Spatio-Temporal Graph Convolutional Networks for Traffic Prediction Considering Multiple Spatio-Temporal Information

1st Jianuo Ji
School of Computer Science and Technology
Harbin Engineering University
Harbin, China
jijianuo@hrbeu.edu.cn

2nd Hongbin Dong
School of Computer Science and Technology
Harbin Engineering University
Harbin, China
donghongbin@hrbeu.edu.cn

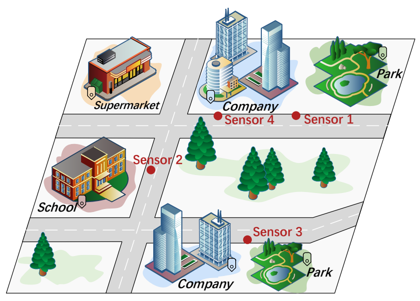
***Abstract*—As an indispensable part of smart city development, traffic prediction's fundamental challenge is effectively modeling complex spatio-temporal dependencies in traffic data. Although previous work has made great efforts to learn the temporal dynamics and spatial dependencies of traffic, the following challenges still exist. Firstly, time series has multi-scale characteristics, meaning that traffic series display different trends at various time scales and change over time. Secondly, the spatial relationships within a traffic network are not singular. The traffic condition of a node is influenced not only by nearby nodes but also potentially by distant nodes. To this end, we propose a novel framework, spatio-temporal graph convolutional networks considering multiple spatio-temporal information (STGCN-MI), for traffic prediction. Specifically, in the temporal dimension, we design a multi-scale local multi-head self-attention module. It more appropriately assigns correlation strengths to data pairs in the temporal dimension by significantly enhancing the ability to represent trends in the series at different time scales, thereby capturing dynamic temporal correlations more accurately. In the spatial dimension, we develop a fusion graph convolution module, which mines the multiple spatial dependencies in the traffic network and fuses them adaptively. Extensive experiments on two real-world datasets demonstrate that the effectiveness and superiority of our methods.**

***Index Terms*—Traffic prediction, Self-attention, Graph convolution, Spatio-temporal prediction**

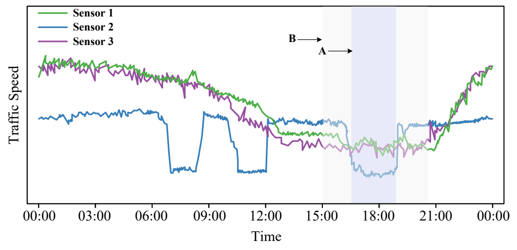
I. Introduction

With the accelerated development of urban transportation infrastructure, Intelligent Transportation Systems (ITSs) have become indispensable and crucial technologies for managing the traffic pressure brought by modern transportation [1]. Traffic prediction, a core area of ITSs, is a significant focus of current research. Accurate traffic flow prediction aids vehicle scheduling and congestion relief, promoting smart city development.

To this end, researchers have tried a variety of methods. Early traffic prediction relied on traditional statistical methods and machine learning models, but these methods struggled with complex, nonlinear spatio-temporal data. Subsequently, deep learning models have been widely used in traffic flow prediction. For example, Yao et al. [2] used convolutional neural networks (CNN) to capture the spatial correlation of grid-based traffic data, and Pan et al. [3] employed recurrent neural networks (RNN) to learn temporal correlations. However, CNN can only capture local spatial correlations in regular grid data, making it challenging to represent the global structure of complex road networks. Additionally, RNN-based models are prone to error accumulation in long-term predictions. To address these issues and extend to non-Euclidean structured space, graph convolutional networks (GCN) [4] were introduced. GCN exhibit a natural advantage for non-Euclidean data by aggregating node information through convolution, becoming a mainstream method for traffic prediction. Meanwhile, as attention mechanism [5] continue to succeed in various fields, they have also been introduced into traffic prediction models to improve prediction accuracy [6], [7].



(a) Location distribution of different attributes



(b) Traffic speed changes detected by sensors 1, 2 and 3

Fig. 1. Distribution of locations in the traffic network and corresponding traffic data curves.

Despite the significant enhancements, there are still some limitations in existing methods. First, in the time dimension, time series have multi-scale properties. Most current studies usually capture local or global time dependencies through time convolution modules or attention mechanism, which, to some extent, ignore the multi-scale nature of time series. Specifically, traffic data series exhibit different trends at different time scales and change over time. As illustrated in Fig. 1(b), sensor 2 exhibits a smooth pattern in region A. Conversely, in region B, the pattern transitions from a decreasing trend to a stable state, followed by an increasing trend. Therefore, time series of different scales contain different temporal information, and focusing on this multi-scale property enables the model to capture long- and short-term temporal dependencies more accurately.

Second, in the spatial dimension, the spatial relationship of the transportation network is not homogeneous. The traffic condition of a node in a road network is not only affected by its neighboring nodes but may also have spatial dependencies with distant nodes. For example, as shown in Fig. 1(a), sensor 1 and sensor 4 are close and located on the same road, indicating that the traffic conditions between them are closely related. The distance between sensors 1 and 2 equals the distance between sensors 2 and 3. However, as shown in the curve of Fig. 1(b), sensors 1 and 3 have more similar trends in the collected traffic data due to their proximity in urban functional areas. However, a direct connection in the distance-based graph may not be established due to their geographically distant locations. This situation, which is common in real-world traffic networks, cannot be effectively addressed even by adaptive graphs widely used for modeling spatial correlations. Therefore, focusing on the multiple spatial relationships in the transportation network can better reflect the interactions between nodes and capture spatial information more accurately.

To address the above two challenges, we propose a novel framework, spatio-temporal graph convolutional networks considering multiple spatio-temporal information (STGCN-MI), for traffic prediction. Specifically, from the perspective of mining both local and global spatio-temporal correlations in the road network, we focus on the multi-scale characteristics of the time series in the time dimension and design a multi-scale local multi-head self-attention module. This approach generates a global receptive field of the time series and enhances the model's ability to represent trends at different scales, thereby better capturing dynamic temporal correlations. In the spatial dimension, we develop a fusion graph convolution module, which mines the multiple spatial dependencies in the traffic network and fuses them adaptively. Additionally, we add a transform layer between the encoder and decoder to minimize error accumulation. The main contributions of our work are summarized as follows:

- We propose a novel model called STGCN-MI for traffic prediction, which considers multiple spatio-temporal information to capture both local and global spatio-temporal dependencies.
- We design a multi-scale local multi-head self-attention module that pays full attention to the multi-scale trend characteristics of traffic data series to capture the dynamic temporal correlation better, and develop a fusion graph convolution module that considers multiple spatial features to model the spatial correlation of road network nodes comprehensively.
- Extensive experiments are conducted on two real-world traffic datasets. The results indicate that STGCN-MI outperforms all baseline models in terms of prediction. Moreover, extensive ablation experiments confirm the effectiveness of the designed modules in the model.

II. Related Work

Traffic prediction is a typical time series prediction problem, and similar problems include demand forecasting for cabs and crowd flow forecasting. Traditional statistical methods like ARIMA [8] struggled with complex spatio-temporal data as they ignored spatial information. Deep learning techniques have recently been widely adopted in various fields, particularly for spatio-temporal prediction problems. For instance, STResNet [9] modeled the traffic network as a grid, capturing spatial correlations using CNN, while 3DCNN [10] captured complex spatio-temporal features of traffic flow data using adaptive 3D convolutional neural networks. However, traditional convolutional methods require input data to be standard 2D or 3D grid data. In contrast, real traffic networks are non-Euclidean with complex topologies, making CNN less effective at capturing spatial dependencies.

Graph neural networks (GNN) extend the convolution operation to non-Euclidean graph data, efficiently extracting topological features. DCRNN [11] was the first model to apply GNN to traffic prediction by replacing the linear operation in GRU with diffusion convolution, cycling it to extract spatial correlations in each time slice. STGCN [12] integrated spatial graph convolution and gated temporal convolution to extract spatial and temporal features, respectively. STFGNN [13] designed the fusion of multiple spatio-temporal graphs by assembling the gated dilation CNN module in parallel with the spatio-temporal fusion graph module to learn the hidden spatio-temporal dependencies. Recent works such as ASTGCN [14], GMAN [15], ASTGNN [16], and ASTGAT [17] introduced sophisticated spatial and temporal attention mechanism to GNN to capture dynamic correlations. However, these methods do not simultaneously consider the multiple spatio-temporal information embedded in the traffic network, and most neglect to model global spatial and temporal correlations. Although STFGNN captured both local and global spatio-temporal dependencies, its design extended the adjacency matrix by three, increasing spatio-temporal complexity and failing to model long-term dependencies effectively.

III. Preliminaries

A. Problem Definition

A traffic network can be represented as a graph $G = (V, E, A)$, where V is a finite set of nodes, $|V| = N$, and $v_i \in V$ denotes the i-th sensor node deployed on the road. E is the set of edges, and $A \in R^{N \times N}$ is the adjacency matrix representing the spatial structure of the graph G. The element A_{ij} denotes the spatial correlation between node

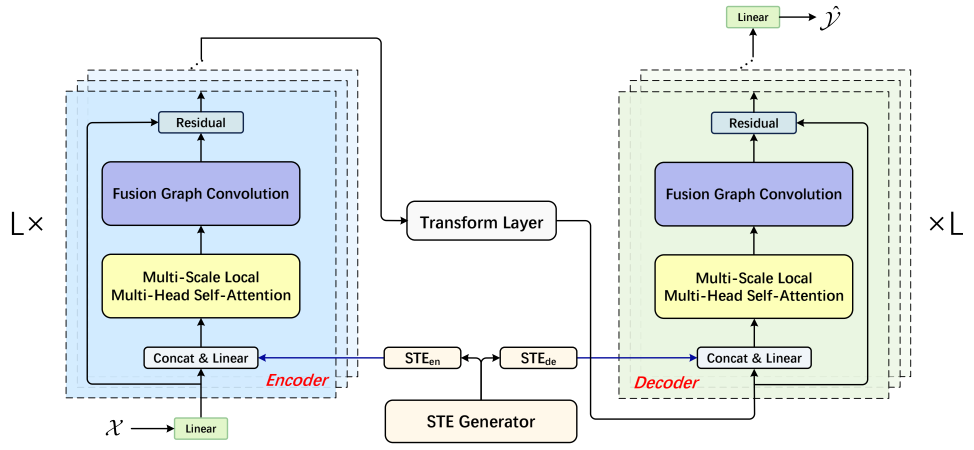


Fig. 2. The overall architecture of STGCN-MI. The model comprises three main components: (1) An encoder and decoder, which encode and decode the data for prediction; (2) A spatio-temporal embedding generator, which provides the preliminary spatio-temporal information for the encoder and decoder; and (3) A transform layer, which mitigates the error transmission effect due to the discrepancy between the historical spatio-temporal features and the predicted future spatio-temporal features.

v_i and node v_j, and is a scalar ranging from 0 to 1. The traffic data observed by the i-th sensor at time t is denoted as the graph signal $x_t^i \in R^F$, and F is the number of features (e.g., flow rate, speed, etc.) of x_t^i. Thus, the traffic data observed by the entire traffic network at time t can be defined as $X_t = \left(x_t^1, ..., x_t^i, ..., x_t^N\right)^T \in R^{N\times F}$. Given a traffic network G and a T_h time-step of historical traffic observations $\mathcal{X} = (X_1, ..., X_t, ..., X_{T_h}) \in R^{T_h\times N\times F}$ on G, the goal of prediction is to learn a function f, to predict the traffic conditions in the future for T_p time-steps:

$$[X_1, ..., X_{T_h}; G] \xrightarrow{f(\cdot)} [\hat{Y}_{T_h+1}, ..., \hat{Y}_{T_h+T_p}] \tag{1}$$

B. Attention Mechanism

The attention mechanism enhances the model's capacity to process sequential data by selectively focusing on the most relevant temporal features from the input traffic data for the current prediction target. In this mechanism, Q represents a vector of features for the current time step, K is used to measure the similarity between each element of the input traffic sequence and Q, and V is a vector associated with K. The model computes the similarity between Q and each K. These similarity values are then utilized as weights to adjust and sum the corresponding V vectors, producing the final output. In this paper, we employ the scaled dot-product attention mechanism [5], defined as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

where d_k is the dimension of vector Q and vector K.

IV. Methodology

A. Encoder-Decoder Architecture

Fig. 2 illustrates our proposed STGCN-MI framework for addressing the traffic prediction task. Before being delivered to the encoder, the historical traffic observations $\mathcal{X} = (X_1, ..., X_t, ..., X_{T_h}) \in R^{T_h\times N\times F}$ are projected into a high-dimensional space $\mathcal{X}_{en}^{(0)} \in R^{T_h\times N\times D}$ using two linear layers to enhance the symbolic power of the model, where D is the dimension of the hidden state. The encoder comprises L identical layers stacked on top of each other, with each layer containing two fundamental blocks: a multi-scale local multi-head self-attention module and a fusion graph convolution module, which capture correlations in the temporal and spatial dimensions, respectively. To prevent the gradient vanishing phenomenon as the number of layers increases, residual connections are added to facilitate faster network convergence and improve the model's generalization ability.

In the l-th encoder, the input $\mathcal{X}_{en}^{(l-1)}$ and the historical spatio-temporal embedding $E_H \in R^{T_h\times N\times D}$ are combined and passed through the concat & linear layer to obtain $\mathcal{X}^{(l-1)} \in R^{T_h\times N\times D}$, which is then passed through two modules to get $\mathcal{X}^{(l)}$. Subsequently, the residual connection is used to obtain the input $\mathcal{X}_{en}^{(l)}$ of the $(l+1)$-th encoder.

After passing through L encoders, the encoded feature $\mathcal{X}_{en}^{(L)}$ is delivered to the transform layer to generate the future representation $\mathcal{X}_{de}^{(0)} \in R^{T_p\times N\times D}$. The number of layers in both the encoder and decoder is denoted by L. The encoded feature is then passed to the first decoder. The structure of the decoders is identical to that of the encoders, except that the historical spatio-temporal embedding is replaced with a future spatio-temporal embedding as the input to the concat & linear layer. This design allows the model to consider both historical and future spatio-temporal information during the decoding process, thereby improving prediction accuracy. After the L-th decoder, the output $\mathcal{X}_{de}^{(L)}$ is processed by two linear layers to generate the prediction result $\hat{\mathcal{Y}} \in R^{T_p\times N\times F}$.

B. Multi-Scale Local Multi-Head Self-Attention Module

Correlations (e.g., periodic, trending) exist between traffic states at different time slices and vary from case to case.

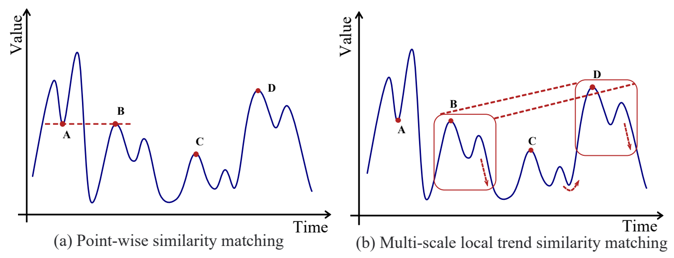


Fig. 3. Comparison of pointwise similarity matching and multi-scale local trend similarity matching for time series.

In this paper, we use an attention mechanism to capture the dynamic temporal correlations of traffic data in the time dimension. Self-attention is a specific form of the attention mechanism where the query Q, key K, and value V are identical. Multi-head self-attention [5] is the most widely used form of self-attention in practice, as it can simultaneously focus on information from different representation subspaces. Multi-head self-attention first linearly projects Q, K, and V into different representation subspaces and then executes the attention functions in parallel. Finally, each set of outputs is concatenated and further projected to produce the final output.

However, the traditional multi-head self-attention mechanism was initially designed for discrete tokens, where the similarity between queries and keys is calculated from point values without considering the multi-scale characteristics of the time series. As a result, it cannot perceive the multi-scale contextual information inherent in continuous time series. For example, Fig. 3(a) shows a traffic data sequence collected by a sensor. Since A and B have the same point-value similarity, the traditional multi-head self-attention mechanism will incorrectly match them. However, A and B have different multi-scale contextual information, such as shape and trend. Specifically, A is at the peak and valley of a period of fluctuation, while B is at the peak of a period of fluctuation, indicating that the local trends of A and B are significantly different. Thus, if the traditional multi-head self-attention mechanism is applied to this sequence, it will assign incorrect weights to the data pairs, resulting in an inaccurate representation of the sequence and affecting prediction performance.

In order to overcome the limitation of traditional multi-head self-attention in capturing multi-scale characteristics of time series, we design a multi-scale local multi-head self-attention module. This module replaces projection operations on queries and keys with multi-scale convolution to capture the complex temporal dynamics of traffic data in the time dimension. By using convolution kernels of different sizes, the model can capture the varying temporal information in the traffic sequence at different scales. Smaller convolution kernels capture short-term fluctuations fine-grained, while larger convolution kernels learn medium- and long-term trends. Suppose ordinary 1D convolution captures local contextual information, as shown in Fig. 3(b). In that case, it may incorrectly match B with C or C with D as the most robust correlation due to their similar short-term local trends. Although correlated, they would not be the most matched pairs of points in that curve. In fact, as the time scale increases, C quickly shows an upward trend, while B and D are still decreasing. Therefore, by combining features using the multi-scale local multi-head self-attention mechanism, the model will more comprehensively learn the multi-scale contextual information of B and C, matching them correctly.

Meanwhile, this paper employs causal convolution instead of projection operations on values to prevent the model from being exposed to future sequence information during the training process. Formally, the multi-scale local multi-head self-attention mechanism (MLSelf-Attn) is defined as follows:

$$MLSelfAttn(Q,K,V) = \oplus(MLhead_1, ..., MLhead_k)W^O \tag{3}$$

$$\begin{aligned} MLhead_i = Attention(&\oplus(\Phi_{i,1}^{Q} \star Q, ..., \Phi_{i,n}^{Q} \star Q), \\ &\oplus(\Phi_{i,1}^{K} \star K, ..., \Phi_{i,n}^{K} \star K), \Psi_i^V \star V) \end{aligned} \tag{4}$$

where k is the number of attention heads, W^O is the final output projection matrix. $\star$ denotes the convolution operation, $\Phi_{i,j}^{Q}$ represents one of the multi-scale convolution kernel parameters, and n indicates the number of multi-scale convolution kernels. Ψ_i^V is the causal convolution kernel parameter.

C. Fusion Graph Convolution Module

1) Multi-graph Construction: As mentioned earlier, various spatial relationships exist within the transportation network. The spatial correlation of traffic data is not only present between neighbouring nodes but also between distant nodes across the entire traffic network. To capture remote spatial dependence, some methods extend the receptive field of Graph Convolutional Networks (GCN) by increasing the number of GCN layers. However, this can lead to the over-smoothing problem [18], negatively affecting prediction performance. To address this issue, we capture spatial connections by measuring the time series similarity between nodes across the transportation network. In summary, we construct multiple graphs to encode the diverse spatial relationships present in the transportation network.

a) Neighborhood graph: In real traffic networks, it is logical to define the adjacency matrix based on the local correlation of the traffic network. To measure the influence of sensors at different distances on the traffic state of a given sensor, we use the neighborhood graph $G_D = (V, E_D, A_D)$ to encode the spatial relationships between sensors from the perspective of geographic distances. The adjacency matrix A_D is constructed based on realistic road network distances and a threshold Gaussian kernel:

$$A_{D,ij} = \begin{cases} \exp\left(-\frac{\text{dist}(v_i,v_j)^2}{\sigma^2}\right), & \text{if } \exp\left(-\frac{\text{dist}(v_i,v_j)^2}{\sigma^2}\right) \geq \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $A_{D,ij}$ denotes the weight of the edge from sensor i to sensor j, determined by the road network distance $dist(v_i, v_j)$ from the i-th sensor to the j-th sensor, σ is the standard deviation of the distance, and ε is the threshold.

b) Pattern similarity graph: Sensors located in similar urban functional areas often have spatial correlations. For example, time series collected by sensors in different business districts may show similar morning and evening peaks. For instance, Area A might reach peak flow at 9:00 a.m. on Mondays, while Area B reaches peak flow at 10:00 a.m. on the same day. Although the peaks do not necessarily co-occur, the flow generally peaks around a specific time of day.

Fast-DTW [13] is an appropriate algorithm for measuring time series similarity, especially when dealing with temporal misalignment. It aligns local patterns by warping the time axis, thereby assessing similarity within local segments of the sequence. From this, the pattern similarity graph $G_F = (V, E_F, A_F)$ is constructed, defining $A_{F,ij}$ as:

$$A_{F,ij} = \begin{cases} 1, & \text{if Fast-DTW}(X_i, X_j) < \eta \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where X_i and X_j denote the time series of node i and node j, respectively.

c) Trend similarity graph: To more accurately measure the long-range spatial correlation between sensors across the entire transportation road network, we construct the trend similarity graph $G_S = (V, E_S, A_S)$ to encode the overall trend correlation of the time series collected by the sensors. In this paper, we use the Spearman correlation coefficient to construct A_S based on the periodic data series observed by each sensor:

$$A_{S,ij} = \frac{\sum_{t=1}^{T} (u_t^i - \bar{u}^i)(u_t^j - \bar{u}^j)}{\sqrt{\sum_{t=1}^{T} \left(u_t^i - \bar{u}^i\right)^2 \sum_{t=1}^{T} \left(u_t^j - \bar{u}^j\right)^2}} \tag{7}$$

where u is the rank of the data sequence, T is the length of the sequence, i denotes the i-th sensor, and $\bar{u}^i$ is the mean of its rank. Set a correlation threshold μ and retain only strongly correlated pairs of nodes. If the correlation between two nodes exceeds μ, retain their correlation measure; otherwise, set their correlation value to 0. Spearman's correlation coefficient is advantageous because it provides a robust measure of the consistency of trends between two variables without assuming a normal distribution of the data.

2) Multi-graph convolutional fusion: For G_D, to capture the features and bi-directional spatial dependencies of the sensors within the S-hop neighborhood, we employ diffusion convolution as proposed in [11] to model these spatial dependencies. The intermediate representation $\mathcal{Z}^{(l-1)} = (Z_1^{(l-1)}, Z_2^{(l-1)}, ..., Z_{T_h}^{(l-1)})$ of the sequence obtained after the multi-scale local multi-head self-attention operation at layer l is input for diffusion graph convolution:

$$\begin{aligned} \text{DGCN}(Z_t^{(l-1)}) = \sum_{s=0}^{S} \Big(& \theta_{s,1} \left(D_O^{-1} A_D\right)^s \\ & + \theta_{s,2} \left(D_I^{-1} A_D^T\right)^s \Big) Z_t^{(l-1)} \end{aligned} \tag{8}$$

Spatial dependencies are modeled using graph convolution [19] for each of the two graph structures G_F and G_S constructed for remote spatial correlations:

$$\text{GCN}(Z_t^{(l-1)}) = \sigma(A Z_t^{(l-1)} W^{(l-1)}) \tag{9}$$

$$A = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \tag{10}$$

where $\tilde{A} = A + I$, A is the adjacency matrix, I is the unit matrix, and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $W^{(l-1)}$ is the projection matrix and σ is the activation function. In the l-th encoder, the output representations of the graph convolutions on G_D, G_F, and G_S are denoted as $\mathcal{X}_D^{(l)}$, $\mathcal{X}_F^{(l)}$, and $\mathcal{X}_S^{(l)}$, respectively. In this paper, we design a fusion operation to adaptively fuse the results of the three graph convolutions:

$$\mathcal{X}_{FS}^{(l)} = \mathcal{X}_F^{(l)} + \mathcal{X}_S^{(l)} \tag{11}$$

$$\mathcal{X}^{(l)} = z \odot \mathcal{X}_D^{(l)} + (1 - z) \odot \mathcal{X}_{FS}^{(l)} \tag{12}$$

$$z = \sigma \left(\mathcal{X}_D^{(l)} W_D + \mathcal{X}_{FS}^{(l)} W_{FS} + b_z \right) \tag{13}$$

where $W_D \in R^{D \times D}$, $W_{FS} \in R^{D \times D}$, and $b_z \in R^D$ are the learnable parameters, $\odot$ is the Hadamard product, σ denotes the sigmoid activation function, and z is the gating mechanism. After adaptive fusion, the output $\mathcal{X}^{(l)} = \left(X_1^{(l)}, X_2^{(l)}, ..., X_{T_h}^{(l)} \right)$ of the fused graph convolution module is obtained.

D. Transform Layer

We incorporate a transform layer between the encoder and the decoder to mitigate the error transmission effect due to the discrepancy between the historical spatio-temporal features and the predicted future spatio-temporal features. This layer transforms the historical information in the encoder into future sequences, inputs them into the decoder, and then further captures the future internal dependencies. Unlike the GMAN [15] model, which uses transform attention to transform traffic data, our model employs the GRU to generate future sequences. This approach avoids the strong assumption in GMAN that spatio-temporal embedding can be directly regarded as traffic data in transform attention.

Specifically, we use the last state $X_{en_{T_h}}^{(L)}$ of the encoder output as the initial hidden state $H_{T_h - 1} \in R^{N \times D}$ and start token $X_{de_{T_h}}^{(0)} \in R^{N \times D}$ to iteratively generate future sequences. The transform layer can be represented as:

$$\begin{cases} r_t = \sigma([X_{\text{de}_t}^{(0)}, h_{t-1}] W_r + b_r) \\ z_t = \sigma([X_{\text{de}_t}^{(0)}, h_{t-1}] W_z + b_z) \\ \tilde{h}_t = tanh([X_{\text{de}_t}^{(0)}, (r_t \odot h_{t-1})] W_h + b_h) \\ h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \\ X_{\text{de}_{t+1}}^{(0)} = h_t W_o + b_o \end{cases} \tag{14}$$

where σ and $\odot$ denote the sigmoid function and Hadamard product, W_r, W_z, W_h, $W_o \in {}^{D \times D}$ and b_r, b_z, b_h, $b_o \in R^{D \times D}$ are learnable parameters. With the transform layer, we can obtain the future sequence $\mathcal{X}_{de}^{(0)} \in {}^{T_p \times N \times D}$.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON PEMS-BAY AND METR-LA DATASETS.

Dataset	Methods	15 min			30 min			1 hour		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PEMS-BAY	ARIMA	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	SVR	1.85	3.59	3.80%	2.48	5.18	5.50%	3.28	7.08	8.00%
	DCRNN	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	STGCN	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	ASTGCN	1.52	3.13	3.22%	2.01	4.27	4.48%	2.61	5.42	6.00%
	STSGCN	1.44	3.01	3.04%	1.83	4.18	4.17%	2.26	5.21	5.40%
	AGCRN	1.37	2.87	2.94%	1.69	3.85	3.87%	1.96	4.54	4.64%
	STFGCN	1.32	2.90	2.81%	1.76	4.12	3.97%	2.15	4.72	4.96%
	GMAN	1.34	2.91	2.86%	1.63	3.76	3.68%	**1.86**	4.32	4.37%
	DSTAGCN	1.36	2.85	2.88%	1.70	3.84	3.83%	2.01	4.60	4.71%
	STID	**1.31**	2.79	2.78%	1.64	3.73	3.73%	1.91	4.42	4.55%
	PDFormer	1.32	2.83	2.78%	1.64	3.79	3.71%	1.91	4.43	4.51%
	STGCN-MI	1.32	**2.75**	**2.77%**	**1.61**	**3.65**	**3.61%**	1.87	**4.29**	**4.36%**
METR-LA	ARIMA	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
	SVR	3.99	8.45	9.30%	5.05	10.87	12.10%	6.72	13.76	16.70%
	DCRNN	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
	STGCN	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	ASTGCN	4.86	9.27	9.21%	5.43	10.61	10.13%	6.51	12.52	11.64%
	STSGCN	3.31	7.62	8.06%	4.13	9.77	10.29%	5.06	11.66	12.91%
	AGCRN	2.87	5.58	7.70%	3.23	6.58	9.00%	3.62	7.51	10.38%
	STFGCN	2.76	5.26	7.26%	3.30	6.89	9.14%	3.95	8.22	11.06%
	GMAN	2.80	5.55	7.41%	3.12	6.49	8.73%	**3.44**	7.35	10.07%
	DSTAGCN	2.74	**5.24**	7.12%	3.14	6.27	8.65%	3.59	7.33	10.26%
	STID	2.82	5.53	7.75%	3.19	6.57	9.39%	3.55	7.55	10.95%
	PDFormer	2.83	5.45	7.77%	3.20	6.46	9.19%	3.62	7.47	10.91%
	STGCN-MI	**2.71**	5.29	**7.05%**	**3.06**	**6.26**	**8.49%**	3.45	**7.25**	**9.97%**

E. Spatial–Temporal Embedding Generator

In order to more efficiently distinguish between sensors at different locations and times, we cascade the spatio-temporal embedding (STE) with the inputs in each layer. The spatial embedding matrix $SE \in R^{N\times D}$ is generated by node2vec [20], which randomly samples neighbors to train a vector for each node in the graph. The temporal embedding matrix $TE \in R^{T\times d_{te}}$ is generated by one-hot coding, where d_{te} denotes the time slots in a day and the value at the current time is 1. We then input the spatial and temporal embedding metrics into the two fully-connected layers to obtain the spatial embedding $SE \in R^{N\times D}$ and temporal embedding $TE \in R^{T\times D}$, respectively. They are summed up by the broadcast mechanism to produce the historical spatio-temporal embedding $E_H \in {}^{T_h\times N\times D}$ and the future spatio-temporal embedding $E_P \in {}^{T_p\times N\times D}$.

V. EXPERIMENTS

A. Datasets

We conduct experiments on two publicly available large-scale traffic datasets, METR-LA and PEMS-BAY, that record traffic speed on the Los Angeles County and Bay Area in America from March 1st, 2012 to June 30th, 2012, and January 1st, 2017 to May 31st, 2017, respectively. Both them aggregate traffic speed observations into 5-minute windows. We adopt the same data preprocessing procedure as in [11], using the Z-score method to normalize the data. All the datasets were split in chronological order with 70% for training, 10% for validation, and 20% for testing.

B. Experiment Settings

We evaluate the performance of the model using three evaluation metrics commonly used in traffic prediction: (1) Mean Absolute Error (MAE), (2) Root Mean Square Error (RMSE), and (3) Mean Absolute Percentage Error (MAPE).

We implement our experiments in the Pytorch framework with one NVIDIA 3090 GPU. STGCN-MI is trained using the Adam optimizer with an initial learning rate of 0.001, with the batch size set to 32, and the loss function used for training is MAE. The number of layers in the encoder and decoder, L, is set to 2, and the number of attention heads, k, is set to 8. The attention head dimension d is set to 8, and the model dimension D is 64. The causal convolution kernel size is set to 3, and the multi-scale convolution kernel sizes are set to 1, 3, and 5. The diffusion step for the diffusion convolution is set to 2. The number of layers in the GRU is set to 1.

C. Baseline Models

We compare the STGCN-MI with the following baseline models:

- ARIMA [8]: Auto-regressive integrated moving average model with kalman filter.
- SVR [21]: Support vector regression utilizes a linear support vector machine to perform regression.
- DCRNN [11]: Diffusion convolutional recurrent neural network, which combines graph convolutional networks with recurrent neural networks in an encoder-decoder architecture.

- STGCN [12]: Spatio-temporal graph convolution network uses the TCN and GCN to capture temporal and spatial dependencies, respectively.
- ASTGCN [14]: A model combines attention mechanism, CNN and GCN to make predictions.
- AGCRN [22]: A GNN and RNN based model that captures spatial correlations using an adaptive graph and captures temporal correlations using a GRU.
- GMAN [15]: A model that employs an encoder-decoder architecture based on attention mechanism, incorporating spatial, temporal, and transform attention.
- STSGCN [23]: Spatial–temporal synchronous graph convolutional networks. It utilizes the local spatiotemporal graph to capture localized correlations.
- STFGNN [13]: Spatial–temporal fusion graph neural networks, which assemble the graph module with a gated dilated CNN module to capture spatio-temporal dependencies.
- DSTAGCN [24]: The model connects the latest time slice with the past time slice to construct a spatio-temporal graph that captures the global spatio-temporal correlation.
- STID [25]: It encodes spatio-temporal information based on a simple MLP layer. It offers a cleaner architecture and significant efficiency advantages.
- PDFormer [26]: A Transformer architecture dynamically learns long-term spatial dependencies and explicitly models the time delay of spatial information propagation.

D. Experimental Results

We evaluate STGCN-MI at 3 steps, 6 steps and 12 steps on the two datasets. The results represent the predictive effectiveness of the model in the short, medium, and long term at T_h = 12, respectively, and are compared with 12 baseline methods. As can be seen from Table I, overall, STGCN-MI achieves state-of-the-art performance on the two datasets.

We can observe that traditional statistical methods such as ARIMA tend to perform poorly because they cannot effectively handle complex spatio-temporal data. On the other hand, DCRNN and STGCN, as classical spatio-temporal graph convolutional models, consider both temporal and spatial correlations, leading to significantly improved prediction performance. However, both DCRNN and STGCN are limited in handling long-term temporal information. In contrast, the attention mechanism of the STGCN-MI model proposed in this paper allows the model to focus on global factors, providing an advantage in long-term prediction.

Methods like AGCRN and STFGNN, which capture hidden spatial correlations using backpropagation-learned adaptive graphs, further improve performance. However, the semantic information in the adaptive graph is often biased, failing to learn the relationships between distant but highly correlated nodes. Additionally, STFGNN extends the adjacency matrix by a factor of three, increasing spatio-temporal complexity, which is also not conducive to long-term prediction. STGCN-MI also performs better overall than the GMAN model, which uses the attention mechanism to model spatio-temporal features. This

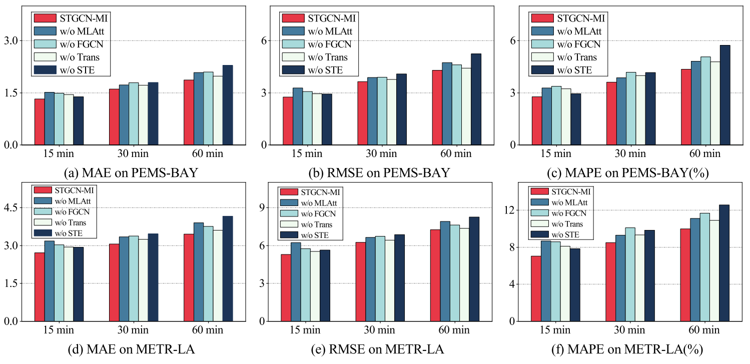


Fig. 4. Performance of different modules on PEMS-BAY and METR-LA.

result highlights the effectiveness of our attention module in better modeling long-term temporal dependencies. However, GMAN's sensitivity to long-term information and its neglect of local spatio-temporal information of the nodes make it less effective in short-term prediction despite its good performance in long-term prediction.

Compared with recent works such as DSTAGCN, STID, and PDFormer, STGCN-MI exhibits excellent performance. This is mainly because our approach captures both local and global spatio-temporal dependencies while considering multiple spatio-temporal information. In conclusion, the results in the table validate the superiority of STGCN-MI.

E. Component Analysis

We design the following variants of STGCN-MI to verify the effectiveness of each part proposed in our model:

- w/o STE: This model removes the STE generator in our STGCN-MI.
- w/o Trans: This model removes the transform layer in our STGCN-MI.
- w/o MLAtt: This model replaces the multi-scale local multi-head self-attention module with the traditional multi-head self-attention mechanism.
- w/o FGCN: This model replaces the fusion graph convolution module with conventional graph convolution using a distance-based construction of the adjacency matrix.

All variants have the same settings as STGCN-MI, except the differences mentioned above.

Fig. 4 shows the results of the ablation study of STGCN-MI and other variants on the two datasets, and we have the following findings. Removing the spatio-temporal embedding module decreases performance, indicating that fully utilizing the spatio-temporal location information of nodes is crucial. Removing the transform layer also decreases performance, demonstrating that the transform layer effectively mitigates errors arising from the gap between historical and future information. Using the traditional multi-head self-attention mechanism significantly worsens performance compared to STGCN-MI, proving the effectiveness of our proposed multi-scale local multi-head self-attention module in multi-scale trend modeling of time series. This module fills the gap that traditional self-attention mechanism fail to consider the multi-scale properties of time series when learning them, which better captures dynamic temporal correlations. The performance of graph convolution using the distance-weighted adjacency matrix based on distance weighting is poor because it aggregates local spatial information only from the distance perspective, which is less effective in modeling spatial correlations. In contrast, our proposed fusion graph convolution module mines multiple spatial relationships and skillfully fuses them adaptively to capture spatial dependencies comprehensively.

VI. Conclusion

This paper proposes a novel model STGCN-MI that considers multiple spatio-temporal information for traffic prediction. Specifically, we focus on the multi-scale nature of time series and design a multi-scale local multi-head self-attention mechanism to better capture dynamic temporal correlations. To comprehensively model the spatial dependencies of road network nodes, we develop a fusion graph convolution module, which mines multiple rich spatial relationships in the traffic network and fuses them adaptively. In addition, we add a transform layer between the encoder and decoder to minimize the error transfer effect. Extensive experiments on two publicly available traffic datasets demonstrate the superiority of our STGCN-MI. As future work, we will apply STGCN-MI to other spatial-temporal prediction tasks, such as air quality forecasting.

References

[1] H. Yuan and G. Li, "A survey of traffic prediction: from spatio-temporal data to intelligent transportation," *Data Science and Engineering*, vol. 6, no. 1, pp. 63–85, 2021.

[2] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5668–5675.

[3] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1720–1730.

[4] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert systems with applications*, vol. 207, p. 117921, 2022.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6910–6920, 2020.

[7] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention convlstm for spatiotemporal prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 531–11 538.

[8] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[9] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[10] H. Li, X. Li, L. Su, D. Jin, J. Huang, and D. Huang, "Deep spatio-temporal adaptive 3d convolutional neural networks for traffic flow prediction," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 2, pp. 1–21, 2022.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[12] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[13] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4189–4196.

[14] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[15] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[16] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2021.

[17] Y. Wang, C. Jing, S. Xu, and T. Guo, "Attention based spatiotemporal graph attention networks for traffic flow forecasting," *Information Sciences*, vol. 607, pp. 869–883, 2022.

[18] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International conference on machine learning*. PMLR, 2020, pp. 1725–1735.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[20] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[21] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, 1996.

[22] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

[23] C. Song, Y. Lin, S. Guo, and H. Wan, "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, no. 01, 2020, pp. 914–921.

[24] Q. Zheng and Y. Zhang, "Dstagcn: Dynamic spatial-temporal adjacent graph convolutional network for traffic forecasting," *IEEE Transactions on Big Data*, vol. 9, no. 1, pp. 241–253, 2022.

[25] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4454–4458.

[26] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 4, 2023, pp. 4365–4373.