2023 IEEE 39th International Conference on Data Engineering (ICDE) | 979-8-3503-2227-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDE55515.2023.00046

When Spatio-Temporal Meet Wavelets: Disentangled Traffic Forecasting via Efficient Spectral Graph Attention Networks

Yuchen Fang[1], Yanjun Qin[1], Haiyong Luo[2†],
Fang Zhao[1†], Bingbing Xu[2], Liang Zeng[3], Chenxing Wang[1]
[1]Beijing University of Posts and Telecommunications, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, China
[3]Tsinghua University, China
{fangyuchen, qinyanjun, zfsse, wangchenxing}@bupt.edu.cn,
{yhluo, xubingbing}@ict.ac.cn, zengl18@mails.tsinghua.edu.cn

***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 of the traffic data. To capture these intricate dependencies, spatio-temporal networks, such as recurrent neural networks with graph convolution networks, graph convolution networks with temporal convolution networks, and temporal attention networks with full graph attention networks, are applied. However, 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. On the other hand, the efficient and effective algorithm for modeling spatial correlations is still lacking in prior networks.**

In this paper, rather than proposing yet another end-to-end model, we aim to provide a novel disentangle-fusion framework STWave to mitigate the distribution shift issue. The framework first decouples the complex traffic data into stable trends and fluctuating events, followed by a dual-channel spatio-temporal network to model trends and events, respectively. Finally, reasonable future traffic can be predicted through the fusion of trends and events. Besides, we incorporate a novel query sampling strategy and graph wavelet-based graph positional encoding into the full graph attention network to efficiently and effectively model dynamic spatial correlations. Extensive experiments on six traffic datasets show the superiority of our approach, *i.e.*, the higher forecasting accuracy with lower computational cost.

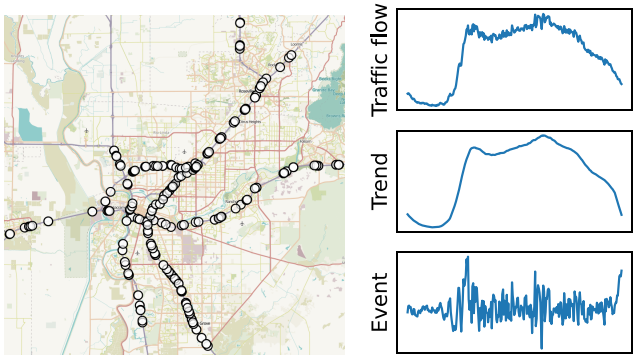
***Index Terms*—traffic forecasting, spatio-temporal data, graph attention network**

I. 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.

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

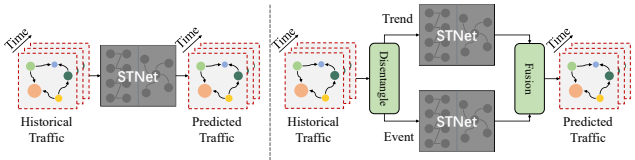


(a) Sensor locations (b) Time series

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

†Corresponding author.

2375-026X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDE55515.2023.00046



(a) End-to-end framework (b) Disentangle-fusion framework

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

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, the graph-based deep learning methods have recently been adopted for capturing spatial correlations in traffic forecasting, such as graph convolutional networks (GCN) [9] based methods [3]–[6], [10], [11], graph attention networks (GAT) [12] based methods [13], [14], and full GAT (*i.e.*, considering relationships between all sensors) based methods [15], [16]. Although the full GAT-based methods can dynamically capture the global spatial information and have shown state-of-the-art performance in various tasks [17], [18], 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 [19].

Motivated by the above analysis, in this paper, rather than proposing yet another end-to-end STNet, we aim to provide a novel disentangle-fusion framework to mitigate the distribution shift of the traffic time series. 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 propose a novel traffic forecasting framework named STWave, which first applies the discrete wavelet transform (DWT) to disentangle the traffic time series into the dual-scale trend-event representations because DWT can decompose data into various components, such as the main information (*i.e.*, trend information) and details (*i.e.*, event information) of the data [20]. Then STWave designs a STNet that utilizes corresponding sequential and graph-based methods on the different information to capture the various temporal changes and spatial correlations. Specifically, the causal convolution with small kernel size, temporal attention with the global temporal receptive field, and the state-of-the-art full GAT are adopted on events, trends, and both of them to capture fluctuating temporal changes, stable temporal changes, and different temporal changes-based dynamic global spatial correlations, respectively. Moreover, to address the high complexity and insufficient structure information in the full GAT, a novel query sampling strategy and a novel graph wavelet-based graph positional encoding are proposed in STWave. The query sampling strategy reduces the complexity while maintaining the global receptive field according to the hierarchical nature of the traffic system [21], and the graph wavelet-based graph positional encoding brings the local-global balanced structure information based on the spectral graph theory [22]. 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 state-of-the-arts and demonstrate the validity of each proposed component.

We summarize key contributions of this paper as follows:

- To the best of our knowledge, STWave is the first study that proposes a framework to learn disentangled representations of traffic time series by using the discrete wavelet transform, which is more suitable for non-stationary traffic compared with end-to-end models.
- We design an appropriate STNet for STWave, which utilizes causal convolution, temporal attention, and full GAT to capture the correct spatio-temporal information of traffic. Moreover, the novel query sampling strategy and graph wavelet-based graph positional encoding are innovatively adopted in the STNet to reduce complexity and improve the structure-aware ability of the full GAT.
- STWave outperforms the existing 15 traffic forecasting approaches by a considerable margin with high efficiency on six real-world datasets. Furthermore, various experiments are conducted to demonstrate the effectiveness of our framework and STNet.

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. Section IV, Section V, and Section VI details method, introduces experiments, and concludes the paper, respectively.

II. Related Work

A. Traffic Forecasting

Researchers utilized statistical methods to forecast traffic in the early year, *e.g.*, Historical Average [23], Vector AutoRegression [24], and AutoRegressive Integrated Moving Average [25], yet these methods rely on linear assumptions and thus fail to extract non-linear correlations of the traffic data.

[26], [27] 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 [28] in some research areas [29], [30], a line of traffic forecasting methods modeled temporal patterns in the traffic data for each sensor individually, such as LSTM [31], TCN [32], and Transformer [33]. However, they ignored the intricate spatial correlations between sensors on the traffic road network, *e.g.*, 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 [10] and AGCRN [11] replaced the pre-defined graph with the adaptive graph in GCNs to capture global and accurate spatial dependencies in the traffic data through back-propagation. At the same time, they lost the guidance of prior knowledge and were easy to under- or over-fitting [34]. 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 [35] in the deep GCN. Despite some methods proposed general approaches such as plugin networks [1] and the covariance loss [36] to improve the performance of GCN-based models. However, most GCN-based methods ignore that correlations between sensors on the road network are constantly changing over time.

B. Graph Attention Network for Traffic Forecasting

To extract time-varying spatial correlations between traffic time series recorded by sensors, ST-CGA [13] utilized the graph attention network (GAT) to capture dependencies between neighbor sensors for each time slice individually. ASTGCN [37] 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 [14] 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 [15] 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 [16] paralleled the full GAT and temporal attention but without the cross-attention. Particularly, ST-GRAT, CDGNet [38], and ASTTN [39] utilized the LINE [40], Node2Vec [41], 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.

TABLE I: Notations and explanations.

Notations	Explanations
$\mathcal{X}$	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	score matrix
P	a trainable projector
S, idx	the number and index of sampled queries
Φ, λ	eigenvector and eigenvalue of the graph Laplacian
ψ, G	graph wavelet and scaling matrix
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	the input and output length of traffic
Θ, θ	learnable parameters of STWave and causal convolution
W, b	learnable parameters of projection

III. Preliminaries

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 $\mathcal{G}$.

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 $\mathcal{X} = \{X_1, ..., X_{T_1}\} \in \mathbb{R}^{T_1 \times N \times C}$ 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) System Overview: Figure 3 elaborates the framework of our STWave, which consists of the following three important components:

- Disentangling flow layer: Given the historical traffic data of all sensors, STWave first utilizes the multi-level discrete wavelet transform (DWT) to separate the entangled historical time series of all sensors into a low-frequency

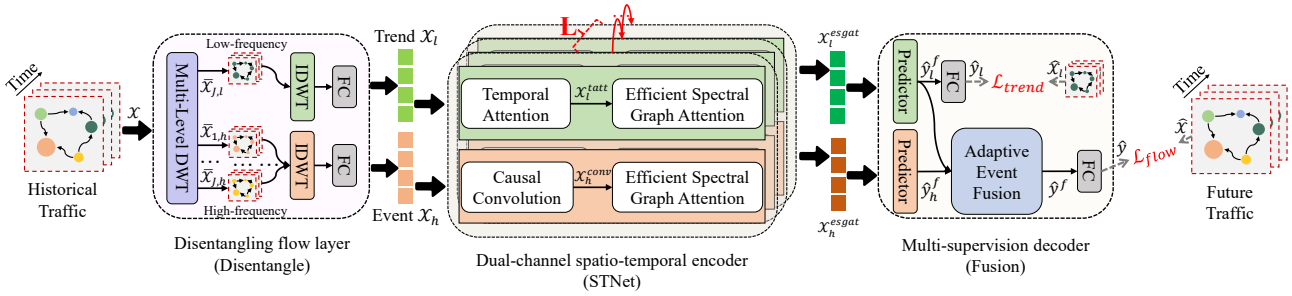


Fig. 3: The architecture of the proposed STWave. FC: fully-connected layer, DWT: discrete wavelet transform.

component and multi-high-frequency components, which can avoid the interference between low-frequency and high-frequency components. To consist of the input dimension and enhance the representation power, STWave chronologically adopts the inverse DWT (IDWT) and fully-connected layer on disentangled 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. For dynamic spatial dependency learning in the spatio-temporal traffic data, STWave uses two efficient spectral graph attention networks on trends and events to effectively and efficiently reveal the time-varying correlations between sensors under different temporal information.
- Multi-supervision 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. Different from only supervising the traffic flow or speed in the literature, we add an auxiliary loss on the stable trends to handle the distribution shift in events.

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 I.

IV. Methodology

A. Disentangling Flow Layer

As mentioned in [42], the excellent representation of the intricate data made up of multiple sources should be disentangled into diverse explanatory sources, enhancing the robustness of the model on richly structured variations. Inspired by Bayesian Structural Time Series models [43] and the independent mechanisms assumption [44], 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 [20], *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,

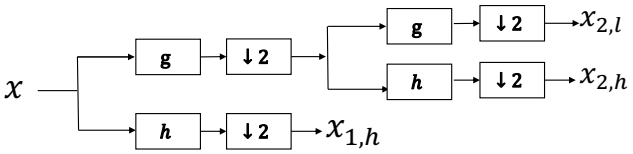


Fig. 4: Example of two-level DWT.

and parallel processing of all high-frequency components will introduce more computation needs. Then we utilize the fully-connected neural network to transform the 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:

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

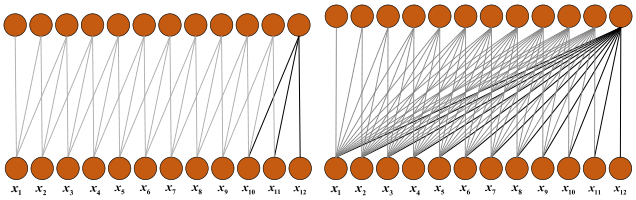
where $W^g, W^h \in \mathbb{R}^{C \times d}$ and $b^g, b^h \in \mathbb{R}^d$ are learnable parameters. After the disentangling flow layer, we obtain the disentangled trend-event representations of traffic data, they can be parallel processed in the next.

B. 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.

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 input $x \in \mathbb{R}^T$ with a filter $f \in \mathbb{R}^K$, the causal convolution operation of x with f at time step t can be formulated as:

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



(a) Causal convolution (b) Temporal attention

Fig. 5: Example of causal convolution and temporal attention.

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 of projections, $\alpha_{t,i}^n$ denotes the correlation between trends at time slice t and i, and $exp(\cdot)$ denotes the exponential function.

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

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.

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

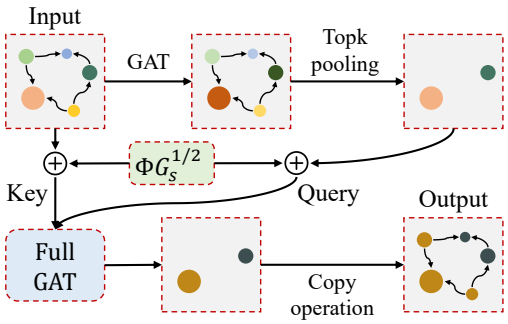


Fig. 6: An illustration of the proposed ESGAT.

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 an ESGAT with a novel query sampling strategy and graph positional encoding. The architecture of ESGAT is shown in Figure 6.

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 [45]. 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 S active sensors that receive max information from other sensors through the GAT as sparse queries. The number of S is controlled by a constant sampling factor e, we set $S = \lceil elogN \rceil$, 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:

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

where $rank(\cdot)$ returns the index of the top S largest values, and $idx_t \in \mathbb{R}^S$ indicates the index of sampled queries at time slice t. Finally, the full GAT on the sampled sensors and the copy operation on the 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 [46], 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, [47] 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 [22]. Different from the graph Laplacian eigenvectors, the graph wavelet [48] corresponds to graph Laplacian eigenvectors diffused away from a central node with a scaling matrix on the graph and can reflect the localization 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 can show not only the structure information but also the localization property of graphs. Furthermore, we set the scale s as the learnable parameter to avoid misleading inductive bias through back-propagation inspired by the adaptive graph learning technology in previous works [10], [11]. Finally, our efficient spectral graph attention can be formulated as follows:

$$\begin{aligned} \tilde{x}_t^i &= x_t^i + \rho^i, \text{ where } i \in [1, ..., N] \\ x_t^{esgat^n} &= \sum_{i=1}^{N} \beta_t^{n,i}(W^{V_S}\tilde{x}_t^i), \text{ where } n \in idx_t \\ \beta_t^{n,i} &= \frac{exp((W^{Q_S}\tilde{x}_t^n)^T(W^{K_S}\tilde{x}_t^i))}{\sum_{k=1}^{N} exp((W^{Q_S}\tilde{x}_t^n)^T(W^{K_S}\tilde{x}_t^k))} \\ x_t^{esgat^n} &= x_t^{esgat^c},\ c = rank(\beta_t^{:,n}, 1), \text{ where } n \notin idx_t. \end{aligned} \tag{12}$$

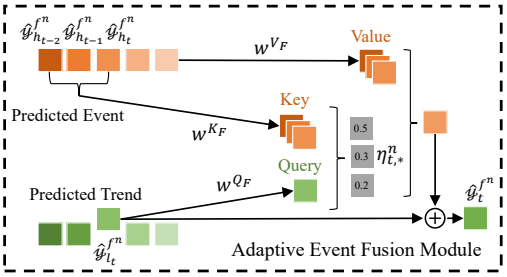


Fig. 7: An illustration of the proposed adaptive event fusion.

After extracting spatial correlations, we obtain learned representations $\mathcal{X}_h^{esgat}, \mathcal{X}_l^{esgat} \in \mathbb{R}^{T_1 \times N \times d}$ of trends and events.

C. Multi-Supervision Decoder

1) Disentangled Feature Prediction: 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 the trends and events. Then we utilize a fully-connected layer to project trends into the 1D value $\hat{\mathcal{Y}}_l \in \mathbb{R}^{T_2 \times N \times C}$ and supervise it in our model to gain knowledge from the much stable temporal changes, thus our model can effectively handle the distribution shift in the events and yielding better performance. The supervision is implemented by the $L1$ loss:

$$\mathcal{L}_{trend} = \sum_{t=T_1+1}^{T_1+T_2} \sum_{n=1}^{N} |x_{l_t}^n - \hat{y}_{l_t}^n|. \tag{13}$$

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.

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:

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

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 $t \leftarrow 1$ **to** $T - T_1 - T_2$ **do**
  Append $\mathcal{T}_{t:t+T_1}$ to input sample set $\mathcal{X}$;
  Append $\mathcal{T}_{t+T_1:t+T_2}$ 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 label sample $\mathcal{Y}_{bs}$;
  Use DWT to calculate the trend $\mathcal{X}_{l_{bs}}$ and event $\mathcal{X}_{h_{bs}}$ of the input sample;
  Feed $\mathcal{X}_{l_{bs}}$, $\mathcal{X}_{h_{bs}}$, Φ, and λ into STWave;
  Optimize Θ by minimizing the objective function;
until *met model stop criteria*;
Result: Learned STWave model.

D. Objective Function

Therefore, by considering the traffic forecasting loss and trend supervision loss, STWave aims to jointly minimize the following objective function:

$$\mathcal{L} = \mathcal{L}_{flow} + \mathcal{L}_{trend}, \tag{16}$$

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

E. Complexity Analysis

The complexity of our spatio-temporal encoder is $O(L(TNK + NT^2 + TNlogN))$, where causal convolution, temporal attention, ESGAT cost $O(TNK)$, $O(NT^2)$, $O(TNlogN)$ complexity, and L represents the number of stacked layers. The complexity of the disentangling flow layer and the decoder is $O(NT)$ and $O(NT^2)$, respectively. Besides, despite the complexity of calculating eigenvectors and eigenvalues of graph Laplacian is $O(N^3)$, it can be quickly preprocessed before training without affecting the model complexity. Therefore, STWave achieves comparable time and memory complexity during the training phase compared to GCN-based models.

V. Experiments

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

- **RQ1:** Does our STWave outperform baselines?

TABLE II: Dataset statistics.

Datasets	#Nodes	#Samples	Time Range	Type
PeMSD3	358	26208	09/2018-11/2018	Traffic flow
PeMSD4	307	16992	01/2018-02/2018	Traffic flow
PeMSD7	883	28224	05/2017-08/2017	Traffic flow
PeMSD8	170	17856	07/2016-08/2016	Traffic flow
PeMSD7(M)	228	12672	05/2012-06/2012	Traffic speed
PeMSD7(L)	1026	12672	05/2012-06/2012	Traffic speed

- **RQ2:** How do our framework and different components in STWave (*e.g.*, ESGAT) affect model performance?
- **RQ3:** How do hyper-parameters affect STWave?
- **RQ4:** Does our ESGAT efficient and effective?
- **RQ5:** Can STWave provide reasonable traffic forecasting results?

A. Experimental Setup

1) Datasets: We evaluate our model on six real-world datasets collected from the California Transportation Agencies (CalTrans) Performance Measurement System (PeMS), named PeMSD3, PeMSD4, PeMSD7, PeMSD8, PeMSD7(M), and PeMSD7(L). 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 II. Following [37], we use the observations traffic from the previous 12 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.

2) Metrics: In this paper, three widely used metrics are used for all tasks, namely, Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE), and Root Mean Squared Errors (RMSE).

3) Baselines: We compare our proposed STWave with the following 15 baseline models:

- HA [23]: It utilizes average value of history to iterative predict the future.
- ARIMA [25]: It integrates moving average into the Autoregressive model.
- VAR [24]: It is a statistical model that can capture spatial dependencies.
- SVR [26]: It utilizes the support vector machine to perform traffic forecasting.
- LSTM [31]: It is a advanced version of the RNN with the long-term memory.
- TCN [32]: 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 [10]: It combines the gated TCN and the adaptive graph-based GCN to capture spatio-temporal dependencies simultaneously.
- ASTGCN [37]: It performs the attention mechanism on the temporal and spatial convolutions to extract dynamic spatio-temporal correlations.
- LSGCN [14]: 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 [49]: It uses a spatio-temporal synchronous technology to extract the local spatio-temporal correlations.
- AGCRN [11]: 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.

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 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 ESGAT is set to 1, and the number of layers L in spatio-temporal encoder is set as 2. Besides, we set different discrete wavelet for different dataset: Symlets wavelet for PeMSD3 and PeMSD7(L), Daubechies wavelet for PeMSD4 and PeMSD7, Coiflets wavelet for PeMSD8 and PeMSD7(M).

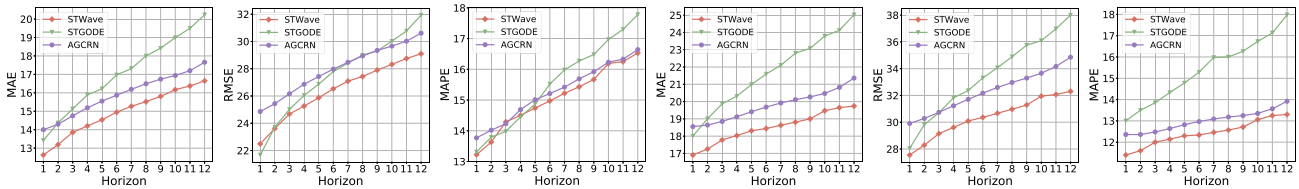
5) Implementation Details: We implement STWave on Python 3.8.10 using PyTorch 1.9.1. All experiments are conducted on a machine, running Ubuntu 20.04.3 LTS, with one Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz and one Tesla A100 GPU card. The source code of STWave is available here: https://github.com/LMissher/STWave.

B. Performance Comparison (RQ1)

The results under three metrics of STWave and baselines across six datasets are reported in Table III and Table V, and results under three metrics for each time slice are shown in Figure 8 and Figure 9. From the Tables, 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 cannot 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

TABLE III: 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%
LSGCN†	16.76	28.41	16.28%	20.50	32.59	13.56%	26.21	40.38	10.89%	16.82	25.99	10.52%
AGCRN†	15.24	27.19	15.01%	18.95	31.00	12.64%	21.40	35.48	8.81%	15.00	24.72	9.80%
STGODE†	15.71	27.30	15.84%	20.01	31.99	13.54%	21.63	36.39	9.52%	15.91	25.17	10.23%
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%**



(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.

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 best performance on all datasets. There are three main reasons: 1) STWave disentangles the trend and the event from the traffic time series and proposes a spatio-temporal encoder to process each term individually; 2) Our model designs the adaptive fusion module and the multi-supervision function to fully incorporate and exploit information of trends and events; and 3) STWave proposes a novel graph wavelet positional encoding to effectively reveal global-local balanced spatial dependencies by injecting graph structure information. Besides, as shown in the Figures, 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.

C. Ablation Study (RQ2)

In order to verify the effectiveness of our proposed disentangle-fusion framework for traffic forecasting, we replace our STNet with LSGCN, AGCRN, and STGODE to form three variants LSGCN†, AGCRN†, and STGODE†. The experimental results of these variants are shown in Table III and V. Compared with the end-to-end manner, using our proposed framework achieves better results on all tasks, because our framework can effectively mitigate the terrible influence introduced by the distribution shift of events. Moreover, their performance worse than our STWave indicates that our STNet is excellent for traffic forecasting.

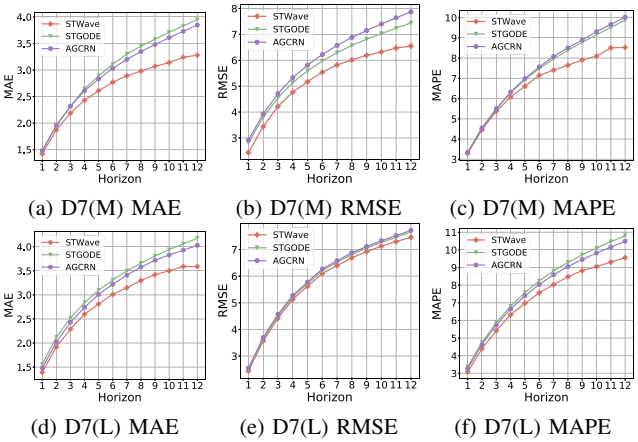
To investigate the effectiveness of different components in STWave, we compare STWave with five different variants:

TABLE IV: 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	15.56	27.18	15.78%	19.44	31.84	12.98%	20.22	34.24	8.59%	13.86	24.29	9.25%
w/o AF	15.49	26.68	15.64%	19.63	31.79	13.35%	21.61	35.87	9.08%	14.48	24.55	9.46%
w/o MS	15.26	27.22	15.68%	19.22	31.28	12.92%	20.35	34.61	8.51%	14.04	24.70	9.23%
w/o Tem	15.91	28.53	16.15%	19.70	31.93	13.50%	20.98	35.03	8.85%	13.86	24.72	9.40%
w/o Spa	16.28	28.02	16.05%	21.44	34.83	14.59%	21.93	37.08	9.07%	14.89	25.97	9.60%
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%**

TABLE V: Comparison of STWave and baselines on two traffic speed datasets. **Bold**: Best, underline: Second best.

Methods	PeMSD7(M)			PeMSD7(L)		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	4.59	8.63	14.35%	4.84	9.03	14.90%
ARIMA	7.27	13.20	15.38%	7.51	12.39	15.83%
VAR	4.25	7.61	10.28%	4.45	8.09	11.62%
SVR	4.09	7.47	10.03%	4.41	8.11	11.58%
LSTM	4.16	7.51	10.10%	4.66	8.20	11.69%
TCN	4.36	7.20	9.71%	4.05	7.29	10.43%
STGCN	3.86	6.79	10.06%	3.89	6.83	10.09%
DCRNN	3.83	7.18	9.81%	4.33	8.33	11.41%
GWN	3.19	6.24	8.02%	3.75	7.09	9.41%
ASTGCN(r)	3.14	6.18	8.12%	3.51	6.81	9.24%
LSGCN	3.05	5.98	7.62%	3.49	6.55	8.77%
STSGCN	3.01	5.93	7.55%	3.61	6.88	9.13%
AGCRN	2.99	5.84	7.42%	3.13	6.04	7.75%
STFGNN	2.93	5.74	7.28%	3.07	5.96	7.71%
STGODE	2.97	5.66	7.36%	3.22	5.98	7.94%
LSGCN†	2.92	5.71	7.45%	3.25	6.01	8.07%
AGCRN†	2.78	5.59	7.08%	2.96	5.93	7.41%
STGODE†	2.75	5.47	6.93%	3.01	5.86	7.65%
STWave	**2.66**	**5.39**	**6.76%**	**2.88**	**5.87**	**7.25%**



(a) D7(M) MAE (b) D7(M) RMSE (c) D7(M) MAPE

(d) D7(L) MAE (e) D7(L) RMSE (f) D7(L) MAPE

Fig. 9: Prediction for each time slice on PeMSD7(M) and PeMSD7(L) (abbreviated as D7(M) and D7(L)) datasets.

- "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 MS": STWave without the trend supervision, *i.e.*, results of it are only supervised by the traffic and may make an unreasonable prediction without the stationary supervision signal.
- "w/o Tem": STWave no longer equips temporal neural network, *i.e.*, fails to capture the temporal changes.
- "w/o Spa": STWave without the ESGAT, *i.e.*, fails to capture the spatial correlations.

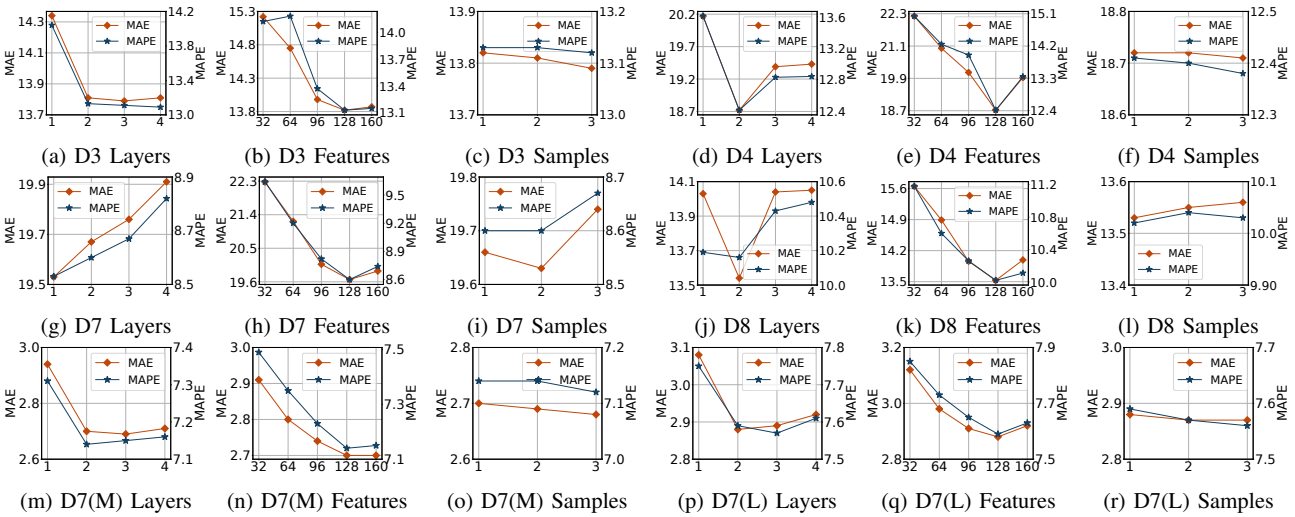
TABLE VI: Performance comparison for variants of STWave on PeMSD7(M) and PeMSD7(L) datasets.

Methods	PeMSD7(M)			PeMSD7(L)		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o DF	2.70	5.48	6.84%	2.91	5.96	7.37%
w/o AF	2.71	5.50	6.88%	2.92	5.99	7.40%
w/o MS	2.68	5.43	6.81%	2.90	5.93	7.34%
w/o Tem	2.70	5.47	6.85%	2.91	5.96	7.36%
w/o Spa	2.87	5.89	7.47%	3.06	6.17	7.69%
STWave	**2.66**	**5.39**	**6.76%**	**2.88**	**5.87**	**7.25%**

Table IV and Table VI show the comparison results on all datasets. It is clear that the original STWave can achieve 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, both "w/o AF" and "w/o MS" underperform STWave, indicating the advantages of adding a smooth supervision signal and removing incorrect events. In conclusion, STWave benefits from the exquisitely-devised components and framework.

D. Parameter Sensitivity Analysis (RQ3)

Figure 10 and Figure 11 depict the results of STNet and framework hyper-parameter sensitivity analysis on all datasets. We search the layers of our dual-channel encoder, the number of features in STWave, and the sampling factor of ESGAT from a search space of $[1, 2, 3, 4]$, $[32, 64, 96, 128, 160]$, and $[1, 2, 3]$. 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 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

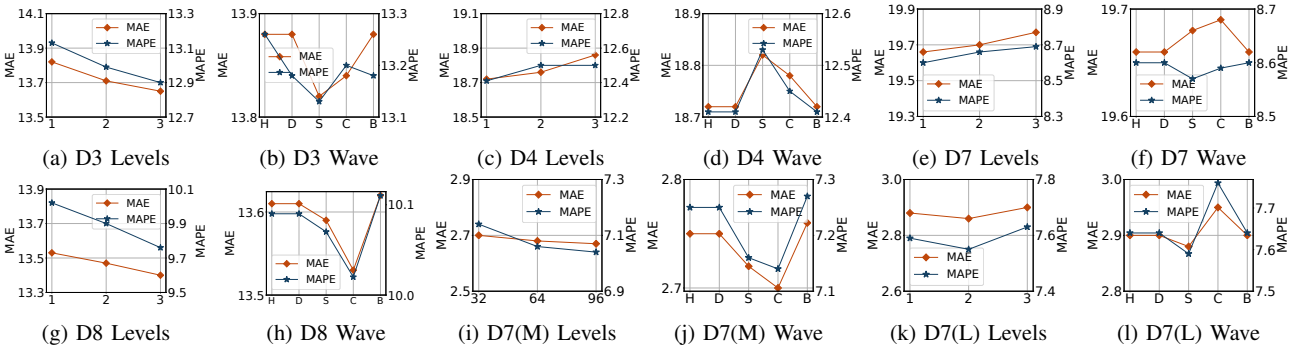


(a) D3 Layers (b) D3 Features (c) D3 Samples (d) D4 Layers (e) D4 Features (f) D4 Samples

(g) D7 Layers (h) D7 Features (i) D7 Samples (j) D8 Layers (k) D8 Features (l) D8 Samples

(m) D7(M) Layers (n) D7(M) Features (o) D7(M) Samples (p) D7(L) Layers (q) D7(L) Features (r) D7(L) Samples

Fig. 10: STNet hyper-parameter study on all datasets. PeMSD is abbreviated as D.



(a) D3 Levels (b) D3 Wave (c) D4 Levels (d) D4 Wave (e) D7 Levels (f) D7 Wave

(g) D8 Levels (h) D8 Wave (i) D7(M) Levels (j) D7(M) Wave (k) D7(L) Levels (l) D7(L) Wave

Fig. 11: Framework hyper-parameter study on all datasets. PeMSD is abbreviated as D.

sub-optimal performance. Besides, 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. Moreover, we search the level and wavelet of DWT from a search space of $[1, 2, 3]$ and [Haar,Daubechies,Symlets,Coiflets,Biorthogonal], abbreviated as [H,D,S,C,B]. 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.

E. ESGAT Study (RQ4)

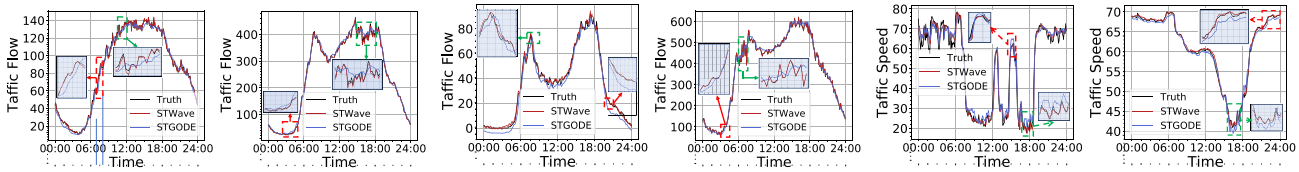
To display the effectiveness and efficiency of ESGAT, we show the performance of STWave, the attention-based LSGCN, state-of-the-art baselines STGODE and STFGNN, and one variant of STWave in three tasks.

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

Figure 13 shows the trade-off between traffic forecasting performance, training speed, and memory usage on the large-scale graph-based dataset PeMSD7(L). 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. Previous 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, the ESGAT makes a better trade-off in terms of speed and performance, while having reasonable memory usage.

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

- "w/o GPE": It no longer uses graph positional encoding (GPE).



(a) PeMSD3 (b) PeMSD4 (c) PeMSD7 (d) PeMSD8 (e) PeMSD7(M) (f) PeMSD7(L)

Fig. 12: Case study on all datasets.

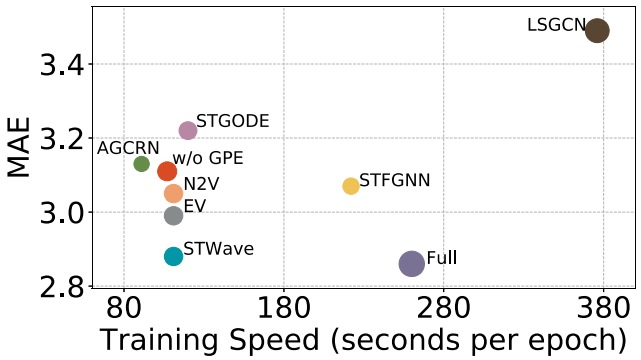


Fig. 13: Computation needs comparison on PeMSD7(L). Performance (y axis), speed (x axis), and memory footprint (size of the circles) of methods.

- "EV": It utilizes graph Laplacian eigenvectors as the GPE.
- "N2V": It uses the Node2vec [41] to learn local-aware graph positional encoding.

as shown in Figure 13, "w/o GPE" performs worst because it lacks the soft inductive bias, *i.e.*, graph structure information. The reason why "N2V" and "EV" perform worse than graph wavelet is that they cannot balance the global graph property and localization information. The learned attention weights between sparse queries and all sensors of STWave, "EV", and "N2V" are visualized in Figure 14, we can observe weights of "EV" and "N2V" are more dense and sparse than STWave, *i.e.*, "EV" considers a few useless correlations and "N2V" neglects a lot essential correlations, demonstrating the equilibrium of our graph wavelet positional encoding.

F. Case Study (RQ5)

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.

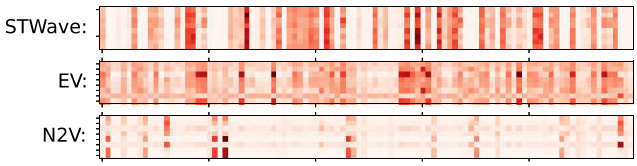


Fig. 14: Visualization for the learned weights on PeMSD4.

VI. 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, thus the two independent components do not disturb each other, 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 ESGAT. Furthermore, with the ESGAT, STWave extracts dynamic correlations under the global spatial receptive field efficiently and effectively. Finally, STWave utilizes the multi-supervision decoder to predict the stable trends and traffic time series simultaneously with an adaptive event fusion module, which can guarantee a reasonable prediction regardless of the distribution. Performance on six traffic benchmarks demonstrates the superiority of STWave over several baselines. Henceforth, STWave will be transferred into other spatio-temporal predicting tasks, such as weather and air quality forecasting.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61872046 and 62261042, the Beijing Natural Science Foundation under Grant 4232035, and the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences.

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] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of NeurIPS*, 2016.

[10] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of IJCAI*, 2019.

[11] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proceedings of NeurIPS*, 2020.

[12] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proceedings of ICLR*, 2018.

[13] 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.

[14] 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.

[15] 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.

[16] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of AAAI*, 2020.

[17] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proceedings of ECCV*, 2020.

[18] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of ICCV*, 2021.

[19] 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.

[20] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.

[21] 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.

[22] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph wavelet neural network," in *Proceedings of ICLR*, 2019.

[23] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[29] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[30] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[31] S. Elmi, "Deep stacked residual neural network and bidirectional lstm for speed prediction on real-life traffic data," in *Proceedings of ECAI*, 2020.

[32] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of NeurIPS*, 2014.

[33] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of AAAI*, 2021.

[34] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proceedings of AAAI*, 2020.

[35] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of ICML*, 2020.

[36] B. Yoo, J. Lee, J. Ju, S. Chung, S. Kim, and J. Choi, "Conditional temporal neural processes with covariance loss," in *Proceedings of ICML*, 2021.

[37] 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.

[38] Y. Fang, Y. Qin, H. Luo, F. Zhao, L. Zeng, B. Hui, and C. Wang, "Cdgnet: A cross-time dynamic graph-based deep learning model for traffic forecasting," *arXiv preprint arXiv:2112.02736*, 2021.

[39] A. Feng and L. Tassiulas, "Adaptive graph spatial-temporal transformer network for traffic forecasting," in *Proceedings of CIKM*, 2022, pp. 3933–3937.

[40] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of WWW*, 2015.

[41] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of SIGKDD*, 2016.

[42] 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.

[43] 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.

[44] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[45] 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.

[46] 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.

[47] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *arXiv preprint arXiv:2012.09699*, 2020.

[48] 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.

[49] 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.