanism.

(2) We utilize a hierarchical structure of stacked Spatial-Temporal Attention Blocks (STA-Blocks) to methodically extract and integrate spatial-temporal correlations. This hierarchical structure enhances the receptive field of local multi-headed self-attention, thereby achieving a level of performance equivalent to full attention.

(3) We have evaluated the predictive performance of our proposed model using two real-world traffic datasets. Our model has demonstrated a significant improvement in the accuracy of traffic forecasting when compared to the baseline models. Additionally,

we have carried out ablation studies to assess the impact of our model's components on its overall performance.

2. Related Works

Traffic forecasting necessitates the consideration of both temporal correlation and spatial dependency. In recent years, the application of deep learning to traffic flow forecasting, which is used to represent spatial-temporal correlation, has seen the emergence of methods that can be categorized as RNN-based, GCN-based, and those based on attention mechanisms.

RNN-based: Given the exceptional predictive capabilities of Recurrent Neural Networks (RNN) [7] and their variants, such as Long Short-Term Memory networks (LSTM) [8] and Gated Recurrent Units (GRU), these have been extensively utilized for time series forecasting tasks, including traffic flow prediction. The Conv-LSTM model [9] was developed by integrating Convolution and bi-directional LSTM to address the issue of long-term dependencies. Furthermore, a multi-dimensional LSTM network [10–12] was designed to capture both temporal and spatial correlations in traffic flows, demonstrating superior performance when compared experimentally with other prominent models. The Diffusion Convolutional Recurrent Neural Network (DCRNN) [13] employs bidirectional random drifts on the graph to capture spatial dependencies, and an encoder–decoder structure to capture temporal dependencies.

GCN-based: Multiple research studies have utilized Convolutional Neural Networks (CNN) for spatial modelling to characterize the spatial correlations of traffic flow. However, CNN has limitations when applied to non-Euclidean spaces. To overcome this, graph convolution has been employed to model the spatial correlations of complex road networks. STGCN [14] proposed a combination of Graph Convolutional Networks (GCN) and Recurrent Neural Networks (RNN) for traffic flow prediction. This approach utilizes graph convolution to extract spatial features and temporal gated convolution to extract temporal features. T-GCN [3] introduced temporal graph convolution networks, which leverage GRU to address the issues of gradient explosion and disappearance during training. It combines graph convolution with GRU to extract spatial and temporal features. Graph WaveNet (GWN) [15] presented a novel graph neural network model for spatial-temporal graph modelling, capturing hidden spatial dependencies accurately. Considering the periodicity of traffic flow data, MCSTGCN [16] proposed multi-component spatial-temporal graph convolutional networks. The network is divided into three components to extract temporal-dependent features at recent, daily, and weekly intervals. STSGCN [17] proposed spatial-temporal synchronous graph convolutional networks to predict traffic flow by constructing local spatial-temporal graphs, integrating information across nodes at different time points. AGCRN [18] devised a new GCN parameter factorization module and combined it with a recurrent network for traffic forecasting. MTGNN [19] utilized the innate dependencies among multiple road segments for traffic forecasting. Consequently, the GCN-based method, which captures temporal correlation using RNN and spatial dependency using graph convolution, has gained popularity as a model for traffic flow forecasting due to its exceptional representation capacity.

Attention mechanism-based: Attention mechanisms enable convolutional neural networks to dynamically focus on important information. In 2018, GAT [20] first combined the attention mechanism with a graph structure. The Graph Attention LSTM Network [21], on the other hand, combines the graph attention mechanism with the LSTM network. This combination allows for learning the spatial features of the graph and using LSTM to predict and fit the prediction values. TPA-LSTM [22] improves traffic flow prediction by optimizing the hidden state output in the LSTM network and then utilizing the attention mechanism to extract key time nodes. Similarly, ConsTGAT [23] employs the attention mechanism and a convolutional approach to learn external characteristics and extract spatial correlation, ultimately enhancing vehicle arrival time estimation on the Baidu map. AttConvLSTM [24] proposes integrating the attention mechanism to emphasize the impact of specific parts on traffic flow predictions. ASTGCN [25] incorporates CNN in the time series component

and leverages the GAT model for spatial correlation extraction, resulting in improved outcomes across multiple traffic datasets. Building upon ASTGCN, DGCN [26] constructs a dynamic Laplace matrix of the graph using the spatial attention mechanism for each input sequence data. A3TGCN [4], an extension of TGCN, achieves significant prediction performance enhancement by incorporating attention to adjust the significance of time points in the time series. GMAN [27] employs an encoder–decoder architecture to model the influence of spatial-temporal factors on traffic conditions. The model comprises multiple spatial-temporal attention blocks that consist of a spatial attention mechanism, a temporal attention mechanism, and a gated fusion mechanism. Similarly to GMAN, STTN [5] contains stacked spatial-temporal blocks and prediction layers. Each spatial-temporal block involves a spatial transformer and a temporal transformer to jointly extract spatial-temporal correlations. ASTGNN [28] and traffic transformer [29] adopt the transformer architecture and utilize self-attention to model complex spatial-temporal correlations. Both models employ encoder–decoder structures to form a deep model of hierarchical features. ASTTN [6] provides a comprehensive analysis of local spatial-temporal attention mechanisms and full spatial-temporal attention mechanisms in the construction of the spatial-temporal transformer on a dynamic graph.

In conclusion, the self-attention mechanism is proficient in encapsulating the dynamic spatial-temporal correlations inherent in traffic conditions. Moreover, the hierarchical local self-attention mechanism enhances the accuracy of capturing local spatial dependencies and short-term temporal correlations. It also efficiently amalgamates local and global spatial dependencies, as well as short-term and long-term temporal correlations. This serves as the foundation for the current study.

3. Methodology

This section proposes the framework of HSTAN for traffic flow forecasting. We first introduce some definitions. Then, we introduce our proposed model and describe how it works for the traffic.

3.1. Problem Formulation

We represent an urban road network G as a weighted graph of the form $G = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is a set of nodes and $|\mathcal{V}| = N$, corresponding to N roads, and $\mathcal{E}$ is a set of edges and $|\mathcal{E}| = E$, reflecting connections between roads. The $A \in \mathbb{R}^{N \times N}$ represents an adjacency matrix of traffic flow data. After observations at time step t in a road network, a traffic signal matrix $X_t = \left(x_t^1, x_t^2, \ldots, x_t^N\right) \in \mathbb{R}^{N \times C}$ can be obtained, where C represents the number of features. The goal of traffic flow prediction is to learn a function f that can predict the feature matrix for T' future time steps, given a graph G and a feature matrix of historical T steps. The function mapping relationship is shown in Equation (1).

$$[X_{t-T+1:t}, G] \xrightarrow{f} X_{t+1:t+T'} \tag{1}$$

Here, $X_{t-T+1:t} \in \mathbb{R}^{T \times N \times C}$ and $X_{t+1:t+T'} \in \mathbb{R}^{T' \times N \times C}$, we assume that G is fixed and independent of time.

3.2. Overview of Model Architecture

This paper introduces a hierarchical spatial-temporal attention networks (HSTAN) model, depicted in Figure 2. The model primarily comprises an input embedding layer, a hierarchical spatial-temporal self-attention module (STA-Blocks), and an output layer, as shown in the left part of Figure 2. The STA-Blocks, stacked in multiple layers, each include a temporal attention layer and a spatial attention layer. The temporal attention layer identifies the temporal correlation of historical data on future time points, while the spatial attention layer discerns the spatial dependency of locally adjacent nodes. The hierarchical structure of the residual association facilitates the capture of more profound spatial-temporal correlations, thereby enhancing the model's predictive performance. Lastly, a two-layer fully

connected neural network is employed to transform the feature space and generate traffic forecasts for all nodes within the target road network.

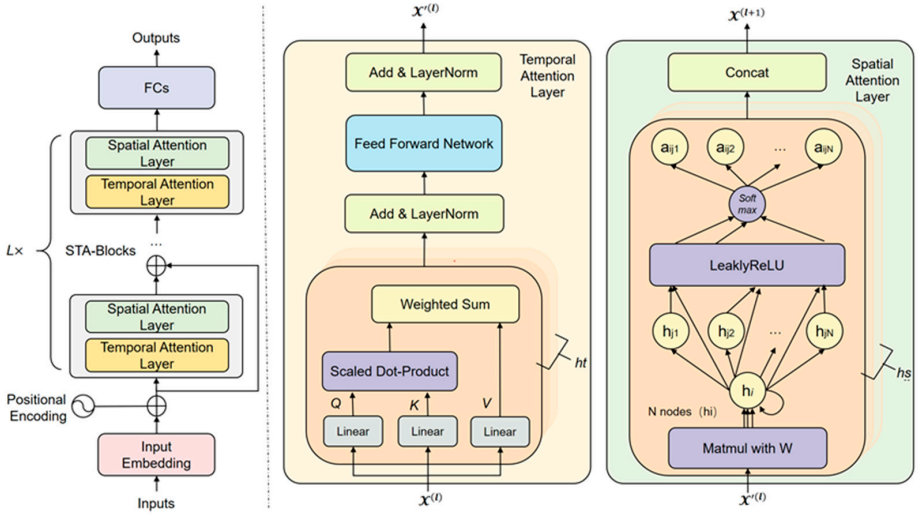


Figure 2. Overall hierarchical architecture of spatial-temporal attention networks. It consists of stacked STA-Blocks and input/output layers. Each STA-Blocks module consists of a spatial attention layer and a temporal attention layer, which uses a multi-headed self-attention mechanism.

3.3. Input Embedding Layer

The input for a traffic prediction model is represented as $\boldsymbol{\mathcal{X}} = [X_{t-T+1}, \ldots, X_t]$, where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T \times N \times C}$. In this representation, N nodes from the traffic network are chosen to be included as input to the model, forming a time series with T steps. The goal of the input embedding layer is to convert the input tensor into $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T \times N \times D}$ using a fully connected network, where D represents the length of the embedding. To analyze the temporal correlation, it is necessary to accurately identify each point in the time series. In this paper, we utilize the position encoding method introduced in [30], represented by Equations (2) and (3). This method incorporates the same dimension D as the input embedding module. Finally, the input embedding vector and the position encoding are combined to produce the output of the input embedding layer.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10,000^{\frac{2i}{D}}}\right) \tag{2}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10,000^{\frac{2i}{D}}}\right) \tag{3}$$

Here, pos represents the data location, and i represents the data dimension. In the position encoding, the sine function curve represents the variation along different dimensions of the input data, forming a geometric series ranging from 2π to $10,000 \cdot 2\pi$.

3.4. STA-Blocks

This paper proposes STA-Blocks, which comprise a temporal attention layer and a spatial attention layer. These layers work together to extract spatial-temporal correlations. The temporal attention layer models non-linear temporal correlations, while the spatial attention layer captures dynamic spatial dependencies.

3.4.1. Temporal Attention Layer

In a traffic network, there is typically a correlation between traffic conditions at different time steps within a sequence at each location. This correlation is known as temporal correlation, and it can be classified into two categories: short-term and long-term. Specifically, "short-term" refers to a smooth variation or, in other words, a tendency to

remain stable for a relatively extended period of time. On the other hand, "long-term" indicates significant variations in impact.

The temporal correlations in this layer are modeled using a multi-headed self-attentive mechanism, as shown in the middle part of Figure 2, utilizing an input time series $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T \times N \times D}$. Here, T represents the length of the time series, and D denotes the dimensionality after input embedding. This layer includes the multi-headed self-attention module, the residuals and layer normalization module, and the feed-forward network module (FFN).

The multi-headed self-attention module computes temporal correlations using three dynamic high-latitude subspaces. These subspaces consist of the query vector, the keyword vector K, and the value vector V, computed as shown in Equations (4) and (5).

$$Q = W_q \times \boldsymbol{\mathcal{X}}; K = W_k \times \boldsymbol{\mathcal{X}}; V = W_v \times \boldsymbol{\mathcal{X}} \tag{4}$$

$$Att(\boldsymbol{\mathcal{X}}) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

We define the parameter matrices of linear mapping as $W_q \in \mathbb{R}^{N \times T \times D}$, $W_k \in \mathbb{R}^{N \times T \times D}$, $W_v \in \mathbb{R}^{N \times T \times D}$. Additionally, the query vector Q is represented as $Q = [q_1, \ldots, q_N]$, the key vector as $K = [k_1, \ldots, k_N]$, and the value vector as $V = [v_1, \ldots, v_N]$. The self-attention mechanism is computed using a scaled dot product, where each element in the key vector is divided by $\sqrt{d_k}$. This is then followed by applying a ***softmax*** function to obtain the weight coefficients corresponding to each time step.

The multi-headed self-attention mechanism (MSA) calculates the correlation between all input elements and generates a weight matrix. This mechanism utilizes multiple attention modules, each with different weights assigned to the same query. The concept is to generate multiple queries by linearly transforming the input query using different weight matrices. Consequently, the attention model is capable of incorporating additional information into the computation of contextual vectors. The multi-headed self-attention can be mathematically expressed as Equation (6).

$$MSA(\boldsymbol{\mathcal{X}}) = \frac{1}{ht}||_{i=1}^{ht}[Att(\boldsymbol{\mathcal{X}})] \tag{6}$$

Here, ht represents the number of attention heads. The feature vector is upscaled to a higher dimension and subsequently split into ht multi-dimensional vectors, which are later averaged. The symbol $\|$ corresponds to the operation of the Concat function.

In the module of residuals and layer normalization, the output of the MSA module undergoes a projection or dropout operation, resulting in residual and layer normalization. Layer normalization is defined by Equation (7). The first reason for this normalization is to adapt the model to a data distribution with a variance between 0 and 1, improving the neural network's generalization. The second reason is to apply layer normalization before the FFN, preventing data from saturating and reducing the issue of gradient vanishing.

$$\boldsymbol{\mathcal{X}} = LayerNorm(\boldsymbol{\mathcal{X}} + MSA(\boldsymbol{\mathcal{X}})) \tag{7}$$

The FFN module performs a non-linear transformation on the output of the preceding module, consisting of three steps: linear, non-linear, and linear sub-transformation, thus providing a comprehensive representation of temporal dependence as illustrated in Equation (8). The paper employs the ***Relu*** function as the activation function.

$$\hat{\boldsymbol{\mathcal{X}}} = FC(FC(Relu(\boldsymbol{\mathcal{X}}))) \tag{8}$$

Finally, the output of FFN via another layer normalization module, as shown in Equation (9).

$$\hat{\boldsymbol{\mathcal{X}}} = LayerNorm\left(\boldsymbol{\mathcal{X}} + \hat{\boldsymbol{\mathcal{X}}}\right) \tag{9}$$

3.4.2. Spatial Attention Layer

In a traffic network, the traffic conditions between different locations exhibit interactive and dynamic behavior, referred to as spatial dependency. The main objective of the

spatial attention mechanism is to identify the features of neighboring nodes and capture the interdependency among them. This layer utilizes GAT [20], a form of multi-headed self-attention mechanism, as shown in the right part of Figure 2, to capture the spatial dependency. The GAT is expressed in Equations (10)–(13).

$$e_{ij} = a\left(W\vec{h}_i, W\vec{h}_j\right) \tag{10}$$

Let h represent the output of the temporal attention layer, where $h = \left\{\vec{h}_1, \vec{h}_2, \ldots\ldots \vec{h}_N\right\}$, and $\vec{h}_i \in R^F$. Here, N denotes the number of nodes, and F indicates the number of features. Moreover, W symbolizes the shared parameter matrix utilized for linearly transforming the features of each node, and it belongs to the space of $W \in \mathbb{R}^{F' \times F}$.

$$\alpha_{ij} = softmax\left(j\left(e_{ij}\right)\right) = \frac{exp(e_{ij})}{\sum_{k \in Ni} exp(e_{ik})} \tag{11}$$

where α_{ij} means the importance of node j's features to node i, as $\alpha_{ij} \in \mathbb{R}^{N \times N}$.

$$\alpha_{ij} = \frac{exp(LeakyReLU\left(\vec{a}^T\left[W\vec{h}_i \middle\| W\vec{h}_j\right]\right))}{\sum_{k \in Ni} exp(LeakyReLU\left(\vec{a}^T\left[W\vec{h}_i \middle\| W\vec{h}_j\right]\right))} \tag{12}$$

where $\vec{a}$ represents a single-layer FFN, parameterized by a weight vector $\vec{a} \in \mathbb{R}^{2F'}$ and non-linearized by applying the ***LeakyRelu*** function with a negative input slope, α = 0.2.

$$\vec{h}_j^{\,l+1} = \sigma\left\{\sum_{j \in N_i} \alpha_{ij} W\vec{h}_j\right\} \tag{13}$$

where α_{ij} represents the correlation coefficient between nodes, which $W\vec{h}_j$ signifies the weight of the feature vectors propagated from the previous layer.

To enhance the generalization of the spatial attention layer and promote training stability, we incorporate K groups of self-attention mechanisms with identical dimensions across the input. These mechanisms are joined together using a splicing function, and dimension alignment is achieved through the mean operation. By employing multiple heads, we foster stronger spatial dependency between nodes, enabling a more rational assignment of weight coefficients. This is particularly beneficial for key nodes and leads to improved overall generalization. The mathematical representation of the multi-headed spatial attention mechanism can be found in Equation (14).

$$\vec{h}_j' = ||_{i=1}^{hs} \sigma\left(\sum_{j \in N_i} a_{ij}^k W^k \vec{h}_j\right) \tag{14}$$

where hs represents the number of heads on GAT. The normalized coefficient, a_{ij}^k, is calculated by the k_{th} group of attention mechanism. W^k denotes the parameter matrix of the k_{th} group. The symbol $\|$ corresponds to the operation of the ***Concat*** function.

3.5. Hierarchical Structure

The STA-Blocks can efficiently capture spatial-temporal correlations, and they are employed in the hierarchical structure to form deep models of spatial-temporal features. Within this model, residual connections are added between the layers to prevent the gradient from vanishing throughout the hierarchy. The hierarchical structure with residuals is presented in Equation (15).

$$\boldsymbol{\mathcal{X}}^{(l+1)} = \boldsymbol{\mathcal{X}}^{(l)} + \mathcal{F}\left(\boldsymbol{\mathcal{X}}^{(l)}, W_l\right) \tag{15}$$

where $\mathcal{F}$ represents the modules in the STA-Blocks, and W_l represents the parameter in the STA-Blocks, and $\boldsymbol{\mathcal{X}}^{(l+1)}$ is the input for the next STA-Blocks, using the residual connection.

3.6. Output Layer

The output layer comprises a two-layer fully connected structure, designed to enable multi-step traffic prediction, as shown in Equation (16).

$$\hat{Y} = FC\left(FC\left(\boldsymbol{\mathcal{X}}^{(l)}\right)\right) \tag{16}$$

where $\boldsymbol{\mathcal{X}}^{(l)}$ represents the output of the lth STA-Blocks, and $\hat{Y}$ represents the predicted traffic flow, where $\hat{Y} \in \mathbb{R}^{N \times T}$.

In this paper, we employ the absolute error between the true and predicted values and adjust the optimization by adding a regularization term. Therefore, the loss function is represented in Equation (17). The first term in Equation (17) computes the minimum error between the true speed values and the predicted speeds, while the second term represents a normalization term.

$$Loss = \frac{1}{n}\sum_{i=1}^{n}\left|y_{true} - y_{pred}\right| + \lambda L_{reg} \tag{17}$$

where y_{true} represents the true speed value at a given time step, and y_{pred} represents the predicted speed value, where λ is the hyper-parameter used for normalization.

3.7. Algorithm Description

Algorithm 1 provides an outline of the training process of the HSTAN model. First, training instances are constructed by constructing training instances from raw data. Then, the model is trained to utilize backpropagation and the Adam algorithm.

Algorithm 1. HSTAN Training Algorithm

Input: The historical traffic flow sequence by input embedding: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T \times N \times D}$; number of train epoch: epochs; count of heads on temporal attention mechanism: *ht*; count of heads on spatial attention mechanism: *hs*; layers with residuals: L; loss function: loss
Output: traffic flow prediction value: $\hat{y}$

```
For e ∈ epochs do
    x ∈ batch
    x ← x + postion − encode(x)
    y ∈ batch
    For k ∈ K do
        For t ∈ ht do        # Temporal Attention
            Initialize W_q, W_k, W_v
            q ← W_q × x
            k ← W_k × x
            v ← W_v × x
            Att_t ← ComputerAttention(q, k, v)
            x_t^k ← Layer normal(Att_t)
            x^t ← Concat(x_t^k) / k
        End For
        For s ∈ hs do        # Spatial Attention
            Initialize W_h, W_j
            Att_s ← ComputerCorrelation(W_h x^t, W_j x^t)
            Att'_s ← Softmax(Att_s)
            x_s^k ← Att'_s x^t
            x^s ← Concat(x_s^k) / k
        End For
        ŷ ← FC(FC(x^s))
    End For
    loss ← L(y, ŷ)
End For
```

4. Experiments Datasets

4.1. Datasets

This study used two real-world datasets for traffic forecasting, namely SZ-taxi and Los-loop.

1. **SZ-taxi:** The dataset is the Shenzhen taxi trajectory from 1 January to 31 January 2015. The 156 major roads in Luohu District were selected as the study area, and the experimental data consisted of two main parts: one was a 156 × 156 adjacency matrix describing the spatial relationships between roads, with each row representing a road and the values in the matrix, indicating the connectivity between roads; the other was a feature matrix describing the change in speed over time on each road, with each row representing a road and each column representing the speed of traffic on the road at different times of the day. The speed of traffic on each road is calculated every 15 min.
2. **Los-loop:** This dataset is collected in real time using loop detectors on motorways in Los Angeles County. It consists of 207 sensors with traffic speeds collected from 1 March 2012 to 7 March 2012. These traffic speed data are aggregated every 5 min.

4.2. Benchmarks

Our proposed approach is compared with five benchmark methods. These benchmarks are described below:

- ARIMA: Autoregressive Integrated Moving Average (ARIMA) is a well-known model used to understand and predict future values in the time series data.
- T-GCN [3]: Temporal Graph Convolutional Network, which combines GCN with GRU for the extraction of spatial-temporal correlations for traffic forecasting.
- STGCN [14]: Spatial-Temporal Graph Convolution Network, which uses graph convolution and one-dimensional convolution to capture spatial-temporal dependency.
- AGCRN [18]: Adaptive Graph Convolutional Recurrent Network, which uses adaptive graphs and a combination of recurrent networks to capture spatial-temporal correlations.
- A3TGCN [4]: Attention Temporal Graph Convolutional Network, based on T-GCN, which adds an attention mechanism to capture both dynamics of global spatial-temporal correlations.

4.3. Experiment Settings

Our experiments are conducted on a computer with Inter(R) Core(TM) i7-10750 CPU, 3.50 GHz with NVIDIA GeForce RTX 3090, and GPU with 24 G of memory. The number of layers of the STA-Blocks is set to 6. The number of heads of the multi-head attention on the spatial term is set to 8, and on the temporal term it is set to 8 too. Adam is used as an optimizer with a weight decay of 1×10^{-4} to train with a loss function to minimize mean absolute errors (MAE). The learning rate for all experimental models was set to 0.001, and each experiment was trained for 50 epochs. To demonstrate the performance of the proposed model, we compare it with the results reported in [4], where the current 12 observations (60 min) are used to predict the traffic conditions in the next 15, 30, 45, and 60 min using SZ-taxi and 15, 30, 45, and 60 min using Los-loop.

During the experiments, three measures were employed to evaluate the accuracy of the prediction models: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), which are shown in Equations (18)–(20).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{18}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{19}$$

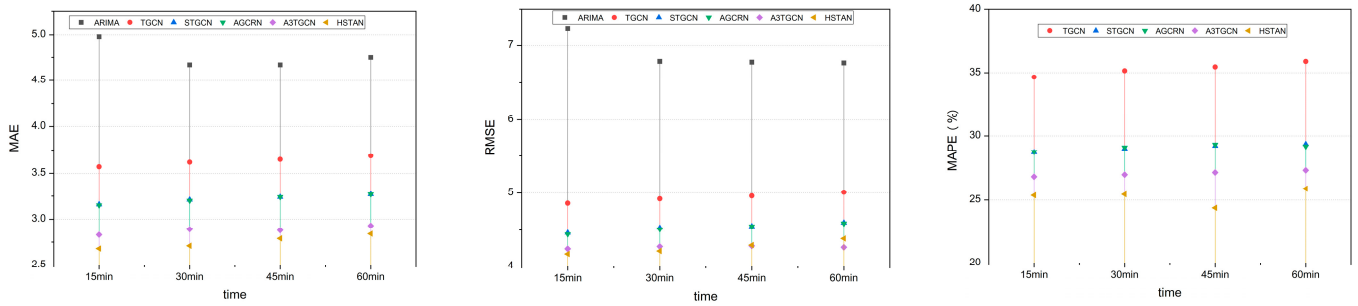
$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (20)$$

4.4. Experiment Result Analysis

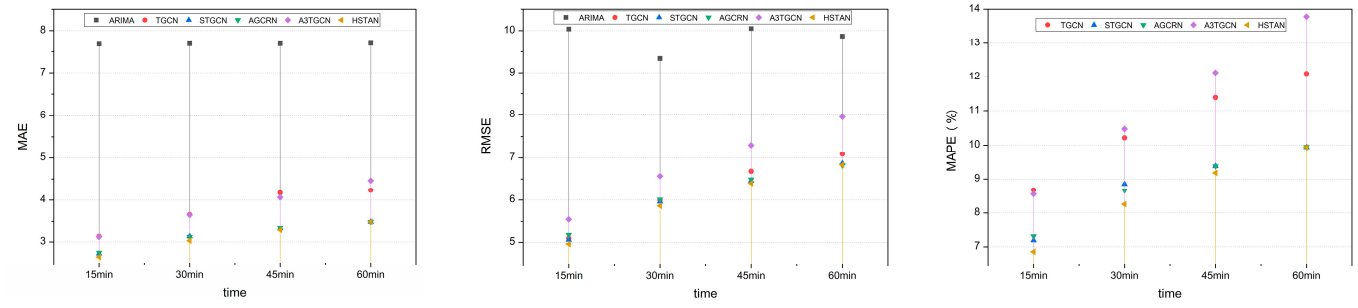
We compare the performance of the HSTAN model with benchmark models listed in Table 1 and Figure 3 for time intervals of 15 min (3 steps), 30 min (6 steps), 45 min (9 steps), and 60 min (12 steps) using the SZ-Taxi and Los-loop datasets.

Table 1. Traffic prediction on the SZ-taxi and Los-loop datasets.

Data	Models	15 min			30 min			45 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SZ-Taxi	ARIMA	4.98	7.24	--	4.67	6.79	--	4.67	6.78	--	4.75	6.77	--
	TGCN	3.57	4.86	34.65	3.62	4.92	35.16	3.65	4.96	35.46	3.69	5.01	35.90
	STGCN	3.16	4.46	28.70	3.21	4.53	28.99	3.24	4.55	29.22	3.27	4.60	29.35
	AGCRN	3.15	4.44	28.74	3.20	4.51	29.09	3.24	4.55	29.31	3.27	4.59	29.17
	A3TGCN	2.83	4.24	26.82	2.89	4.27	26.98	2.88	**4.28**	27.14	2.93	**4.26**	27.31
	HSTAN (Ours)	**2.68**	**4.17**	**25.37**	**2.71**	**4.21**	**25.45**	**2.79**	4.29	**24.38**	**2.84**	4.38	**25.89**
Los-loop	ARIMA	7.68	10.04	--	7.69	9.34	--	7.69	10.05	--	7.70	9.87	--
	TGCN	3.13	5.09	8.67	3.66	5.99	10.21	4.17	6.68	11.39	4.23	7.09	12.10
	STGCN	2.73	5.06	7.18	3.13	5.96	8.85	3.32	6.46	9.38	**3.49**	6.87	**9.94**
	AGCRN	2.75	5.17	7.30	3.12	6.01	8.66	3.33	6.49	9.37	**3.49**	**6.82**	9.95
	A3TGCN	3.12	5.55	8.55	3.65	6.57	10.46	4.06	7.30	12.13	4.46	7.95	13.78
	HSTAN (Ours)	**2.65**	**4.96**	**6.86**	**3.03**	**5.86**	**8.25**	**3.28**	**6.40**	**9.18**	**3.49**	6.83	**9.94**



(**a**)



(**b**)

Figure 3. Comparison of our model and baselines using the SZ-taxi and Los-loop datasets, which are listed as (**a**) MAE, RMSE, and MAPE for the SZ-taxi set; (**b**) MAE, RMSE, and MAPE for the Los-loop dataset.

Our findings indicate that (1) our model consistently achieves the best or second-best prediction performance compared to the benchmarks on both datasets from 3 steps to 12 steps, with the advantages being more pronounced in short-term horizon predictions. (2) Deep learning models that incorporate spatial dependency as graph structures consis-

tently outperform conventional statistical method ARIMA. This demonstrates the power of deep learning models and highlights the importance of incorporating graph structures into temporal prediction. (3) Both A3TGCN and our model outperform conventional graph deep learning models, namely T-GCN, STGCN, and AGCRN, using the SZ-Taxi dataset. This demonstrates the significance of the global feature-capturing capability offered by the self-attention mechanism. The relatively low prediction error can possibly be attributed to the superior global spatial-temporal feature-capturing capability of the self-attention mechanism in traffic prediction compared to the local convolutional operation of the GCN-based methods. This is true regardless of the predefined graph structure or the use of adaptive graph structure by the GCN-based methods. (4) The superiority of our model over the A3TGCN model indicates the necessity of fusing local and global spatial-temporal correlations via the hierarchical structure to enhance model performance. On the SZ-taxi dataset, our model exhibits reductions of approximately 5.3% in MAE, 1.65% in RMSE, and 5.4% in MAPE compared to the A3TGCN model in the 15 min time series; approximately 6.22% in MAE, 1.4% in RMSE, and 5.67% in MAPE in the 30 min time series; approximately 3.12% in MAE, 0.2% in RMSE, and 10.16% in MAPE in the 45 min time series,; and approximately 3.07% in MAE, 2.81% in RMSE, and 5.2% in MAPE in the 60 min time series. (5) Our model outperforms other GCN models, specifically T-GCN, STGCN, and AGCRN, with the advantages being more pronounced in short-term horizon predictions. In the 15 min time series, our model demonstrates reductions of approximately 15.33% in MAE, 2.55% in RMSE, and 20.87% in MAPE compared to the T-GCN model; approximately 2.93% in MAE, 1.97% in RMSE, and 4.45% in MAPE compared to the STGCN model; and approximately 3.63% in MAE, 4.06% in RMSE, and 6.02% in MAPE compared to the AGCRN model using the Los-loop dataset. In the 60 min time series, our model exhibits reductions of approximately 17.49% in MAE, 3.66% in RMSE, and 17.85% in MAPE compared to the T-GCN model. Additionally, our model performs at a similar level to the STGCN and AGCRN models in terms of MAE, RMSE, and MAPE metrics. The relatively high performance in short-term prediction can possibly be attributed to the increased importance of local spatial attention due to the relatively small changes in the temporal dimension.

To examine the influence of each component in our model, we proceed to assess the performance of the model variants by eliminating the temporal attention layer (HSTAN w/o TA) and the spatial attention layer (HSTAN w/o SA). Table 2 clearly demonstrates that HSTAN consistently outperforms its variants, underscoring the significance of the spatial attention layer and the temporal attention layer in capturing intricate spatial-temporal dependencies. Hence, the prediction error of HSTAN w/o SA is lower than that of HSTAN w/o TA, thereby establishing the primacy of the temporal attention layer over the spatial attention layer.

Table 2. Comparisons of model variants using the SZ-taxi and Los-loop datasets.

Data	Model	MAE					RMSE				
		15 min	30 min	45 min	60 min	Avg.	15 min	30 min	45 min	60 min	Avg.
SZ-Taxi	HSTAN w/o TA	4.80	4.82	4.84	4.88	4.83	6.24	6.29	6.60	6.36	6.37
	HSTAN w/o SA	3.57	3.65	3.83	3.95	3.75	4.77	4.89	5.17	5.60	5.10
	HSTAN	2.68	2.71	2.79	2.84	2.75	4.17	4.21	4.29	4.38	4.26
Los-loop	HSTAN w/o TA	7.64	7.72	7.78	7.93	7.76	10.71	10.86	10.88	11.01	10.86
	HSTAN w/o SA	3.03	3.87	5.36	5.12	4.34	5.47	6.93	8.67	8.81	7.47
	HSTAN	2.65	3.03	3.28	3.49	3.11	4.96	5.86	6.40	6.83	6.01

Figure 4 clearly demonstrates that HSTAN consistently outperforms its variants. The temporal attention layer is crucial for both datasets, providing a 43% and 33.1% reduction in MAE and RMSE for the SZ-taxi data, and a 57.2% and 44.6% reduction in MAE and RMSE for the Los-loop data. This indicates that temporal correlations play an important role in the traffic data imputation problem for urban road networks. With the temporal

attention layer ablated, the MAE and RMSE values of the SZ-taxi data increase to 26.6% and 16.5%, whereas the MAE and RMSE values of the Los-loop data increase to 28.3% and 19.5%. We can find that temporal dependencies contribute more to the traffic data imputation problem than spatial correlations.

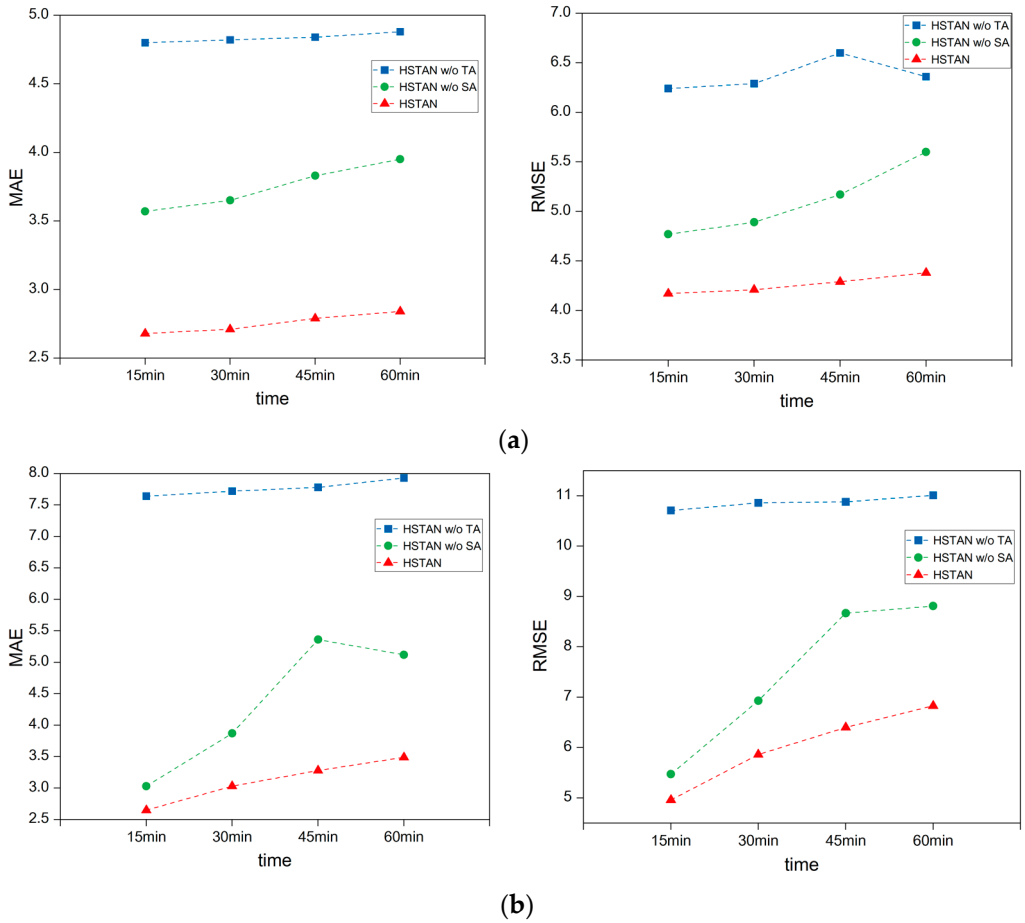
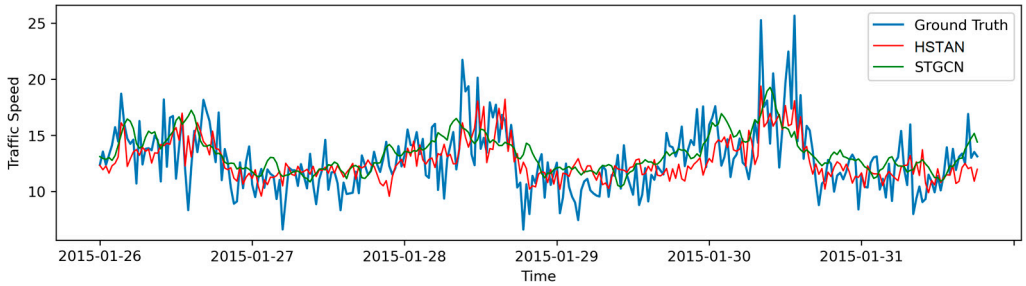


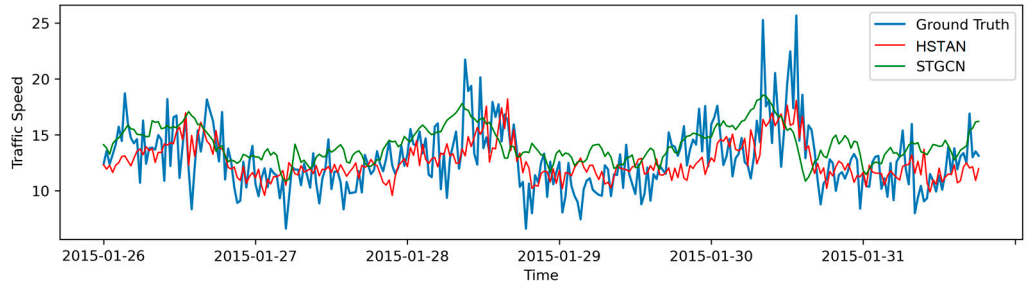
Figure 4. Comparison of model variants using the SZ-taxi and Los-loop datasets, which are listed as (**a**) MAE and RMSE for the SZ-taxi set; (**b**) MAE and RMSE for the Los-loop dataset.

4.5. Visualized Analysis

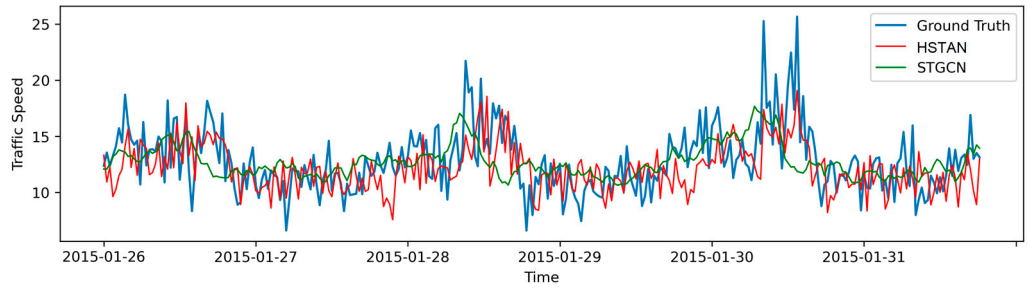
The HSTAN and STGCN models both employ stacked hierarchies to capture spatial and temporal information. However, there are differences in the specific techniques used within each spatial-temporal block. The STGCN model utilizes 1-D temporal convolutions and spectral-based graph convolutions in each spatial-temporal block. On the other hand, the HSTAN model incorporates the temporal attention layer and spatial attention layer based on the attention mechanism. Thus, the forecasting outcomes of the HSTAN and STGCN models, employing two real datasets, are visually presented to facilitate a comprehensive understanding of the models' performance. For the SZ-taxi dataset, we selected one road segment and visualized the results from 27 January to 1 February 2015. The visualization results for the 15, 30, 45, and 60 min time series are displayed in Figure 5. Similarly, for the Los-loop dataset, we visualized the data from one loop detector. The visualization results for the 15, 30, 45, and 60 min intervals are shown in Figure 6.



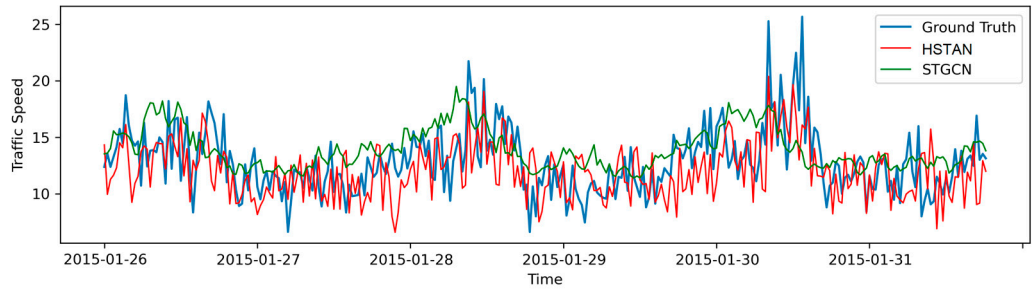
(**a**) 15 min



(**b**) 30 min



(**c**) 45 min

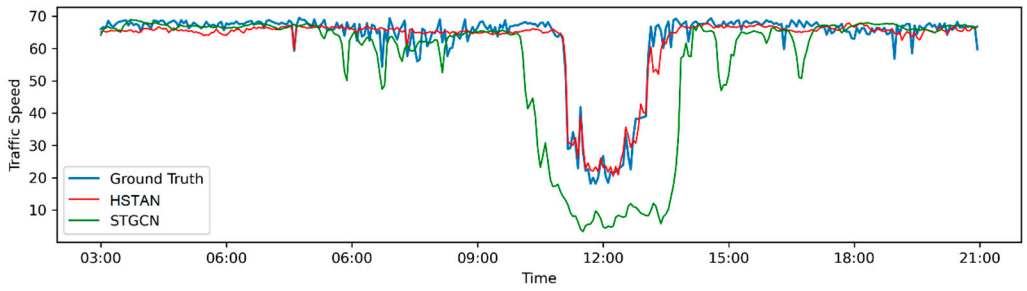


(**d**) 60 min

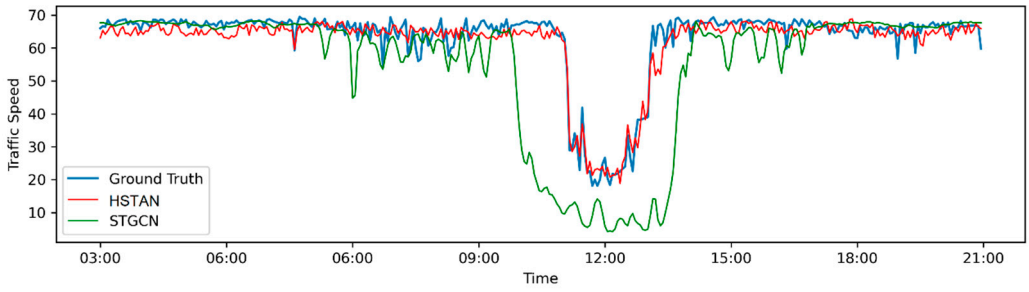
Figure 5. Comparison of the prediction effect of the SZ-taxi dataset, which are listed as (**a**) description of the predication in 15 min; (**b**) description of the predication in 30 min; (**c**) description of the predication in 45 min; (**d**) description of the predication in 60 min.

Overall, the predicted traffic speeds exhibit a consistent variation trend compared to the actual traffic speeds across different time series lengths. This indicates the competence of our model in the task of traffic forecasting. In Figure 5, our model aptly captures the variation trends in traffic speed and identifies the start and end points of rush hours. It also demonstrates a smoother traffic prediction than the STGCN model, showcasing its enhanced generalization ability. Moreover, the superiority of the multi-headed self-attention mechanism and graph attention mechanism over traditional graph convolution and time series convolution is clearly evident, as they enable more precise traffic prediction.

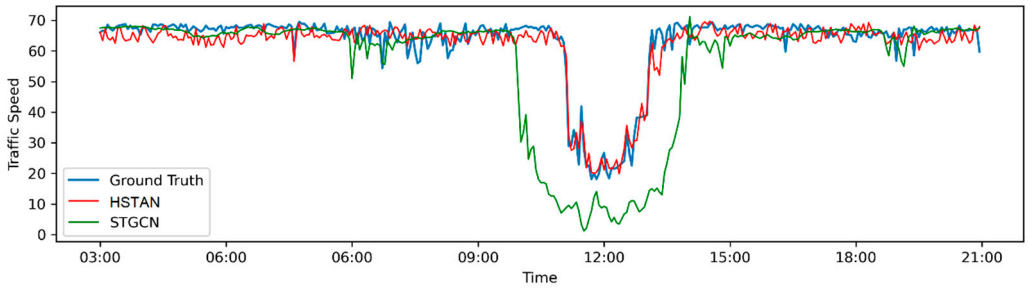
Figure 6 exemplifies the accurate anticipation of traffic congestion during rush hours, further validating the efficacy of our model in real-time traffic forecasting.



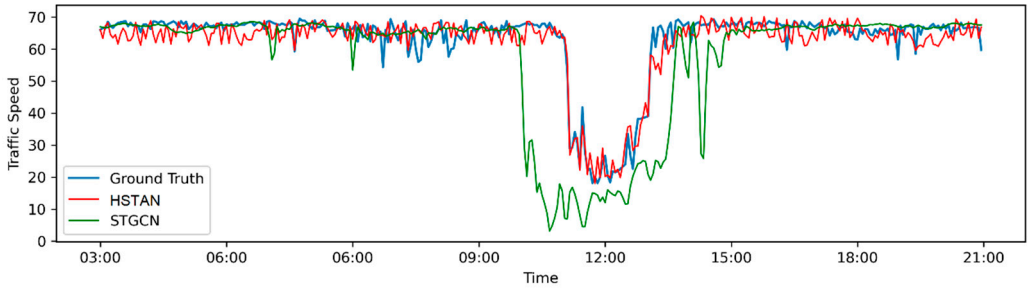
(**a**) 15 min



(**b**) 30 min



(**c**) 45 min



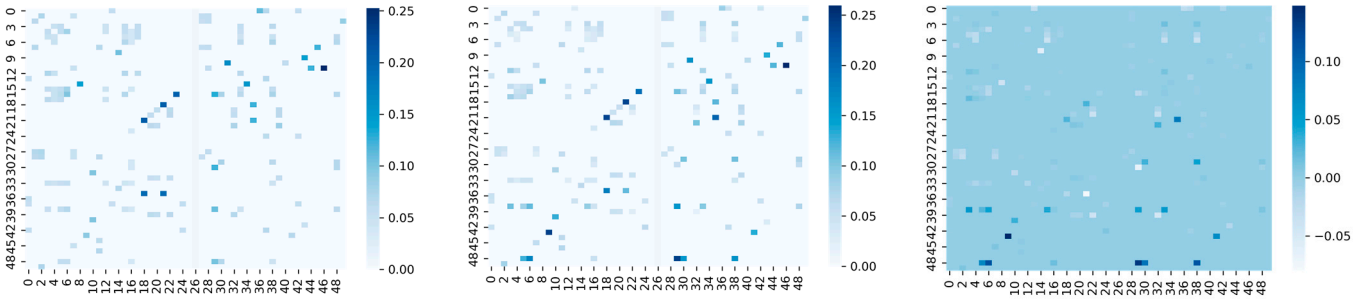
(**d**) 60 min

Figure 6. Comparison of the prediction effect of the Los-loop dataset, which are listed as (**a**) description of the predication in 15 min; (**b**) description of the predication in 30 min; (**c**) description of the predication in 45 min; (**d**) description of the predication in 60 min.

4.6. Effect of Spatial-Temporal Attention

We demonstrate the impact of diverse spatial-temporal attention mechanisms, as discussed in Section 3.4. Additionally, we conduct an analysis of the attention weight matrix of the multi-headed self-attention mechanism in the spatial attention layer. We examined the data from 0:00 a.m. to 12:00 p.m. of one day in the SZ-taxi test set. The heat maps of the weight matrices, involving the top 50 nodes, are presented in Figure 7. Our

findings indicate that (1) the spatial interdependence among the graph nodes changes over time, rendering spatial attention dynamic, and (2) the attention mechanism illustrates the temporal dynamics of spatial interdependence more effectively than the widely recognized graph convolution.



(**a**) (**b**) (**c**)

Figure 7. Heat maps of weight matrix on spatial attention. It consists of a heat map at 0:00 a.m. (**a**), a heat map at 12:00 p.m. (**b**), and the difference between the above two times (**c**).

5. Conclusions

In this paper, we propose an efficient algorithm for traffic forecasting that aims to enhance the characterization of spatial-temporal correlations. Our proposed method effectively captures spatial-temporal correlations by utilizing a multi-headed self-attention mechanism in both temporal and spatial terms. The hierarchical structure allows for the extraction of local spatial-temporal features and the fusion of global spatial-temporal features. We evaluate the performance of our model on two commonly used real traffic datasets, where it outperforms five benchmark models. Furthermore, we conduct ablation experiments to examine the individual components and analyze the attention weight matrix to gain insights into the inner workings of our model. In comparison to previous CNN, RNN, transformer, and other networks, our proposed method proves to be more effective, featuring larger receptive fields and fewer training parameters, leading to more efficient and rapid training. In our future work, we plan to utilize the HSTAN model for updating the weighted adjacency matrix based on the output of each layer. One of our primary objectives is to investigate the impact of attention on the spatial-temporal graph with a dynamic topology.

Author Contributions: Conceptualization, Q.L. and W.S.; methodology, Q.L.; software, W.D.; data curation, W.S. and W.D.; writing—original draft preparation, Q.L.; writing—review and editing, Q.L. and W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Two real-world traffic data sets are available in reference [4]. Shenzhen dataset (SZ-taxi) and ring detector dataset (Los-loop) are used in Los Angeles. There are restrictions on the availability of these data, which are used with permission in this study.

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

References

1. Veres, M.; Moussa, M. Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3152–3168. [CrossRef]
2. Jiang, W.; Luo, J. Graph Neural Network for Traffic Forecasting: A Survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [CrossRef]

3. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3848–3858. [CrossRef]
4. Bai, J.; Zhu, J.; Song, Y.; Zhao, L.; Hou, Z.; Du, R.; Li, H. A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 485. [CrossRef]
5. Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.J.; Xiong, H. Spatial-Temporal Transformer Networks for Traffic Flow Forecasting. *arXiv* **2020**, arXiv:2001.02908.
6. Feng, A.; Tassiulas, L. Adaptive Graph Spatial-Temporal Transformer Network for Traffic Flow Forecasting. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM'22), New York, NY, USA, 17–21 October 2022; pp. 3933–3937. [CrossRef]
7. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
8. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
9. 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]
10. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; pp. 324–328. [CrossRef]
11. Tran, Q.H.; Fang, Y.-M.; Chou, T.-Y.; Hoang, T.-V.; Wang, C.-T.; Vu, V.T.; Ho, T.L.H.; Le, Q.; Chen, M.-H. Short-term traffic speed forecasting model for a parallel multi-lane arterial road using gps-monitored data based on deep learning approach. *Sustainability* **2022**, *14*, 6351. [CrossRef]
12. Yang, D.; Chen, K.; Yang, M.; Zhao, X. Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features. *IET Intell. Transp. Syst.* **2019**, *13*, 1475–1482. [CrossRef]
13. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In Proceedings of the 6th International Conference on Learning Representations (ICLR'18), Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–16. [CrossRef]
14. 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 (IJCAI'18), Stockholm, Sweden, 13–19 July 2018; pp. 3634–3640. [CrossRef]
15. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19), Macao, China, 10–16 August 2019; pp. 1907–1913. [CrossRef]
16. Feng, N.; Guo, S.N.; Song, C.; Zhu, Q.C.; Wan, H.Y. Multi-component spatial-temporal graph convolution networks for traffic flow forecasting. *Ruan Jian Xue Bao/J. Softw.* **2019**, *30*, 759–769. [CrossRef]
17. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 914–921. [CrossRef]
18. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *arXiv* **2020**, arXiv:2007.02842.
19. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 753–763. [CrossRef]
20. Velikovi, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, arXiv:1710.10903.
21. Wu, T.; Feng, C.; Yun, W. Graph Attention LSTM Network: A New Model for Traffic Flow Forecasting. In Proceedings of the 5th International Conference on Information Science and Control Engineering (ICISCE'18), Zhengzhou, China, 20–22 July 2018.
22. Shih, S.Y.; Sun, F.K.; Lee, H.Y. Temporal Pattern Attention for Multivariate Time Series Forecasting. *arXiv* **2019**, arXiv:1809.04206.
23. Fang, X.; Huang, J.; Wang, F.; Zeng, L.; Wang, H. ConSTGAT: Contextual Spatial-Temporal Graph Attention Network for Travel Time Estimation at Baidu Maps. In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'20), Virtual Event, 6–10 July 2020. [CrossRef]
24. Liu, C.H.; Piao, C.; Ma, X.; Yuan, Y.; Leung, K.K. Modeling citywide crowd flows using attentive convolutional lstm. In Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE'21), Chania, Greece, 19–22 April 2021; pp. 217–228. [CrossRef]
25. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 922–929. [CrossRef]
26. Luo, X.; Zhu, C.; Zhang, D.; Li, Q. Dynamic Graph Convolution Network with Spatio-Temporal Attention Fusion for Traffic Flow Prediction. *arXiv* **2023**, arXiv:2302.12598.
27. Zheng, C.; Fan, X.; Wang, C.; Qi, J. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 1234–1241. [CrossRef]
28. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5415–5428. [CrossRef]

29. Yan, H.; Ma, X.; Pu, Z. Learning Dynamic and Hierarchical Traffic Spatiotemporal Features With Transformer. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 22386–22399. [CrossRef]
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.

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.