STWave$^+$: A Multi-Scale Efficient Spectral Graph Attention Network with Long-Term Trends for Disentangled Traffic Flow Forecasting

Yuchen Fang, Yanjun Qin, Haiyong Luo, *Member, IEEE*, Fang Zhao, and Kai Zheng*, *Senior Member, IEEE*

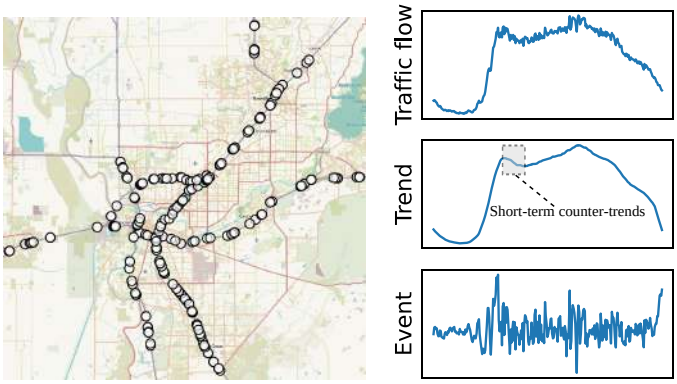
Abstract—Traffic forecasting is crucial for public safety and resource optimization, yet is very challenging due to the temporal changes and the dynamic spatial correlations. To capture these intricate dependencies, spatio-temporal networks, such as recurrent neural networks with graph convolution networks, are applied. However, traffic forecasting is still a non-trivial task because of three major challenges: 1) Previous spatio-temporal networks are based on end-to-end training and thus fail to handle the distribution shift in the non-stationary traffic time series. 2) Existing methods always utilize the one-hour input to forecast future traffic and the long-term historical trend knowledge is ignored. 3) The efficient and effective algorithm for modeling multi-scale spatial correlations is still lacking in prior networks. Therefore, in this paper, rather than proposing yet another end-to-end model, we provide a novel disentangle-fusion framework STWave$^+$ to mitigate the distribution shift issue. The framework first decouples the complex one-hour traffic data into stable trends and fluctuating events, followed by a dual-channel spatio-temporal network to model trends and events, respectively. Moreover, long-term trends are used as a self-supervised signal in STWave$^+$ to teach overall temporal information into one-hour trends through a contrastive loss. Finally, reasonable future traffic can be predicted through the adaptive fusion of one-hour trends and events. Additionally, we incorporate a novel query sampling strategy and multi-scale graph wavelet positional encoding into the full graph attention network to efficiently and effectively model dynamic hierarchical spatial correlations. Extensive experiments on four traffic datasets show the superiority of our approach, *i.e.*, the higher forecasting accuracy with lower computational cost.

Index Terms—Traffic forecasting, spatio-temporal data, graph attention network, contrastive learning.

1 Introduction

As the technological rising in the past, more and more inexpensive diversity sensors have been deployed in monitoring systems to bring an intelligent world by leveraging record values [1]. For instance, many sensors, *e.g.*, speed cameras and loop detectors, have been deployed in road networks by the transportation management department to constantly record helpful traffic information, *e.g.*, traffic flow and traffic speed, thus generating a great deal of traffic time series. Figure 1a shows an example of deploying traffic flow sensors on the highways of Northern Central California.

Given the observed traffic time series and underlying road networks, traffic forecasting aims to predict a sequence of traffic time series in the future, which benefits daily travel, traffic management, and risk assessment [2]. Despite its importance, traffic forecasting is very challenging because of the intricate temporal changes in the traffic time series and the dynamic spatial correlations between sensors under the time-varying traffic environment. Therefore, it has become a pressing need to capture the temporal changes and spatial correlations for accurately forecasting traffic in the future. As shown in Figure 2a, a common solution to the task of traffic forecasting is to directly feed the traffic data into a spatio-temporal network (STNet, *i.e.*, the combination of sequential and graph-based deep learning methods) to handle spatio-temporal dependencies simultaneously with an end-to-end training manner. Although the inspiring results of previous end-to-end STNets [3]–[7], traffic forecasting is still demanding for the following reasons.



(a) Sensor locations (b) Time series

Fig. 1: Example of sensors on the road network and the traffic flow time series with its components.

- *Y. Fang and K. Zheng are with the Yangtze Delta Region Institute (Quzhou), School of Computer Science and Engineering, Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, China. Email: fyclmiss@gmail.com, zhengkai@uestc.edu.cn. *Corresponding author: K. Zheng.*
- *Y. Qin is with the Department of Electronic Engineering, Tsinghua University, China. Email: qinyanjun@mail.tsinghua.edu.cn.*
- *H. Luo is with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, China. Email: yhluo@ict.ac.cn.*
- *F. Zhao is with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, China. Email: zfsse@bupt.edu.cn.*

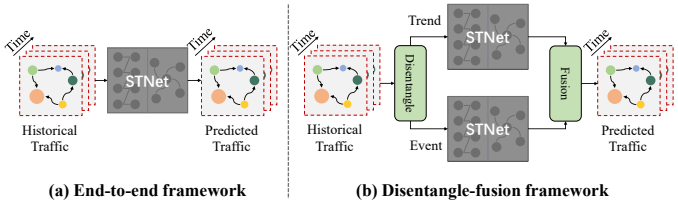


Fig. 2: The end-to-end and our proposed disentangle-fusion traffic forecasting framework, where STNet denotes the spatio-temporal network.

For the temporal aspect, traffic time series may result in end-to-end STNets over-fitting because it is entangled with multiple local independent modules and a local independent module may experiences a distribution shift [8]. As shown in Figure 1b, the recorded traffic flow time series is entangled with a stable trend series and a fluctuating event series. It is obvious that if the fluctuating event series has experienced a distribution shift, a reasonable prediction can be still made based on the invariant stable trend series. However, it is arduous for end-to-end STNets to handle the distribution shift on the fluctuating event series. In summary, the learned prediction associations from the end-to-end STNets are unable to generalize well on the non-stationary traffic time series. Therefore, we want to learn disentangled trend-event representations which are more helpful for traffic forecasting.

For the spatial aspect, graph-based deep learning methods have recently been adopted for capturing spatial correlations in traffic forecasting, such as graph convolutional networks (GCN) based methods [3]–[6], [9], [10], graph attention networks (GAT) based methods [11], [12], and full GAT (*i.e.*, considering relationships between all sensors) based methods [13], [14]. Although full GAT-based methods can dynamically capture the global spatial information and have shown state-of-the-art performance, the capability of these methods in traffic forecasting is limited by: 1) neglecting the learning efficiency of full GAT, *i.e.*, the time and space complexity of training model is $O(N^2)$, which introduces heavy computational needs and hinders the application on large-scale datasets; 2) only considering the value-based spatial semantic information and lacks the prior structure knowledge to prevent over-fitting [15].

To address the above issues, rather than proposing yet another end-to-end STNet, we provide a novel disentangle-fusion framework to mitigate the distribution shift issue. As shown in Figure 2b, the framework first disentangles the complex traffic data into stable trends and fluctuating events, followed by a dual-channel spatio-temporal network to capture the dual-scale temporal changes and spatial correlations. Therefore, the procession of trends is not violated by the non-stationary events and reasonable results can be predicted through the fusion of trends and some useful information in events. Following this principle, we design a novel model named STWave, which first applies the discrete wavelet transform (DWT) to disentangle the traffic into dual-scale trend-event representations. Then STWave proposes a STNet that utilizes the causal convolution, temporal attention, and the state-of-the-art full GAT on events, trends, and both of them to capture fluctuating temporal changes, stable temporal changes, and dynamic global spatial correlations, respectively. Moreover, to reduce the high complexity and improve the structure information of the full GAT, a novel query sampling strategy and graph wavelet positional encoding are used in STWave according to the hierarchical nature of the traffic system [16] and the spectral graph theory [17]. Finally, an adaptive event fusion module is used in STWave to merge useful information from inaccurate forecast events into easily predict trends. Experimental results on six real-world datasets show STWave significantly outperforms the state-of-the-art.

In this paper, we propose STWave$^+$ to extend the conference version STWave [18] from the following two aspects.

First, STWave only utilizes the short-term one-hour input to forecast the future, which ignores long-term trend knowledge. Compared with one-hour input, the long-term trend is more stable and robust in the face of a short-term counter-trend. For example, as shown in Figure 1b, the whole trend of historical traffic is upward, we can infer that the future traffic is very likely to be consistent with the overall trend that is also upward, even if the short-term counter-trend is downward. Therefore, to improve the accuracy of traffic forecasting, we introduce a self-supervised signal that contains long-term historical trend information to guide the training of STWave. Specifically, we use the multi-level DWT to derive long-term historical trends from the long-term traffic, which are used to predict future trends. Then a contrastive loss is computed between the hidden states of the long short-term trends to make the short-term representations consistent with the long-term and thus the long-term knowledge is acquired.

The second issue of STWave is that it fails to extract multi-scale spatial information. As [19] says, intricate spatial correlations of practical traffic scenarios can be decoupled into the road-scale and the area-scale components, *i.e.*, traffic in the business area may come from neighbor roads, roads in the neighbor residential area, and roads in the remote residential area. However, STWave can only extract the local information at the fixed scale such as neighbor roads because of the single-scale graph wavelet positional encoding-based full GAT. To further improve the capability of capturing hierarchical spatial structure dependencies, we propose a multi-scale graph wavelet positional encoding, which is obtained with different scaling sizes graph wavelet and provides multiple valuable local properties under the global information.

We summarize the new contributions of this extension as follows:

- We identify and study in depth two limitations in our previous models, which ignore the long-term trend knowledge on the temporal dimension and the hierarchical structure information on the spatial dimension.
- We design a contrastive loss to align representations of long short-term trends to inject long-term trend knowledge into the one-hour input-based model.
- We propose a multi-scale graph wavelet positional encoding to capture hierarchical spatial dependencies.
- Extensive experimental results on four real-world traffic flow datasets demonstrate the effectiveness of STWave$^+$ and its components.

The remainder of the paper is organized as follows. First, we show the literature review in Section II. Second, we give the problem formulation of traffic forecasting and elaborate system overview in Section III. The proposed STWave$^+$ is presented in Section IV. The experimental results are discussed in Section V. Section VI concludes the paper.

2 Related Work

2.1 Traffic Forecasting

Researchers utilized statistical methods to forecast traffic in the early year, *e.g.*, Historical Average [20], Vector AutoRegression [21], and AutoRegressive Integrated Moving Average [22], yet these methods rely on linear assumptions and thus fail to extract non-linear correlations of the traffic data. [23], [24] applied machine learning methods such as Support Vector Regression and K-Nearest Neighbors algorithms in traffic forecasting, but the hand-craft features limit their ability of generalization. With the success of deep learning in some research areas, a line of traffic forecasting methods modeled temporal patterns in the traffic data for each sensor individually, such as LSTM, TCN, and Transformer. However, they ignored the intricate spatial correlations between sensors on the traffic road network, *i.e.*, the traffic recorded by a sensor is influenced by the environment. Another line further combined GCNs with sequential methods to capture spatio-temporal patterns simultaneously, such as STGCN [4] and DCRNN [3]. Subsequently, to erase the impact of the pre-defined graph according to the traffic road network, GWN [9] and AGCRN [10] replaced the pre-defined graph with the adaptive graph in GCNs to capture global and accurate spatial dependencies through back-propagation. However, they lost the guidance of prior knowledge and were easy to under- or over-fitting. Compared with them, STFGNN [5] can effectively leverage the structure and semantic prior knowledge in the traffic road network and historical traffic values by the spatio-temporal fusion graph. Consequently, based on the STFGNN, STGODE [6] utilizes the tensor-based neural ODE to relieve the over-smoothing issue [25] of the deep GCN. However, most GCN-based methods ignore that correlations between sensors on the road network are constantly changing over time.

2.2 Graph Attention Network for Traffic Forecasting

To extract time-varying spatial correlations between traffic time series recorded by sensors, ST-CGA [11] utilized the graph attention network (GAT) to capture dependencies between neighbor sensors for each time slice individually. ASTGCN [26] further utilized the attention mechanism on spatio-temporal convolutions to dynamically adjust their weights. The central issue of GAT is that it only considers the spatial structure information and ignores the rich spatial semantic information, *e.g.*, sensors on the roads with the same functions or under the same environments may be highly correlated. Subsequently, LSGCN [12] dropped the input graph used in the vanilla GAT to derive the full GAT, where the full GAT can mitigate the impact of hard inductive bias brought by the input graph and mine global spatial information. Then LSGCN combined the novel cosine-based full GAT and the graph convolution as the spatial gated block to capture long- and short-range spatial dependencies, respectively. Consequently, ST-GRAT [13] designed a Transformer architecture-based model to forecast traffic speed, which stacked the full GAT and temporal attention to extract dynamic spatio-temporal information. Similar to ST-GRAT, GMAN [14] paralleled the full GAT and temporal attention but without the cross-attention. Particularly, ST-GRAT, GMAN [14], and ASTTN [27] utilized the LINE, Node2Vec, and graph Laplacian eigenvectors to generate graph positional encoding according to the traffic road network, thus bringing structure information into the model. Compared with previous graph attention-based methods, our STWave$^+$ can achieve higher accuracy with lower complexity through the query sampling strategy and the multi-scale graph wavelet positional encoding.

2.3 Wavelet Transform for Traffic Forecasting

The wavelet transform plays an essential role in the time series multi-scale analysis, which can decompose time series into different components with different frequencies. In the early years, [28]–[30] used the variants of wavelet transform such as stationary wavelet transform to decompose traffic time series into multi-components and then parallel process these components by utilizing forecasting methods such as artificial neural network, Kernel Extreme Learning Machine, and multi-linear regression, yet they are inefficient due to the multi-path architecture. [31] subsequently used the one-level discrete wavelet transform on traffic to decompose it into the changing trend and the discrete quantity and then feed them into the neural network. However, the multi-resolution analysis is missed in this method. Moreover, [32], [33] proposed the graph wavelet-based graph convolution network to extract spatial information for traffic forecasting, but the dynamic extracting ability is constrained because of the static graph. In this paper, we propose a novel discrete wavelet transform-based dual-channel framework to analyze multi-scale traffic and incorporate the graph wavelet with the graph attention to dynamically construct spatial connections.

3 Preliminaries

3.1 Traffic Network

Given the real-world traffic road network and the deployed sensors that record traffic information on the traffic road network, we formulate the traffic road network as a directed graph $\mathcal{G} = (V, E, A)$ in our paper to predict traffic, where V is the set of sensors, E is the set of edges between neighboring sensors on the traffic road network, and $A \in \mathbb{R}^{N \times N}$ corresponds to the adjacency matrix of $\mathcal{G}$.

3.2 Problem Definition

The traffic forecasting problem aims to forecast the future traffic on the traffic road network through the known historical traffic data recorded by the deployed sensors. Specifically, $x_t^i \in \mathbb{R}^C$, where $C = 1$, represents the traffic flow or speed value of the ith sensor on the traffic network at time step t, and $X_t = [x_t^1, ..., x_t^i, ..., x_t^N]^T \in \mathbb{R}^{N \times C}$ represents values of all sensors on the traffic network at time step t. Given historical T_1 time slices traffic data $\mathcal{X} = \{X_1, ..., X_{T_1}\} \in$

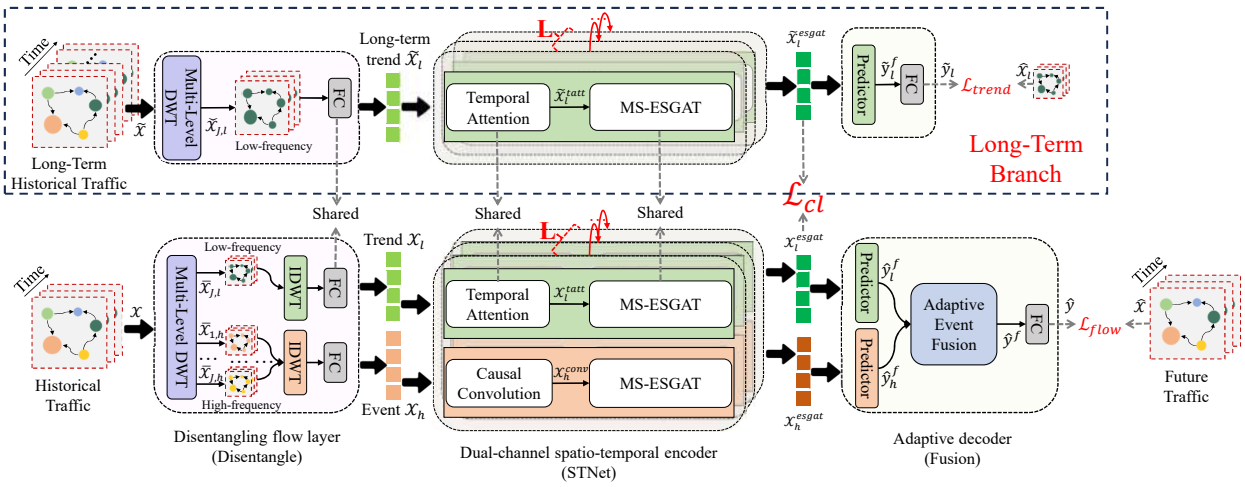


Fig. 3: The architecture of the proposed STWave$^+$. FC: fully-connected layer, DWT: discrete wavelet transform.

TABLE 1: Notations and explanations.

Notations	Explanations
$\mathcal{X}$, $\widetilde{\mathcal{X}}$	one-hour and long-term historical traffic time series
$\hat{\mathcal{X}}$, $\hat{\mathcal{Y}}$	future and predicted traffic time series
$\star$	causal convolution
$\mathbf{g}$, $\mathbf{h}$, f	low-pass, high-pass, and causal filter
α, η	temporal correlation of time slices
β, γ	spatial correlation of sensors
M, P	the score matrix and a trainable projector
E, idx	the number and index of sampled queries
Φ, λ	eigenvector and eigenvalue of the graph Laplacian
ψ, G	graph wavelet and scaling matrix
S	the set of scales of the graph wavelet
ρ	graph positional encoding
$\mathcal{L}$	objective function
A, N	the adjacency matrix and the number of sensors
C, d	the number of input features and STWave features
J, K	the level of DWT and the kernel size of causal convolution
T_1, T_2, T_3	the input, output, and long-term input length of traffic
Θ, θ	learnable parameters of STWave$^+$ and causal convolution
W, b	learnable parameters of projection

$\mathbb{R}^{T_1\times N\times C}$ of all sensors and the graph $\mathcal{G}$ of traffic network, the purpose of our paper is to learn a function $\mathcal{F}$ to forecast the traffic data of all sensors in the future T_2 time slices, namely $\hat{\mathcal{Y}} = \{\hat{Y}_{(T_1+1)}, ..., \hat{Y}_{(T_1+T_2)}\} \in \mathbb{R}^{T_2\times N\times C}$, and its ground truth is denoted by $\hat{\mathcal{X}} = \{X_{(T_1+1)}, ..., X_{(T_1+T_2)}\} \in \mathbb{R}^{T_2\times N\times C}$. The task can be formulated as:

$$\{\hat{Y}_{(T_1+1)}, ..., \hat{Y}_{(T_1+T_2)}\} = \mathcal{F}_\Theta(\{X_1, ..., X_{T_1}\}, \mathcal{G}), \quad (1)$$

where Θ denotes the learnable parameters in our model.

3.3 System Overview

Figure 3 elaborates the framework of our STWave$^+$, which consists of the four important components in two branches:

- Disentangling flow layer: Given the historical traffic data of all sensors, STWave$^+$ utilizes the multi-level DWT to separate the entangled historical traffic of all sensors into a low-frequency component and multi-high-frequency components, which can avoid the interference between these components. To consist of input dimension and enhance representation power, STWave$^+$ chronologically adopts the inverse DWT (IDWT) and fully-connected layer on these components to derive stable trends and fluctuating events.
- Dual-channel spatio-temporal encoder: Based on the stable and fluctuating properties of the disentangled trends and events, STWave$^+$ uses the temporal attention and causal convolution on trends and events to capture the stable and fluctuating temporal changes, respectively. For learning dynamic spatial dependencies in the spatio-temporal traffic data, STWave$^+$ uses two multi-scale efficient spectral graph attention networks (MS-ESGAT) on trends and events to effectively and efficiently reveal the time-varying correlations between sensors under different temporal environments.
- Adaptive decoder: Given the learned representation of historical trends and events, STWave$^+$ utilizes two predictors on them to forecast trends and events in the future, and then uses an adaptive event fusion module on them to derive the future traffic.
- Long-term branch: Different from only using the one-hour input to forecast traffic, the long-term branch in STWave$^+$ utilizes the multi-level DWT to derive the long-term trends as the input. To avoid introducing heavy computation needs, the IDWT is not used. Next, as with one-hour trends, the fully-connected layer, temporal attention, and MS-ESGAT are utilized to process long-term trends. Finally, an auxiliary loss on the stable trends and a contrastive loss between the long short-term trends are computed in STWave$^+$ to handle the distribution shift in events and inject the long-term knowledge into the one-hour input-based branch.

The details of each component will be shown in Section IV. Besides, we summarize notations used in this paper for reading convenience, as shown in Table 1.

4 Methodology

4.1 Disentangling Flow Layer

As mentioned in [34], the excellent representation of the intricate data made up of multiple sources should be disentangled into diverse explanatory sources, enhancing the

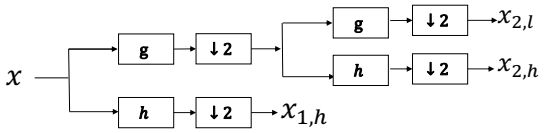
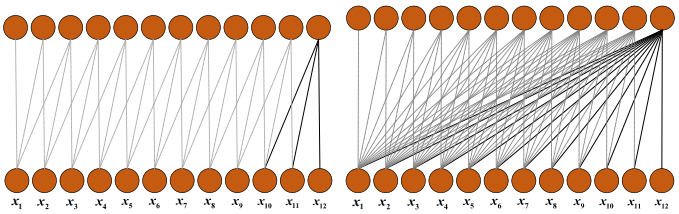


Fig. 4: Example of two-level DWT.

robustness of the model on richly structured variations. Inspired by Bayesian Structural Time Series models [35] and the independent mechanisms assumption [36], we can see that the traffic time series is composed of stable trends and fluctuating events, moreover, the trends and events do not influence each other. Therefore, when one component of the traffic time series changes because of a distribution shift, others will keep unchanged. The idea of disentangling traffic time series into trends and events results in better generalization in non-stationary temporal changes. To implement this idea, we introduce the discrete wavelet transform (DWT) into our framework to disentangle the traffic time series. The reason why we adopt DWT is that it plays an essential role in the time series multi-scale analysis when the distribution of time series varies greatly over time [37], *i.e.*, DWT can separate multiple components from the input signal according to the different frequencies by using filters of wavelets, such as slowly changes in stable trends correspond to the low-frequency. Figure 4 shows an example of two-level DWT, which decomposes the input signal $x \in \mathbb{R}^T$ into a low-frequency component $x_{2,l} \in \mathbb{R}^{\frac{T}{4}}$ including trends and two high-frequency components $x_{2,h} \in \mathbb{R}^{\frac{T}{4}}$ and $x_{1,h} \in \mathbb{R}^{\frac{T}{2}}$ that save events, where $\mathbf{g}$ and $\mathbf{h}$ represent the low-pass filter and high-pass filter of a wavelet, and we can utilize the most suitable wavelet from widely used wavelets such as Haar wavelet for disentangling traffic time series through experiments. Therefore, given the traffic time series $\mathcal{X} \in \mathbb{R}^{T_1 \times N \times C}$, we can utilize the multi-level DWT to obtain smooth enough low- and multi-high-frequency components through filters, where the low- and high-frequency components can represent the stable trends and fluctuate events in the traffic time series. For brevity, we only show the two-level DWT process, and it can be generalized to more levels with only slight changes. The DWT on the input traffic data $\mathcal{X}$ can be formulated as:

$$\begin{aligned} \bar{\mathcal{X}}_{2,l} &= (\mathbf{g} \star (\mathbf{g} \star \mathcal{X})_{(\downarrow 2)})_{(\downarrow 2)}, \\ \bar{\mathcal{X}}_{2,h} &= (\mathbf{h} \star (\mathbf{g} \star \mathcal{X})_{(\downarrow 2)})_{(\downarrow 2)}, \\ \bar{\mathcal{X}}_{1,h} &= (\mathbf{h} \star \mathcal{X})_{(\downarrow 2)}, \end{aligned} \tag{2}$$

where $\star$ is the convolution operation and $\downarrow 2$ means the output is down-sampled by 2. After the DWT, we can note that the time slices in the low-frequency component and the high-frequency component are reduced by the down-sampling operation in DWT. To consist length with input and return the different frequency data into the time domain, the up-sampling operation and IDWT with inverse low- and high-pass filters $\mathbf{g}^T, \mathbf{h}^T$ are applied in this layer. Moreover, we add all inverse high-frequency components as events to keep using non-stationary information and without many channels, *i.e.*, drop high-frequency components may lose some useful information, and parallel processing of all high-frequency components will introduce more computation needs. Then we utilize the fully-connected layer to transform trends and events into high-dimensional $\mathcal{X}_l, \mathcal{X}_h \in \mathbb{R}^{T_1 \times N \times d}$, which can improve the representation power of the following spatio-temporal network. The IDWT and fully-connected module are formulated as:



(a) Causal convolution (b) Temporal attention

Fig. 5: Causal convolution and temporal attention.

$$\begin{aligned} \mathcal{X}_l &= W^g \mathbf{g}^T \star (\mathbf{g}^T \star (\bar{\mathcal{X}}_{2,l})_{\uparrow 2})_{\uparrow 2} + b^g, \\ \mathcal{X}_h &= W^h (\mathbf{g}^T \star (\mathbf{h}^T \star (\bar{\mathcal{X}}_{2,h})_{\uparrow 2})_{\uparrow 2} \\ &\quad + \mathbf{h}^T \star (\bar{\mathcal{X}}_{1,h})_{\uparrow 2}) + b^h, \end{aligned} \tag{3}$$

where $W^g, W^h \in \mathbb{R}^{C \times d}$ and $b^g, b^h \in \mathbb{R}^d$ are learnable parameters. After the disentangling flow layer, we obtain the disentangled trend-event representations of traffic data, they can be parallel processed in the next.

4.2 Dual-Channel Spatio-Temporal Encoder

The dual-channel spatio-temporal encoder is elaborately designed to capture fluctuating temporal changes, stable temporal changes, and spatial correlations by stacking the causal convolution, temporal attention, and efficient spectral graph attention network (ESGAT) L times.

4.2.1 Temporal Changes Extraction

Different from previous works directly using a single sequential method to model the intricate temporal patterns in the entangled traffic time series, we disentangle the traffic into trends and events. It is obvious the temporal changes of trends and events are quite different. The temporal changes of trends are stable and persistent, while the temporal changes of events are fluctuating and sudden, therefore the distant time slices in the trend still have a strong correlation, and only the consecutive time slices in the event are related. As shown in Figure 5, the causal convolution with a small kernel size can only involve a little historical information, and the temporal attention can interact with all historical information with the global receptive field, they are perfectly suited to the characteristics of trends and events. Therefore, we employ the causal convolution of kernel size K with stride 1 and the temporal attention on events and trends to capture fluctuating and stable temporal changes, respectively. The causal convolution can be seen as a special 1D convolution, which slides over time slices with a local window filter, as illustrated in Figure 5a. Mathematically, given a 1D sequence $x \in \mathbb{R}^T$ with a filter $f \in \mathbb{R}^K$, the causal convolution of x with f at time step t can be formulated as:

$$x \star f(t) = \sum_{k=0}^{K} f(k) x(t-k), \tag{4}$$

in this paper, the causal convolution for the event representation $\mathcal{X}_h$ can be represented as:

$$\mathcal{X}_h^{conv} = ReLU(\theta \star \mathcal{X}_h), \tag{5}$$

where θ is a learnable parameter, $ReLU(\cdot)$ denotes the rectified linear unit. Moreover, we utilize the temporal attention on the trend representation $\mathcal{X}_l$ because trends are stable and all historical time slices have strong correlations to the future. The temporal attention for the trend representation of sensor n at time slice t can be formulated as:

$$\begin{aligned} x_{l_t}^{tatt^n} &= \sum_{i=1}^{t} \alpha_{t,i}^n (W^{V_T} x_{l_i}^n) \\ \alpha_{t,i}^n &= \frac{exp((W^{Q_T} x_{l_t}^n)^T (W^{K_T} x_{l_i}^n))}{\sum_{k=1}^{t} exp((W^{Q_T} x_{l_t}^n)^T (W^{K_T} x_{l_k}^n))}, \end{aligned} \tag{6}$$

where $W^{Q_T}, W^{K_T}, W^{V_T} \in \mathbb{R}^{d \times d}$ are learnable parameters, $\alpha_{t,i}^n$ denotes the correlation between trends at time slice t and i, and $exp(\cdot)$ denotes the exponential function.

After extracting temporal changes, representations $\mathcal{X}_h^{conv}, \mathcal{X}_l^{tatt} \in \mathbb{R}^{T_1 \times N \times d}$ of trends and events are obtained.

4.2.2 Spatial Correlations Extraction

For the multi-variate traffic forecasting task, many works have proven that capturing spatial correlations between sensors on the road network is an effective way to improve performance, and a large number of graph-based models have been proposed. The graph-based models can be roughly divided into three categories: GCN-based models, GAT-based models, and full GAT-based models. However, GCN-based models fail to capture the time-varying spatial correlations and GAT-based models can only dynamically capture the spatial correlations between neighbors. Therefore, the full GAT may be an excellent spatial correlation modeling technology for traffic forecasting because it can dynamically capture spatial correlations between all sensors, where the full GAT on the learned representations $\mathcal{X}_h^{conv}, \mathcal{X}_l^{tatt}$ is shown as follows. For simplicity, we remove the superscript and subscript in $\mathcal{X}_h^{conv}, \mathcal{X}_l^{tatt}$ and utilize a unified representation $\mathcal{X}$ in this section.

$$\begin{aligned} x_t^n &= \sum_{i=1}^{N} \beta_t^{n,i} (W^{V_S} x_t^i) \\ \beta_t^{n,i} &= \frac{exp((W^{Q_S} x_t^n)^T (W^{K_S} x_t^i))}{\sum_{k=1}^{N} exp((W^{Q_S} x_t^n)^T (W^{K_S} x_t^k))}, \end{aligned} \tag{7}$$

where $W^{Q_S}, W^{K_S}, W^{V_S} \in \mathbb{R}^{d \times d}$ are learnable parameters of projections. $\beta_t^{n,i}$ denotes the correlation between sensor n and i at time slice t. However, we observe two major limitations of the original full GAT. First, the original full GAT has a quadratic calculation complexity about the sensor number N, and N is very large in the real-world datasets, thus bringing unaffordable computation needs. Second, the original full GAT only calculates value-based spatial semantic correlations and lacks the structural information of the graph, which may result in over-fitting. To address these limitations, we propose a MS-ESGAT with a query sampling strategy and multi-scale graph positional encoding. The architecture of MS-ESGAT is shown in Figure 6.

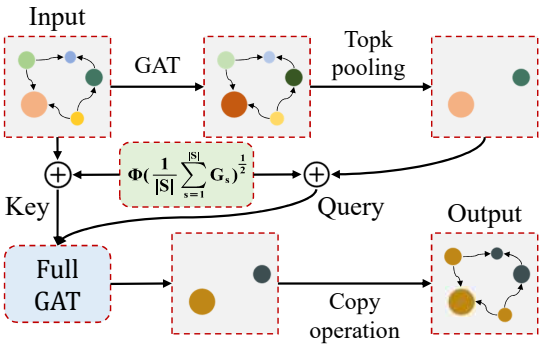


Fig. 6: An illustration of the proposed MS-ESGAT.

Query Sampling Strategy: A direct way to reduce the complexity of the original full GAT is to gain information from only neighbors, which degenerates into the vanilla GAT and loses the global information. To maintain global receptive field, a query sampling strategy is proposed to select active sensors as sparse queries to absorb information from all sensors. The results of unsampled sensors are copied from a sampled sensor that has the highest correlation between them. This strategy is inspired by the fact that sensors located in a region or community always have similar functions and flow under the hierarchical traffic system [19]. Therefore, we first utilize a GAT to pass messages between traffic time series, and then use a topk-pooling to select active sensors on behalf of regions or communities. The GAT can be formulated as:

$$\begin{aligned} m_t^n &= \sum_{i \in \mathcal{N}_n} \gamma_t^{n,i} (W^{V_M} x_t^i) \\ \gamma_t^{n,i} &= \frac{exp((W^{Q_M} x_t^n)^T (W^{K_M} x_t^i))}{\sum_{k \in \mathcal{N}_n} exp((W^{Q_M} x_t^n)^T (W^{K_M} x_t^k))}, \end{aligned} \tag{8}$$

where $\mathcal{N}_n$ and $M_t \in \mathbb{R}^{N \times d}$ denote the index of neighbors of sensor n on the road network and the scores of sensors at time step t. Then we utilize the topk-pooling to sample E active sensors that receive max information from other sensors as sparse queries. The number of E is controlled by a constant sampling factor e, we set $E = \lceil elogN \rceil$, which makes the followed full GAT only need to calculate $O(NlogN)$ dot-product, and the layer memory usage maintains $O(NlogN)$. Specifically, to evaluate how much information from other sensors can be retained, we employ a trainable projection vector $P \in \mathbb{R}^{d \times 1}$ to project the score matrix to 1D and sample sensors according to values:

$$idx_t = rank(\frac{M_t P}{\|P\|}, E), \tag{9}$$

where $rank(\cdot)$ returns the index of top E largest values and therefore the left term $idx_t \in \mathbb{R}^E$ is a set contains E indices of top E largest scores at time slice t. Finally, the full GAT on sampled sensors and the copy operation on unsampled sensors are formulated as:

$$\begin{aligned} x_t^{esgat^n} &= \sum_{i=1}^{N} \beta_t^{n,i} (W^{V_S} x_t^i), \text{ where } n \in idx_t, \\ x_t^{esgat^n} &= x_t^{esgat^c}, c = rank(\beta_t^{:,n}, 1), \text{ where } n \notin idx_t. \end{aligned} \tag{10}$$

Graph Positional Encoding: To effectively inject structure information into the full GAT, we propose a novel

graph positional encoding. In the vanilla Transformer architecture [38], the positional encoding of sequences are always sine and cosine functions, which is an important part of the self-attention to distinguish time slices. However, sinusoids cannot be clearly defined in graphs, since there is no clear notion of position along an axis. In graph-based tasks, [39] uses graph Laplacian eigenvectors as the graph positional encoding because eigenvectors of the graph Laplacian are the natural equivalent of sine functions, which can reveal the structure information in the graph. However, the influence of the eigenvectors on the signal of one node is not localized in its neighborhood [17]. Different from the graph Laplacian eigenvectors, the graph wavelet [40] corresponds to graph Laplacian eigenvectors diffused away from a central node with a scaling matrix on the graph and can reflect the local property compared with eigenvectors, where the graph wavelet ψ_s at scale s can be formulated as:

$$\psi_s = \Phi G_s \Phi^T, \tag{11}$$

where Φ contains eigenvectors of the graph Laplacian. $G_s = diag(exp(s\lambda_1), ..., exp(s\lambda_d))$ is the scaling matrix at scale s, and λ_i is the ith lowest graph Laplacian eigenvalues. It is obvious that the graph wavelet ψ_s can be seen as the dot-product of $\Phi G_s^{\frac{1}{2}}$ and its transpose, and spatial correlations between sensors are also calculated by the dot-product. Therefore, we can set $\Phi G_s^{\frac{1}{2}}$ as our graph positional encoding $\rho \in \mathbb{R}^{N\times d}$, which shows not only the structure information but also the local property of graphs. Although the graph wavelet positional encoding can bring the local property to the full GAT, in the real-world traffic system, the traffic on the road is not only affected by roads at a fixed distance from it but also by roads from many different distances simultaneously. However, the single-scale graph wavelet fails to extract the hierarchical structure information. Inspired by the multi-scale graph convolution network [41], we propose a novel multi-scale graph wavelet positional encoding, which is obtained with different scales and provides valuable local properties under the global graph information. Specifically, the single-scale graph wavelet is replaced with $|S|$ parallel units of different scale graph wavelets, where S is the set of scales. The multi-scale graph wavelet ψ can be defined as follows:

$$\psi = \frac{1}{|S|}\sum_{s=1}^{|S|} \Phi G_s \Phi^T = \Phi(\frac{1}{|S|}\sum_{s=1}^{|S|} G_s)\Phi^T, \tag{12}$$

according to Eq. (12), the multi-scale graph positional encoding is $\rho = \Phi(\frac{1}{|S|}\sum_{s=1}^{|S|} G_s)^{\frac{1}{2}} \in \mathbb{R}^{N\times d}$. Furthermore, we set the $|S|$ parallel scales as learnable parameters to avoid misleading inductive bias through back-propagation, which is inspired by the adaptive graph learning technology [9], [10]. Finally, our multi-scale efficient spectral graph attention can be formulated as follows:

$$\begin{aligned}
\dot{x}_t^i &= x_t^i + \rho^i, \text{where } i \in [1, ..., N] \\
x_t^{esgat^n} &= \sum_{i=1}^{N} \beta_t^{n,i}(W^{V_S}\dot{x}_t^i), \text{where } n \in idx_t \\
\beta_t^{n,i} &= \frac{exp((W^{Q_S}\dot{x}_t^n)^T(W^{K_S}\dot{x}_t^i))}{\sum_{k=1}^{N} exp((W^{Q_S}\dot{x}_t^n)^T(W^{K_S}\dot{x}_t^k))} \\
x_t^{esgat^n} &= x_t^{esgat^c}, c = rank(\beta_t^{:,n}, 1), \text{where } n \notin idx_t.
\end{aligned} \tag{13}$$

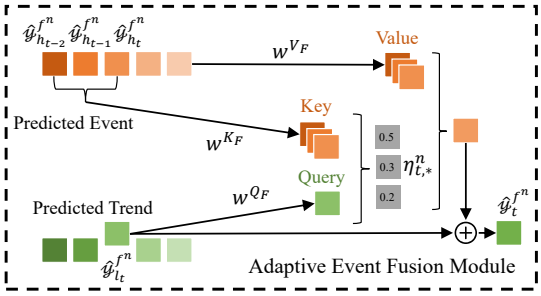


Fig. 7: An illustration of the proposed adaptive event fusion.

After extracting spatial correlations, representations $\mathcal{X}_h^{esgat}, \mathcal{X}_l^{esgat} \in \mathbb{R}^{T_1\times N\times d}$ are obtained.

4.3 Adaptive Decoder

4.3.1 History-Future Transform

To transform the learned representations encoded by the dual-channel encoder into the future, we utilize predictors (*i.e.*, fully-connected neural networks) on the temporal dimension of $\mathcal{X}_l^{esgat}, \mathcal{X}_h^{esgat} \in \mathbb{R}^{T_1\times N\times d}$ to derive the future representations $\hat{\mathcal{Y}}_l^f, \hat{\mathcal{Y}}_h^f \in \mathbb{R}^{T_2\times N\times d}$ of trends and events.

4.3.2 Adaptive Event Fusion

Unlike stable trends that can be reasonably predicted most of the time, fluctuating events often have a distribution shift that skews the predicted results. Therefore, we need to keep intentional events and discard useless events. As shown in Figure 7, we make a weighted sum on the events for each time slice in the trends, and the weight is calculated by the attention and can be learned through the back-propagation, *i.e.*, we use a data-driven way to adaptively judge whether the event is accurately predicted. The adaptive event fusion can be formulated as:

$$\begin{aligned}
\hat{y}_t^{f^n} &= \hat{y}_{l_t}^{f^n} + \sum_{i=T_1+1}^{T_1+t} \eta_{t,i}^n (W^{V_F}\hat{y}_{h_i}^{f^n}) \\
\eta_{t,i}^n &= \frac{exp((W^{Q_F}\hat{y}_{l_t}^{f^n})^T(W^{K_F}\hat{y}_{h_i}^{f^n}))}{\sum_{k=T_1+1}^{T_1+t} exp((W^{Q_F}\hat{y}_{l_t}^{f^n})^T(W^{K_F}\hat{y}_{h_k}^{f^n}))},
\end{aligned} \tag{14}$$

the future representation of traffic $\hat{\mathcal{Y}}^f \in \mathbb{R}^{T_2\times N\times d}$ is obtained by the fusion.

4.3.3 Traffic Forecasting

Finally, we first use a fully-connected neural network to transform the future representation of traffic $\hat{\mathcal{Y}}^f$ into the expected prediction $\hat{\mathcal{Y}} \in \mathbb{R}^{T_2\times N\times C}$, and then utilize the $L1$ loss to supervise traffic forecasting:

$$\mathcal{L}_{flow} = \sum_{t=T_1+1}^{T_1+T_2}\sum_{n=1}^{N} |x_t^n - \hat{y}_t^n|. \tag{15}$$

4.4 Long-Term Branch

The long-term historical trends are vital for the future, which can provide overall temporal changes that are not covered in the original one-hour input, *e.g.*, overall temporal changes of traffic flow in a day are first rising and

then falling. Unfortunately, most previous traffic forecasting methods followed the paradigm that using one-hour input to predict the future, *i.e.*, utilizing long-term historical information in traffic forecasting is still under-explored. The intuitive way to inject long-term historical trend knowledge into a model is directly using the long-term historical traffic as the input, yet introducing heavy computation needs, which is unbearable for the current computing resources. Therefore, we consider utilizing the long-term historical trend as a self-supervised signal to guide the training of the original model, which can avoid inefficient inference stage. Moreover, to keep training as efficient as possible, we use the multi-level DWT to derive the long-term historical trends, which are consistent with the length of one-hour input. Specifically, given the long-term historical traffic data $\widetilde{\mathcal{X}} \in \mathbb{R}^{T_3 \times N \times C}$, where $T_3 > T_1$, the multi-level DWT with down-sampling operation is used on it to get the long-term historical trends $\check{\mathcal{X}}_l \in \mathbb{R}^{T_1 \times N \times C}$. Similar to the one-hour input, we use the fully-connected layer to transform the long-term trends into high-dimensional $\widetilde{\mathcal{X}}_l \in \mathbb{R}^{T_1 \times N \times d}$ to improve the representation power of the following spatio-temporal network. The high-dimensional long-term trends then pass through an encoder that contains temporal attention and MS-ESGAT to obtain the spatio-temporal representations $\widetilde{\mathcal{X}}_l^{esgat} \in \mathbb{R}^{T_1 \times N \times d}$. Moreover, the input fully-connected layer and the encoder of the long-term branch and the one-hour trend branch share weights in this paper. This is because they have the same structure and thus the long-term branch can influence the one-hour trend during the training process. For transferring the more stable and robust knowledge of the long-term trends into one-hour inputs, we want to make short-term representations as similar as possible to long-term representations. To achieve this goal, we utilize the contrastive loss [42] to align representations at the same time, and thus representations of the whole sequence of long short-term trends will be similar. The contrastive loss is formulated as follows:

$$\mathcal{L}_{cl} = \frac{1}{T_1} \sum_{t=1}^{T_1} -log \frac{exp(sim(X_{l_t}^{esgat}, \widetilde{X}_{l_t}^{esgat}))}{\sum_{n=1}^{T_1} exp(sim(X_{l_t}^{esgat}, \widetilde{X}_{l_n}^{esgat}))}, \quad (16)$$

where $sim(\cdot, \cdot)$ indicates the similarity function (*e.g.*, inner product). After the encoder, we chronologically utilize a predictor to transform the historical long-term trends $\widetilde{\mathcal{X}}_l^{esgat} \in \mathbb{R}^{T_1 \times N \times d}$ into the future $\widetilde{\mathcal{Y}}_l^f \in \mathbb{R}^{T_2 \times N \times d}$ and a fully-connected layer to project future trends into the 1D value $\widetilde{\mathcal{Y}}_l \in \mathbb{R}^{T_2 \times N \times C}$. Finally, the predicted future trends are supervised in our model by the $L1$ loss to align long-term historical trends with future stable temporal changes. The trend loss is expressed as follows:

$$\mathcal{L}_{trend} = \sum_{t=T_1+1}^{T_1+T_2} \sum_{n=1}^{N} |x_{l_t}^n - \tilde{y}_{l_t}^n|, \quad (17)$$

where $x_{l_t}^n$ is the ground truth of future trends.

4.5 Objective Function

Therefore, by considering the traffic forecasting loss, contrastive loss, and trend loss, STWave$^+$ aims to jointly minimize the following objective function:

$$\mathcal{L} = \mathcal{L}_{flow} + \mathcal{L}_{cl} + \mathcal{L}_{trend}, \quad (18)$$

Algorithm 1: Training procedure of STWave$^+$.

Data: Road network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$;
Time slices T of train set;
Train data of all observing sensors $\mathcal{T} \in \mathbb{R}^{T \times N \times C}$;
All hyperparameters;
for $t \leftarrow T_3 + 1$ **to** $T - T_2$ **do**
  Append $\mathcal{T}_{t-T_1:t}$ to input sample set $\mathcal{X}$;
  Append $\mathcal{T}_{t-T_3:t}$ to long-term input sample set $\widetilde{\mathcal{X}}$;
  Append $\mathcal{T}_{t:t+T_2}$ to label sample set $\mathcal{Y}$;
end
Initialize all learnable parameters Θ in STWave$^+$;
Calculate eigenvectors Φ and eigenvalues λ of graph Laplacian of A;
repeat
  Randomly select a batch of input sample $\mathcal{X}_{bs}$;
  Randomly select a batch of long-term input sample $\widetilde{\mathcal{X}}_{bs}$;
  Randomly select a batch of label sample $\mathcal{Y}_{bs}$;
  Use DWT to calculate the long-term trend $\widetilde{\mathcal{X}}_{l_{bs}}$, one-hour trend $\mathcal{X}_{l_{bs}}$, and one-hour event $\mathcal{X}_{h_{bs}}$ of the input sample;
  Feed $\mathcal{X}_{l_{bs}}$, $\widetilde{\mathcal{X}}_{l_{bs}}$, $\mathcal{X}_{h_{bs}}$, Φ, and λ into STWave$^+$;
  Optimize Θ by minimizing the objective function;
until *met model stop criteria*;
Result: Learned STWave$^+$ model.

moreover, we show the training procedure of our STWave$^+$ in Algorithm 1.

4.6 Complexity Analysis

The complexity of causal convolution, temporal attention, and MS-ESGAT are $O(TNK)$, $O(NT^2)$, $O(TNlogN)$. Thus the complexity of the long-term branch encoder and the one-hour encoder is $O(L(NT^2 + TNlogN))$ and $O(L(TNK + NT^2 + TNlogN))$, where L represents the number of stacked layers. Moreover, the complexity of the disentangling flow layer and the decoder is $O(NT)$ and $O(NT^2)$, respectively. Besides, the complexity of calculating d smallest eigenvectors and eigenvalues of graph Laplacian is $O(Nd + d^2)$ [43], it can be quickly preprocessed without affecting the model complexity. Therefore, STWave$^+$ achieves comparable time and memory complexity compared to GCN-based models.

5 Experiments

We investigate the effectiveness of our STWave$^+$ with the goal of answering the following research questions:

- **RQ1:** Does our STWave$^+$ outperform baselines?
- **RQ2:** How do framework and components in STWave$^+$ (*e.g.*, MS-ESGAT) affect model performance?
- **RQ3:** How do hyper-parameters affect STWave$^+$?
- **RQ4:** Does our MS-ESGAT efficient and effective?
- **RQ5:** Can STWave$^+$ provide reasonable results?

TABLE 2: Dataset statistics.

Datasets	#Nodes	#Edges	#Samples	Time Range
PeMSD3	358	1093	26208	09/2018-11/2018
PeMSD4	307	680	16992	01/2018-02/2018
PeMSD7	883	548	28224	05/2017-08/2017
PeMSD8	170	1732	17856	07/2016-08/2016

5.1 Experimental Setup

5.1.1 Datasets

We evaluate our model on four real-world traffic flow datasets collected from the California Transportation Agencies Performance Measurement System, named PeMSD3, PeMSD4, PeMSD7, and PeMSD8. They are sampled in real-time every 5 minutes and widely used in previous studies [4], [5]. Descriptive statistics for these datasets are presented in Table 2. Following the traditional paradigm, we use the observations traffic from the previous 12 (*i.e.*, one-hour) and long-term time slices to predict the next 12 slices, and split them into a training set (60%), validation set (20%), and test set (20%) in chronological order.

5.1.2 Metrics

In this paper, we utilize three widely used metrics, namely, Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE), and Root Mean Squared Errors (RMSE).

5.1.3 Baselines

We compare STWave$^+$ with the following 16 baselines:

- HA [20]: It utilizes average value of history to iterative predict the future.
- ARIMA [22]: It integrates moving average into the Autoregressive model.
- VAR [21]: It is a statistical model that can capture spatial dependencies.
- SVR [23]: It utilizes the support vector machine to perform traffic forecasting.
- LSTM [44]: It is a advanced version of the RNN with the long-term memory.
- TCN [45]: It integrates the dilated kernel into the causal convolution.
- STGCN [4]: It joints the causal convolution network with the graph convolution network to extract spatial-temporal dependencies simultaneously.
- DCRNN [3]: It integrates pre-defined graph-based GCN into the encoder-decoder architecture-based recurrent network to predict multi-slice traffic.
- GWN [9]: It combines the gated TCN and the adaptive graph-based GCN to capture spatio-temporal dependencies simultaneously.
- ASTGCN [26]: It performs the attention mechanism on the temporal and spatial convolutions to extract dynamic spatio-temporal correlations.
- LSGCN [12]: It uses a gated graph block to satisfy the long- and short-range spatial dependencies, which contains a graph convolution network and a novel cosine graph attention network.
- STSGCN [46]: It uses a spatio-temporal synchronous technology to extract the local spatio-temporal correlations.
- AGCRN [10]: It integrates the adaptive graph-based GCN into the encoder-decoder architecture-based recurrent network.
- STFGNN [5]: It designs a dynamic time warping-based temporal graph to mine functional-aware spatial relationships.
- STGODE [6]: It re-writes the GCN into the neural ODE from to relieve the over-smoothing issue in the deep GCN. Besides, it uses temporal and pre-defined graph to represent spatial correlations.
- STWave [18]: It is the conference version of STWave$^+$. Specifically, it does not equip the long-term branch and utilizes the single-scale ESGAT.

5.1.4 Hyper-Parameter Settings

We train STWave$^+$ using the Adam optimizer for 200 epochs with a batch size of 64 and an initial learning rate of 0.001. Moreover, the learning rate decays to $\frac{1}{10}$ when loss does not decrease through 20 epochs during the training. We list the default settings of our model as follows: the number of features d in STWave is set as 128, the kernel size K in the causal convolution is set as 2, the level J of DWT is set as 1, the sampling factor e of MS-ESGAT is set to 1, the scale S of MS-ESGAT is set to 3, the long-term historical length T_3 is set to 48, *i.e.*, the input of the long-term branch is four hours, and the number of layers L in spatio-temporal encoder is set as 2. Besides, we set different discrete wavelets for different datasets: Symlets wavelet for PeMSD3, Daubechies wavelet for PeMSD4 and PeMSD7, Coiflets wavelet for PeMSD8.

5.1.5 Implementation Details

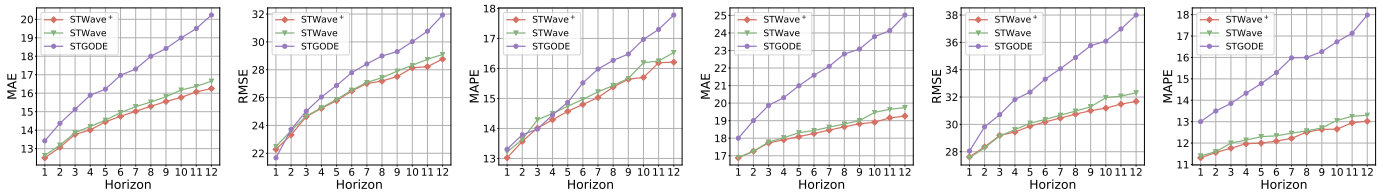
We implement STWave$^+$ on Python 3.8.10 using PyTorch 1.9.1. All experiments are conducted on a machine, running Ubuntu 20.04.3 LTS, with one Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz and one Tesla A100 GPU card.

5.2 Performance Comparison (RQ1)

The results under three metrics of STWave$^+$ and baselines across four datasets are reported in Table 3, and results under three metrics for each time slice are shown in Figure 8. From the Table, we get the following observations. First, HA performs worst in all tasks and thus provides a lower bound of traffic forecasting. Moreover, the results of ARIMA, VAR, and SVR are much worse than the neural network-based models because they fail to capture non-linear dependencies and need hand-craft features. Second, we can see that graph-free methods like LSTM are usually inferior to graph-based baselines (*e.g.*, GWN), demonstrating the assistance of graphs in capturing spatial dependencies. Third, as for graph-based algorithms, ASTGCN and LSGCN outperform previous methods, which tells us the effectiveness of extracting dynamic relationships between traffic time series. Finally, STFGNN and STGODE are better than other graph-based methods, as they carefully propose the temporal graph and the GODE to increase their spatial receptive field. However, they fail to mine global spatial correlations and thus are inferior to AGCRN. All in all, our method obtains the best performance on all datasets. There are three main reasons: 1) STWave$^+$ disentangles the trend and the event from the traffic time series and

TABLE 3: Comparison of STWave and baselines on four traffic flow datasets. **Bold**: Best, underline: Second best.

Methods	PeMSD3			PeMSD4			PeMSD7			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	31.58	52.39	33.78%	38.03	59.24	27.88%	45.12	65.64	24.51%	34.86	59.24	27.88%
ARIMA	35.41	47.59	33.78%	33.73	48.80	24.18%	38.17	59.27	19.46%	31.09	44.32	22.73%
VAR	23.65	38.26	24.51%	24.54	38.61	17.24%	50.22	75.63	32.22%	19.19	29.81	13.10%
SVR	21.97	35.29	21.51%	28.70	44.56	19.20%	32.49	50.22	14.26%	23.25	36.16	14.64%
LSTM	21.33	35.11	23.33%	26.77	40.65	18.23%	29.98	45.94	13.20%	23.09	35.17	14.99%
TCN	19.32	33.55	19.93%	23.22	37.26	15.59%	32.72	42.23	14.26%	22.72	35.79	14.03%
STGCN	17.55	30.42	17.34%	21.16	34.89	13.83%	25.33	39.34	11.21%	17.50	27.09	11.29%
DCRNN	17.99	30.31	18.34%	21.22	33.44	14.17%	25.22	38.61	11.82%	16.82	26.36	10.92%
GWN	19.12	32.77	18.89%	24.89	39.66	17.29%	26.39	41.50	11.97%	18.28	30.05	12.15%
ASTGCN(r)	17.34	29.56	17.21%	22.93	35.22	16.56%	24.01	37.87	10.73%	18.25	28.06	11.64%
LSGCN	17.94	29.85	16.98%	21.53	33.86	13.18%	27.31	41.46	11.98%	17.73	26.76	11.20%
STSGCN	17.48	29.21	16.78%	21.19	33.65	13.90%	24.26	39.03	10.21%	17.13	26.80	10.96%
AGCRN	15.98	28.25	15.23%	19.83	32.26	12.97%	22.37	36.55	9.12%	15.95	25.22	10.09%
STFGNN	16.77	28.34	16.30%	20.48	32.51	16.77%	23.46	36.60	9.21%	16.94	26.25	10.60%
STGODE	16.50	27.84	16.69%	20.84	32.82	13.77%	22.59	37.54	10.14%	16.81	25.97	10.62%
STWave	14.93	26.50	15.05%	18.50	30.39	12.43%	19.94	33.88	8.38%	13.42	23.40	8.90%
LSGCN†	16.63	28.31	16.19%	20.41	32.50	13.48%	26.07	40.26	10.77%	16.72	25.84	10.41%
AGCRN†	15.16	27.10	14.94%	18.82	30.89	12.53%	21.26	35.33	8.72%	14.89	24.60	9.72%
STGODE†	15.66	27.23	15.80%	19.87	31.89	13.42%	21.46	36.25	9.43%	15.83	25.12	10.17%
STWave+	**14.71**	**26.31**	**14.88%**	**18.25**	**30.14**	**12.21%**	**19.59**	**33.53**	**8.17%**	**13.21**	**23.04**	**8.63%**



(a) PeMSD3 MAE (b) PeMSD3 RMSE (c) PeMSD3 MAPE (d) PeMSD4 MAE (e) PeMSD4 RMSE (f) PeMSD4 MAPE

(g) PeMSD7 MAE (h) PeMSD7 RMSE (i) PeMSD7 MAPE (j) PeMSD8 MAE (k) PeMSD8 RMSE (l) PeMSD8 MAPE

Fig. 8: Prediction for each time slice on PeMSD3, PeMSD4, PeMSD7, and PeMSD8 datasets.

proposes a encoder to process each term individually. 2) Our model designs the adaptive fusion module and the long-term branch to fully incorporate and exploit information on events and long-term trends. 3) STWave+ proposes a novel multi-scale graph wavelet positional encoding to effectively reveal multi-scale spatial dependencies by injecting graph structure information into the model. Besides, as shown in the Figure, the bias between the truth and the future value is highly correlated with the length of prediction. We can see STWave+ shows a smaller bias than baselines for all time slices, especially in long-term traffic forecasting.

5.3 Ablation Study (RQ2)

In order to verify the effectiveness of our proposed disentangle-fusion framework and the self-supervised signal for traffic forecasting, we replace our STNet in the model with LSGCN, AGCRN, and STGODE to form three variants LSGCN†, AGCRN†, and STGODE†. The experimental results of these variants are shown in Table 3. Compared with the end-to-end manner, using our proposed framework achieves better results on all tasks, because our framework can not only effectively mitigate the terrible influence introduced by the distribution shift of events but also acquire long-term knowledge. Moreover, their performance worse than our STWave+ indicates that our STNet is excellent for traffic forecasting.

To investigate the effectiveness of different components in STWave+, we compare STWave+ with the following five different variants:

- "w/o DF": STWave+ without the disentangling flow layer, *i.e.*, follows the end-to-end paradigm and directly feeds the traffic into the model.
- "w/o AF": STWave+ replaces the adaptive fusion with the addition operation, *i.e.*, it directly adds events and trends and ignores the spurious forecasting of events.
- "w/o Tem": STWave+ no longer equips temporal neural network, *i.e.*, fails to capture the temporal changes.
- "w/o Spa": STWave+ without the MS-ESGAT, *i.e.*, fails

TABLE 4: Performance comparison for variants of STWave on PeMSD3, PeMSD4, PeMSD7, and PeMSD8 datasets.

Methods	PeMSD3			PeMSD4			PeMSD7			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o DF	15.27	26.95	15.60%	19.21	31.58	12.81%	20.04	34.00	8.35%	13.74	24.12	9.13%
w/o AF	15.21	26.39	15.49%	19.45	31.61	13.17%	21.44	35.62	8.91%	14.28	24.14	9.12%
w/o Tem	15.63	28.14	15.86%	19.41	31.56	13.12%	20.73	34.88	8.69%	13.59	24.30	9.27%
w/o Spa	15.93	27.67	15.78%	21.10	34.59	14.21%	21.66	36.81	8.93%	14.51	25.36	9.18%
w/o LT	14.84	26.42	14.96%	18.37	30.27	12.34%	19.81	33.73	8.30%	13.31	23.27	8.79%
w/ LTL1	14.82	26.39	14.93%	18.31	30.22	12.27%	19.75	33.68	8.28%	13.24	23.12	8.70%
STWave$^+$	**14.71**	**26.31**	**14.88**%	**18.25**	**30.14**	**12.21**%	**19.59**	**33.53**	**8.17**%	**13.21**	**23.04**	**8.63**%

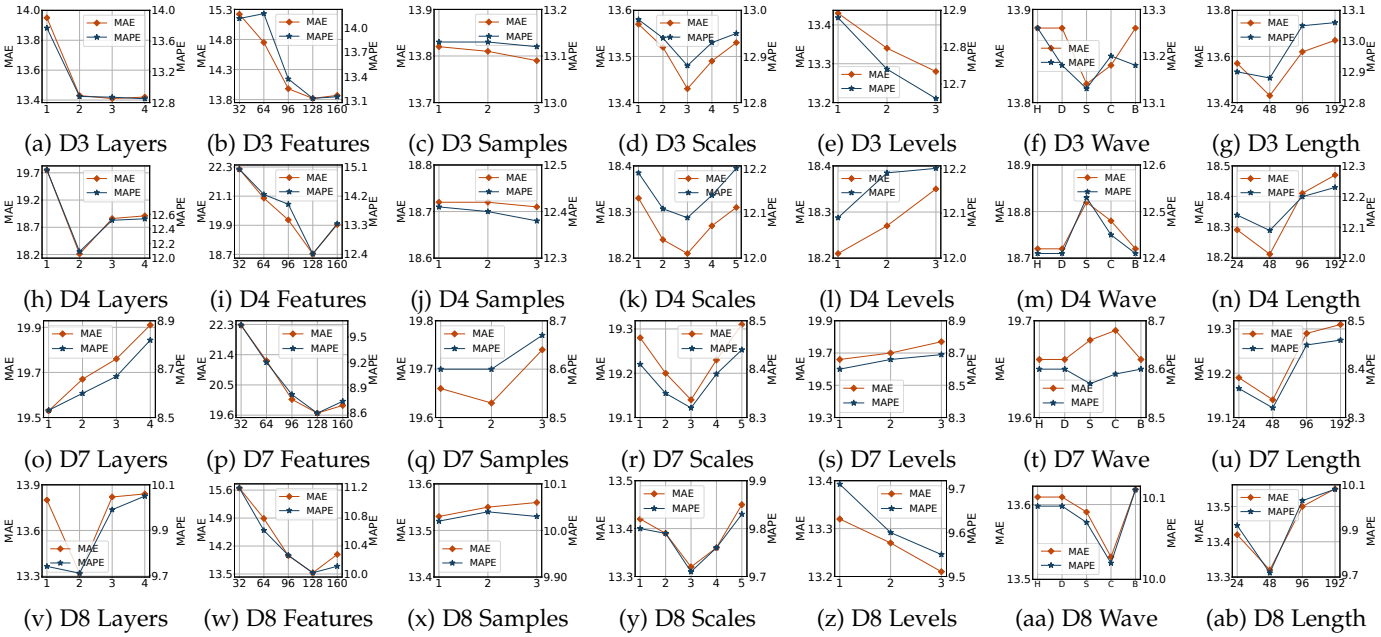


Fig. 9: Hyper-parameter study on all datasets. PeMSD is abbreviated as D.

to capture the spatial correlations.

- "w/ LTL1": STWave$^+$ replaces the contrastive loss with a knowledge distillation loss, *i.e*, the L1 loss.
- "w/o LT": STWave$^+$ without the long-term branch, *i.e.*, the model is not guided by long-term knowledge and stable trends and may be affected by the short-term counter-trends and the events.

Table 4 shows the comparison results on all datasets. It is clear that the original STWave$^+$ can achieve the best performance compared to its variants. Generally, the results worse of "w/o Spa" far outperforms that of "w/o Tem" on most tasks, indicating that the spatial dimension plays a more vital role than the temporal dimension in multivariate traffic forecasting tasks. We also observe that "w/o DF" performs worse than STWave$^+$ because it ignores disentangling the independent components in the traffic time series and may faces over-fitting. Moreover, "w/o AF" underperform STWave$^+$, indicating the advantages of selecting helpful event information and removing incorrect events. Finally, we can see that the performance of "w/o LT" and "w/ LTL1" is lower compared with the original model, which denotes the benefits of the long-term branch and the contrastive loss, *i.e.*, the overall trends and the non-alignment time series representations that are pushed away are useful. In conclusion, STWave$^+$ benefits from the exquisitely-devised components and framework.

5.4 Parameter Sensitivity Analysis (RQ3)

Figure 9 depicts the results of hyper-parameter sensitivity on all datasets. We search the layers of our dual-channel encoder, the number of features in STWave$^+$, the sampling factor of MS-ESGAT, and the scales of MS-ESGAT from a search space of $[1, 2, 3, 4]$, $[32, 64, 96, 128, 160]$, $[1, 2, 3]$, and $[2, 3, 4, 5]$. First, the performance of our model improves as the layers of our dual-channel encoder increase and tends to be stable when there are 2 layers. Second, when the number of features is 128, our model can achieve the best performance. Obviously, increasing the neural network size can improve representation ability, but too many features may introduce noise in learned representations and result in sub-optimal performance. Third, the general performance increases a little with the increase of sampled sensors. It verifies our query sparsity assumption that sensors in the same region have the same traffic and a few active sensors can on behalf of all regions. Besides, we can observe that STWave$^+$ performs the best with 3 scales of the graph wavelet. This is because as the adaptive scale increases, model optimization is more difficult and thus decreasing the forecasting performance.

Moreover, we search the level of DWT, the wavelet of DWT, and the input length of long-

TABLE 5: Computation needs comparison on PeMSD7.

Methods	Memory Size (Training / Inference)	Time (Training / Inference)	MAE
LSGCN	56723M / 56723M	574s / 106s	27.31
AGCRN	34281M / 34281M	207s / 32s	22.37
STFGNN	34111M / 34111M	481s / 91s	23.46
STGODE	46951M / 46951M	224s / 34s	22.59
STwave	30587M / 30587M	240s / 40s	19.94
Full	65959M / 44835M	370s / 45s	19.54
w/o GPE	49839M / 30211M	337s / 39s	20.37
EV	50161M / 30586M	338s / 40s	19.96
w/o MS	50163M / 30587M	338s / 40s	19.71
STWave$^+$	50169M / 30590M	338s / 40s	19.59

(a) STWave$^+$ (b) EV (c) w/o MS

Fig. 10: Visualization for the most important neighbors of STWave$^+$ and its variants. Red sensor: central node, blue sensors: most important neighbors.

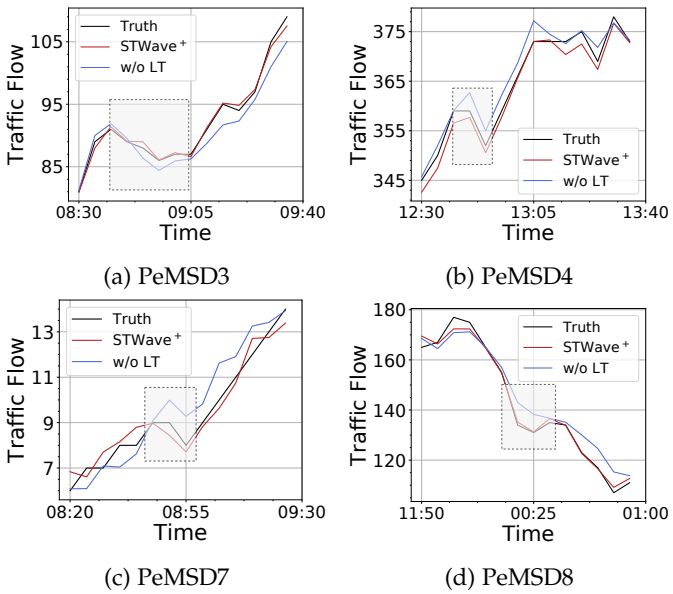
term branch from a search space of $[1, 2, 3]$, [Haar,Daubechies,Symlets,Coiflets,Biorthogonal] (abbreviated as [H,D,S,C,B]), and $[24, 48, 96, 192]$. For the level of DWT, STWave$^+$ with one level of DWT can achieve the best performance on PeMSD4 and PeMSD7 datasets. Although other datasets need more levels to obtain stable trends, we only use one-level DWT in our model for the trade-off between computation needs and performance. For the wavelet of DWT, different wavelet functions have different disentangle performances, in which Daubechies, Symlets, and Coiflets are respectively applicable to different traffic datasets. For the input length of the long-term branch, the performance is first upward and the downward with increasing input length. This phenomenon points out the positive effect of integrating overall temporal changes into traffic forecasting and the too-long historical input will introduce negative information.

5.5 MS-ESGAT Study (RQ4)

To display the effectiveness and efficiency of MS-ESGAT, we show the performance of STWave$^+$, LSGCN, STGODE, STFGNN, and one variant of STWave$^+$.

- "Full": STWave$^+$ without the query sampling strategy in MS-ESGAT, *i.e.*, calculates all spatial correlations.

Table 5 shows the forecasting performance, training speed, inference speed, and memory usage on the large-scale graph-based dataset PeMSD7 with the same feature dimension. While "Full" performs well, its speed is slow and memory usage is large due to the quadratic complexity. On the other hand, AGCRN and STGODE are fast and slow at the cost of lower quantitative performance due to the one-layer and multi-layer GCN. The attention-based LSGCN is the worst in all aspects because it calculates spatial correlations between all sensors and mines temporal information insufficiently. Among these models, our MS-ESGAT makes a better trade-off in terms of speed and performance, while having reasonable memory usage.



(a) PeMSD3 (b) PeMSD4

(c) PeMSD7 (d) PeMSD8

Fig. 11: Long-term study on all datasets.

To show the usefulness of multi-scale graph wavelet positional encoding, we propose three variants of our model:

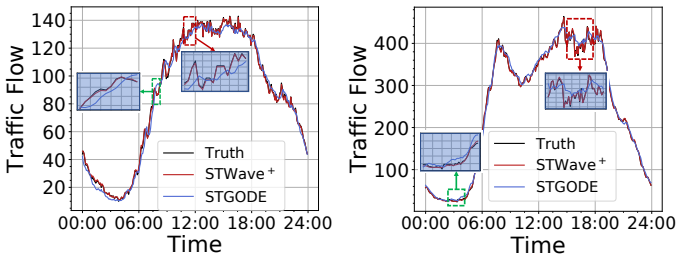
- "w/o GPE": It no longer uses graph positional encoding (GPE).
- "EV": It uses graph Laplacian eigenvectors as the GPE.
- "w/o MS": The GPE is single-scale graph wavelet.

as shown in Table 5, "w/o GPE" performs worst because it lacks the soft inductive bias, *i.e.*, graph structure information. The reason why "EV" and "w/o MS" perform worse than graph wavelet is that they fail to balance the global graph property and multi-scale local information. The most important neighbors of STWave$^+$, "EV", and "w/o MS" are visualized in Figure 10, we can observe sensors of "EV" is more sparse on the road network and "w/o MS" fails to focus multi-scale local neighbors, *i.e.*, "EV" is insensitive in the local and "w/o MS" neglects a lot essential correlations at different scales, demonstrating the equilibrium of our graph wavelet positional encoding.

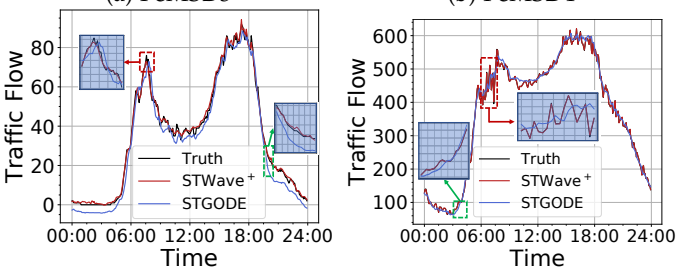
5.6 Visualization Study (RQ5)

5.6.1 Long-Term Study

To verify the effectiveness of our long-term self-supervised learning, we visualize some predicted curves of STWave$^+$ and STWave that contain the short-term counter-trend (*i.e.*, the shaded curve) in Figure 11. As shown in the figure, we can observe that the predicted results of STWave deviate more from the ground truth after the short-term counter-trend compared with STWave$^+$. This is because STWave$^+$ can gain overall temporal changes during the training stage and this knowledge teaches the model to avoid the influence of short-term counter-trends. Besides, as shown in Table 5, our model not only receives the overall temporal changes but also does not decrease inference speed.



(a) PeMSD3 (b) PeMSD4



(c) PeMSD7 (d) PeMSD8

Fig. 12: Case study on all datasets.

5.6.2 Case Study

To show our framework that disentangling traffic into trends and events can make reasonable results, we conduct a case study on all datasets, *i.e.*, we visualize some predicted curves of traffic time series and correspond ground truth in Figure 12. As shown in Figure 12, the forecast curves of the stable trends (*e.g.*, red rectangles) of our model are more precise than that of STGODE because STWave$^+$ disentangles the traffic into different components and the easy to predict trends are not disturbed by the fluctuating events. Particularly, our model substantially exceeds STGODE for the fluctuation time slices (*e.g.*, green rectangles) because it obtains useful information from the predicted events by using the adaptive event fusion module.

6 Conclusion

In this paper, we propose a novel disentangle-fusion framework for traffic forecasting, namely STWave$^+$, which does not follow the paradigm of modeling the intricate traffic end-to-end. Specifically, STWave$^+$ first disentangles the traffic time series into trends and events through DWT, whereby a dual-channel spatio-temporal encoder is proposed to capture the stable temporal changes, fluctuate temporal changes, and spatial correlations under different temporal environments by the causal convolution, temporal attention, and our MS-ESGAT. Furthermore, with the MS-ESGAT, STWave$^+$ extracts global dynamic correlations efficiently and effectively. Finally, STWave$^+$ utilizes the adaptive event fusion to predict traffic. Besides, a long-term historical self-supervised signal is used to improve the robust of our model. Performance on six traffic datasets demonstrates the superiority of STWave$^+$ over baselines. Henceforth, we will focus on learning discrete wavelet functions through back-propagation to reduce hyperparameters and thus make STWave$^+$ easier to tune. Additionally, compressing and reducing the memory requirements and time consumption caused by self-supervised signal in the training phase will be the most important research direction of STWave$^+$.

Acknowledgments

This work is partially supported by NSFC (No. 61972069, 61836007, 61832017, 62272086), Shenzhen Municipal Science and Technology R&D Funding Basic Research Program (JCYJ20210324133607021), Municipal Government of Quzhou under Grant No. 2022D037, and Key Laboratory of Data Intelligence and Cognitive Computing, Longhua District, Shenzhen.

References

[1] R.-G. Cirstea, T. Kieu, C. Guo, B. Yang, and S. J. Pan, "Enhancenet: Plugin neural networks for enhancing correlated time series forecasting," in *Proceedings of ICDE*, 2021.
[2] H. Liu, C. Jin, B. Yang, and A. Zhou, "Finding top-k optimal sequenced routes," in *Proceedings of ICDE*, 2018.
[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proceedings of ICLR*, 2018.
[4] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of IJCAI*, 2018.
[5] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of AAAI*, 2021.
[6] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ode networks for traffic flow forecasting," in *Proceedings of SIGKDD*, 2021.
[7] Y. Fang, F. Zhao, Y. Qin, H. Luo, and C. Wang, "Learning all dynamics: Traffic forecasting via locality-aware spatio-temporal joint transformer," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 433–23 446, 2022.
[8] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *Proceedings of ICLR*, 2022.
[9] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of IJCAI*, 2019.
[10] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proceedings of NeurIPS*, 2020.
[11] X. Zhang, C. Huang, Y. Xu, and L. Xia, "Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting," in *Proceedings of CIKM*, 2020.
[12] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "Lsgcn: Long short-term traffic prediction with graph convolutional networks." in *Proceedings of IJCAI*, 2020.
[13] C. Park, C. Lee, H. Bahng, Y. Tae, S. Jin, K. Kim, S. Ko, and J. Choo, "St-grat: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proceedings of CIKM*, 2020.
[14] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of AAAI*, 2020.
[15] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" in *Proceedings of NeurIPS*, 2021.
[16] J. Han, H. Liu, H. Xiong, and J. Yang, "Semi-supervised air quality forecasting via self-supervised hierarchical graph neural network," *IEEE Transactions on Knowledge and Data Engineering*, 2022.
[17] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph wavelet neural network," in *Proceedings of ICLR*, 2019.
[18] Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, L. Zeng, and C. Wang, "When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks," in *Proceedings of ICDE*, 2023, pp. 515–527.
[19] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical graph convolution networks for traffic forecasting," in *Proceedings of AAAI*, 2021.

[20] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.
[21] Z. Lu, C. Zhou, J. Wu, H. Jiang, and S. Cui, "Integrating granger causality and vector auto-regression for traffic prediction of large-scale wlans," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 10, no. 1, pp. 136–151, 2016.
[22] 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.
[23] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.
[24] J. Van Lint and C. Van Hinsbergen, "Short-term traffic and travel time prediction models," *Artificial Intelligence Applications to Critical Transportation Issues*, vol. 22, no. 1, pp. 22–41, 2012.
[25] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of ICML*, 2020.
[26] 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 AAAI*, 2019.
[27] A. Feng and L. Tassiulas, "Adaptive graph spatial-temporal transformer network for traffic forecasting," in *Proceedings of CIKM*, 2022, pp. 3933–3937.
[28] S. Dunne and B. Ghosh, "Weather adaptive traffic prediction using neurowavelet models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 370–379, 2013.
[29] R. Liu, Y. Wang, H. Zhou, and Z. Qian, "Short-term passenger flow prediction based on wavelet transform and kernel extreme learning machine," *Ieee Access*, vol. 7, pp. 158 025–158 034, 2019.
[30] A. Khazaee Poul, M. Shourian, and H. Ebrahimi, "A comparative study of mlr, knn, ann and anfis models with wavelet transform in monthly stream flow prediction," *Water Resources Management*, vol. 33, pp. 2907–2923, 2019.
[31] H. Li, Z. Lv, J. Li, Z. Xu, Y. Wang, H. Sun, and Z. Sheng, "Traffic flow forecasting in the covid-19: A deep spatial-temporal model based on discrete wavelet transformation," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 5, pp. 1–28, 2023.
[32] Z. Cui, R. Ke, Z. Pu, X. Ma, and Y. Wang, "Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction," *Transportation Research Part C: Emerging Technologies*, vol. 115, p. 102620, 2020.
[33] Z. Li, W. Li, and K. Hwang, "Adaptive graph convolution networks for traffic flow forecasting," *arXiv preprint arXiv:2307.05517*, 2023.
[34] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
[35] J. Qiu, S. R. Jammalamadaka, and N. Ning, "Multivariate bayesian structural time series model." *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 2744–2776, 2018.
[36] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.
[37] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.
[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NeurIPS*, 2017.
[39] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *arXiv preprint arXiv:2012.09699*, 2020.
[40] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
[41] M. Behmanesh, P. Adibi, S. M. S. Ehsani, and J. Chanussot, "Geometric multimodal deep learning with multiscaled graph wavelet convolutional network," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[42] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proceedings of Neurips*, 2020, pp. 5812–5823.
[43] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.
[44] S. Elmi, "Deep stacked residual neural network and bidirectional lstm for speed prediction on real-life traffic data," in *Proceedings of ECAI*, 2020.
[45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of NeurIPS*, 2014.
[46] 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 AAAI*, 2020.

Yuchen Fang is a research assistant at University of Electronic Science and Technology of China. His general research interests are in spatio-temporal data mining, graph neural networks, and urban computing, with a special focus on traffic forecasting. He has published several papers in top journals and conference proceedings, such as ICDE, SIGIR, AAAI, and TITS.

Yanjun Qin is currently a post-doctor at the Department of Electronic Engineering, Tsinghua University. She received the Ph.D. degree with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, China. Her current main interests include autonomous Driving, smart Transportation, and Spatio-Temporal data mining.

Haiyong Luo received the Ph.D. degree in computer science from the University of Chines Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with the Institute of Computer Technology, Chinese Academy of Science, Beijing, China. His main research interests are location based services, pervasive computing, mobile computing, and Internet of Things.

Fang Zhao received the Ph.D. degrees in computer science and technology from the Beijing University of Posts and Telecommunication, Beijing, China, in 2009. She is currently a Professor with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include mobile computing, location based services, and computer networks.

Kai Zheng is a Professor of Computer Science with University of Electronic Science and Technology of China. He received his PhD degree in Computer Science from The University of Queensland in 2012. He has been working in the area of spatial-temporal databases, uncertain databases, social-media analysis, inmemory computing and blockchain technologies. He has published over 100 papers in prestigious journals and conferences in data management field such as SIGMOD, ICDE, VLDB Journal, ACM Transactions and IEEE Transactions. He is a senior member of IEEE.