Spatiotemporal-aware Trend-Seasonality Decomposition Network for Traffic Flow Forecasting

Lingxiao Cao, Bin Wang, Guiyuan Jiang, Yanwei Yu*, Junyu Dong

Faculty of Information Science and Engineering, Ocean University of China
caolingxiao@stu.ouc.edu.cn, {wangbin9545,jiangguiyuan,yuyanwei,dongjunyu}@ouc.edu.cn

Abstract

Traffic prediction is critical for optimizing travel scheduling and enhancing public safety, yet the complex spatial and temporal dynamics within traffic data present significant challenges for accurate forecasting. In this paper, we introduce a novel model, the **S**patiotemporal-aware **T**rend-Seasonality **D**ecomposition **N**etwork (STDN). This model begins by constructing a dynamic graph structure to represent traffic flow and incorporates novel spatio-temporal embeddings to jointly capture global traffic dynamics. The representations learned are further refined by a specially designed trend-seasonality decomposition module, which disentangles the trend-cyclical component and seasonal component for each traffic node at different times within the graph. These components are subsequently processed through an encoder-decoder network to generate the final predictions. Extensive experiments conducted on real-world traffic datasets demonstrate that STDN achieves superior performance with remarkable computation cost. Furthermore, we have released a new traffic dataset named JiNan, which features unique inner-city dynamics, thereby enriching the scenario comprehensiveness in traffic prediction evaluation.

Introduction

With technological advancements, a diverse array of sensors has been increasingly integrated into monitoring systems to bolster modern intelligent transportation systems (ITS) (Cirstea et al. 2021; Ji et al. 2022; Dai et al. 2023). Transportation authorities deploy a variety of sensors, such as electronic police cameras and bayonet detectors, across road networks to continuously collect essential traffic data, including flow and speed. Utilizing historical traffic flow data and road network topology, traffic forecasting aims to predict future flow variations, thereby improving daily travel and traffic management (Dai et al. 2021; Li et al. 2023).

In the realm of traffic forecasting, considerable efforts are devoted to modeling traffic dynamics. Methods based on deep learning, especially the Spatio-Temporal Graph Neural Networks (STGNNs), have proven more effective than statistical time series analysis and shallow machine learning techniques in addressing traffic forecasting challenges. To address temporal dynamics, sequential models like RNN-based variants (Graves and Graves 2012; Deng et al. 2022, 2024a) and non-sequential Transformers (Vaswani et al. 2017) have been profoundly studied. For spatial dynamics, recent progress has been made with the Graph Neural Networks (GNNs) (Yin et al. 2021), which represent sensors as nodes within a graph, leveraging graph structure to capture traffic patterns. Despite these substantial advancements, our study suggests that current methods still exhibit considerable potential for improvement in two critical aspects.

Firstly, the prominent spatio-temporal characteristics in traffic flow can be more effectively modeled with the appropriate inductive bias. Although previous studies have demonstrated (Zhang, Zheng, and Qi 2017; Yu et al. 2019; Guo et al. 2019), traffic patterns are strongly influenced by the specific temporal periodicities, most efforts have roughly incorporated these temporal features into the models without explicitly modeling the synergy between the long and short periodicities (Chen et al. 2018; Deng et al. 2024b). Regarding spatial aspects, while different locations exhibit unique spatial characteristics, most GNN methods construct static graphs based solely on the distances between two nodes, fail to consider the global interactions among all nodes in the graph (Yin et al. 2021).

Secondly, effective trend-seasonality decomposition of traffic flow can greatly enhance the representation learning of traffic nodes (Wu et al. 2021; Fang et al. 2023). Utilizing this methodology improves the prediction of traffic flow by distinguishing systematic patterns and noise components. However, the application of trend-seasonality decomposition predominantly to individual nodes in a traffic network overlooks the interactions among global nodes, thereby diminishing the quality of node representations learned by GNNs.

To bridge these research gaps, we introduce a **S**patiotemporal-aware **T**rend-Seasonality **D**ecomposition **N**etwork (STDN), which enhances global node representations through a novel trend-seasonality decomposition incorporating spatio-temporal embeddings. It features three key modules: *(1) Module of Spatio-Temporal Embedding Learning* models the temporal periodicity by learning the temporal embedding including specific weeks and minutes, and acquires an initial spatial location embedding from the eigenvalues and eigenvectors of the graph Laplacian matrix.

*Corresponding author: Yanwei Yu.

(2) Module of Dynamic Relationship Graph Learning explores the global dynamic interaction among traffic nodes, enhanced by the spatio-temporal embedding, thereby capturing the high-order relationships between each traffic node. *(3) Module of Trend-Seasonality Decomposition* aims to refine the node representations by disentangling the traffic flow into the trend-cyclical and seasonal components, which are further processed through an encoder-decoder network. The contributions of our study are summarised as follows:

- We propose the STDN model, a novel dynamic GCN-based framework for traffic flow prediction. To our knowledge, this is the first approach to learn disentangled representations of traffic flow in view of the spatio-temporal embeddings.
- We develop a novel trend-seasonality decomposition mechanism. Each component of decomposition is designed to be aware of the spatio-temporal embeddings, enriching the model's capability to capture high-order node interactions across time and space.
- We conduct comprehensive multi-step traffic flow prediction experiments on three real-world datasets. The experimental results demonstrate that our method consistently surpasses various competing baselines. Additionally, the effectiveness of each module is verified through the ablation study.
- We release a new urban dataset named JiNan, which, unlike popular highway traffic datasets (e.g., PeMS) focuses more on the spatio-temporal dynamics of inner-city traffic. We believe this dataset will further enrich the scenario comprehensiveness in traffic flow prediction evaluations.

Related Works

We discuss related works from three categories of methods highly relevant to our study.

GCN-based Models. Methods such as STGCN (Yu, Yin, and Zhu 2017) and DCRNN (Li et al. 2018) combine graph convolution network (**GCN**) with sequential data to model traffic flows. To address the limitations of predefined graphs based on the traffic network, GWNet (Wu et al. 2019) and AGCRN (Bai et al. 2020) introduce adaptive graphs within GCNs, enhancing the capture of global and accurate spatial dependencies in traffic data. STGODE (Fang et al. 2021) utilizes tensor-based neural ODEs to mitigate the oversmoothing problem in deep GCNs. Subsequently, dynamic graph convolution (Han et al. 2021; Lan et al. 2022; Zhao et al. 2023) are utilized to incorporate more intrinsic dynamic information within the spatial structure. However, they lose the guidance of prior knowledge, which may lead to underfitting and overfitting.

Attention-based Models. Attention-based models such as GMAN (Zheng et al. 2020) and ASTGCN (Guo et al. 2019) directly capture global temporal correlations between two time slices through attention mechanisms. Subsequent models like SSTBAN, STWave, and PDformer (Guo et al. 2023; Fang et al. 2023; Jiang et al. 2023) extend this approach by integrating intrinsic spatial correlations with temporal dynamics through similar mechanisms. Although these models have achieved success in several aspects, they have not yet to fully leverage the potential of spatio-temporal embeddings within traffic sequence data.

Decomposition-based Methods. Decomposing time sequence data into the trend-cyclical and seasonal components has proven effective for prediction (Wu et al. 2021; Wang et al. 2024). D^{2}STGNN (Shao et al. 2022) distinguishes two different types of hidden temporal signals: diffusion signals and intrinsic signals, which intriguingly parallel the trend-cyclical and seasonal parts. Other models treat seasonal part as delay signals (Jiang et al. 2023; Long et al. 2024). STWave (Fang et al. 2023) adopts a unique approach by starting with the data itself, decomposing spatio-temporal data into trend and event components using wavelet transformations. However, these models overlook the unique characteristics across time and space.

Problem Definition

In this section, we define the key components and objectives of our study, focusing on the structure of traffic networks and the goals of traffic forecasting.

Definition 1 (Traffic Network) *Given the real-world traffic scenarios, we define the traffic network as a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$, where $\mathcal{V}$ denotes a set of N nodes, each corresponding to a different sensor within the road network. $\mathcal{E}$ represents a set of edges that denote the connectivity among the nodes. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix that models the connectivity between nodes.*

Definition 2 (Traffic Forecasting) *Given historical traffic time series and the road topology, the objective of traffic flow forecasting is to predict future values of traffic time series. Specially, we represent the historical time series as a signal tensor $\boldsymbol{\mathcal{X}} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T] \in \mathbb{R}^{T \times N \times C}$, where T is the length of historical traffic time series and C is the number of dimensions of node attributes. We aim to construct a function $f(\cdot)$ that maps the historical time series over T time steps to predict the subsequent T' time steps:*

$$[\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T; \mathcal{G}] \xrightarrow{f(\cdot)} \left[\hat{\mathbf{X}}_{T+1}, \hat{\mathbf{X}}_{T+2}, \ldots, \hat{\mathbf{X}}_{T+T'}\right]. \tag{1}$$

Methodology

STDN consists of three principal modules, which are introduced as follows.

Module 1: Dynamic Relationship Graph Learning

To address the limitations of simple distance-based connectivity, which overlooks the high-order relationships between each node, we construct a dynamic relationship graph that considers different time steps and nodes. This approach allows us to model the complex higher-order spatio-temporal relationships among all traffic nodes effectively.

Inspired by Han et al. (2021), we design three learnable matrices and a learnable core tensor to streamline the constructing of the dynamic graph. These components include: a time slot embedding $\mathbf{E}^t \in \mathbb{R}^{N_t \times D}$, a starting

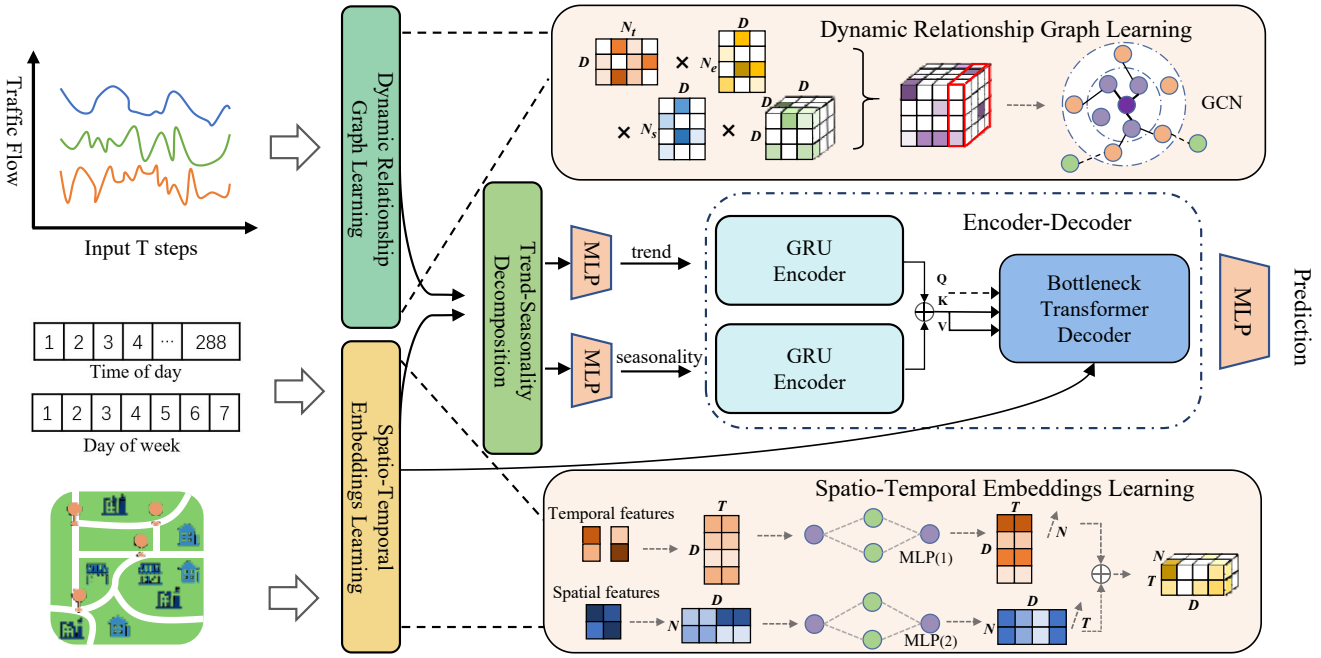

Figure 1: The overview of the proposed framework. MLP: multi-layer perceptron, GCN: graph convolution network.

node embedding $\mathbf{E}^s \in \mathbb{R}^{N_s \times D}$, an ending node embedding $\mathbf{E}^e \in \mathbb{R}^{N_e \times D}$ and the core tensor $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{D \times D \times D}$. Here, N_t, N_s, N_e and the D denote the number of time slots, starting nodes, ending nodes and the dimension of the embeddings, respectively. The specific calculations are as follows:

$$\begin{aligned} \mathbf{A}'_{t,i,j} &= \sum_{o=1}^{D}\sum_{q=1}^{D}\sum_{r=1}^{D} \boldsymbol{\mathcal{K}}_{o,q,r}\mathbf{E}^t_{t,o}\mathbf{E}^e_{i,q}\mathbf{E}^s_{j,r}, \\ \mathbf{A}''_{t,i,j} &= \max(0, \mathbf{A}'_{t,i,j}), \\ \mathbf{A}_{t,i,j} &= \frac{e^{\mathbf{A}''_{t,i,j}}}{\sum_{n=1}^{N_s} e^{\mathbf{A}''_{t,i,n}}}. \end{aligned} \tag{2}$$

Through this methodology, we derive a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N_t \times N_s \times N_e}$ that encapsulates the high-order connectivity among global nodes across various time steps. Notably, since the starting nodes can also serve as ending nodes, we designate $N = N_s = N_e$ for simplification.

Subsequently, to establish and analyze the relationships between different nodes, we employ a dynamic graph convolution. This process involves the matrix multiplication of the constructed adjacency matrix, the hidden states of the nodes, and the learnable parameters across different graphs at various time. The hidden states of the nodes are updated by aggregating the hidden states of their neighbors through weighted links at each time step:

$$\boldsymbol{\mathcal{H}}_L = \sum_{l=0}^{L} \left(\mathbf{A}_{(t)}\right)^l \mathbf{H}^t_l \mathbf{W}_l, \tag{3}$$

where $\mathbf{A}_{(t)}$ denotes the adjacency matrix extracted from the tensor $\boldsymbol{\mathcal{A}}$ at time slot t. The term $(\mathbf{A}_{(t)})^l$ refers to the matrix multiplication operation. $\mathbf{H}^t_l$ represents the output hidden states at the l-th layer, which serves as input for the dynamic graph convolution in the l+1-th layer, $\mathbf{W}_l$ are the parameters specific to the l-th layer, and $\mathbf{H}^1_0$ represents the normalized traffic sequence data at first time step. Consequently, through these operations, we generate the final output $\boldsymbol{\mathcal{H}}_L \in \mathbb{R}^{T \times N \times D}$ which constitutes the processed sequence data.

Module 2: Spatio-Temporal Embeddings Learning

To more effectively capture the temporal correlations and periodicities in traffic flow, we design a temporal context embedding learning module. Given the time of day, represented as $\mathbf{z}^d_h \in \mathbb{R}^T$ and the day of the week, represented as $\mathbf{z}^w_h \in \mathbb{R}^T$, we extract temporal features and encode them by one-hot. By concatenating these encoded features (denoted by [,]), we generate an initial temporal embedding as below:

$$\mathbf{Z}^t_h = \sigma(\mathbf{W}[onehot(\mathbf{z}^d_h), onehot(\mathbf{z}^w_h)]), \tag{4}$$

where $\mathbf{Z}^t_h \in \mathbb{R}^{T \times D}$, σ denotes the ReLU activation function. $\mathbf{W}$ comprises the trainable parameters. D specifies the dimensionality of the temporal embedding.

To enhance the ability of the temporal embedding to learn multi-resolution temporal features and to facilitate its integration with traffic time series, we refine the initial temporal embedding using a bias-free MLP:

$$\mathbf{M}^t_h = \sigma_2(\mathbf{W}_2\sigma_1(\mathbf{W}_1\mathbf{Z}^t)), \tag{5}$$

where σ_1 denotes the ReLU activation function, σ_2 is the sigmoid activation function, and $\mathbf{W}_1$, $\mathbf{W}_2$ are the trainable parameters. The resulting temporal embedding $\mathbf{M}^t_h \in \mathbb{R}^{T \times D}$ is utilized in subsequent stages of the model.

To effectively model the structure of the road network, we utilize the normalized Laplacian matrix, defined as:

$$\Delta = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}, \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, $\mathbf{D}$ is the degree matrix, and $\mathbf{I}$ is the identity matrix. The Laplacian matrix's eigenvalues and eigenvectors encapsulate the spatial graph information in Euclidean space. By decomposing the Laplacian, we obtain the eigenvalue matrix and the eigenvector matrix:

$$\Delta = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}, \tag{7}$$

where $\mathbf{U}$ is the matrix of eigenvector and $\mathbf{\Lambda}$ is the matrix of eigenvalue. To construct an initial spatial embedding, we select the k_r smallest nontrivial eigenvectors, resulting in $\mathbf{Z}^s \in \mathbb{R}^{N \times k_r}$. To capture global information as comprehensively as possible, we set k_r to 32.

To enhance the spatial embedding to represent spatial features and facilitate its integration with subsequent traffic time series, we process the initial spatial embedding using a bias-free MLP:

$$\mathbf{M}^s = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{Z}^s), \tag{8}$$

where σ is the ReLU activation function, D is the spatial embedding dimension, and $\mathbf{W}_1$, $\mathbf{W}_2$ are the trainable parameters. $\mathbf{M}^s \in \mathbb{R}^{N \times D}$ represents the spatial embedding that will be utilized in sebsequent stages of the model.

To effectively integrate the spatio-temporal embeddings with the traffic sequence data, we further propose a refined spatial-temporal embedding learning module. Specifically, this module involves broadcasting the spatial embedding $\mathbf{M}^s$ across the dimension T to produce $\boldsymbol{\mathcal{M}}^s \in \mathbb{R}^{T \times N \times D}$, and similarly broadcasting the temporal embedding $\mathbf{M}_h^t$ across the dimension N to yield $\boldsymbol{\mathcal{M}}_h^t \in \mathbb{R}^{T \times N \times D}$. These embeddings are then combined as follows:

$$\boldsymbol{\mathcal{M}} = \sigma_1(\boldsymbol{\mathcal{M}}_h^t) + \sigma_2(\boldsymbol{\mathcal{M}}^s), \tag{9}$$

where $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{T \times N \times D}$, σ_1 represents the mathematical sine function, σ_2 denotes the ReLU activation function.

In this manner, we obtain the refined spatio-temporal information embeddings for each node at different times. The use of the sine function is particularly beneficial as it not only normalizes the data to a range between -1 and 1 but also enables the model to effectively capture non-linear changes. Specifically, when we generate the initial temporal embedding $\mathbf{M}_h^t$, we employ the sigmoid activation function, as shown in equation 5. Consequently, after the application of the sine function, the data is normalized to a range between 0 and 1, which is crucial for effectively fusing the embedding with the traffic sequence data in subsequent processing steps.

Module 3: Trend-Seasonality Decomposition

Given the strong correlation between the trend-seasonality of a traffic node and its spatio-temporal context, the Trend-Seasonality Decomposition module is designed based on spatio-temporal embeddings learning. This module effectively disentangles the traffic flow of each node into distinct trend and seasonal components, adjusting the representations to better suit the forecasting task.

When processing the traffic sequence data with trend-seasonality decomposition, since the dimensionality of the traffic hidden states and the spatio-temporal embeddings has been aligned, we initially derive the trend component by through multiplication-wise interaction between the hidden states of the nodes $\boldsymbol{\mathcal{H}}_L$ and the spatio-temporal embeddings $\boldsymbol{\mathcal{M}}$. The residual part constitutes the seasonal component. The calculations are as follows:

$$\begin{aligned} \boldsymbol{\mathcal{X}}_t &= \boldsymbol{\mathcal{H}}_L \odot \boldsymbol{\mathcal{M}}, \\ \boldsymbol{\mathcal{X}}_s &= \boldsymbol{\mathcal{H}}_L - \boldsymbol{\mathcal{X}}_t, \end{aligned} \tag{10}$$

where $\boldsymbol{\mathcal{X}}_s, \boldsymbol{\mathcal{X}}_t \in \mathbb{R}^{T \times N \times D}$ denote the seasonal and the extracted trend-cyclical part respectively. $\odot$ is the Hadamard product. The trend component of traffic flow is generally smooth and strongly influenced by the time and location of the traffic node. Our approach is based on the mild assumption that nearby times and locations yield similar trends, while distinct times and locations exhibit unique traffic patterns.

Encoder-Decoder Architecture

In this module, we utilize an encoder-decoder network to extract deeper spatio-temporal features. For the encoding process, we employ Gated Recurrent Unit (GRU) due to its effectiveness in capturing temporal dependencies. For the decoding process, we opt for the Transformer architecture, primarily because of its unique multi-head attention mechanism, which generates predictions with superior performance. We denote use the GRU$(\cdot)$ to denote the Gated Recurrent Unit as below:

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_t &= GRU(\boldsymbol{\mathcal{X}}_t), \\ \boldsymbol{\mathcal{Y}}_s &= GRU(\boldsymbol{\mathcal{X}}_s), \end{aligned} \tag{11}$$

where $\boldsymbol{\mathcal{Y}}_s, \boldsymbol{\mathcal{Y}}_t \in \mathbb{R}^{T \times N \times D}$ represent the outputs of the GRU encoder for the seasonal and trend components, respectively. These outputs are then combined through element-wise summation to produce the final output $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{T \times N \times D}$.

The Transformer architecture utilizes multi-head attention mechanisms, which first projects the queries, keys and values into h different d-dimensional subspaces, and then execute the attention function in parallel:

$$\begin{aligned} \mathrm{MHSA}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) &= \oplus(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O, \\ \mathrm{head}_j &= \mathrm{softmax}(\frac{(\mathbf{Q}W_j^Q)(\mathbf{K}W_j^K)^{\top}}{\sqrt{d}})(\mathbf{V}W_j^V), \end{aligned} \tag{12}$$

where W is the training parameter.

Inspired by Guo et al. (2023), we incorporate a component known as the Bottleneck Transformer Block (BT block) into our architecture, strategically designed to reduce both temporal and spatial complexity.

For the predicted time of day and day of the week, we obtain the predicted time embedding $\mathbf{M}_p^t \in \mathbb{R}^{T' \times D}$ according to equation 4 and equation 5. To align with the default settings, we maintain $T = T'$. Additionally, we extend the temporal embedding $\mathbf{M}_p^t$ across the dimension N, resulting in $\boldsymbol{\mathcal{M}}_p^t \in \mathbb{R}^{T' \times N \times D}$.

To effectively prompt the decoder to focus on capturing the high-order dynamics between time and space, we concatenate the predicted temporal embedding and spatial embedding with the output from $(l-1)^{th}$ BT block as $\boldsymbol{\mathcal{Z}}^{(l-1)} \in$

Datasets	Nodes	Edges	Time Interval	Time range
PeMS04	307	340	5min	01/2018-02/2018
PeMS07	883	866	5min	05/2017-08/2018
JiNan	406	324	5min	08/2016-09/2016

Table 1: Statistics of the datasets.

$\mathbb{R}^{T' \times N \times D}$ to obtain $\boldsymbol{\mathcal{H}}^{(l)} = (\boldsymbol{\mathcal{Z}}^{(l-1)} || \boldsymbol{\mathcal{M}}_p^t || \boldsymbol{\mathcal{M}}^s) \in \mathbb{R}^{T' \times N \times 3D}$. Besides, $\boldsymbol{\mathcal{H}}_{:,v}^{(l-1)} \in \mathbb{R}^{T' \times 3D}$ represents the input pertaining to node v across all time slices.

The specific BT block employs defined T' vectors $\mathbf{IT}^{(l)} \in \mathbb{R}^{T' \times 3D}$ as trainable parameters within the model. The transformation within the BT block is described by the following equations:

$$\begin{aligned} \mathbf{IT}' &= \mathrm{MHSA}(\mathbf{IT}^{(l)}, \boldsymbol{\mathcal{H}}_{:,v}^{(l-1)}, \boldsymbol{\mathcal{H}}_{:,v}^{(l-1)}) \in \mathbb{R}^{T' \times 3D}, \\ \boldsymbol{\mathcal{Z}}_{:,v}^{(l)} &= \mathrm{MHSA}(\boldsymbol{\mathcal{H}}_{:,v}^{(l-1)}, \mathbf{IT}', \mathbf{IT}') \in \mathbb{R}^{T' \times 3D}. \end{aligned} \tag{13}$$

The initial input to the first BT block is $\boldsymbol{\mathcal{Y}}$ sourced from the encoder. The output from the final l BT block is denoted as $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{T' \times N \times D}$, which represents the projected future representation of traffic flow.

Ultimately, to derive the expected traffic predictions, we employ a fully-connected neural network that transforms the future representation of traffic flow into the predicted values $\hat{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{T' \times N \times C}$. We utilize the $L1$ as the training loss function as below:

$$\mathcal{L} = \sum_{t=T+1}^{T+T'} \sum_{n=1}^{N} |\hat{x}_t^n - y_t^n|. \tag{14}$$

Experiments

Datasets

To evaluate the performance of STDN, we utilize three real-world datasets, each offering unique traffic flow dynamics. The dataset JiNan, which is first released by us, derived from actual traffic flow statistics in a city, mirrors the setup in (Song et al. 2020), where the time interval is set to 5 minutes. Detailed descriptions of these datasets are provided in Table 1.

Baseline Methods

To assess the performance of STDN, we compare it against a diverse range of established baselines:

- **HA** (Hamilton 2020) uses the historical average of input data for prediction.
- **ARIMA** (Box et al. 2015) is a well-known statistical model widely employed for time series forecasting.
- **VAR** (Lütkepohl 2005): is another traditional method for time series forecasting.
- **SVR** (Wu, Ho, and Lee 2004) employs support vector regression for predictive modeling.
- **LSTM** (Graves and Graves 2012) is a deep learning model that captures temporal dependencies but does not account for spatial correlations.
- **DCRNN** (Li et al. 2018) integrates diffusion convolution into GRU layers to enhance spatial-temporal correlation capture.
- **STGCN** (Yu, Yin, and Zhu 2017) combines graph and temporal convolutions to handle spatial-temporal data.
- **GWNet** (Wu et al. 2019) combines dilated convolution with diffusion graph convolution and introduces a self-adaptive adjacency matrix.
- **GMAN** (Zheng et al. 2020) employs multi-attention to capture both spatial and temporal dynamics.
- **ASTGCN** (Guo et al. 2019) applies attention mechanisms on both temporal and spatial convolutions to dynamically capture spatio-temporal correlations.
- **AGCRN** (Bai et al. 2020) focuses on extracting node-specific features and uncovering hidden node interdependencies.
- **DMSTGCN** (Han et al. 2021) learns dynamic spatial dependencies and builds a multi-faceted fusion module for complex traffic data features.
- **STGODE** (Fang et al. 2021) adopts ordinary differential equations for traffic flow forecasting.
- **STGNCDE** (Choi et al. 2022) designs two neural controlled differential equations for prediction.
- **D^{2}STGNN** (Shao et al. 2022) models traffic flow by separating it into the diffusion component and the inherent component.
- **DSTAGNN** (Lan et al. 2022) constructs a spatio-temporal graph and utilizes multi-head attention to represent dynamic spatial relevance.
- **SSTBAN** (Guo et al. 2023) implements a self-supervised learning approach with a masking method for prediction.
- **STWave** (Fang et al. 2023) employs wavelets to decompose traffic data into stable trends and fluctuating events.

Evaluation Metrics and Experimental Settings

In our evaluation, we employ the mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) to quantify the performance of different methods.

Our experiments are conducted on a server with NVIDIA RTX 4090 GPU cards, running CUDA version 12.2. All the models are implemented using PyTorch. The datasets are split in a 6:2:2 ratio for training, validation, and testing, respectively. We use historical data from the past hour to predict the traffic flow for the next hour, corresponding to using the past 12 time steps to forecast the next 12 steps.

To prevent overfitting, an early-stopping strategy is employed with a patience setting of 10. We use Adam optimizer with an initial learning rate of 0.001. The standard batch size for all experiments is set to 64. If GPU memory constraints occur, the batch size is reduced to 32, and further to 16 if necessary, until the programs can run efficiently. The number of dimensions of node attribute on three datasets is C = 1. Totally, there are 3 hyperparameters in our model, *i.e.*, the numbers of bottleneck transformer block L, the number of attention heads h, and the dimensionality d of each attention

Model	PeMS04			PeMS07			JiNan		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	38.03	59.24	27.88%	45.12	65.64	24.51%	13.23	22.85	46.83%
ARIMA	33.73	48.80	24.18%	38.17	59.27	19.46%	14.89	26.22	48.53%
VAR	24.54	38.61	17.24%	50.22	75.63	32.22%	12.56	20.60	40.54%
SVR	28.70	44.56	19.20%	32.49	50.22	14.26%	11.98	20.82	39.56%
LSTM	26.77	40.65	18.23%	29.98	45.94	13.20%	11.30	19.40	38.02%
DCRNN	21.02	33.44	14.17%	25.22	38.61	11.82%	11.40	20.54	41.74%
STGCN	21.16	34.89	13.83%	25.33	39.34	11.21%	9.41	16.08	36.42%
GWNet	24.89	39.66	17.29%	26.39	41.50	11.97%	9.39	16.13	35.83%
GMAN	19.14	31.60	13.19%	20.97	34.02	9.05%	10.15	16.98	38.32%
ASTGCN(r)	22.93	35.22	16.56%	24.01	37.87	10.73%	10.06	17.23	38.95%
AGCRN	19.83	32.26	12.97%	22.37	36.55	9.12%	9.52	16.15	40.27%
DMSTGCN	20.01	32.18	14.50%	23.73	36.01	12.21%	8.78	14.61	34.52%
STGODE	20.84	32.82	13.77%	22.59	37.54	10.14%	9.25	15.75	37.68%
STGNCDE	19.21	31.09	12.76%	20.53	33.84	8.80%	9.14	16.60	36.89%
D^2STGNN	19.55	31.99	12.82%	21.55	34.83	9.39%	9.12	15.97	35.65%
DSTAGNN	19.30	31.46	12.70%	21.42	34.51	9.01%	8.95	14.99	36.87%
SSTBAN	18.89	31.21	12.62%	20.45	37.55	9.31%	8.41	14.62	33.79%
STWave	21.20	34.47	14.32%	20.33	34.03	**8.58%**	9.23	16.10	36.12%
STDN	**18.40***	**30.41***	**12.21%***	**20.08***	**33.73***	9.29%	**8.32***	**14.52***	**32.19%***

Table 2: Performance comparison of all models on three real-world datasets. Marker $*$ indicates the results are statistically significant (t-test with p-value < 0.01).

head, where the total number of features $D = h \times d$. The optimal settings for our model on PeMS04 and JiNan datasets are $L = 2$, $h = 8$, $d = 16$ ($D = 128$). For the PeMS07 dataset, the best performance is achieved with $L = 2$, $h = 8$, $d = 12$ ($D = 96$). All source code and data are available at https://github.com/roarer008/STDN

Performance Comparison

Table 2 presents the results from graph-based baselines and grid-based baselines. The best results are highlighted in bold, and the second-best results are underlined. Based on these results, several key conclusions can be drawn:

- STDN achieves state-of-the-art performance, particularly evident in the PeMS04 and JiNan datasets. Traditional machine learning methods such as ARIMA typically perform poorly, as they are unable to capture the non-linear correlations present in the spatio-temporal traffic data.
- Among the GCN-based models, AGCRN demonstrates strong performance. Compared to other models, STDN excels in capturing the structure of the road network by effectively integrating eigenvalues from the Laplacian matrix with traffic flow data.
- Attention-based models generally perform near optimally among all baselines. Notably, STDN distinguishes itself by integrating spatio-temporal embeddings with traffic flow data and the decoder, significantly enhancing traffic forecasting accuracy.
- Compared to the baseline models, we incorporate multi-resolution temporal features, such as "time of day" and "day of week", for temporal embeddings, alongside a geospatial directed graph for spatial embeddings. Based on these spatiotemporal embeddings, traffic flow is disentangled into trend and seasonality parts. This novel disentangling method significantly enhances the traffic forecasting accuracy.

Model	PeMS04			JiNan		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o TE	19.10	30.89	12.87%	8.47	14.67	34.18%
w/o SE	18.55	30.46	12.52%	8.44	14.68	35.15%
w/o STE	18.93	30.70	12.97%	8.55	14.79	35.38%
w/o DRG	18.61	30.90	12.23%	8.37	14.57	33.91%
w/o STD	18.60	31.08	12.33%	8.75	15.14	37.88%
STDN	**18.40**	**30.41**	**12.21%**	**8.32**	**14.52**	**32.19%**

Table 3: Ablation study on PeMS04 and JiNan datasets.

Ablation Study

To evaluate the effectiveness of different components in STDN, we conducted the ablation study with several variants of the STDN:

- **w/o TE**: This variant removes the temporal embedding modeling, meaning the decoder operates solely with spatial embedding cues.
- **w/o SE**: This variant removes the spatial embedding modeling, meaning the decoder operates solely with temporal embedding cues.
- **w/o STE**: This variant eliminates spatio-temporal embedding, thus the traffic flow is not decomposed into trend-seasonality components. As a result, the decoder does not incorporate any spatio-temporal cues.
- **w/o DRG**: This variant eliminates the dynamic relationship graph learning.
- **w/o STD**: Instead of using spatiotemporal-aware decomposition to disentangle the traffic sequence data, this vari-

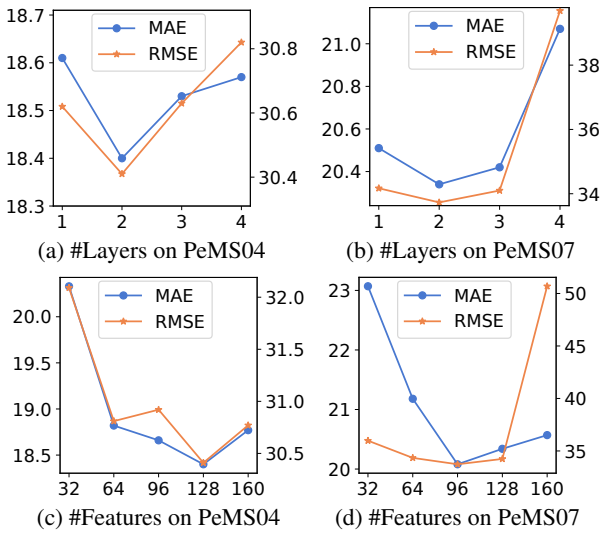

Figure 2: Parameter sensitivity study on PeMS04 and PeMS07 datasets.

ant adopts the decomposition method utilized by Autoformer (Wu et al. 2021).

Table 3 presents the comparison of STDN and its variants on PeMS04 and JiNan datasets. From this comparison, we can draw several conclusions: (1) The origin STDN consistently achieves the best performance relative to its variants, underscoring the effectiveness of its full configuration. (2) The results show that the variant "*w/o TE*" generally outperforms "*w/o SE*" across most tasks. This suggests that temporal information, particularly periodic information, plays a more critical role than spatial information. (3) The "*w/o DRG*" underperforms STDN, indicating the importance of the dynamic relationship graph learning module. (4) The performance of "*w/o STD*" underlines the necessity of the trend-seasonality decomposition module aware of spatio-temporal embeddings.

Parameter Sensitivity Study

Figure 2 illustrates the results of hyper-parameter sensitivity analysis for our STDN on PeMS04 and PeMS07 datasets. This study involved varying the number of decoder layers and the number of features in STDN, exploring options within the ranges of [1, 2, 3, 4] for layers and [32, 64, 96, 128, 160] for features. From this analysis, we can draw several conclusions: (1) The performance of our model improves with an increasing in the number of decoder layers but stabilizes at 2 layers. (2) Optimal performance is achieved with 128 features on the PeMS04 dataset and 96 features on the PeMS07 dataset. This finding highlights that while increasing the number of features generally enhances the model's capability to represent complex traffic patterns, but excessive features may introduce noise and degrade model performance.

Model Efficiency Study

To demonstrate the efficiency of our model, we benchmark STDN against DMSTGCN, SSTBAN, and STWave, which have achieved suboptimal results at the PeMS04 and JiNan datasets. Figure 3 displays the average training time per epoch and inference time for each model. Figure 4 illustrates the MAE curves on the validation part of PeMS04 dataset during the training process. The following observations can be made: (1) STDN not only trains faster but also infers quicker than the compared models. (2) STDN demonstrates a faster convergence rate, achieving better performance in fewer epochs. In our experiments, while STWave reaches its best performance at epoch 100, its MAE is still higher than the lowest MAE achieved by our STDN at just epoch 19.

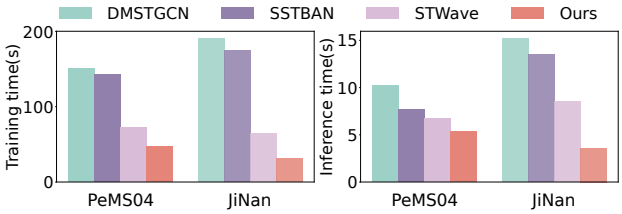

Figure 3: The computational time cost on PeMS04 and JiNan datasets.

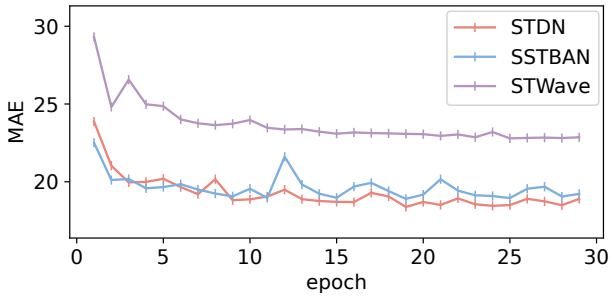

Figure 4: The MAE on the validation part of PeMS04 dataset during the training process.

The computational complexity of our STDN encoder-decoder module is $O(TD+LND)$, with the encoder and decoder contributing complexities of $O(TD)$ and $O(LND)$, respectively. Here, L denotes the number of bottleneck transformer blocks. The complexity of the spatio-temporal embedding module is given by $O((T+N)D+N^3)$. Although calculating the eigenvectors and eigenvalues of the graph Laplacian is computationally intensive, marked by a complexity of $O(N^3)$, this process can be efficiently handled through preprocessing prior to training. Therefore, STDN maintains comparable time and memory complexity during training, ensuring efficiency without compromising performance.

Conclusion

In this paper, we introduce a novel spatiotemporal-aware trend-seasonality decomposition network (STDN), which marks a pioneering approach in employing spatio-temporal embeddings to learn disentangled representations of traffic flow. The empirical evaluations conducted across three real-world datasets demonstrate the superior performance of STDN over existing models. The release of the new inner-city dataset JiNan can also enrich the scenario comprehensiveness in traffic forecasting evaluations.

Acknowledgments

This work is partially supported by the Fundamental Research Funds for the Central Universities under Grant No 202442005, the National Natural Science Foundation of China under Grant Nos. 62176243 62372421, and 62402463, and the National Key R&D Program of China under Grant No 2022ZD0117201.

References

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33: 17804–17815.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Chen, L.; Zhong, Q.; Xiao, X.; Gao, Y.; Jin, P.; and Jensen, C. S. 2018. Price-and-time-aware dynamic ridesharing. In *2018 IEEE 34th international conference on data engineering (ICDE)*, 1061–1072. IEEE.

Choi, J.; Choi, H.; Hwang, J.; and Park, N. 2022. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, 6, 6367–6374.

Cirstea, R.-G.; Kieu, T.; Guo, C.; Yang, B.; and Pan, S. J. 2021. EnhanceNet: Plugin neural networks for enhancing correlated time series forecasting. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1739–1750. IEEE.

Dai, S.; Wang, J.; Huang, C.; Yu, Y.; and Dong, J. 2021. Temporal multi-view graph convolutional networks for citywide traffic volume inference. In *2021 IEEE International Conference on Data Mining (ICDM)*, 1042–1047. IEEE.

Dai, S.; Wang, J.; Huang, C.; Yu, Y.; and Dong, J. 2023. Dynamic multi-view graph neural networks for citywide traffic inference. *ACM Transactions on Knowledge Discovery from Data*, 17(4): 1–22.

Deng, J.; Chen, X.; Jiang, R.; Song, X.; and Tsang, I. W. 2022. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 35(8): 7665–7680.

Deng, J.; Chen, X.; Jiang, R.; Yin, D.; Yang, Y.; Song, X.; and Tsang, I. W. 2024a. Disentangling Structured Components: Towards Adaptive, Interpretable and Scalable Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Deng, J.; Ye, F.; Yin, D.; Song, X.; Tsang, I.; and Xiong, H. 2024b. Parsimony or Capability? Decomposition Delivers Both in Long-term Time Series Forecasting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Fang, Y.; Qin, Y.; Luo, H.; Zhao, F.; Xu, B.; Zeng, L.; and Wang, C. 2023. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 517–529. IEEE.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 364–373.

Graves, A.; and Graves, A. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 922–929.

Guo, S.; Lin, Y.; Gong, L.; Wang, C.; Zhou, Z.; Shen, Z.; Huang, Y.; and Wan, H. 2023. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1585–1596. IEEE.

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

Han, L.; Du, B.; Sun, L.; Fu, Y.; Lv, Y.; and Xiong, H. 2021. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 547–555.

Ji, J.; Wang, J.; Jiang, Z.; Jiang, J.; and Zhang, H. 2022. STDEN: Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4, 4048–4056.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, 4, 4365–4373.

Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, 11906–11917. PMLR.

Li, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; and Li, Y. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, 17(1): 1–21.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR '18)*.

Long, Q.; Fang, Z.; Fang, C.; Chen, C.; Wang, P.; and Zhou, Y. 2024. Unveiling Delay Effects in Traffic Forecasting: A Perspective from Spatial-Temporal Delay Differential Equations. In *Proceedings of the ACM on Web Conference 2024*, 1035–1044.

Lütkepohl, H. 2005. *New introduction to multiple time series analysis*. Springer Science & Business Media.

Shao, Z.; Zhang, Z.; Wei, W.; Wang, F.; Xu, Y.; Cao, X.; and Jensen, C. S. 2022. Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting. *Proc. VLDB Endow.*, 15(11): 2733–2746.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, 01, 914–921.

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

Wang, B.; Wang, P.; Zhang, Y.; Wang, X.; Zhou, Z.; Bai, L.; and Wang, Y. 2024. Towards Dynamic Spatial-Temporal Graph Learning: A Decoupled Perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8, 9089–9097.

Wu, C.-H.; Ho, J.-M.; and Lee, D.-T. 2004. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4): 276–281.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.

Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; and Yin, B. 2021. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4927–4943.

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

Yu, Y.; Tang, X.; Yao, H.; Yi, X.; and Li, Z. 2019. Citywide traffic volume inference with surveillance camera records. *IEEE Transactions on Big Data*, 7(6): 900–912.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, 1.

Zhao, Y.; Luo, X.; Ju, W.; Chen, C.; Hua, X.-S.; and Zhang, M. 2023. Dynamic hypergraph structure learning for traffic flow forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 2303–2316. IEEE.

Zheng, C.; Fan, X.; Wang, C.; and Qi, J. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, 1234–1241.