ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240789

STPDN: Spatio-Temporal Pattern Decomposition Network with Fluctuation Awareness for Robust Traffic Flow Forecasting

Xudong Zhang[a,*], Peng Peng[a], Xuewen Chen[a], Yulei Wu[b], Lingdong Shen[a] and Haina Tang[a,]**

[a]School of Artificial Intelligence, University of Chinese Academy of Sciences
[b]School of Electrical, Electronic and Mechanical Engineering, University of Bristol

Abstract. The significance of traffic prediction in modern urban life has become increasingly prominent. Accurate traffic forecasting improves urban traffic management and enhances road resource utilization. In recent years, many models have introduced spatio-temporal contextual embeddings to distinguish between different time steps and spatial nodes. However, these models often overlook anomalous fluctuations in traffic data due to data imbalance. Consequently, performance declines when encountering uncommon situations, especially those caused by unexpected traffic accidents. To maintain overall performance while being aware of anomalous fluctuations, we propose STPDN, a dual-branch Spatio-Temporal Pattern Decomposition Graph Neural Network. Specifically, We introduce latent variables to characterize the distribution of latent patterns in traffic sequences, enabling the model to distinguish regular patterns and anomalous fluctuations without supervised information specifically targeting anomalies. Subsequently, we develop a resilient graph generator capable of producing dynamic spatio-temporal graphs, facilitating the propagation of impacts caused by dynamic fluctuations. Finally, we achieve more comprehensive and robust predictions by fusing regular patterns and anomalous fluctuations. Evaluation of real-world and simulated datasets shows that our model outperforms others, offering more reliable prediction solutions for urban traffic management systems, particularly in handling unforeseen traffic events. The code can be found at https://github.com/dhxdla/PyTorch-implementation-of-the-STPDN.git.

1 Introduction

Traffic congestion presents significant economic and environmental challenges in urban areas globally. In response to this pressing issue and to advance Intelligent Transportation Systems(ITS), the artificial intelligence community has turned its attention to traffic prediction. The objective is to forecast the future evolution of traffic systems by leveraging historical data and road networks [27, 32].

The complexity of traffic flow sequences, characterized by intricate spatio-temporal dependencies, renders traffic prediction a highly challenging task. Recently, researchers have made notable progress in this area by employing spatio-temporal graph neural networks [36]. These approaches typically utilize Graph Convolution Networks [11] to capture non-Euclidean spatial dependencies and incorporate sequence models or convolution operations to capture temporal dependencies. Despite these advancements, predefined graph structures in GCNs may fail to adapt to the evolving nature of traffic dynamics over time. To address this limitation, recent efforts have focused on designing data-driven dynamic spatio-temporal graphs through attention mechanisms [13, 5, 21]. However, this approach may introduce noisy connections into the network structure due to anomalous fluctuations within the traffic data.

Many studies have also recognized the presence of spatio-temporal regularities in traffic data. As depicted in Figure 1-a, traffic flow sequences exhibit significant periodicity and seasonality. Nodes often display similar characteristics during the same periods in different cycles, indicating the existence of temporal low-rank properties. To capture these regular patterns, some studies have introduced temporal embedding [26, 3]. Furthermore, as shown in Figure 1-b, similar trends within specific time windows among nodes suggest the presence of common patterns across different nodes. MEGACRN [7] and GMRL [4] respectively introduce meta-learning and Gaussian mixture models to capture this node similarity.

While the aforementioned methods have enhanced spatiotemporal prediction performance by capturing regular patterns, it is also crucial to address the presence of anomalous fluctuations. As depicted in Figure 1-c, irregular fluctuations are evident in traffic flow, suggesting the random occurrence of abnormal events. These methods introduce spatio-temporal embeddings to incorporate additional contextual information[26, 34], providing strong prior knowledge for specific time steps and spatial nodes. However, this prior knowledge may introduce bias. During the learning process, an abundance of regular patterns often overshadows the presence of anomalous fluctuations. While models often excel in learning regular patterns, they may struggle with abrupt events, as models tend to focus on capturing local expectations. Rethinking the problem of traffic prediction, while overall performance is certainly important, the most valuable predictions often lie in scenarios deviating from regular patterns. In this study, our focus lies on how to enhance the model's perception of dynamic anomalous fluctuations while ensuring the identification of regular patterns.

To address this, we propose the Spatio-Temporal Pattern Decomposition Network (STPDN), which partitions the prediction task into two subtasks: predicting regular patterns and forecasting dynamic anomalous fluctuations. Specifically, in the Low-Rank Pattern Aware Block, we introduce latent variables to characterize the distribution

* Email: zhangxudong221@mails.ucas.ac.cn
** Corresponding Author. Email: hntang@ucas.ac.cn

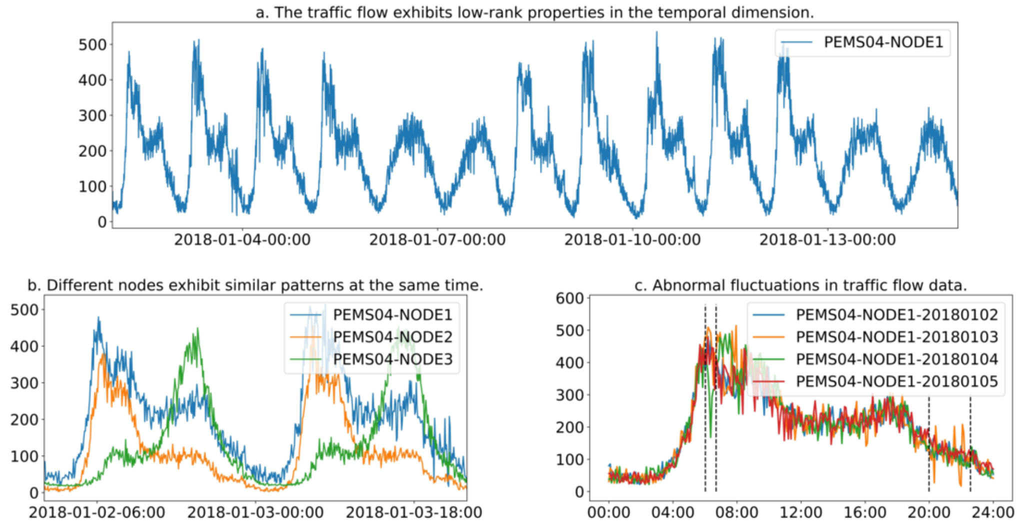

Figure 1. Illustrating Consistent Patterns and Anomalous Fluctuations in Traffic Data.

of traffic flow patterns. With prior knowledge that the majority of traffic are regular, we designed a learnable Latent Pattern Unit to simulate the distribution of latent variables, enabling the learning of regular traffic flow patterns without the need for any supervised information targeting anomalies. In the Fluctuation Pattern Aware Block, our emphasis lies on anomalous fluctuations within the traffic sequence. We design a resilient graph generator to generate dynamic spatio-temporal graphs at each time step unaffected by anomalous influences and propagate the effects of fluctuations using these graphs. Through the integration of regular patterns and anomalous fluctuations, our model achieves superior performance in capturing both expected predictions and unforeseen events, offering a promising solution for reliable urban traffic management systems. Our contributions can be summarized as follows:

- We propose a novel pattern decomposition architecture that divides the prediction task into subtasks: predicting regular patterns and forecasting dynamic anomalous fluctuations, ensuring stable predictions while capturing abnormal events.
- We introduce latent variables to represent patterns in historical information. By employing a Latent Pattern Unit to approximate the distribution of latent variables, we can distinguish between regular patterns and fluctuations without any supervised information tailored to anomalies.
- We design a resilient graph generator capable of generating dynamic spatio-temporal graphs unaffected by anomalies, which are then utilized to propagate information about dynamic anomalous fluctuations.
- In multiple real-world and simulated datasets with elevated noise levels, thorough comparisons with mainstream methods reveal that STPDN excels across various metrics and demonstrates robustness in perceiving abnormal fluctuations.

2 Related work

The transportation system stands as one of the pivotal infrastructures in modern cities. However, the gap between infrastructure service capacity and transportation demand has brought about a series of issues, including traffic congestion and resource wastage[31].

2.1 Traditional Traffic Prediction Methods

Early research on traffic prediction employed traditional statistical methods like ARIMA, and VAR, alongside machine learning techniques under the assumption of time series stationarity [19, 23]. Data-driven machine learning methods, renowned for their robust generalization capabilities, have been introduced into traffic prediction, yielding promising results[18]. Given the formidable nonlinear fitting capabilities of deep learning methods, these models have been widely applied to traffic prediction problems for modeling spatial and temporal correlations. Considering the inherent temporal features of traffic prediction tasks, methods based on RNNs and their variants have found extensive use in extracting temporal dependencies[22]. However, most of the aforementioned methods tend to overlook the acquisition of spatial dependencies. CNN-based methods partition the road network into grids, effectively capturing the spatial dependencies of adjacent grids[37, 17]. In contrast to CNNs, which can only aggregate information in Euclidean space, Graph Neural Networks [11] exhibit capabilities in processing graph-structured data. They fuse information from neighbors with connectivity relationships to enhance traffic prediction performance.

2.2 Spatio-Temporal Graph Neural Networks

Currently, research predominantly focuses on spatio-temporal graph neural networks for traffic prediction [36, 16, 20]. GCN heavily relies on a predefined topology graph, which has the issue of insufficient information. A substantial body of research has designed more rational spatio-temporal graphs, enabling more efficient aggregation of temporal and spatial information [14, 24]. Graph wavenet [34], for instance, addresses this challenge by learning adaptive node embeddings to construct dependency matrices. DMSTGCN, DSTAGNN [5, 13] are committed to developing data-driven dynamic spatio-temporal graphs to cope with the dynamic nature of the temporal dimension. STSGCN [28] constructs local spatio-temporal graphs, connecting individual spatial graphs at adjacent time steps, capable of simultaneously handling temporal and spatial dependencies. Traformer [8] constructs a large spatio-temporal graph, allowing information to flow directly between different nodes and steps.

With the existence of heterogeneity in traffic feature sequences, MegaCRN[7] draws inspiration from the Graph Memory Network [9, 35] and utilizes a module similar to Memory Bank [33, 30] to memorize typical patterns, generating a graph structure based on these patterns. (AST-GCN, GMAN, PDFormer, Trafformer, and GridFormer utilize self-attention in both temporal and spatial dimensions to obtain global spatio-temporal heterogeneity[25, 40, 6, 8, 29]. GMRL designs a Gaussian representation extractor to enhance features by extracting similar patterns[4].

The rapid evolution of urban anomaly detection technologies holds profound and contemplative implications for researchers engaged in the realm of traffic forecasting [39, 15]. Some researchers seeking to unravel the complexities of anomaly detection have embraced a bifurcated approach, employing dual-branch structures to discern abnormal patterns and achieve enhanced efficacy in dealing with abnormal fluctuation.[38].

3 Preliminary

Traffic data $\mathcal{X}$ pertains to spatio-temporal tensor information collected within a road network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{X} \in \mathbb{R}^{N \times L \times C}$. In this context, $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ represents the set of edges, and $\mathcal{A} \in \mathbb{R}^{N \times N}$ signifies the connectivity relations within the road network. Here, N signifies the number of spatial nodes, L denotes the temporal length, and C represents the feature dimension (e.g., traffic flow, traffic speed). The task at hand, traffic flow forecasting, involves utilizing historical traffic flow $X_t = \mathcal{X}_{t-T:t} \in \mathbb{R}^{N \times T \times C}$ to forecast future traffic flow $Y_t = \mathcal{X}_{t:t+T'} \in \mathbb{R}^{N \times T' \times C}$, where T is the historical length and T' is the future length.

4 Methodology

In this section, we initially present the comprehensive architecture of the Spatio-Temporal Pattern Decomposition Graph Neural Network. Subsequently, we offer a detailed exposition of its three pivotal modules: the ST Embedding Block, the Low-Rank Pattern Aware Block, and the Fluctuation Pattern Aware Block.

4.1 Overview

As depicted in Figure 2, the model adopts a dual-branch structure, with each branch dedicated to predicting the regular patterns and dynamic fluctuations respectively. Initially, spatio-temporal embedding is applied to the input data, introducing additional contextual spatio-temporal information. The Low-Rank Pattern Aware Block is utilized to extract regular patterns from spatio-temporal data and generate baseline predictions $\hat{Y}_p$. The Fluctuation Pattern Aware Block is responsible for recognizing dynamic fluctuations and propagating their impact through dynamic spatio-temporal graphs, resulting in anomaly-influenced predictions $\hat{Y}_r$. Finally, $\hat{Y}_p$ and $\hat{Y}_r$ are fused to obtain a more robust prediction $\hat{Y}_t = Linear(\hat{Y}_p + \hat{Y}_r)$, where $Linear$ represents the linear layer.

4.2 Spatio-Temporal Embedding Block

The spatio-temporal embedding approach is widely adopted for traffic prediction tasks, incorporating spatial, periodic, and temporal information. Initially, we map input sequences to a feature dimension D through a fully connected layer:

$$E_x = W_{ep} \cdot X_t + b_{ep} \tag{1}$$

where X_t is the input, W_{ep} and b_{ep} are both Learnable parameters. $E_x \in \mathbb{R}^{N \times D}$ represents the embedding.

We introduce learnable parameters E_w and E_d to capture weekly and daily periodic information and E_s for node-level characteristics. E_w represents the day of the week, and E_d indicates the time of day, incorporating meaningful prior information about the weekly and daily cycles [26]. For daily information (E_d), we initialized a learnable embedding of size $(288, D)$, dividing a day into five-minute intervals, with each component indicating the specific time of day for the current input. For weekly information (E_w), we initialized a learnable embedding of size $(7, D)$, with each component representing the specific day of the week for the current input. These embeddings are integrated as follows:

$$E_p = [E_x \parallel E_s \parallel E_w^i \parallel E_d^j] \tag{2}$$

where $\parallel$ denotes concatenation. E_p encapsulates low-rank spatio-temporal information. E_w^i and E_d^j represent the i-th and j-th components of the learnable periodic embeddings, signifying the current input feature's position in the weekly and daily cycles, respectively.

Through convolutional operations, we obtain E_{xt} to represent dynamic information for each time step, and fuse it with contextual embeddings to get E_r:

$$E_{xt} = \theta_{xt} \star X_t \tag{3}$$

$$E_r = [E_{xt} \parallel E_s \parallel E_w^i \parallel E_d^j \parallel E_{pos}] \tag{4}$$

where $E_{xt} \in \mathbb{R}^{N \times T \times D}$ is the embedding that encapsulates dynamic information for each time step,θ_{xt} denotes the 1D convolutional kernel, with $\star$ representing the convolution operation. We also introduce a positional embedding E_{pos} of size $\mathbb{R}^{T \times D}$ to incorporate position information, allowing the model to distinguish time steps and capture temporal dynamics.

4.3 Low-Rank Pattern Aware Block

Traffic data inherently exhibits diverse spatio-temporal distributions. It is crucial to examine the current state of latent patterns and forecast future traffic states based on these latent patterns. We address this by introducing latent variable z to indicate regular patterns, which transforms the problem into:

$$p_\theta(Y_p \mid X_t) = \int_z p_\theta(z \mid X_t) \, p_\theta(Y_p \mid z, X_t) \, dz \tag{5}$$

By leveraging the idea of variational inference to estimate the posterior distribution $p_\theta(z \mid X_t)$ [10], the problem can be further transformed into optimizing the ELBO:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z \mid X_t, Y_p)} [\log p_\theta(Y_p \mid z, X_t)] - \mathrm{KL}[q_\phi(z \mid X_t, Y_p) \parallel p(z)] \tag{6}$$

Where $q_\phi(z \mid X_t, Y_p)$ serves as the encoder, $p_\theta(Y_p \mid z, X_t)$ as the generator, and $p(z)$ as the prior distribution. Given that traffic data only contains finite common latent patterns, it's unnecessary to enumerate the entire pattern space by sampling $q_\phi(z \mid X_t, Y_p)$ using the reparameterization trick. Instead, we designed a self-learning Latent Pattern Unit to represent the entire latent pattern space $p(z)$. Since the data predominantly comprises regular patterns, the Latent Pattern Unit with limited size can effectively memorize the most common patterns. Maximizing the ELBO intuitively requires encoding $q_\phi(z \mid X_t, Y_p)$ to be as close as possible to the true distribution in the

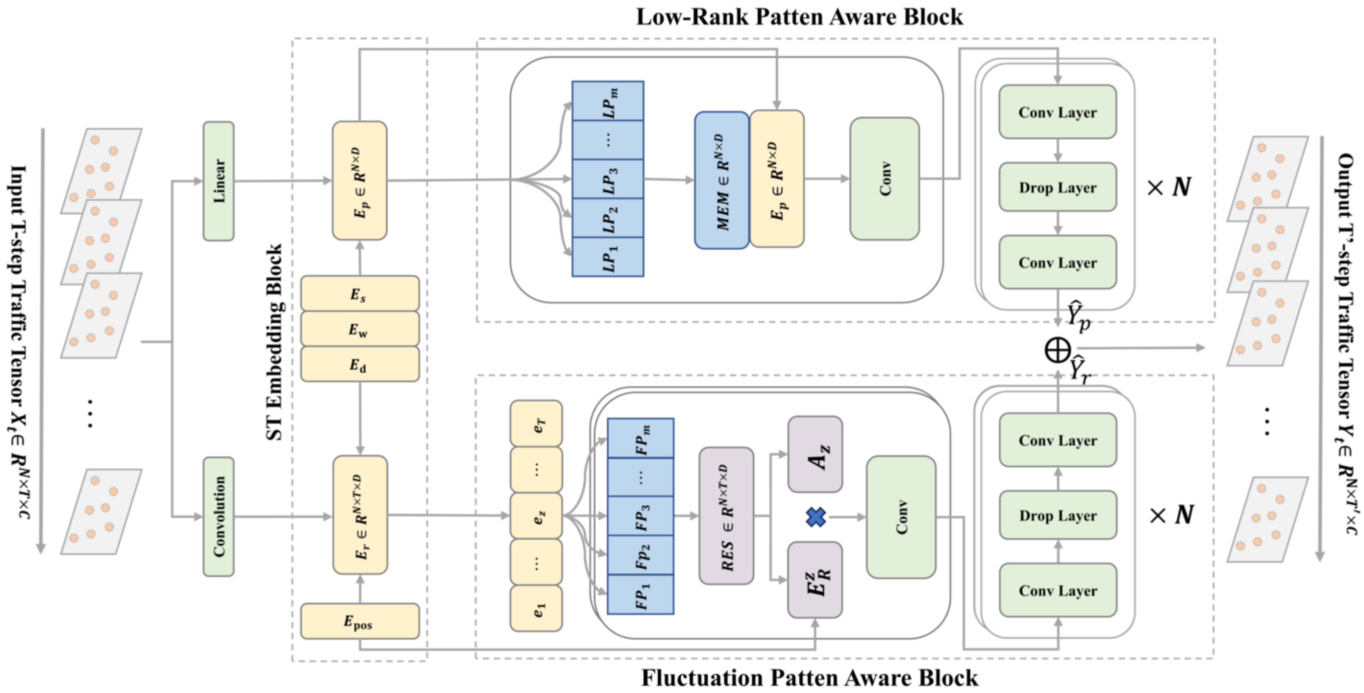

Figure 2. An Overview of the Architecture of the Spatio-Temporal Pattern Decomposition Graph Neural Network.

Latent Pattern Unit, thereby maximizing prediction accuracy based on this representation.

The overall structure is illustrated in Figure 2, depicting the establishment of the Low-Rank Pattern Aware Block for learning regular patterns and expected predictions. The Low-Rank Pattern Aware Block employs Latent Pattern Unit $LP \in \mathbb{R}^{m \times D}$ to indicate the regular patterns in traffic data. Here, m represents the number of items in the Latent Pattern Unit, and D represents the feature dimension. The Latent Pattern Unit, utilizing a minimal number of learnable parameters, provides a comprehensive representation of the overall traffic patterns, with each item representing a regular pattern. To ensure stability in calculating the KL divergence, the Latent Pattern Unit LP is first transformed into a probability distribution form:

$$M_{i,d} = \frac{e^{LP_{i,d}}}{\sum_{d=1}^{D} e^{LP_{i,d}}}, \tag{7}$$

where $LP_{i,d}$ represents the i-th item and d-th characteristic dimension in LP. Next, the similarity between the input embedding E_p and the distribution of items in the Latent Pattern Unit is measured.

$$KL_{i,j} = \sum_{d=1}^{D} M_{i,d} \ln \left(\frac{M_{i,d}}{E_p^{j,d}} \right), \tag{8}$$

where $E_p^{j,d}$ represents the d-th feature of the j-th node in the input embeddings, and $M_{i,d}$ represents the d-th feature of the i-th item. The magnitude of $KL_{i,j}$ measures the similarity between the features of the nodes j and the i-th item in the Latent Pattern Unit. A smaller $KL_{i,j}$ indicates greater similarity. After m calculations, m values of KL_j are obtained, representing the importance of each item for node j.

Then, the item most similar to the input is fused with the original input, achieving the goal of reducing noise and generating expectation prediction, as shown below:

$$E_M = \theta_m * [E_p \parallel MEM], \tag{9}$$

where $E_M \in \mathbb{R}^{N \times D}$ represents the feature fused with regular patterns, $MEM \in \mathbb{R}^{N \times D}$ represents the regular pattern features, $MEM_j = M_{\text{argmin}(KL_j)}$ denotes the item most similar to node j, θ_m is a 1D convolutional kernel, $*$ represents the convolution operation, and $\parallel$ denotes the concatenation operation.

After integrating the regular patterns into the features, further learning of latent features is achieved through multiple convolutional layers. This produces a comprehensive prediction of regular patterns, resulting in a more stable expected value:

$$H_{i+1} = \delta_h^i * (\text{drop}(\text{ReLU}(\theta_h^i * H_i))) + H_i, \tag{10}$$

where $\hat{Y}_p = H_{np}$, $H_0 = E_M$, np represents the number of all prediction modules, δ_h^i, θ_h^i are 1D convolutional kernels, drop represents the dropout operation, ReLU is the activation function. Additionally, we introduce a residual connection, using skip connections to obtain the input for the next layer.

4.4 *Fluctuation Pattern Aware Block*

As traffic data exhibits dynamics, each time step possesses its uniqueness. Therefore, we segment the input embedding $E_r \in \mathbb{R}^{N \times T \times D}$ along the time dimension, dividing it into $e_z \in \mathbb{R}^{N \times D}$, where $z \in [1, T]$. Each sliced tensor is then fed into the Fluctuation Pattern Aware Block for processing.

Given the presence of random incidents in traffic data, the impact of such events is often challenging to capture. The fluctuations generated by these events also propagate along the timeline and spatial structure. As illustrated in Figure 3, the Fluctuation Pattern

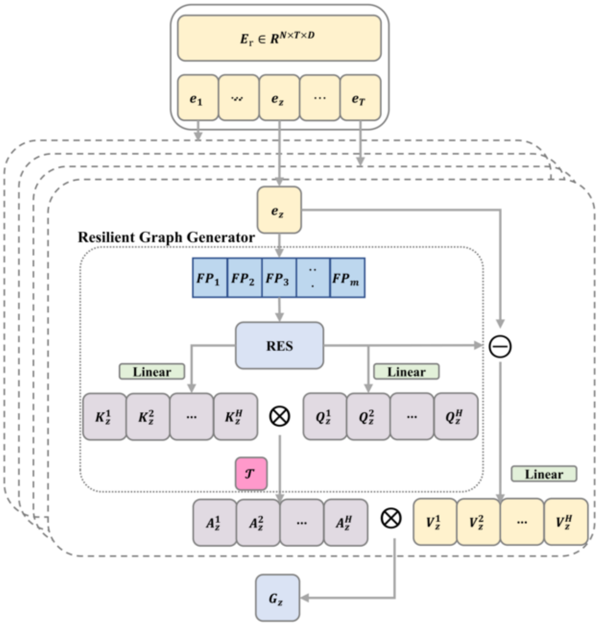

Figure 3. The Comprehensive Architecture of the Fluctuation Pattern-Aware Block.

Aware Block captures the fluctuation patterns and constructs a resilient graph generator to produce dynamic spatio-temporal graphs unaffected by anomalous events. Subsequently, the anomalous fluctuations are propagated through this graph.

The Fluctuation Pattern Aware Block begins by utilizing a designed Latent Pattern Unit $FP \in \mathbb{R}^{m\times D}$ to find the regular patterns for each sliced tensor e_z. The similarity between regular patterns and samples is then measured to filter out similar states.

$$KL_{i,j}^{z} = \sum_{d=1}^{D} R_{i,d} \ln\left(\frac{R_{i,d}}{e_z^{j,d}}\right), \tag{11}$$

where $KL_{i,j}^{z}$ represents the similarity between the j-th node and the i-th item in the z-th time slice. $R_{i,d} = e^{FP_{i,d}} / \sum_{d=1}^{D} e^{FP_{i,d}}$ denotes the feature d of item i in the Latent Pattern Unit, and $e_z^{j,d}$ represents the feature d of node j in the z-th time slice. Once the intrinsic information inherent in the samples is obtained, the next step involves comparing real samples with regular patterns. The differing parts are then fused with the input embedding to indicate the representation of sudden events at each time step.

$$E_R^z = \theta_r \star [e_z \parallel e_z - RES_z], \tag{12}$$

where $E_R^z \in \mathbb{R}^{N\times D}$ represents the embedding containing dynamic fluctuation information at the z-th time step, $e_z - RES_z \in \mathbb{R}^{N\times D}$ represents the filtered irregular fluctuation, $RES_z = FM_{\text{argmin}(KL_j^z)}$ denotes the regular pattern of node j in the z-th time slice, θ_r is a 1D convolutional kernel, $\star$ denotes the convolution operation, and $\parallel$ denotes the concatenation operation.

The resilient graph generator constructs dynamic spatio-temporal graphs for each time step using data that has filtered out anomalous fluctuations, thus ensuring the robustness of the graph structures. Simultaneously, the stable characteristic of the graph structure needs to be considered. We initially analyze the regular patterns RES_z^i of each node e_z^i in the current time step slice. By measuring the similarity between regular patterns of nodes, we determine the node connection relationships at each time step.

$$D_{i,j}^{z,h} = \frac{\left(W_q^{z,h} RES_z^i + b_q^{z,h}\right) \cdot \left(W_k^{z,h} RES_z^j + b_k^{z,h}\right)}{\sqrt{d}}, \tag{13}$$

where $D_{i,j}^{z,h}$ represents the relationship between nodes i and j on the z-th time slice of the h-th set of learnable parameters, and d represents the feature dimension. In the self-generated dynamic graphs, nodes with weak relationships still maintain connections with smaller weights. However, real traffic graphs are sparse. Therefore, we designed a self-learned filtering matrix to filter out non-essential connections.

$$A_{i,j}^{z,h} = hs\left(\mathcal{T}_{i,j}\right) \odot D_{i,j}^{z,h} \tag{14}$$

where $A_{i,j}^{z,h}$ represents the relationship between nodes i and j on the z-th time slice of the h-th set of learnable parameters, and $\odot$ denotes the element-wise product. $\mathcal{T}$ is a learnable parameter matrix of size $N \times N$ used to simulate a more informative graph structure matrix than a topology graph. We map $\mathcal{T}$ to a 0-1 matrix using the Hard Sigmoid function $hs()$, setting unimportant connections to zero, thus achieving graph structure sparsification. $\mathcal{T}_{i,j}$ represents the the relationship between nodes i and j.

Finally, based on the learned dynamic spatio-temporal graphs $A \in \mathbb{R}^{T\times N\times N\times H}$, the captured fluctuation information $E_R \in \mathbb{R}^{N\times T\times D}$ is propagated. This process combines dynamic graphs generated at each time step with graphs produced by each set of attention parameters.

$$G_0 = \sum_{h=1}^{H} w^h \sum_{z=1}^{T} E_R^z A^{z,h} + b^h, \tag{15}$$

where H represents the number of parameter sets, T is the length of the input sequence, and b^h and w^h are both learnable parameters.

After propagating the fluctuation information through the dynamic spatio temporal graphs, multiple convolutional layers with a skip connection structure are used to obtain the final fluctuation prediction $\hat{Y}_r = G_{np}$, where np represents the number of all convolution modules.

$$G_{i+1} = \delta_g^i \star (\text{drop}(\text{ReLU}(\theta_g^i \star G_i))) + G_i, \tag{16}$$

where $\hat{Y}_r = G_{np}$, $\star$ is convolution operation; δ_g^i and θ_g^i are 1D convolutional kernels.

5 Experiment

Table 1. Statistics of PEMS04, PEMS07, PEMS08 Noise401 and Noise402 Datasets.

Dataset	Time Range	Interval	Node	Ratio of Abnormal
PEMS04	1/1/2018 - 2/28/2018	5 min	307	-
PEMS07	5/1/2017 - 8/31/2017	5 min	883	-
PEMS08	5/1/2017 - 8/31/2017	5 min	170	-
Noise401	1/1/2018 - 2/28/2018	5 min	307	10%
Noise402	1/1/2018 - 2/28/2018	5 min	307	20%

In this section, we systematically evaluate the performance of our model across multiple real-world datasets and simulation datasets, comparing it with various methods. Our experimental results demonstrate that our approach achieves superior performance and can effectively identify abnormal fluctuations. Subsequently, we conduct a

Table 2. The Collective Performance Across PEMS04, PEMS07, and PEMS08. The table displays average performance metrics for 12-step predictions on the respective datasets.

model		DCRNN	STGCN	GWN	AGCRN	STFGNN	DSTAGNN	FOGS	MSDR	LightCTS	MegaCRN	**STPDN**
PEMS04	MAE	24.84	22.59	19.88	19.67	20.01	19.30	19.38	19.97	18.75	18.57	**18.54**
	MAPE	17.34%	14.52%	14.27%	12.93%	13.07%	12.64%	12.79%	13.04%	12.99%	12.67%	**12.31%**
	RMSE	38.33	35.47	32.41	32.23	32.40	31.52	31.61	32.32	30.32	30.30	**30.27**
PEMS07	MAE	25.36	24.64	21.26	21.05	22.05	21.90	21.04	22.19	20.81	19.73	**19.66**
	MAPE	11.72%	10.97%	9.26%	8.96%	9.36%	9.39%	8.79%	9.24%	8.91%	8.38%	**8.36%**
	RMSE	25.61	38.22	33.91	34.96	35.74	35.46	35.48	35.93	33.65	33.29	**33.14**
PEMS08	MAE	17.83	18.29	16.25	16.10	16.44	15.83	15.20	16.35	14.97	15.17	**14.41**
	MAPE	11.73%	11.50%	10.48%	10.33%	10.57%	9.94%	9.68%	10.25%	9.65%	9.89%	**9.35%**
	RMSE	27.93	27.74	26.72	25.54	26.19	25.04	25.22	25.82	24.10	24.14	**24.03**

series of ablation experiments to assess the individual contributions of each component in traffic forecasting. Lastly, we delve into parameter sensitivity experiments, specifically the Latent Pattern Unit. These experiments aim to explore the influence of Latent Pattern Unit size on predictive performance. Through rigorous experimentation, we strive to enhance our understanding of the intricacies and dependencies within the proposed model architecture.

5.1 Dataset

We conducted a series of experiments on three real-world datasets with varying numbers of nodes: PEMS04, PEMS07, and PEMS08 [28], all sourced from the Caltrans Performance Measurement System [2] in California. Given that traffic anomalies often have a duration, we designed four noise kernels with two different time lengths to simulate real-world traffic anomalies: gradual traffic increase/decrease and return to normal (10 steps), and sudden traffic surge/drop and return to normal (5 steps). Using PEMS04, we generated two simulation datasets, Noise401 and Noise402, which incorporated simulated pulse noise at proportions of 10% and 20%, respectively. This noise was introduced randomly into the real-world datasets and propagated along the physical topology graph, spreading to first-order neighbors with an intensity of 20%. Details of the noise data generation can be found in the code. The statistics of the four datasets used in our experiments are summarized in Table 1.

5.2 Baseline Model

We compared the performance of our model against several baseline models, and the experiments demonstrated that our model achieved superior performance. DCRNN models traffic flow using a directed graph and integrates GNN into RNN for spatio-temporal predictions [16]. STGCN proposes a spatio-temporal prediction framework using convolutional layers for faster training with fewer parameters [36]. GWN constructs a spatio-temporal graph matrix with adaptive node embeddings to address the lack of predefined graph structure in GNN [34]. AGCRN combines matrix factorization with node embedding to reduce computation while enhancing interpretability [1]. STFGNN designs a graph integrating spatio-temporal relationships for graph convolution [14]. DSTAGNN uses dynamic information from historical traffic data to replace static graphs with dynamic spatial-temporal aware graphs [13]. FOGS combines time similarity and topological graphs for novel graph generation [24]. MSDR improves RNN by considering previous hidden states more in time-dependent feature extraction, combining with GNN for spatio-temporal predictions [20]. MegaCRN introduces a meta-graph learner to address sequence non-stationarity [7]. LightCTS enhances computational efficiency with a stack of temporal and spatial operators [12].

5.3 Experimental Setup

For PEMS04, PEMS07, PEMS08, Noise401, and Noise402 datasets, we adopted a 6:2:2 ratio to split the data into training, validation, and test sets. All datasets were aggregated into 5-minute windows, resulting in 288 data points daily. Additionally, we standardized the input data to ensure training stability. In the prediction task, we utilized one hour of historical data as input to forecast the traffic state for the next hour (12 steps).

The model was implemented using PyTorch, and experiments were conducted on an NVIDIA A100-SXM4-80GB card. Hyperparameters were fine-tuned based on the model's performance on the validation datasets. The hidden dimension D is set to 256, the num heads H to 4, the number of prediction blocks np to 8, the learning rate to 0.001, and the batch size to 32.

We employed three widely used metrics in traffic prediction, namely MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and MAPE (Mean Absolute Percentage Error), to evaluate the overall performance of our model. Additionally, we used PDR (Performance Decreased Ratio), which represents the proportion of performance decrease in MAPE, to assess the magnitude of performance variation of the model under different noise levels.

5.4 Experimental Results

Table 2 presents a comprehensive performance comparison between our model and ten baseline models across the PEMS04, PEMS07, and PEMS08 datasets. The reported metrics, including MAE, MAPE, and RMSE, represent the averaged values for a 12-step prediction horizon. It's important to note that the calculation method for RMSE can vary across different articles. Some researchers calculate RMSE for each batch and then average across batches, which can be influenced by batch size and may result in slightly lower values than the true RMSE. We have standardized the RMSE calculation method by first computing the Mean Squared Error (MSE) for all samples, averaging across batches, and finally taking the square root to obtain RMSE. Due to our model's effective capture of regular patterns and its adept handling of dynamic anomalous fluctuations, experimental results demonstrate that our model outperforms others on all three datasets. On the PEMS04 and PEMS08 datasets, the Mean Absolute Percentage Error (MAPE) is approximately 3% higher compared to the best-performing baseline.

As shown in Table 3, our method maintains optimal performance even after introducing noise in Noise401 and Noise402 datasets. It's worth noting that with an increase in the proportion of noise, the

Table 3. The Collective Performance Across Noise401 and Noise402, Representing 10% and 20% Noise Addition on PEMS04, Respectively.

model	Noise401				Noise402			
	MAE	MAPE	RMSE	PDR	MAE	MAPE	RMSE	PDR
DCRNN	31.45	26.42%	48.62	52.37%	36.57	35.36%	59.39	103.92%
STGCN	26.49	21.45%	42.48	47.71%	30.60	32.11%	51.88	121.16%
GWN	21.08	19.04%	35.10	33.39%	25.03	25.90%	45.21	81.49%
AGCRN	20.87	16.78%	35.97	29.75%	24.83	22.69%	45.09	75.47%
DSTAGNN	21.52	16.78%	36.37	32.71%	25.74	23.89%	46.35	89.00%
LightCTS	21.90	18.47%	36.12	42.22%	25.89	23.49%	46.19	80.83%
MegaCRN	22.18	15.30%	35.78	20.76%	24.03	17.58%	44.88	38.75%
STPDN	**20.67**	**13.39%**	**35.16**	**8.76%**	**23.60**	**14.40%**	**44.32**	**16.96%**

performance of all models declines. However, under 10% noise, our model's performance decrease ratio (PDR) for MAPE is 8.76%, and under 20% noise, the MAPE performance decrease (PDR) is 16.96%. Our model exhibits the smallest performance degradation when faced with a significant amount of noise.

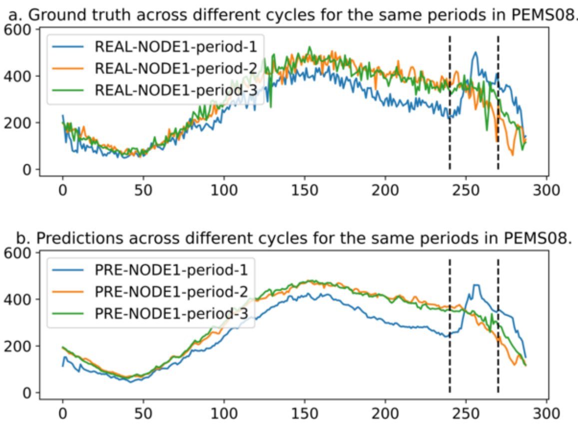

Figure 4. Fluctuations-Aware Traffic Flow Analysis: Ground Truth and Model Predictions for Node 1 Across Different Cycles

As illustrated in Figure 4, we present a typical case showcasing awareness of fluctuations. In Figure 4 a, we depict the ground truth of Node 1 during corresponding periods in different cycles. Notably, the portion within the black line reveals that compared to periods 2 and 1, period 3 exhibits distinct irregular fluctuations. In Figure 4 b, we present the corresponding prediction for this segment. It is evident that the portion within the black line effectively identifies abnormal fluctuations in the traffic flow sequence, indicating our model's capability to recognize and propagate the impact of exceptional fluctuations on traffic predictions. Additionally, our model demonstrates excellent predictive performance for regular patterns, attributed to the robust memory function of our Latent Pattern Unit concerning regular traffic patterns.

In summary, our model exhibits superior overall performance, excelling in both anomalous fluctuations handling and regular pattern prediction.

5.5 *Ablation Study*

We conducted nine ablation experiments on the PEMS04 dataset to investigate the impact of the Latent Pattern Unit, the overall framework, and the resilient graph generator on model performance.

w/o LPL: Removal of the Latent Pattern Unit in the Low-Rank Pattern Aware Block. **w/o FPL:** Elimination of the Latent Pattern Unit in the Fluctuation Pattern Aware Block, utilizing a data-driven approach based on the attention mechanism for dynamic graph generation. **w/o L:** Elimination of all the Latent Pattern Unit in our model. **w/o LP:** Omitte the Low-Rank Pattern Aware Block. **w/o FP:** Omitte the Fluctuation Pattern Aware Block. **w/o MG:** Substitution of dynamic multiple graphs with a single dynamic graph, removing the multi-head attention mechanism. **w/o DG:** Elimination of the computation of a spatio-temporal graph for each time step, using only one graph for all input time steps. **w/o STRU:** Removal of the learnable filtering matrix $\mathcal{T}$. **w/o STRU w TOPO:** Replace the filtering matrix $\mathcal{T}$ with a physical topology graph. Replacement of the learnable filtering matrix with a spatial topological graph.

Table 4. Ablation Study on the Role of Latent Pattern Unit in Models on the PEMS04 Dataset

Model	w/o LPL	w/o FPL	w/o L	w/o LP	w/o FP	STPDN
MAE	18.65	18.75	18.80	60.05	18.65	**18.54**
MAPE	12.33%	12.47%	12.56%	43.96%	12.41%	**12.31%**
RMSE	30.41	30.60	30.88	82.35	30.53	**30.27**

Starting with an examination of the overall structure, as depicted in Table 4, we observe that eliminating the Fluctuation Pattern Aware Block has minimal impact on overall performance. However, removing the Low-Rank Pattern Aware Block results in a significant performance decline. This suggests that while the Low-Rank Pattern Aware Block learns regular pattern-based baseline predictions, the Fluctuation Pattern Aware Block is more attuned to anomalous fluctuations. Furthermore, in Figure 5, we illustrate the impact of removing the Fluctuation Pattern Aware Block on prediction results. In a and b, represent typical traffic patterns, while c exhibits anomalous fluctuations denoted by the black dashed lines. Notably, after removing the Fluctuation Pattern Aware Block, the model notably loses its anomaly detection capability.

Table 5. Ablation Study on PEMS04 Investigating the Influence of Different Graph Structures on Model Performance.

	w/o MG	w/o DG	w/o STRU	w/o STRU w TOPO	STPDN
MAE	19.8	20.86	18.64	18.55	**18.54**
MAPE	13.35%	13.11%	12.44%	12.34%	**12.31%**
RMSE	31.4	32.9	30.39	30.33	**30.27**

In Table 4, we also observe the impact of the Latent Pattern Unit on prediction performance. It is evident that removing any individual branch of the Latent Pattern Unit or removing all Latent Pattern Units

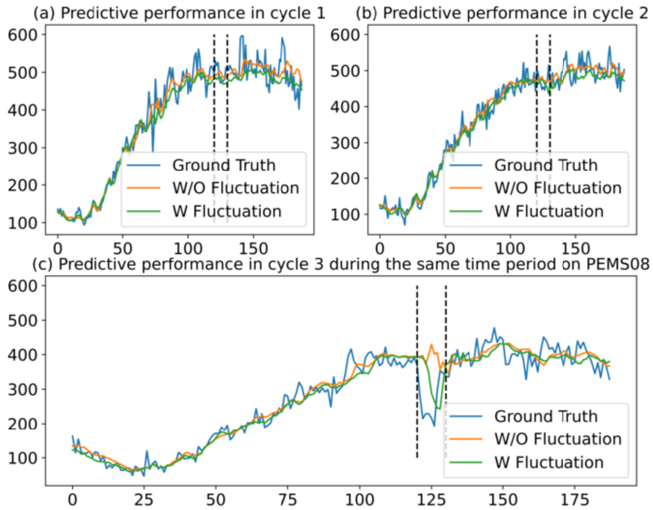

Figure 5. Effect of Removing Fluctuation Pattern Aware Block on Prediction Results.

results in a certain degree of performance decline. This underscores the importance of the Latent Pattern Unit in the model.

In problems involving the resilient graph generator, the importance of dynamic graphs becomes apparent in Table 5. Each time step possesses unique characteristics, making the use of a static graph an unreasonable approach. The introduction of multiple graphs and structural filters contributes to a slight enhancement in model performance. The use of a self-learning structural filter proves to be slightly more effective than directly employing a topological graph structure filter.

5.6 *Parameter Sensitivity Experiment*

In our model, a critical parameter is denoted by m, representing the number of items in the Latent Pattern Unit. We conducted experiments on the PEMS04 and PEMS07 to investigate the impact of m on prediction performance, as depicted in Table 6.

The model achieves optimal performance when m is set to 700 and 800. Values of m that are either too large or too small result in a decline in model performance. This implies that there exists a finite set of patterns within the entire dataset, and using a fixed number of patterns can fully characterize the entire dataset.

6 Conclusion

This paper introduces a dual-branch Spatio-Temporal Pattern Decomposition Graph Neural Network designed to address challenges posed by abnormal fluctuations. Our method focuses on capturing both regular patterns and dynamic anomalous fluctuations in traffic flow sequences, thus offering a comprehensive solution for urban traffic management systems. The first branch termed the Low-Rank Pattern Aware Block, utilizes the Latent pattern unit to capture regular patterns, enabling stable predictions. The Fluctuation Pattern Aware Block removes regular patterns to focus on fluctuations. The resilient graph generator constructed dynamic spatio-temporal graphs based on regular patterns to propagate fluctuation information. Integration of regular patterns and dynamic fluctuations ensures prediction stability and accurate forecasting. Our model was evaluated on multiple datasets, achieving state-of-the-art results. Overall,

Table 6. The Results of Parameter Sensitivity Experiment on PEMS04 and PEMS07.

M	PEMS04			PEMS07		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
M=600	18.9582	13.32%	30.4224	19.7535	8.38%	33.6923
M=700	18.6791	12.46%	30.5412	**19.6596**	8.36%	**33.1396**
M=800	**18.5424**	**12.31%**	**30.2750**	19.7105	**8.27%**	33.6576
M=900	18.5434	12.46%	30.3521	19.7083	8.29%	33.6476
M=1000	18.6425	12.56%	30.5468	19.7596	8.32%	33.5246

STPDN represents a significant step forward in traffic prediction research, offering a promising avenue for the development of intelligent transportation systems.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (NSFC) (Grant No.52071312).

References

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

[2] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001.

[3] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 269–278, 2021.

[4] J. Deng, J. Deng, R. Jiang, and X. Song. Learning gaussian mixture representations for tensor time series forecasting. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023.

[5] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong. 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*, KDD '21, page 547–555, New York, NY, USA, 2021.

[6] J. Jiang, C. Han, W. X. Zhao, and J. Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *AAAI*, 2023.

[7] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, and T. Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8078–8086, 2023.

[8] D. Jin, J. Shi, R. Wang, Y. Li, Y. Huang, and Y.-B. Yang. Trafformer: unify time and space in traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8114–8122, 2023.

[9] A. H. Khasahmadi, K. Hassani, P. Moradi, L. Lee, and Q. Morris. Memory-based graph networks. In *International Conference on Learning Representations*, 2020.

[10] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

[11] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[12] Z. Lai, D. Zhang, H. Li, C. S. Jensen, H. Lu, and Y. Zhao. Lightcts: A lightweight framework for correlated time series forecasting. *Proceedings of the ACM on Management of Data*, 1(2):1–26, 2023.

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

[14] M. Li and Z. Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4189–4196, 2021.

[15] Q. Li, X. Pan, F. Xiao, and B. Bhanu. ESSL: enhanced spatio-temporal self-selective learning framework for unsupervised video anomaly detection. In *Proceedings of the 26th European Conference on Artificial*

Intelligence (ECAI), volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 1398–1405. IOS Press, 2023.

[16] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[17] Y. Liang, K. Ouyang, L. Jing, S. Ruan, Y. Liu, J. Zhang, D. S. Rosenblum, and Y. Zheng. Urbanfm: Inferring fine-grained urban flows. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3132–3142, 2019.

[18] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai. Road traffic speed prediction: A probabilistic model fusing multi-source data. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1310–1323, 2017.

[19] M. Lippi, M. Bertini, and P. Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14 (2):871–882, 2013.

[20] D. Liu, J. Wang, S. Shang, and P. Han. Msdr: Multi-step dependency relation networks for spatial temporal forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1042–1050, 2022.

[21] X. Luo, C. Zhu, D. Zhang, and Q. Li. Dynamic graph convolutional network with attention fusion for traffic flow prediction. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 1633–1640, 2023.

[22] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

[23] B. Pan, U. Demiryurek, and C. Shahabi. Utilizing real-world transportation data for accurate traffic prediction. In *2012 ieee 12th international conference on data mining*, pages 595–604. IEEE, 2012.

[24] X. Rao, H. Wang, L. Zhang, J. Li, S. Shang, and P. Han. FOGS: first-order gradient supervision with learning-based graph for traffic flow forecasting. In L. D. Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 3926–3932, 2022.

[25] S. M. V. Reddy, A. Dammur, and A. Narasimhamurthy. An attention based spatial temporal graph convolutional networks for traffic flow prediction. *International Journal of Intelligent Systems and Applications in Engineering*, 12(15s):481–488, 2024.

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

[27] M. Shaygan, C. Meese, W. Li, X. G. Zhao, and M. Nejad. Traffic prediction using artificial intelligence: review of recent advances and emerging opportunities. *Transportation research part C: emerging technologies*, 145:103921, 2022.

[28] C. Song, Y. Lin, S. Guo, and H. Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 914–921, 2020.

[29] C. Su, C. Wu, and D. Lian. Gridformer: Spatial-temporal transformer network for citywide crowd flow prediction. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 2218–2225. IOS Press, 2023.

[30] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[31] S. Wang, J. Cao, and S. Y. Philip. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 34(8):3681–3700, 2020.

[32] Y. Wang and Y. Luo. Weagan: Weather-aware graph attention network for traffic prediction. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 2591–2598, 2023.

[33] J. Weston, S. Chopra, and A. Bordes. Memory networks. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph wavenet for deep spatial-temporal graph modeling. In S. Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1907–1913, 2019.

[35] J. Xu, E. Dai, X. Zhang, and S. Wang. Hp-gmn: Graph memory networks for heterophilous graphs. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1263–1268, 2022.

[36] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640, 7 2018.

[37] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[38] M. Zhang, T. Li, H. Shi, Y. Li, and P. Hui. A decomposition approach for urban anomaly detection across spatiotemporal data. In *IJCAI International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 2019.

[39] M. Zhang, T. Li, Y. Yu, Y. Li, P. Hui, and Y. Zheng. Urban anomaly analytics: Description, detection, and prediction. *IEEE Transactions on Big Data*, 8(3):809–826, 2020.

[40] C. Zheng, X. Fan, C. Wang, and J. Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020.