Applied Intelligence (2023) 53:12472–12488
https://doi.org/10.1007/s10489-022-04122-x

STGHTN: Spatial-temporal gated hybrid transformer network for traffic flow forecasting

Jiansong Liu[1] · Yan Kang[1] · Hao Li[1] · Haining Wang[1] · Xuekun Yang[1]

Accepted: 26 August 2022 / Published online: 28 September 2022

Abstract

Accurate traffic forecasting is a critical function of intelligent transportation systems, which remains challenging due to the complex spatial and temporal dependence of traffic data. GNN-based traffic forecasting models typically utilize predefined graphical structures based on prior knowledge and do not adapt well to dynamically changing traffic characteristics, which may limit their performance. The transformer is a compelling architecture with an innate global self-attention mechanism, but cannot capture low-level detail very well. In this paper, we propose a novel Spatial-Temporal Gated Hybrid Transformer Network (STGHTN), which leverages local features from temporal gated convolution, spatial gated graph convolution respectively and global features by transformer to further improve the traffic flow forecasting results. First, in the temporal dimension, we take full advantage of the local properties of temporal gated convolution and the global properties of transformer to effectively fuse short-term and long-term temporal dependence. Second, we mutually integrate two modules to complement each representation by utilizing spatial gated graph convolution to extract local spatial dependence and transformer to extract global spatial dependence. Furthermore, we propose a multi-graph model that constructs a road connection graph, a similarity graph, and an adaptive dynamic graph to exploit the static and dynamic associations between road networks. Experiments on four real datasets confirm the proposed method's state-of-the-art performance. Our implementation of the STGHTN code via PyTorch is available at https://github.com/JianSoL/STGHTN.

Keywords Graph convolution network · Temporal convolution network · Transformer · Spatial-temporal forecast

1 Introduction

Traffic forecasting is a core issue in the field of transport planning and management research [1–3], aiming to predict traffic based on historical information. Traffic forecasting theories and methods were introduced in the 1930s and they have seen success through years of research and practice. Quick and accurate traffic flow forecasting is an open research field. The advent of big data and technology such as intelligent transportation systems and sensors enables easy access to spatial-temporal data to facilitate traffic volume forecasting, for sensing congestion, control traffic conditions and enable travelers to choose appropriate travel modes and adjust routes [4–6].

Traffic forecasting has always been a challenge, due to complex temporal and spatial dependence. As shown in Fig. 1, traffic data exhibit temporal, spatial, and spatial-temporal dependence. Temporal dependence refers to the influence of historical moments on future moments, as reflected in the variation of traffic flow over time. Spatial dependence refers to the topology of roads, which directly influence each other at the same time step. This is manifested in the way that traffic flows on upstream roads affect downstream roads, and vice versa. Spatial-temporal dependence refers to the influence between roads at different moments in time. As shown in Fig. 2, temporal

✉ Yan Kang
kangyan@ynu.edu.cn

Jiansong Liu
ljs@mail.ynu.edu.cn

Hao Li
lihao707@ynu.edu.cn

Haining Wang
haining@mail.ynu.edu.cn

Xuekun Yang
kxyang@mail.ynu.edu.cn

[1] School of Software, Yunnan University, Chenggong District, Kunming, 650500, Yunnan, China

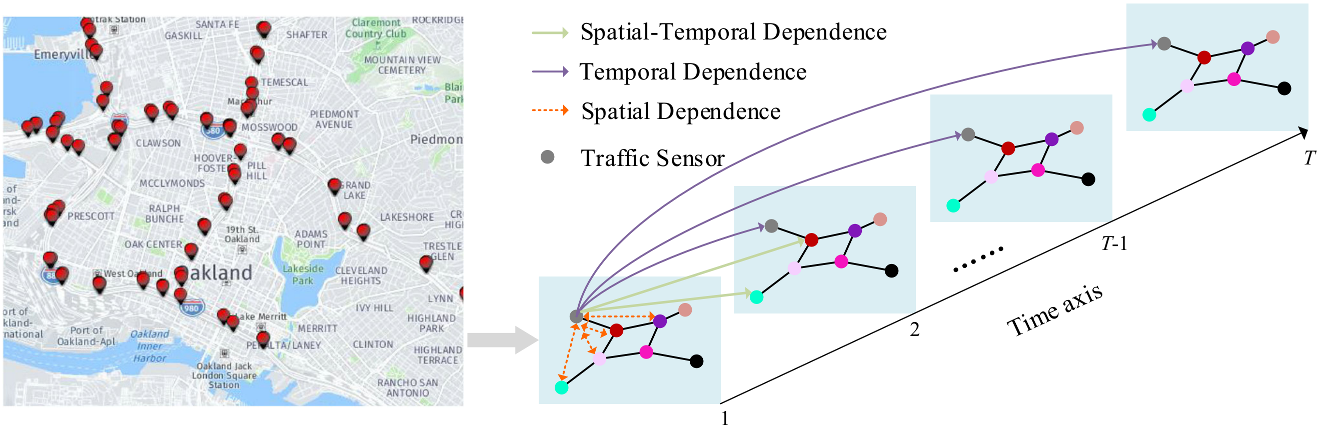


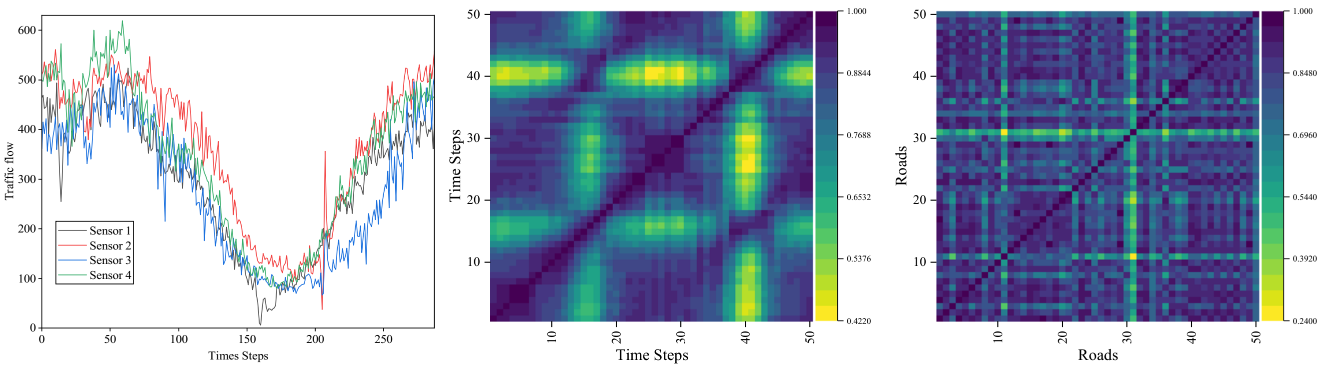
Fig. 1 Complex spatial and temporal dependence of traffic data

dependence usually shows a slight cyclical variation, and spatial dependence between roads shows different correlations depending on the environment.

Earlier traffic prediction methods were mainly based on statistical learning [7–9] and machine learning [10–12], modeled only on temporal dependence, and could not effectively capture the nonlinear dependence of spatial-temporal data. With the development of deep learning, methods such as DCRNN [13], T-GCN [14] and A3T-GCN [15] tend to capture temporal dependence through Recurrent Neural Networks (RNNs) [16] or variants, such as Long-Short Term Memory (LSTM) [17] and Gated Recurrent unit (GRU) [18]. These methods tend to capture only temporal dependence. GNN-based methods, such as STGCN [19], ASTGCN [20] and STFGNN [21] construct adjacency matrices based on prior knowledge, such as those based on road adjacency relationships and time-series similarity. Although the static spatial dependence between roads is fully considered, the spatial dependence should be dynamic, and road connectivity may change due to congestion. STGNN [22] adopts a learnable position attention mechanism to aggregate information about neighbouring roads, and utilises this aggregated information to generate a dynamic graph structure. This dynamic approach takes into account real-world events without considering the inherent spatial dependencies between roads.

The transformer [23] was first proposed in the NLP domain, and has achieved superior results in machine translation due to its strong ability to model the long-term features of sequences [24]. The transformer for the traffic prediction task is a pioneering work based on its key component, the multi-headed self-attention mechanism (MSA). It is noted that some local features and short-term dependence among input sequences cannot be effectively captured by transformer.

To address the above challenges and limitations, we propose a novel Spatial-Temporal Gated Hybrid Transformer Network (STGHTN) for traffic flow forecasting. In the temporal dimension, Temporal Gated Convolution (TGC) and transformer are hybridized to capture short-term and long-term temporal dependencies respectively. In the spatial dimension, Spatial Gate Graph Convolution (SGGC) and transformer are integrated to obtain local and global spatial



(a) Traffic flow (b) Temporal Correlation (c) Spatial Correlation

Fig. 2 Temporal and spatial pearson correlation analysis

dependence. SGGC constructs a multi-graph fusion model that fuses static and dynamic graph features to explore the spatial dependence of local spaces. We summarize our contributions as follows:

- We propose a novel transformer-based network model to effectively capture dynamic complex spatial-temporal features and then solve the prediction problem of spatial-temporal data.
- We design a Temporal Hybrid CNN-Transformer (THCT) to model short-term and long-term temporal dependence by integrating TGC and transformer.
- We propose a Spatial Hybrid GCN-Transformer (SHGT) consisting of SGGC and transformer on multi-graphs to extract local and global spatial dependence. Three different graphs such as road connection graph, similarity graph, and adaptive dynamic graph are constructed to fully exploit the static and dynamic associations between roads.
- We apply the proposed STGHTN to four real-world traffic datasets. STGHTN significantly surpasses SOTA algorithms on all four tasks.

2 Related work

2.1 Spatial-temporal forecasting

Traffic forecasting is an important part of intelligent transportation systems [25]. Early research treated traffic forecasting as a simple time-series problem without considering spatial features. Common traditional machine learning models include the Vector Autoregressive (VAR) model [26], Autoregressive Integrated Moving Average (ARIMA) [27], Support Vector Machine (SVM) [10], and Kalman filtering [9]. It is difficult to model nonlinear spatial-temporal data with these methods. With the development of deep learning, the RNN has been applied to sequence tasks. RNN cannot effectively solve long-term dependencies. And RNN requires iterative propagation, which may lead to cumulative errors.

LSTM and GRU were later proposed to alleviate this problem. LSTM and GRU cannot capture long-term temporal dependence adequately, and their performance may be limited due to an increased sequence length. The emergence of the transformer [23] solves this problem well, allowing better parallel training and capturing the dependence between long sequences. Correlations between multiple domains can be well captured by a self-attention mechanism. To improve the accuracy of the model, the researchers introduced spatial features into the model. Conv-LSTM [28] captures spatial features through a 1-dimensional Convolutional Neural Network (CNN). DMVST-Net [29] captures local features of regions in relation to their neighbors with a local CNN. These methods still capture spatial dependence mainly through CNNs. Although certain results have been achieved, CNNs are not applicable to the non-Euclidean space of roads. Previous network models have handled data with rules on Euclidean space. Graph convolution was created for non-Euclidean space, with great results for several types of tasks based on graph structures. Graph Neural Networks (GNNs) are seeing increased use for modeling spatial-temporal data. DCRNN [13], GCGRU [30] both use GNN to obtain spatial features, and are combined with RNN to obtain temporal features. GraphWaveNet [31] adaptively learns graph structure information without priority knowledge, and efficiently obtains spatial-temporal dependence at the same time.

2.2 Graph convolution networks

GNNs have been widely used in node classification, link prediction, and other graph-structured data tasks. Graph convolution methods include spectral and spatial convolution. Spectral convolution uses a graph Fourier transform for convolution. The original spectral-based approach [32] generalizes CNNs to non-Euclidean spaces. Although this method implements convolution on the graph, it is hard to implement because it is computationally complex and is not localized. ChebNet [33] reduces computational complexity by approximating the convolution kernel with K iterations of Chebyshev polynomials. Spatial convolution is the constant aggregation of information about a node's neighbors. NN4G [34] is the first spatial-based graph convolution neural network. GraphSAGE [35] extracts information about nodes by sampling and aggregating their neighbors, enabling the application of graph convolution networks to large-scale graphs. GAT [36] introduces an attention mechanism to adjust the weight relationship between neighboring nodes.

2.3 Transformer

The attention model has been widely used in deep learning tasks such as natural language processing[37], speech recognition [38], and computer vision [39] , and has become the basic model for deep learning. The attention model was originally implemented on a model of encoders and decoders and was applied to machine translation [40]. The original transformer is a model of an encoder-decoder implemented using MSA, whose parallelized design improves the training speed. Researchers have introduced attention mechanisms to spatial-temporal data prediction, with respectable performance at many tasks. STGNN [22] uses transformer to capture global

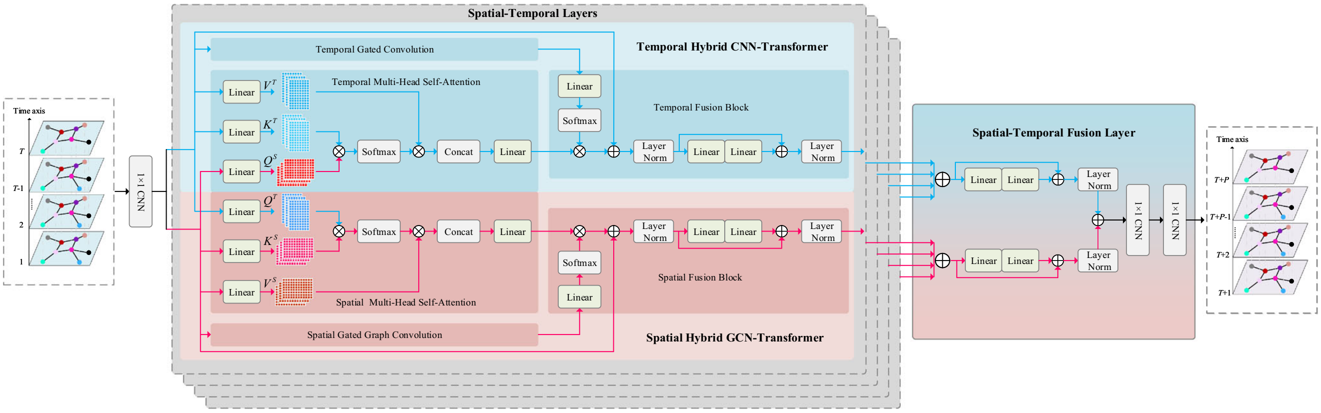

Spatial-Temporal Layers
Temporal Hybrid CNN-Transformer
Temporal Gated Convolution
Temporal Multi-Head Self-Attention
Linear
V^T
K^T
Q^S
Q^T
K^S
V^S
Softmax
Concat
Temporal Fusion Block
Layer Norm
Spatial Fusion Block
Spatial Multi-Head Self-Attention
Spatial Gated Graph Convolution
Spatial Hybrid GCN-Transformer
Time axis
T
T-1
2
1
1×1 CNN
Spatial-Temporal Fusion Layer
T+P
T+P-1
T+2
T+1

Fig. 3 STGHTN consists of four stacked spatial-temporal layers and a spatial-temporal fusion layer

temporal dependence and local temporal dependence using GRU and spatial dependence using a GNN layer with a positional attention mechanism. The transformer-enhanced DetectorNet [41] has a multiview temporal attention module to extract temporal dependence at long and short distances, and a dynamic attention module to extract dynamic spatial dependence.

3 Preliminaries

3.1 Problem definition

In this paper, the goal of traffic forecasting is to predict traffic flow data for future periods based on historical traffic flow data. We define the topology of a real traffic road network as a weighted directed graph $G = (V, E, A)$, where $V = \{v_1, v_2, v_3, ..., v_N\}$ is the set of nodes representing roads, N is the number of road sensor nodes, E denotes the set of edges, and $A \in \mathbb{R}^{N\times N}$ denotes a weighted adjacency matrix, representing the relationship between roads. We express the traffic flow data at time t on the road v_n as X_t^n. The purpose of traffic prediction is to learn a mapping function $\mathcal{F}(\cdot)$ that predicts the traffic data $\mathcal{X} = \{X_1, \ldots, X_T\} \in \mathbb{R}^{T\times N}$ at the next P time steps based on the traffic data $\hat{\mathcal{X}} = \{X_{T+1}, \ldots, Y_{T+P}\} \in \mathbb{R}^{P\times N}$ at the first T time steps. This is defined as follows:

$$[X_{T+1}, \ldots, X_{T+P}] = \mathcal{F}([X_1, \ldots, X_T; G]) \qquad (1)$$

4 Methodology

4.1 Overview

Figure 3 shows the STGHTN framework, which includes an input layer, four stacked Spatial-Temporal Layers (STLs), and a Spatial-Temporal Fusion Layer. The input layer converts the spatial-temporal features to a high-dimensional space by convolution to represent complex spatial-temporal dependence. An STL includes a THCT and a SHGT. THCT extracts short-term and long-term temporal dependence. SHGT is used to extract dynamic spatial dependence on local and global scales. STFL aggregates spatial-temporal features of different granularities to explore spatial dependencies between different time steps and performs downstream prediction tasks using 1×1 convolution.

4.2 Temporal hybrid CNN-transformer

Although RNN-based models are widely used in time-series analysis, RNN still suffers from time-consuming iterations, unstable gradients, and slow response to dynamic changes. THCT consists of a TGC, a Temporal Multi-Head Self-Attention (TMSA), and a temporal fusion block. TGC adopts a 1D dilated causal convolution [42] and gating mechanism [43] to extract short-term temporal dependence. TMSA adopts a self-attention [23] mechanism to extract long-term temporal dependence. The temporal fusion block is used to integrate short-term and long-term temporal dependence.

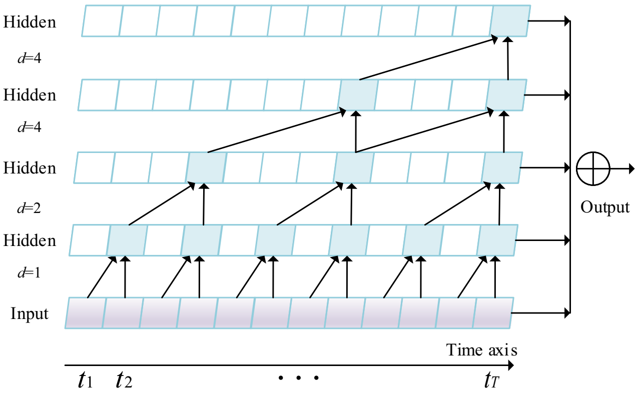


Fig. 4 Example of dilated causal convolution

4.2.1 Temporal gated convolution

Ordinary CNN-based models cannot effectively model sequence-to-sequence problems. We use dilated causal convolution to capture the temporal trend between time steps of road nodes. As shown in Fig. 4, dilated causal convolution introduces dilation in the standard convolution operation and increases the perceptual field through a deeply stacked network. Given the input time-series $X = \{x_1, x_2, ..., x_T\} \in \mathbb{R}^T$ and filters $\mathcal{F} = \{f_1, f_2, ..., f_U\} \in \mathbb{R}^U$, the dilated causal convolution at time step t is calculated as:

$$X * \mathcal{F}(t) = \sum_{u=1}^{U} \mathcal{F}(u) x_{t-d(u-1)} \tag{2}$$

where d is the dilation rate, which indicates the distance between convolution kernels.

The gating mechanism is used in RNNs to control the flow of information. Short-term temporal features are extracted in parallel using the gated convolution on the time-axis, which is more efficient than LSTM for long-term time-series data. Given the input $X^T \in \mathbb{R}^{N \times T \times C}$, where C is the number of channels. It takes the form:

$$TGC(X^T) = Sigmoid(\Phi_1 * X^T + a) \odot Tanh(\Phi_2 * X^T + b) \tag{3}$$

where Φ_1 and Φ_2 are independent 1D dilated causal convolution operations in the time dimension, a and b are model parameters, $\odot$ is the element-wise product, $Sigmoid$ and $Tanh$ are the activation functions.

4.2.2 Temporal multi-head self-attention

We use MSA to model complex temporal dependence. Given the input temporal feature $X^T \in \mathbb{R}^{T \times N \times C}$ and the spatial feature $X^S \in \mathbb{R}^{T \times N \times C}$, we map spatial-temporal features to a high-dimensional space to learn complex temporal dependence. Subspaces $Q_i^T \in \mathbb{R}^{T \times d_\mathcal{U}}$, $K_i^T \in \mathbb{R}^{T \times d_\mathcal{U}}$, and $V_i^T \in \mathbb{R}^{T \times d_\mathcal{V}}$ are generated by linear transformation:

$$Q_i^T = X^S W_{q_i}^T, K_i^T = X^T W_{k_i}^T, V_i^T = X^T W_{v_i}^T \tag{4}$$

where $W_{q_i}^T$, $W_{k_i}^T$, and $W_{v_i}^T$ are the parameters of the learnable. MSA weights are calculated by the scaled dot product, whcih can be expressed as:

$$\begin{aligned} &TMSA(X^S, X^T, X^T) = Concat(head_1^T, head_1^T, \cdots, head_n^T) W^T \\ &where \quad head_i^T = Softmax\left(\frac{Q_i^T (K_i^T)^T}{\sqrt{d_\mathcal{U}}}\right) V_i^T \end{aligned} \tag{5}$$

where n is the number of heads of MSA, W^T is learnable parameters, and $Softmax$ is an activation function.

4.2.3 Temporal fusion block

To consider both short-term and long-term dependence, we fuse the outputs $TMSA(X^T) \in \mathbb{R}^{N \times T \times C}$ of TMSA and $TGC(X^T) \in \mathbb{R}^{N \times T \times C}$ of TGC, with the following form:

$$T_{out} = TMSA(X^S, X^T, X^T) \odot Sigmoid(TGC(X^T) W_1^T) \tag{6}$$

where W_1^T consists of learnable parameters, and $T_{out} \in \mathbb{R}^{T \times N \times C}$ is a post-fusion temporal feature. We increase the expressiveness of the model using residual connections and linear transformations, and further adjust the dependence between time steps, The specific form is as follows:

$$\begin{aligned} &\vec{T}_{out} = LN(T_{out} + X^T) \\ &Ttra = LN(Relu(\vec{T}_{out} W_2^T) W_3^T + \vec{T}_{out}) \end{aligned} \tag{7}$$

where W_2^T and W_3^T are learnable parameters, LN is layer normalization, and $Relu$ is the activation functions.

4.3 Spatial hybrid GCN-transformer

Most of the existing GNN-based methods for capturing spatial dependence suffer from a lack of extraction of global spatial features. And the predefined graph structure information cannot be adapted to dynamic spatial-temporal data. The SHGT consists of an SGGC, a Spatial Multi-Head Self-Attention (SMHA), and a spatial fusion block. SGGC uses multi-graph graph convolution operations to extract local spatial information. SMHA leverages a self-attention mechanism to excavate connections between distant roads to adjust for global spatial dependence. The spatial fusion block is designed to integrate the dependencies between connected roads and between roads that are far apart.

4.3.1 Multi-graph construction

Graph convolution is an aggregation of information on a graph, whose structure can greatly affect the performance of the model. We usually think that interconnected roads in road networks can have similar properties. However, in the real world, two shopping areas that are far apart may also have similar properties. At the same time, there are hidden potential dependence. This cannot be done by the predefined graph structure, and due to the change of external environment, the predefined graph structure often cannot fully reflect the real road relationship. Therefore, we propose a multi-graph fusion scheme that combines a road connection graph, a similarity graph, and an adaptive dynamic graph, considering both static and dynamic links between roads. The multi-graph formalism is $G = \{G_1, G_2, G_3\}$, taking the following form:

- Road connection graph $G_1 = (V, E^s, A^s)$ is constructed based on road connection relationships. Its spatial adjacency matrix $A^s \in \mathbb{R}^{N\times N}$ is deduced from G. It takes the form:

$$A^s_{ij} = \begin{cases} 1, & ((v_i, v_j) \in V) \;\; \& \;\; (v_i, v_j \in E^s) \\ 0, & otherwise \end{cases} \tag{8}$$

- Similarity graph $G_2 = (V, E^t, A^t)$ is constructed based on the Dynamic Time Warping (DTW) algorithm [44], which is widely used in speech recognition. We use Algorithm 1 to calculate the similarity $DTW(v_i, v_j)$ between two roads based on their traffic flow sequences. DTW is more able than Euclidean distance to reflect the similarity of traffic sequences. For example, the change of traffic flow on an upstream road often has a certain lag compared to that on the corresponding downstream road. Euclidean distance cannot effectively measure the similarity between two time-series that have similar shapes but are not synchronized in time. DTW can effectively solve this problem. The temporal similarity matrix $A^t \in \mathbb{R}^{N\times N}$ is generated using the following form:

$$A^t_{ij} = \begin{cases} 1, & exp(-DTW(v_i, v_j)) > \rho \\ 0, & otherwise \end{cases} \tag{9}$$

where ρ is a threshold.

- The adaptive dynamic graph $G_3 = (V, E^{adp}, A^{adp})$ is generated based on A^s and A^t. We use learnable parameters $\mathcal{E}^s$ and $\mathcal{E}^t$, which are respectively the source and target node, to capture the potential and dynamic dependence between two roads. $\mathcal{E}^s$ and $\mathcal{E}^t$ generate dynamic spatial dependence by matrix multiplication. The adaptive adjacency matrix is as follow:

$$A^{adp} = Softmax(ReLU(\mathcal{E}_s \mathcal{E}_t^T)) \odot (A^s + A^t) \tag{10}$$

where $ReLU$ and $Softmax$ are activation functions, used respectively to reduce and normalize the effect of slight perturbations.

4.3.2 Spatial gated graph convolution

GCN can efficiently exploit the feature information of nodes. We use this to capture the dependence of the spatial relationships between roads [45], where the representation of nodes is calculated by aggregating information from direct neighbors. Given the input $x \in \mathbb{R}^{N\times C}$, a single-layer GCN expressed as:

$$\begin{aligned} GCN(A, x) &= ReLU(D^{-1/2}\hat{A}D^{1/2}xW_1) \\ \hat{A} &= A + I_N \\ D_{ii} &= \sum_j \hat{A}_{ij} \end{aligned} \tag{11}$$

where D is the degree matrix of $\hat{A}$, and W_1 is a learnable parameter.

```
Input: $X = \{x_1, x_2, ...x_N\} \in \mathbb{R}^N$, $Y = \{y_1, y_2, ...y_N\} \in \mathbb{R}^N$, cumulative distance matrix $\mathbf{W}_{N\times N} = 0$;
Output: Temporal similarity $DTW(X, Y)$;
for $i = 1; i <= N; i++$ do
    for $j = 1; j <= N; j++$ do
        $dis_{i,j} = (x_i - y_i)^2$;
        if $i = 1 \& j = 1$ then $\mathbf{W}_{i,j} = dis_{i,j}$;
        else if $i = 1$ then $\mathbf{W}_{i,j} = dis_{i,j} + \mathbf{W}_{i,j-1}$;
        else if $j = 1$ then $\mathbf{W}_{i,j} = dis_{i,j} + \mathbf{W}_{i-1,j}$;
        else $\mathbf{W}_{i,j} = dis_{i,j} + min(\mathbf{W}_{i-1,j-1}, \mathbf{W}_{i-1,j}, \mathbf{W}_{i,j-1})$;
        end if
    end for
end for
$DTW(X, Y) = \mathbf{W}_{N,N}$;
```

Algorithm 1 Temporal similarity calculation.

As shown in Fig. 5, to increase the expressiveness of GCN representation, we add skip connections and linear transformations to the network. Given input features X^S, we can obtain the static graph convolution to generate static spatial features $STA(X^S)$, and dynamic graph convolution generate dynamic spatial features $DYN(X^S)$. The specific form is as follows:

$$\begin{aligned} STA(X^S) &= Concat(GCN(A^s, X^S), GCN(A^t, X^S), X^S)W_{sta} \\ DYN(X^S) &= Concat(GCN(A^{adp}, X^S), X^S)W_{dyn} \end{aligned} \tag{12}$$

where W_{sta} and W_{dyn} are learnable parameters to mitigate GCN over-smoothing. We simultaneously consider static and dynamic graph convolution through a gated fusion mechanism, which obtains a tensor that lies between 0 and 1 by means of a sigmoid activation function. The specific form is as follows:

$$\begin{aligned} z &= Sigmoid(STA(X^S)W_{g1} + DYN(X^S)W_{g2} + b) \\ GFS(X^S) &= z \odot STA(X^S) + (1 - z) \odot DYN(X^S) \end{aligned} \tag{13}$$

where W_{g1}, W_{g2}, and b are learnable parameters, $GFS(X^S) \in \mathbb{R}^{N\times T\times C}$ is a post-fusion feature, and $z \in \mathbb{R}^{N\times T\times C}$ is the gating value.

4.3.3 Spatial multi-head self-attention

In the spatial dimension, we use MSA to capture spatial dependence on a global scale. Given the spatial feature $X^S \in \mathbb{R}^{N\times T\times C}$ and the temporal feature $X^T \in \mathbb{R}^{N\times T\times C}$, subspaces $Q_i^T \in \mathbb{R}^{N\times d_{\mathcal{K}}}$, $K_i^T \in \mathbb{R}^{N\times d_{\mathcal{K}}}$, and $V_i^T \in \mathbb{R}^{N\times d_{\mathcal{M}}}$ are generated by linear transformation. They take the form:

$$Q_i^S = X^T W^S_{q_i}, K_i^S = X^S W^S_{k_i}, V_i^S = X^S W^S_{v_i} \tag{14}$$

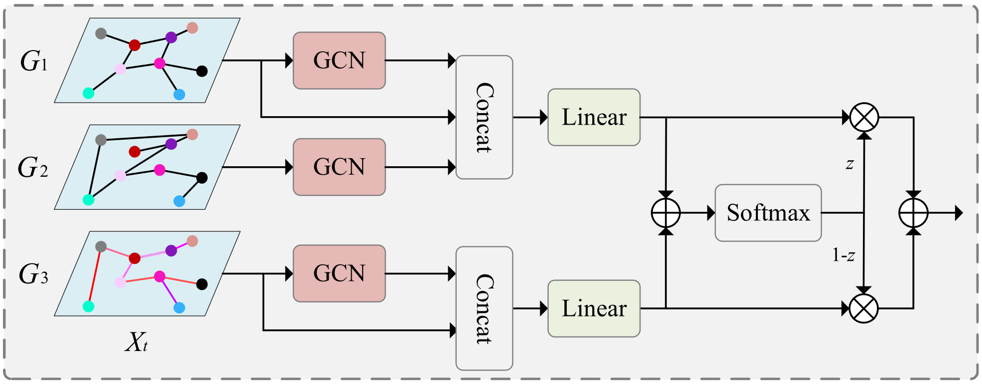


Fig. 5 Module of spatial gated graph convolution

where $W_{q_i}^S$, $W_{k_i}^S$, and $W_{v_i}^S$ are learnable parameters. MSA weights are calculated by the scaled dot product, whcih can be expressed as:

$$\begin{aligned} &SMSA(X^T, X^S, X^S) = Concat(head_1^S, head_1^S, \cdots, head_n^S)W^S \\ &where \quad head_i^S = Softmax\left(\frac{Q_i^S(K_i^S)^T}{\sqrt{d_K}}\right)V_i^S \end{aligned} \tag{15}$$

where W^S is a learnable parameter.

4.3.4 Spatial fusion block

Since GCN aggregates information about neighbors around a node, it is local in nature, while MSA captures global features and can effectively capture spatial dependence between two roads that are far from each other. We adopt a similar approach to THCT to fuse local and global spatial dependence. Specifically, it takes the form:

$$S_{out} = SMSA(X^T, X^S, X^S) \odot Sigmoid(GFS(X^S)W_1^S) \tag{16}$$

where W_1^S consists of learnable parameters, and $S_{out} \in \mathbb{R}^{N \times T \times C}$ is the spatial feature after fusion. Further, we adjust the spatial dependence between roads using residual connections and linear transformations. The following form is used:

$$\begin{aligned} &\vec{S}_{out} = LN(S_{out} + X^S) \\ &Stra = LN(Relu(\vec{S}_{out}W_2^S)W_3^S + \vec{S}_{out}) \end{aligned} \tag{17}$$

where W_2^S and W_3^S are learnable parameters.

4.4 Spatial-temporal fusion layer

After extracting high-level features from the STLs, we aggregate the temporal feature $T_{tra}^{(k)} \in \mathbb{R}^{N \times T \times C}$ and spatial features $S_{tra}^{(k)} \in \mathbb{R}^{N \times T \times C}$ from each STL. To preserve the original features of the data as much as possible, we sum up the features extracted from each layer. The dependence between sequences is further adjusted using linear transformations and residual concatenation to increase the expressiveness of the model. The temporal and spatial features are summed to obtain the fused spatial-temporal feature $FUS(T_{tra}^{(k)}, S_{tra}^{(k)}) \in \mathbb{R}^{N \times T \times C}$. The specific format is as follows:

$$\begin{aligned} &X_{out}^T = LN(W_a^T(W_b^T\sum_{k=0}^{K} T_{tra}^{(k)}) + \sum_{k=0}^{K} T_{tra}^{(k)}) \\ &X_{out}^S = LN(W_a^S(W_b^S\sum_{k=0}^{K} S_{tra}^{(k)}) + \sum_{k=0}^{K} S_{tra}^{(k)}) \\ &FUS(T_{tra}^{(k)}, S_{tra}^{(k)}) = X_{out}^T + X_{out}^S \end{aligned} \tag{18}$$

where W_a^T, W_b^T, W_a^S, and W_b^S are learnable parameters, and K is the number of layers in the STLs.

Further, we use a two-layer 1 × 1 convolution operations to complete the multi-step prediction, which is more efficient than single-step prediction. The specific format is as follows:

$$\vec{Y} = \Theta_2 \star (\Theta_1 \star FUS(T_{tra}^{(k)}, S_{tra}^{(k)})) \tag{19}$$

where Θ_1 and Θ_2 are independent 1×1 convolution operations, and $Y \in \mathbb{R}^{N \times T}$ is the multi-step predicted value.

Since the traffic data collection process is error-prone, we use a Huber loss function, which is robust and insensitive to outliers. It is expressed as:

$$\mathcal{L}(Y, \vec{Y}) = \begin{cases} \frac{1}{2}(Y - \vec{Y})^2, |Y - \vec{Y}| \leq \varpi \\ \varpi|Y - \vec{Y}| - \frac{1}{2}\varpi^2, |Y - \vec{Y}| > \varpi \end{cases} \tag{20}$$

where Y is real traffic flow, and $\vec{Y}$ is predicted traffic flow, and ϖ is a threshold. We summarize the training process of STGHTN in Algorithm 2.

5 Experiment

5.1 Datasets

We used four large-scale datasets from STSGCN [46]: PEMS03, PEMS04, PEMS07, and PEMS08, which come from four regions of California. Table 1 shows the details of the four datasets. Sensor data was aggregated at 5-minute

Table 1 Details of four datasets

Dataset	Nodes	Time steps	Time range	Missing ratio
PEMS03	358	26208	9/1/2018-11/30/2018	0.672%
PEMS04	307	16992	1/1/2018-2/28/2018	3.182%
PEMS07	883	28224	5/1/2017-8/31/2017	0.452%
PEMS08	170	17856	7/1/2016-8/31/2016	0.696%

```
Input: Road network graph $G = (G_1, G_2, G_3)$, traffic flow data $\mathcal{X}$, input data times steps $T$, output data times steps $P$, number of training epochs $epochs$, number of layers of STLs $K$;
Output: learned STGHTN;
Randomly initialize learnable parameters of STGHTN;
$D_{train} \leftarrow \emptyset$
Sliding window to construct the training set from the traffic flow data $D_{train} \leftarrow (\mathcal{X}_{train}, \mathcal{Y}_{train})$;
for $epoch = 0; epoch < epochs; epoch++$ do
    Randomly select a batch of input sample $X \in \mathbb{R}^{N \times T}$ and $Y \in \mathbb{R}^{N \times P}$ from $D_{train}$;
    Gets input feature $X^T$ and $X^S$ from $X$ using $1 \times 1$ CNN;
    for $k = 0; k < K; k++$ do
        Get temporal features $T_{tra}^{(k)}$ from $X^T$ and $X^S$ using THCT;
        Get spatial features $S_{tra}^{(k)}$ from $X^T$ and $X^S$ using SHGT;
    end for
    Get fused spatial-temporal features $\overrightarrow{Y}$ from $\sum_{k=0}^{K} T_{tra}^{(k)}$ and $\sum_{k=0}^{K} S_{tra}^{(k)}$ using STFL;
    Compute loss $\mathcal{L}$ from $Y$ and $\overrightarrow{Y}$ using eq. (20);
    Update trainable parameters with gradient descent;
end for
```

Algorithm 2 Training process of STGHTN.

intervals. We used the first 12 time steps to predict the next 12 time steps. The input data was normalized using the Z-Score as follows:

$$\hat{X} = \frac{X - mean(X_{train})}{std(X_{train})} \tag{21}$$

where $mean(X_{train})$ and $std(X_{train})$ are the mean and standard deviation, respectively, of the training data.

5.2 Baseline methods

We compare the STGHTN model with seven state-of-the-art baselines:

- **SVR** [11]: Linear support vector regression uses SVM to make regression predictions on traffic flow sequences, ignoring spatial dependence.
- **LSTM** [17]: Long short Term Mempry Network to model time series.
- **DCRNN** [13]: Diffusion Convolution Recurrent Neural Network, which models spatial dependence using bidirectional random walks and temporal dependence using encoders and decoders.
- **STGCN** [19]: Spatial-Temporal Graph Convolutional Networks, which use graph convolution and gated causal convolution and do not rely on LSTM or GRU.
- **ASTGCN** [20]: Attention Based Spatial Temporal Graph Convolutional Networks, where a spatial-temporal attention mechanism captures spatial relationships through graph convolution and temporal relationships through ordinary convolution.
- **Graph WaveNet** [31]: Graph WaveNet for Deep Spatial-Temporal Graph Modeling uses adaptive adjacency matrices and learns by node embedding. Temporal dependence captures using 1D dilated convolution.
- **STSGCN** [46]: Spatial-Temporal Synchronous Graph Convolutional Networks capture spatial-temporal relationships and use the same component for both time and space.
- **STFGNN** [21]: Spatial-Temporal Fusion Graph Neural Networks use a spatial-temporal fusion module that a new graph structure constructs based on a data-driven approach.

5.3 Experiment settings

The datasets were divided into training, testing, and validation sets at a 6:2:2 ratio. Experiments were conducted in a Windows environment. The processor was an Intel Xeon E5-2680 v4 CPU @2.40 GHz with 128 GB RAM. The GPU was a single NVIDIA RTX2080Ti. We used the grid search strategy to determine the optimal hyperparameters. We trained our model using an Adam optimizer. The batch size was 16 and the learning rate was 0.001. There were four spatial-temporal layers and four heads of MSA. The number of channels was 32. The evaluation metrics were Mean Absolute Error (MAE), Mean Absolute Percentage

Error (MAPE), and Root Mean Squared Error (RMSE). The specific format is as follows:

$$MAE = \frac{1}{N \times T} \sum_{t=1}^{T} \sum_{n=1}^{N} |Y_t^n - \vec{Y}_t^n|$$

$$RMSE = \sqrt{\frac{1}{N \times T} \sum_{t=1}^{T} \sum_{n=1}^{N} \left(Y_t^n - \vec{Y}_t^n \right)^2}$$

$$MAPE = \frac{1}{N \times T} \sum_{t=1}^{T} \sum_{n=1}^{N} \frac{|Y_t^n - \vec{Y}_t^n|}{Y_t^n} \tag{22}$$

5.4 Experimental results

Table 2 shows the prediction performance of the different models on the four datasets at 15 minutes, 30 minutes, 60 minutes and on average. STGHTN shows the best performance in both short-term and long-term forecasting.

The traditional machine learning method SVR tends to have less-than-ideal prediction performance due to a lack of nonlinear representation capability. Deep learning methods have strong nonlinear representation abilities. Among these, LSTM had poor prediction performance because it

Table 2 Performance comparison of different methods on PEMS03, PEMS04, PEMS07, and PEMS08 datasets

Datestes	Methods	15 min			30 min			60 min			Average		
		MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
PEMS03	SVR	17.14	29.63	17.07	21.16	35.41	21.39	29.77	47.12	30.17	21.88	36.84	22.11
	LSTM	15.73	26.39	16.20	20.04	31.72	18.71	27.37	42.37	28.50	20.64	33.02	20.40
	DCRNN	15.57	25.88	14.84	18.35	30.47	17.20	24.11	38.69	22.67	18.77	31.23	17.67
	STGCN	17.43	28.43	16.99	19.25	31.33	18.40	23.48	37.51	21.95	19.62	31.95	18.77
	ASTGCN	15.71	26.48	15.31	17.75	29.84	16.85	22.12	36.30	21.70	18.02	30.39	17.42
	Graph WaveNet	16.16	25.90	16.36	19.21	30.56	17.46	24.73	38.42	24.01	19.64	31.34	19.43
	STSGCN	15.85	25.65	15.91	17.56	28.58	16.61	20.72	33.75	19.44	17.67	28.90	16.89
	STFGNN	15.32	25.86	15.21	16.80	28.44	16.23	19.37	32.57	18.37	16.84	28.59	16.33
	STGHTN	**14.18**	**24.94**	**13.41**	**15.41**	**26.88**	**14.17**	**17.13**	**29.26**	**16.07**	**15.46**	**26.79**	**14.28**
PEMS04	SVR	21.93	33.94	14.55	26.49	40.25	17.64	36.44	53.64	24.67	27.39	42.06	18.33
	LSTM	21.85	33.98	14.44	25.91	39.64	17.32	34.80	51.52	24.49	26.67	41.13	18.07
	DCRNN	20.37	31.94	13.81	23.62	36.51	15.96	30.49	45.99	20.72	24.17	37.64	16.38
	STGCN	21.35	33.99	14.10	23.60	37.08	15.55	28.80	44.00	19.08	24.06	37.84	15.92
	ASTGCN	19.20	30.56	12.72	20.38	32.35	13.52	23.57	36.83	15.50	20.63	32.75	13.63
	Graph WaveNet	21.76	33.94	14.69	24.83	38.31	17.73	30.55	46.31	23.33	25.06	38.62	18.00
	STSGCN	19.64	31.47	12.79	21.16	33.89	13.70	24.21	38.36	15.63	21.34	34.22	13.84
	STFGNN	18.78	30.14	12.48	19.82	31.95	13.12	21.55	34.60	14.26	19.82	31.94	13.12
	STGHTN	**18.40**	**29.53**	**12.05**	**19.38**	**31.33**	**12.57**	**20.98**	**33.73**	**13.90**	**19.35**	**30.93**	**12.66**
PEMS07	SVR	24.33	37.32	10.32	29.97	45.41	12.84	41.33	60.38	18.31	30.75	47.06	13.30
	LSTM	23.86	36.56	10.28	28.68	43.48	12.22	39.38	57.57	17.63	29.53	45.07	12.82
	DCRNN	22.86	35.08	9.99	26.63	40.54	11.69	34.82	51.32	16.23	27.19	41.57	12.17
	STGCN	25.12	39.61	10.81	27.73	43.48	11.92	33.80	51.74	14.75	28.20	44.25	12.22
	ASTGCN	22.42	34.69	9.53	25.32	38.95	10.80	31.37	47.21	13.66	25.61	39.56	10.99
	Graph WaveNet	22.68	35.12	9.64	26.70	40.96	11.31	34.62	51.86	15.68	27.24	41.52	11.92
	STSGCN	21.12	34.18	8.84	23.50	38.41	9.86	28.10	45.88	11.90	23.72	38.95	9.97
	STFGNN	20.30	32.89	8.49	21.95	35.94	9.14	24.77	40.55	10.33	21.98	36.03	9.17
	STGHTN	**19.63**	**31.58**	**8.27**	**21.06**	**34.37**	**9.01**	**23.55**	**38.28**	**10.27**	**21.00**	**33.99**	**8.93**
PEMS08	SVR	17.56	26.94	10.78	21.62	33.21	13.28	30.55	45.21	18.80	22.40	34.69	13.80
	LSTM	17.61	26.98	10.97	21.10	32.52	12.73	28.98	43.27	18.05	21.76	33.73	13.44
	DCRNN	15.48	23.53	10.33	17.65	26.71	11.68	21.76	32.41	14.19	17.87	27.18	11.81
	STGCN	17.07	26.07	11.58	18.64	28.60	12.31	22.49	34.05	14.54	19.01	29.16	12.63
	ASTGCN	16.19	24.96	9.87	18.12	27.84	11.00	22.12	33.33	13.48	18.31	28.20	11.16
	Graph WaveNet	16.62	25.83	10.37	18.77	29.48	11.81	23.01	35.67	15.37	18.88	29.78	12.36
	STSGCN	15.72	24.25	10.38	17.13	26.75	11.22	19.56	30.58	12.55	17.19	26.89	11.20
	STFGNN	15.20	23.60	9.87	16.48	25.97	10.56	18.83	29.66	11.90	16.56	26.10	10.63
	STGHTN	**14.54**	**22.62**	**9.33**	**15.42**	**24.42**	**9.64**	**17.11**	**27.20**	**10.76**	**15.30**	**24.27**	**9.88**

Smallest errors are bolded and second-best are underlined

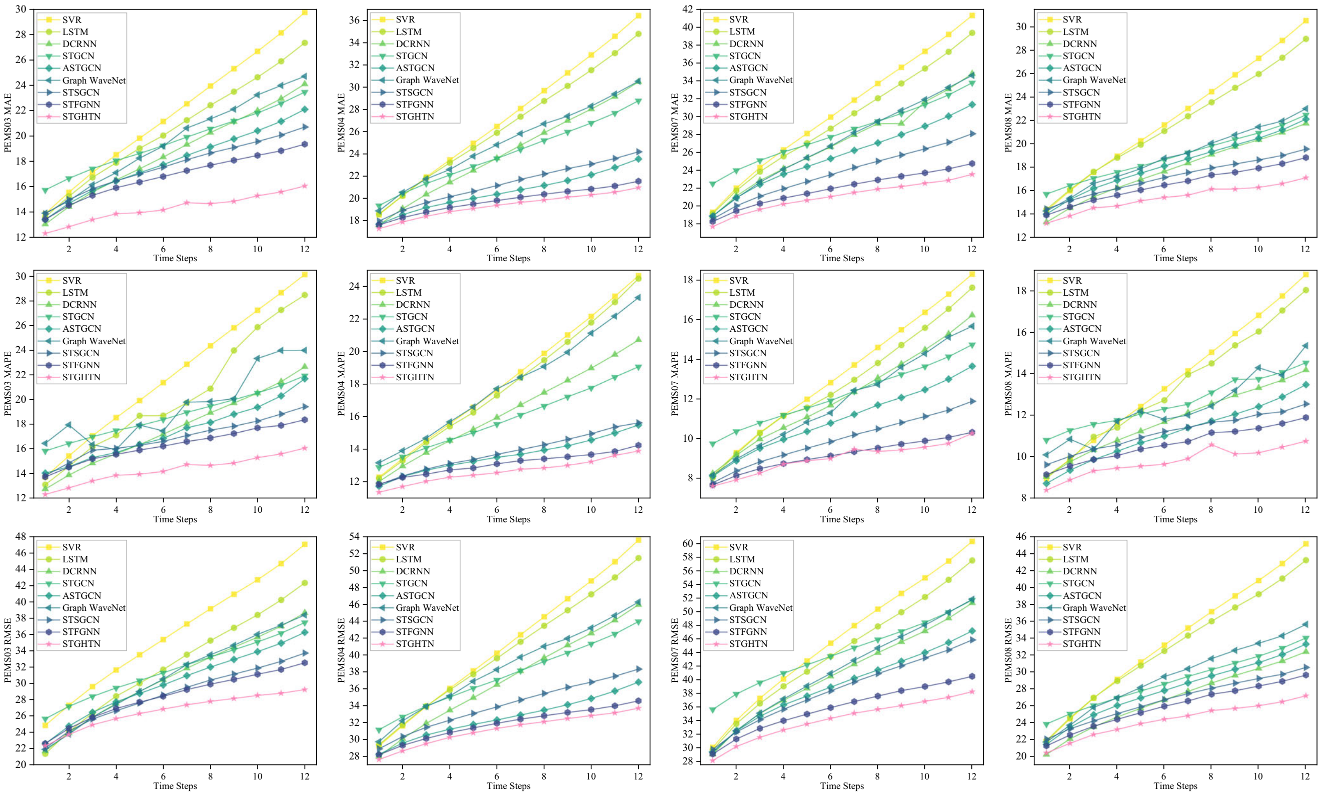


Fig. 6 The performance of the different models on the four datasets varies with the prediction time step

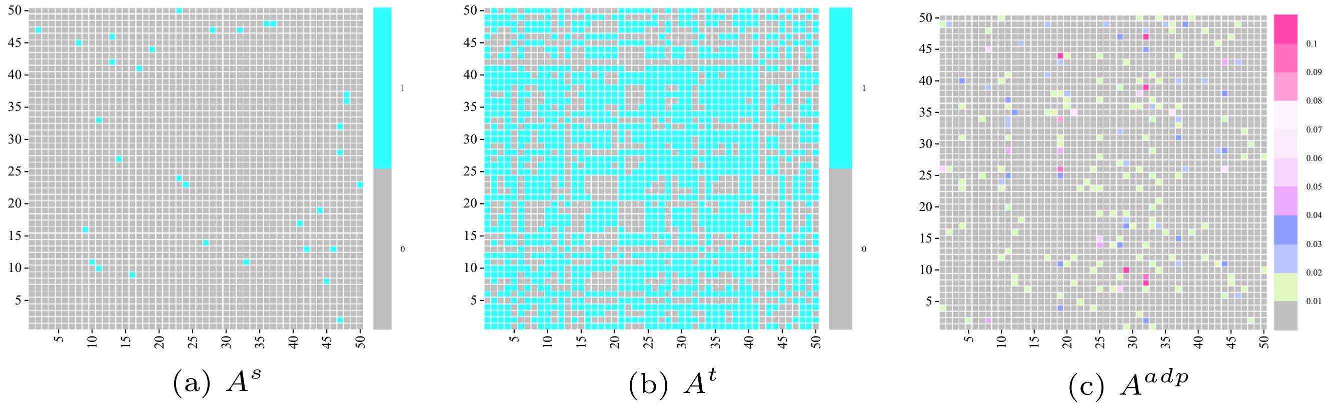
only considers temporal correlations and ignores spatial correlations. DCRNN, STGCN, ASTGCN, Graph WaveNet, STSGCN, and STFGNN considered both temporal and spatial correlations to improve prediction performance. The more recent model STFGNN incorporates a spatial graph, temporal graph, and temporal connected graph in a baseline approach that showed greater performance. We propose a model that fuses A^s, A^t, and A^{adp}, considering both static and dynamic graph feature fusion. The best performance is demonstrated on all datasets.

In the average prediction result, the improvement ratios of the RMSE metrics compared with the baseline optimal model STFGNN on the four datasets are 6.30%, 3.16%, 5.66%, and 7.01%, respectively. The relatively large missing rate in the PEMS04 dataset had a certain impact on the performance of the model. Our proposed model generally has stronger prediction performance because it more comprehensively considers spatial-temporal correlations. Figure 6 shows the predicted results for the 12 time steps in more detail. The results show that the prediction performance of the model decreases as the range of predictions increases. The LSTM model modelling only temporal features has the fastest degradation in prediction performance. Our model shows the best performance overall

Table 3 Combinations of different graphs on PEMS04 and PEMS08

Dataset	Adjacency Matrix	MAE	MAPE(%)	RMSE
PEMS04	$[A^s]$	19.68	12.83	31.34
	$[A^t]$	19.58	12.62	31.57
	$[A^{adp}]$	19.52	12.82	31.21
	$[A^s, A^t]$	19.42	12.91	30.95
	$[A^s, A^t, A^{adp}]$	**19.35**	**12.66**	**30.93**
PEMS08	$[A^s]$	15.59	10.45	24.75
	$[A^t]$	15.48	10.27	24.51
	$[A^{adp}]$	15.42	9.94	24.32
	$[A^s, A^t]$	15.60	10.15	24.75
	$[A^s, A^t, A^{adp}]$	**15.30**	**9.88**	**24.27**

Smallest errors are bolded



(a) A^s (b) A^t (c) A^{adp}

Fig. 7 Heatmap of three kinds of adjacency matrix for 50 nodes on PEMS04

and also the slowest decline in performance as the prediction horizon increases, as we consider both modelling long-term temporal dependence and short-term temporal dependence.

5.5 Ablation study

5.5.1 Evaluation on multi-graph fusion

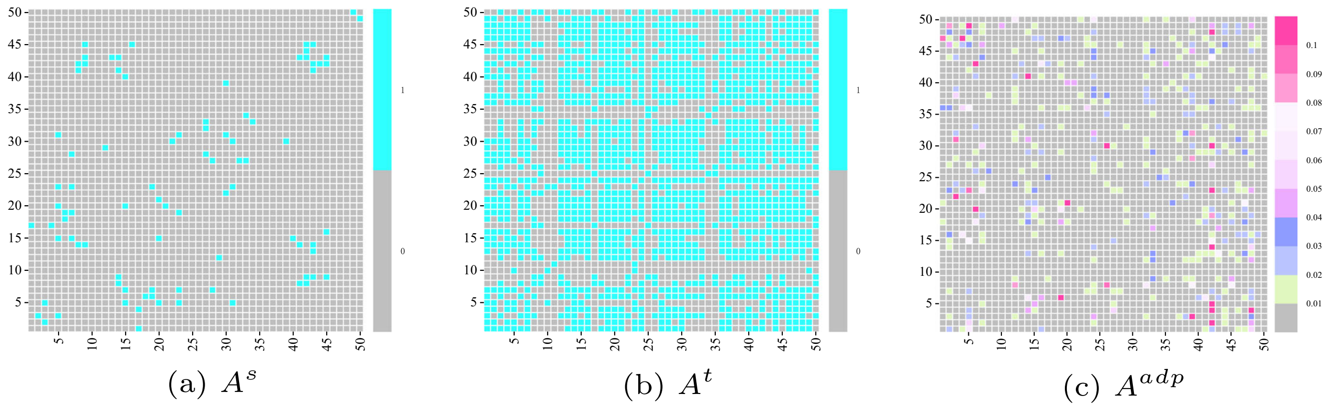
To verify the effectiveness of the multi-graph, we designed four combinations of graphs for comparison experiments on PEMS04 and PEMS08, whose results are shown in Table 3. It can be found that the similarity graph method and the adaptive dynamic graph method perform better than the road connection graph method. In the absence of expert domain knowledge, we can achieve relatively effective predictions using only the adaptive adjacency matrix. When we consider the fusion of three adjacency matrices simultaneously, a combination of the predefined prior knowledge and potential information mined from traffic data, the optimal evaluation score is finally achieved. Figures 7 and 8 visualize the three graphs. It can be found that the adjacency matrix A^s constructed based on the road connection relationship is sparse compared to the graph A^t constructed by the DTW method. A^s is only a local spatial connection relationship, and A^t can dig out the spatial dependence relationship at a long distance. The adaptive adjacency matrix A^{adp} can dynamically adjust the correlations relationship between roads.

5.5.2 Ablation experiments

To verify the validity of each part of our model, we designed four variant models and conducted ablation experiments with our models on the PEMS04 and PEMS08 datasets. The variant models are as follows:

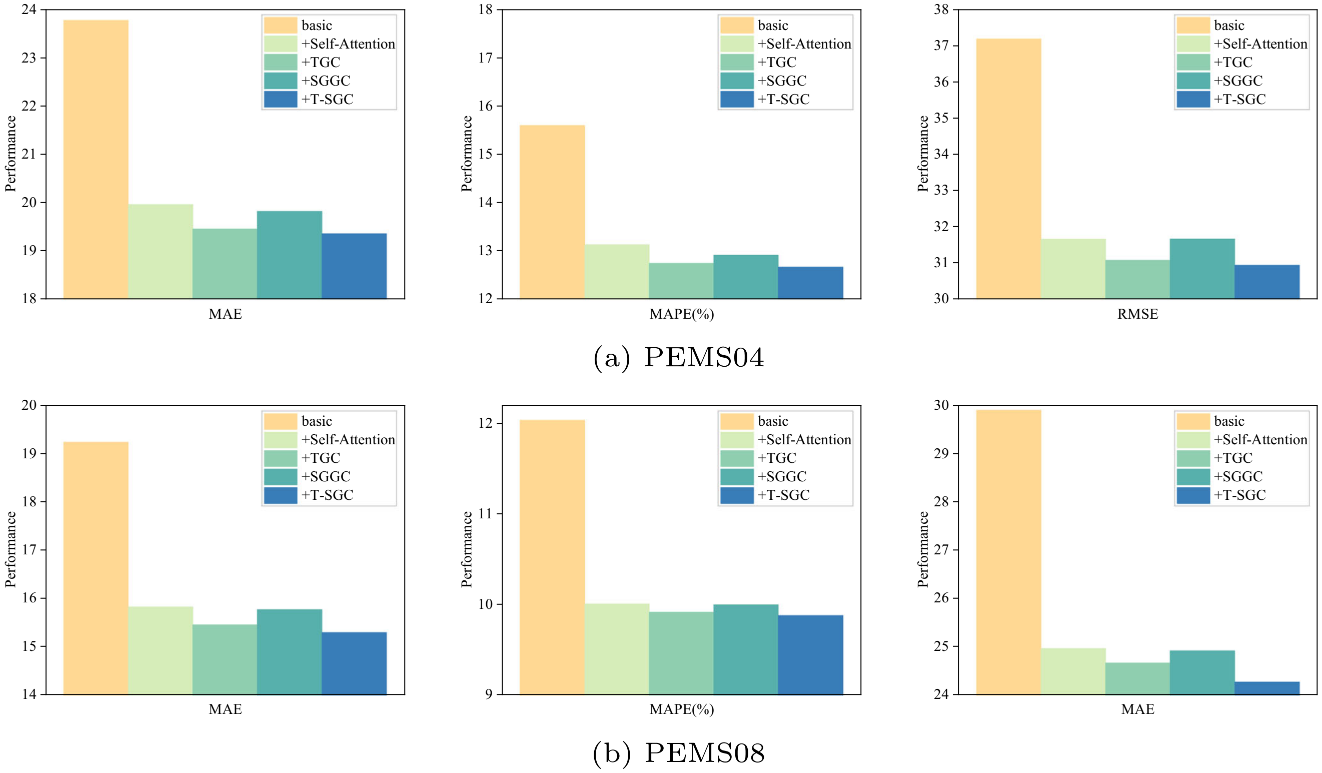
- Basic: This basic framework does not include TMSA, SMSA, TGC and SGGC.
- +Self-Attention: This model adds TMSA and SMSA to the basic model.
- +TGC: TGC is added to the +Self-Attention model.
- +SGGC: SGGC is added to the +Self-Attention model.
- +T-SGC: This model adds both TGC and SGGC to the +Self-Attention model.

As shown in Fig. 9, the performance after introducing the attention mechanism is clearly due to the basic model,



(a) A^s (b) A^t (c) A^{adp}

Fig. 8 Heatmap of three kinds of adjacency matrix for 50 nodes on PEMS08



(a) PEMS04

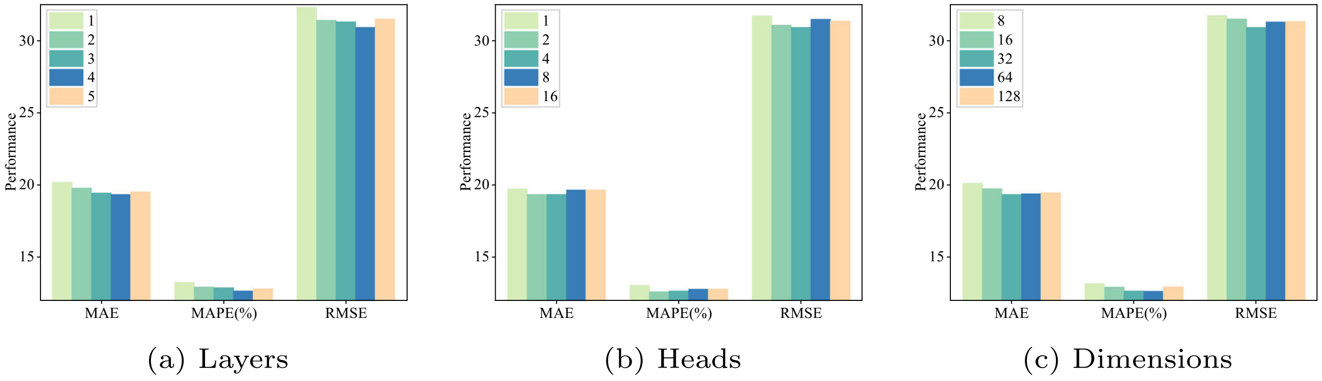
(b) PEMS08

Fig. 9 Ablation study on PEMS04 and PEMS08 datasets

because we use the self-attention mechanism to extract the remote temporal dependence in the temporal dimension and the direct spatial dependence of different roads in the spatial dimension, respectively. TMSA can capture long-term temporal dependence. We add TGC to further capture short-term temporal dependence. SMSA mines dependence from the data to introduce real-world semantic connections. We further use SGGC to model the direct spatial dependence of different roads. It can be found that +T-SGC has a respectable improvement compared to +Self-Attention, while confirming the effectiveness of our model design.

5.6 Effect of hyperparameters

Figures 10 and 11 show the prediction results of our model on PEMS04 and PEMS08 with different hyperparameters. When we adjust one parameter, the others are set optimally by default. Layer denotes the number of STLs. Head indicates the number of heads of MSA. Dimensions indicate the number of channels. It can be found that the appropriate increase of layer, head, and dimensions can improve the performance of the model. However, this will make the model too complex and reduce computational efficiency, while leading to overfitting, which reduces performance.

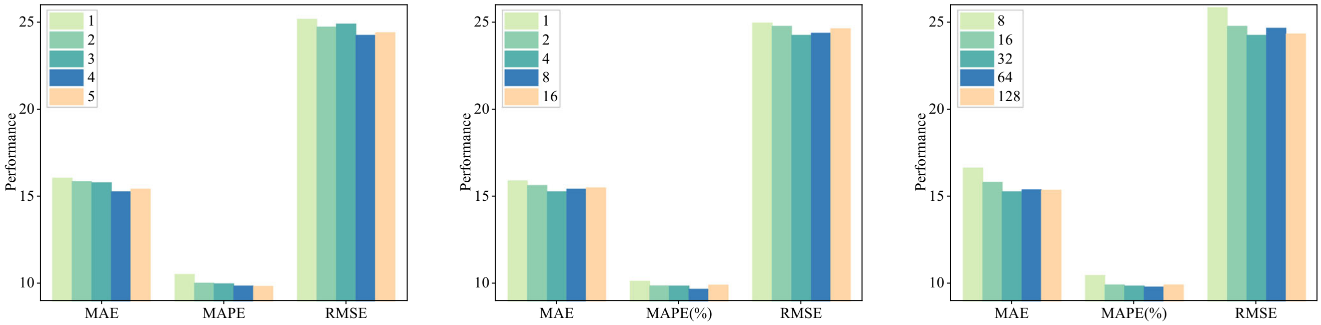


(a) Layers

(b) Heads

(c) Dimensions

Fig. 10 Impact of hyperparameter settings on PEMS04



(a) Layers (b) Heads (c) Dimensions

Fig. 11 Impact of hyperparameter settings on PEMS08

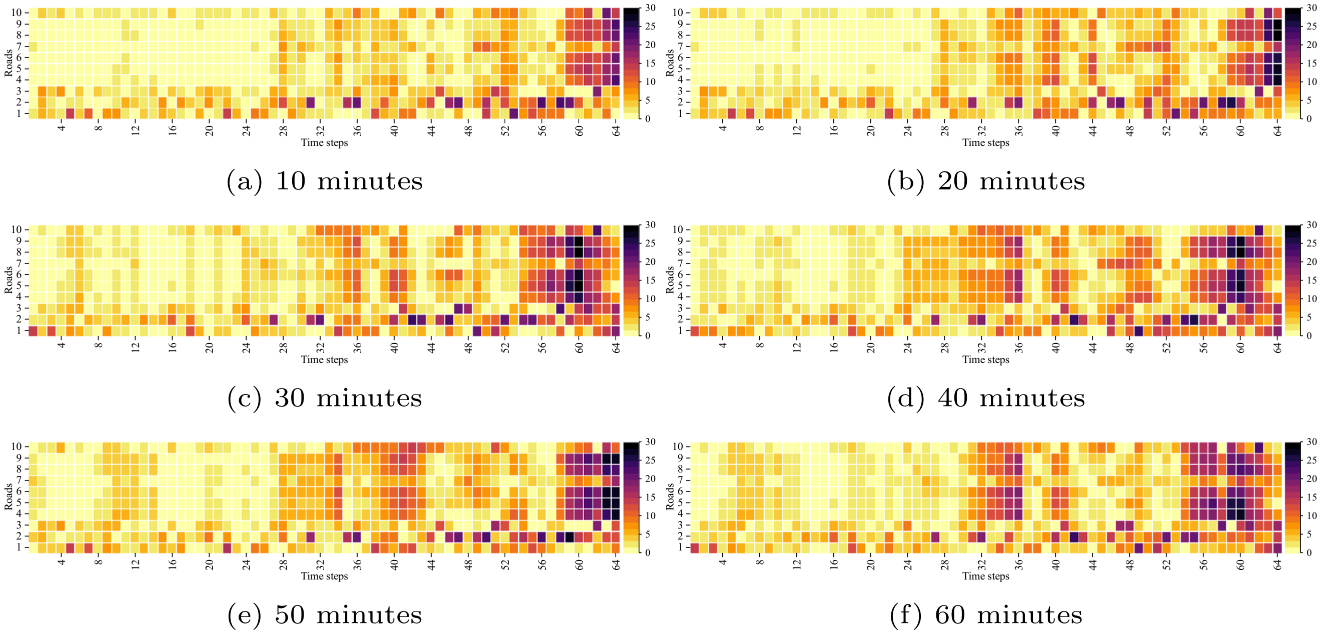
5.7 Visualization

Figures 12 and 13 show the absolute errors of the STGHTN model on the 10 minutes, 20 minutes, 30 minutes, 40 minutes, 50 minutes, and 60 minutes prediction tasks on PEMS04 and PEMS08.

We can find that the STGHTN model shows promising forecasting performance for both short-term and long-term forecasts. It can effectively capture the temporal trend of traffic flow. Because real-world traffic conditions are complex and variable, the prediction effectiveness of the model decreases as the prediction range increases. Figure 14 shows the prediction results of our model and the baseline model STFGNN, STSGCN at 500 time steps on the PEMS04 and PEMS08 datasets. We can find that although both exhibit excellent prediction performance, our model more accurately predicts the beginning and end of a traffic peak. Our model convolves separately in the temporal and spatial dimensions and incorporates a self-attention mechanism to respond more quickly to the dynamically changing traffic flow.

6 Conclusion

We considered traffic flow forecasting as a spatial-temporal forecasting problem, and proposed STGHTN for traffic flow prediction. We separately performed TGC and SGGC

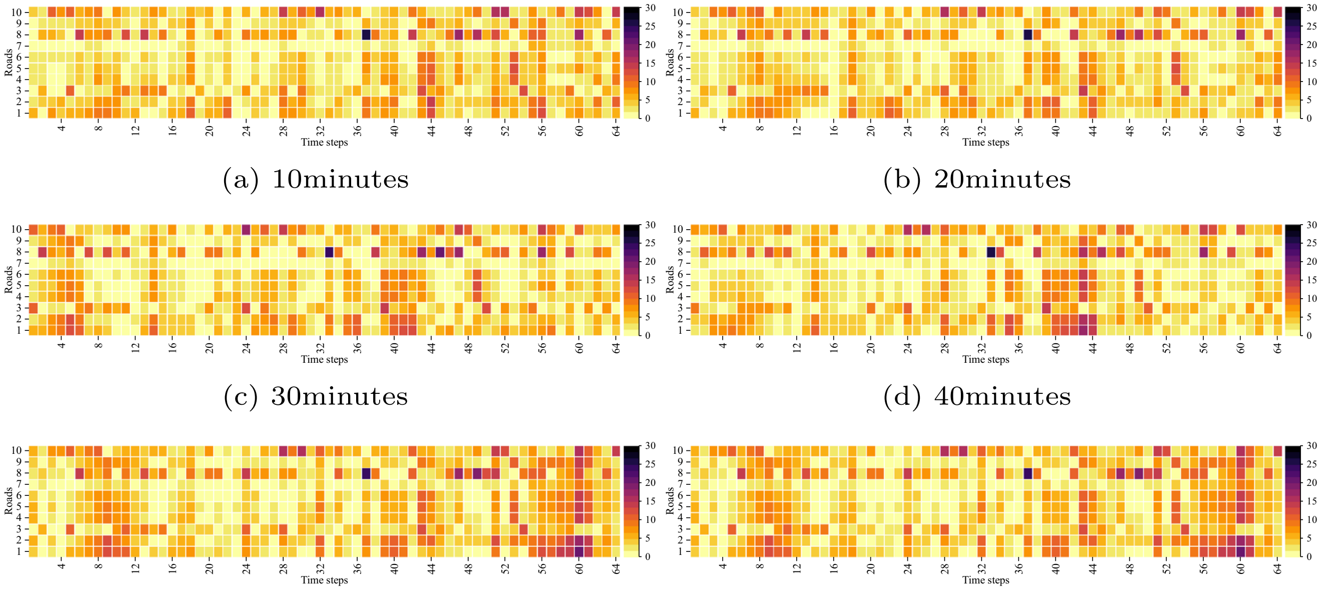


(a) 10 minutes (b) 20 minutes

(c) 30 minutes (d) 40 minutes

(e) 50 minutes (f) 60 minutes

Fig. 12 Heatmap shows the absolute errors between true and predicted values for different prediction horizons on PEMS04



(a) 10minutes

(b) 20minutes

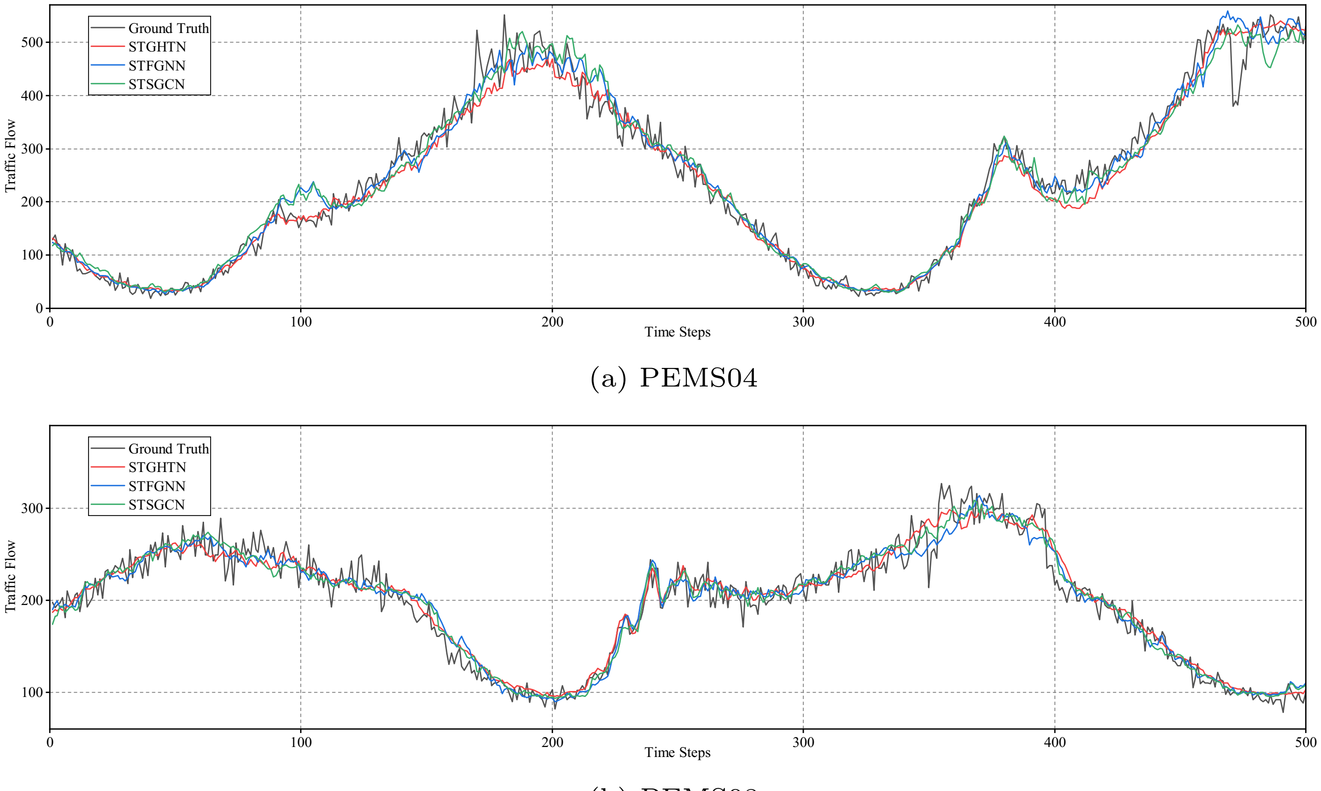
(c) 30minutes

(d) 40minutes

(e) 50minutes

(f) 60minutes

Fig. 13 Heatmap shows the absolute errors between true and predicted values for different prediction horizons on PEMS08



(a) PEMS04

(b) PEMS08

Fig. 14 Visualization results of different models on PEMS04 and PEMS08

in the temporal and spatial dimensions to extract local temporal and spatial dependence. These dependencies are complementary to the global dependence obtained by transformer. Multiple spatial graph structures were constructed to exploit the static and dynamic associations between roads. Moreover, we employ STFL to explore spatial-temporal dependence across different time periods.

We conducted rich experiments on four real datasets and achieved the optimal performance compared to the state-of-the-art. In future work, we will exploit more effectively fusing mechanisms for transformer, and apply our model to more spatial-temporal sequence tasks, such as rainfall predictions.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grant No. 61762092); the Open Foundation of the Key Laboratory in Software Engineering of Yunnan Province (Grant No. 2020SE303); the Major Science and Technology Project of Precious Metal Materials Genome Engineering in Yunnan Province (Grant No. 2019ZE001-1 and 202002AB080001-6); Yunnan provincial major science and technology: Research and Application of key Technologies for Resource Sharing and Collaboration Toward Smart Tourism (Grant No. 202002AD080047); and the Postgraduate Scientific Research Innovation Project of Hunan Province.

References

1. Wang Y, Zhang D, Liu Y, Dai B, Lee LH (2019) Enhancing transportation systems via deep learning: a survey. Transp Res Part C Emerg Technol 99:144–163
2. Pu B, Liu Y, Zhu N, Li K, Li K (2020) Ed-acnn: Novel attention convolutional neural network based on encoder–decoder framework for human traffic prediction. Appl Soft Comput 97:106688
3. Kong X, Zhang J, Wei X, Xing W, Lu W (2022) Adaptive spatial-temporal graph attention networks for traffic flow forecasting. Appl Intell 52(4):4300–4316
4. Zhao Z, Chen W, Wu X, Chen PC, Liu J (2017) Lstm network: a deep learning approach for short-term traffic forecast. IET Intell Transp Syst 11(2):68–75
5. Kuang Y, Yen BT, Suprun E, Sahin O (2019) A soft traffic management approach for achieving environmentally sustainable and economically viable outcomes: an australian case study. J Environ Manag 237:379–386
6. Yan H, Ma X, Pu Z (2021) Learning dynamic and hierarchical traffic spatiotemporal features with transformer. IEEE Transactions on Intelligent Transportation Systems
7. Williams BM, Hoel LA (2003) Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. J Transp Eng 129(6):664–672
8. Hamed MM, Al-Masaeid HR, Said ZMB (1995) Short-term prediction of traffic volume in urban arterials. J Transp Eng 121(3):249–254
9. Okutani I, Stephanedes YJ (1984) Dynamic prediction of traffic volume through kalman filtering theory. Transport Res B-Meth 18(1):1–11
10. Wu C-H, Ho J-M, Lee D-T (2004) Travel-time prediction with support vector regression. IEEE Trans Intell Transp Syst 5(4):276–281
11. Drucker H, Burges CJ, Kaufman L, Smola A, Vapnik V (1996) Support vector regression machines. Adv Neural Inf Process Syst 9
12. Van Lint J, Van Hinsbergen C (2012) Short-term traffic and travel time prediction models. Artif Intell Appl Critical Transp Issues 22(1):22–41
13. Huang Y, Weng Y, Yu S, Chen X (2019) Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting. In: 2019 18th IEEE International conference on trust, security and privacy in computing and communications/13th IEEE International conference on big data science and engineering (TrustCom/BigDataSE), pp 678–685. IEEE
14. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2019) T-gcn: a temporal graph convolutional network for traffic prediction. IEEE Trans Intell Transp Syst 21(9):3848–3858
15. Bai J, Zhu J, Song Y, Zhao L, Hou Z, Du R, Li H (2021) A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. ISPRS Int J Geo-Infor 10(7):485
16. Seo Y, Defferrard M, Vandergheynst P, Bresson X (2018) Structured sequence modeling with graph convolutional recurrent networks. In: International conference on neural information processing, pp 362–373. Springer
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Cho K, van Merrienboer B, Gulcehre C, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: Conference on empirical methods in natural language processing (EMNLP 2014)
19. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: IJCAI
20. Guo S, Lin Y, Feng N, Song C, Wan H (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on artificial intelligence, vol 33, pp 922–929
21. Li M, Zhu Z (2021) Spatial-temporal fusion graph neural networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on artificial intelligence, vol 35, pp 4189–4196
22. Wang X, Ma Y, Wang Y, Jin W, Wang X, Tang J, Jia C, Yu J (2020) Traffic flow prediction via spatial temporal graph neural network. In: Proceedings of the Web conference 2020, pp 1082–1092
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30
24. Lim B, Arık SÖ, Loeff N, Pfister T (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecasting 37(4):1748–1764
25. Yang B, Kang Y, Yuan Y, Huang X, Li H (2021) St-lbagan: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation. Knowl-Based Syst 215:106705
26. Kamarianakis Y, Prastacos P (2003) Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. Transp Res Rec 1857(1):74–84
27. Smith BL, Williams BM, Oswald RK (2002) Comparison of parametric and nonparametric models for traffic flow forecasting. Transp Res Part C Emerg Technol 10(4):303–321
28. Liu Y, Zheng H, Feng X, Chen Z (2017) Short-term traffic flow prediction with conv-lstm. In: 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), pp 1–6. IEEE
29. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand

prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 32

30. Xu C, Zhang A, Xu C, Chen Y (2021) Traffic speed prediction: spatiotemporal convolution network based on long-term, short-term and spatial features. Applied Intelligence, pp 1–19
31. Wu Z, Pan S, Long G, Jiang J, Zhang C (2019) Graph wavenet for deep spatial-temporal graph modeling. In: IJCAI
32. Bruna J, Zaremba W, Szlam A, Lecun Y (2014) Spectral networks and locally connected networks on graphs. In: International conference on learning representations (ICLR2014), CBLS, April 2014
33. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems 29
34. Micheli A (2009) Neural network for graphs: a contextual constructive approach. IEEE Trans Neural Netw 20(3):498–511
35. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Adv Neural Inf Process Syst 30
36. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. stat 1050:20
37. Zhang P, Ge N, Chen B, Fan K (2019) Lattice transformer for speech translation. In: Proceedings of the 57th Annual meeting of the association for computational linguistics, pp 6475–6484
38. Zhang Q, Lu H, Sak H, Tripathi A, McDermott E, Koo S, Kumar S (2020) Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 7829–7833. IEEE
39. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International conference on computer vision, pp 10012–10022
40. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: Transformers for image recognition at scale ICLR
41. Li H, Zhang S, Li X, Su L, Huang H, Jin D, Chen L, Huang J, Yoo J (2021) Detectornet: Transformer-enhanced spatial temporal graph neural network for traffic prediction. In: Proceedings of the 29th International conference on advances in geographic information systems, pp 133–136
42. Guo K, Hu Y, Sun Y, Qian S, Gao J, Yin B (2021) Hierarchical graph convolution network for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 35, pp 151–159
43. Dauphin YN, Fan A, Auli M, Grangier D (2017) Language modeling with gated convolutional networks. In: International conference on machine learning, pp 933–941. PMLR
44. Lu B, Gan X, Jin H, Fu L, Zhang H (2020) Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In: Proceedings of the 29th ACM International conference on information & knowledge management, pp 1025–1034
45. Jiang B, Zhang Z, Lin D, Tang J, Luo B (2019) Semi-supervised learning with graph learning-convolutional networks. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp 11313–11320
46. Song C, Lin Y, Guo S, Wan H (2020) Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 914–921

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jiansong Liu received the B.D. degree in Mining Engineering, Wuhan University of Science and Technology, Wuhan, China, in 2020. He is currently working toward the M.Sc. degree in software engineering from the National Pilot School of Software, Yunnan University, Kunming, China. His main research interests include spatial-temporal data forecasts, artificial intelligence, and natural language processing.

Yan Kang received Ph.D. degree in Computer Software and Theory from Institute of Software Chinese Academy of Sciences, Beijing, China, in 2003. She is a senior software architect, an associate professor of the Software Institute of Yunnan University, an innovative team of Yunnan Province, and a core member of the Key Laboratory of Software Engineering & Data Science of Yunnan Province. Her research interests include software engineering, system optimization, big data processing and mining.

Hao Li Professor, MS in Computer Science from the University of Essex, UK, Ph.D. in Computer Science from the University of Huddersfield, UK, Visiting Scholar at California Institute of Technology. Mainly engaged in distributed computing, grid and cloud computing research; familiar with software engineering, and has all been researching enterprise ERP and information construction.

Haining Wang received the B.D. degree in College of Information Science and Technology, Hunan Institute of Technology, Yueyang, China, in 2016. He is currently working toward the M.Sc. degree in software engineering from the National Pilot School of Software, Yunnan University, Kunming, China. His research interests include evolutionary computation and machine learning.

Xuekun Yang received the BS degree from Yunnan University, KunMing, Yunnan, China, in 2020 and he is now a master's student at the School of Software, Yunnan University. His main research interests include natural language processing, traffic flow prediction and protein site prediction.