A Seasonal-Trend Dynamic Interactive Integration Network for Traffic Forecasting

Jianuo Ji[1], Hongbin Dong[1](✉), and Xiaoping Zhang[2]

[1] College of Computer Science and Technology,
Harbin Engineering University, Harbin, China
donghongbin@hrbeu.edu.cn
[2] Traditional Chinese Medicine Data Center,
China Academy of Chinese Medical Sciences, Beijing, China

Abstract. As a fundamental technology in Intelligent Transportation Systems, accurate traffic prediction has become a critical challenge for real-time applications. Traffic series typically exhibit both seasonal fluctuations and long-term trends, and the complex interaction between these components presents significant forecasting challenges. However, existing methods often either model the raw time series directly, thereby overlooking the entangled effects or perform decomposition without fully exploiting the interactions between the decomposed components. Moreover, many methods rely on a single or predefined graph structure to characterize spatial associations, limiting their ability to capture dynamic spatial dependencies effectively. To this end, we propose a Seasonal-Trend Dynamic Interactive Integration Network, namely STDIINet, for accurate traffic prediction. Specifically, the trend and seasonal parts of the time series data, are decoupled, modeling each according to its distinct characteristics. For the seasonal component, we employ a dynamic spatial learning module based on a graph construction method, combined with a temporal convolution module, to capture seasonal features. For the trend component, we utilize a multilayer perceptron to extract long-term trend information. Furthermore, we design a dynamic interactive integration framework that integrates these components while supporting bidirectional dynamic fusion between trend and seasonal features, thereby enhancing the model's capability to learn complex spatio-temporal representations. Experiments on three real-world traffic datasets demonstrate that STDIINet outperforms the state-of-the-art methods.

Keywords: Spatio-Temporal data mining · Traffic forecasting · Series decomposition

1 Introduction

With the rapid proliferation of location-based services and the widespread adoption of smart mobile devices, the volume of spatio-temporal data has grown exponentially, posing significant challenges to Intelligent Transportation Systems (ITS). Among the core tasks in ITS, traffic flow prediction has garnered considerable attention from both academia and industry. Accurate traffic prediction

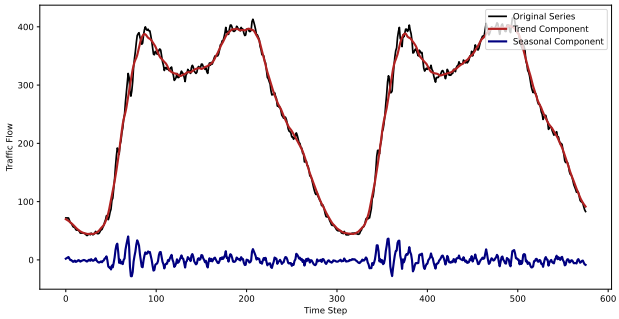

Fig. 1: Decomposition of time series (from Traffic Flow in PeMS08 dataset).

has been shown to play a pivotal role in a range of transportation-related applications, such as route planning [4,15], vehicle dispatching, and traffic congestion mitigation [12], enhancing decision-making efficiency and service quality.

Traffic prediction aims to forecast future traffic conditions based on historical traffic data. Such data can be collected from various sources, including road sensors, surveillance cameras, and GPS devices, and these observation nodes collectively form an integrated transportation network [21]. Consequently, traffic data exhibit complex spatio-temporal characteristics with dynamic correlations. In recent years, deep learning methods have gained increasing popularity in the field of traffic prediction, gradually replacing traditional statistical approaches [16,10]. These methods are particularly advantageous for handling complex nonlinear relationships, offering superior generalization capabilities and predictive accuracy. Among them, Spatio-Temporal Graph Neural Networks (STGNNs) have shown remarkable performance. STGNNs typically integrate hybrid neural structures [13,24] and attention mechanisms [25,5] to effectively capture spatial dependencies. Meanwhile, Recurrent Neural Networks (RNNs) [9,8] and Temporal Convolutional Networks (TCNs) [19,18] are often employed to model temporal correlations.

Although considerable progress has been made in leveraging spatio-temporal data for traffic prediction, most existing methods overlook a common phenomenon in real-world traffic systems: traffic data is essentially a composite signal composed of multiple temporal components. Many current approaches oversimplify the modeling of time series data, primarily capturing surface-level spatial and temporal variations while failing to account for the more intricate underlying patterns. As illustrated in Fig. 1, real-world traffic time series often exhibit both periodic seasonal fluctuations and long-term trend variations. The trend component reflects general, continuous changes in traffic flow over time, such as a gradual increase or decrease, while the seasonal component captures recurring patterns driven by human travel behaviors, including morning and evening rush hours, or weekday–weekend differences. These components typically coexist and

are superimposed, creating a complex mixed signal that poses a significant challenge to accurately modeling spatio-temporal dependencies. Furthermore, most existing methods rely on a single or predefined static graph structure to represent spatial correlations, which limits their ability to capture the dynamic nature of spatial relationships in traffic networks.

To address the above issues, in this paper, we propose a Seasonal-Trend Dynamic Interactive Integration Network, namely STDIINet, for accurate traffic prediction. Specifically, we employ a time series decomposition module to disentangle the input traffic data into seasonal and trend components, which are then modeled separately. For the seasonal component, we introduce a Dynamic Spatial Learning (DSL) module based on a graph construction approach, coupled with a temporal convolution (TConv) module to effectively capture periodic fluctuations. The dynamic graph structure enables the model to better explore dynamic spatial relationships. In contrast, the trend component is processed using a multilayer perceptron (MLP) to extract long-term temporal patterns. To fully leverage the unique characteristics of both components, we propose a dynamic interactive integration framework that supports bidirectional fusion between seasonal and trend features. This framework allows the trend component to offer a global evolutionary context for guiding seasonal fluctuations, while the seasonal component provides localized feedback to refine trend modeling. This interactive design significantly enhances the model's capacity to capture complex spatio-temporal dependencies. In summary, the main contributions of this paper are as follows.

- A dynamic graph construction method is developed to model the evolving associations within the transportation network, enabling the effective extraction of unique and latent spatial relationships among nodes.
- To comprehensively capture the spatio-temporal characteristics of seasonal fluctuations and trend components, we propose a seasonal-trend dynamic interactive integration module. This module leverages TConv, DSL, and MLP to extract features from the decomposed time series. Furthermore, a fusion unit is introduced to enable dynamic interactions, allowing the trend to guide seasonal variations and seasonal features to refine the trend, thereby enhancing forecasting accuracy.
- We conduct extensive experiments on three real-world traffic datasets, and the results demonstrate that our model significantly outperforms the baseline methods. Moreover, various experimental analyses are performed to validate the superior performance of STDIINet and to verify the effectiveness of each individual module.

2 Related Work

Traffic prediction is inherently a spatio-temporal forecasting problem, sharing similarities with tasks such as wind speed prediction and pedestrian flow estimation. Traditional statistical methods like ARIMA [16] and SVM have been

widely adopted for time series prediction. However, due to their inability to incorporate spatial information, they struggle to handle the complexities of spatio-temporal data. In recent years, deep learning approaches have become increasingly popular for modeling the nonlinear and complex nature of traffic data. Convolutional Neural Networks (CNNs) have been applied to traffic prediction by treating traffic flow readings as grid-based images, where each cell records the number of vehicles passing through during a given time interval [11,20]. This representation allows spatial correlations between regions to be captured using techniques originally developed for image recognition [14]. To model temporal dependencies more effectively, Recurrent Neural Networks (RNNs), with their ability to retain sequence information, were introduced and rapidly adopted for traffic flow prediction [22]. Subsequently, with the rising interest in graph-based data representations, Graph Neural Networks (GNNs) have become a mainstream approach for traffic forecasting due to their superior ability to model spatial dependencies in non-Euclidean domains [6,13]. Several notable GNN-based models have emerged in this context. DCRNN [9] integrates diffusion graph convolution into the GRU architecture to enhance spatio-temporal modeling. GWNet [19] and AGCRN [1] employ learnable node embeddings, initializing them randomly and optimizing them through gradient descent to generate adaptive adjacency matrices. MTGNN [18] introduces a graph learning module that constructs dynamic graphs by computing similarities between learnable node embeddings. DSTAGNN [7] further enhances spatio-temporal modeling by constructing dynamically-aware graphs that replace traditional static structures. DGCRN [8] leverages hypernetworks and dynamic filters to utilize and extract evolving node attributes, thereby generating dynamic graphs that better reflect the data's inherent variability. However, most models learn spatio-temporal relationships under entangled time series, which limits the performance of current spatio-temporal prediction methods.

3 Preliminaries

Definition 1 (Traffic Topology Graph). In a given road network, a traffic topology graph is indicated by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ represents the set of nodes ($|\mathcal{V}| = N$), each node corresponding to a road sensor; $\mathcal{E}$ represents the set of edges, indicating the physical connectivity between sensors; and $\mathcal{A} \in \mathbb{R}^{N \times N}$ is defined as the adjacency matrix of the graph, with elements indicating the connectivity between pairs of nodes.

Definition 2 (Traffic Signal Tensor). The traffic signal tensor of N nodes over P time steps is denoted by $\mathcal{X} = (\mathrm{X}_1, ..., \mathrm{X}_t, ..., \mathrm{X}_P) \in \mathbb{R}^{P \times N \times C}$. Here, $\mathrm{X}_t \in \mathbb{R}^{N \times C}$ refers to the traffic signals of N nodes at a specific time step t, and C denotes the number of traffic-related features (e.g., $C = 3$ for traffic flow, speed, and occupancy). The traffic prediction problem is formalized as Eq. 1. Given a series of observations from the N sensors over the past P time steps,

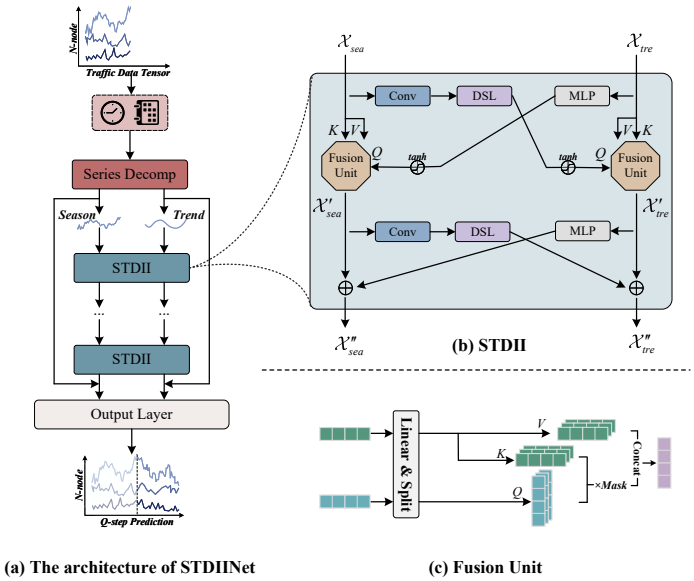

Fig. 2: Detailed framework of STDIINet.

the objective is to predict the traffic signals for the subsequent Q time steps using a mapping function $\mathcal{F}$:

$$[\mathrm{X}_{t+1}, ..., \mathrm{X}_{t+Q}] = \mathcal{F}([\mathrm{X}_{t-P+1}, ..., \mathrm{X}_t]) \tag{1}$$

4 Methodology

The proposed STDIINet framework is illustrated in Fig. 2, and its technical details are elaborated in this section. Initially, the raw input data is processed by a temporal embedding layer, which enhances the periodicity and trends in the traffic data and provides a higher-dimensional representation. A time series decomposition module is employed to decouple the data into seasonal and trend components, which are then modeled separately through the Seasonal Trend Dynamic Interactive Integration (STDII) module. The seasonal component is processed using the Dynamic Spatial Learning (DSL) module, based on a graph construction method, and the Time Convolution (TConv) module, which captures periodic seasonal fluctuations. The trend component is modeled using a Multi-Layer Perceptron (MLP) for long-term trend extraction. Subsequently, a fusion unit is employed to facilitate bidirectional dynamic fusion, enabling the trend component to influence seasonal fluctuations and vice versa. After stacking multiple STDII modules, the final prediction results are generated through the output layer.

4.1 Temporal Embedding

Temporal embedding is designed to implicitly represent temporal features in traffic data, thereby enhancing the model's ability to capture temporal patterns and cyclical variations. Let $T^w \in \mathbb{R}^{N_w \times d}$ and $T^d \in \mathbb{R}^{N_d \times d}$ denote the learnable embedding matrices for the day of the week and time of day, respectively, where $N_\mathrm{w} = 7$ is the number of days in a week. N_d represents the number of samples collected per day by the sensors. For instance, with a sampling interval of 5 minutes, $N_d = 288$ samples are collected per day. $W_t \in \mathbb{R}^P$ and $D_t \in \mathbb{R}^P$ denote the day-of-week and time-of-day indices, respectively, for the traffic time series within the interval $t - P + 1 : t$. These indices are used to extract the corresponding embeddings $T_t^w \in \mathbb{R}^{P \times d}$ and $T_t^d \in \mathbb{R}^{P \times d}$ from their respective embedding matrices. T^D and T^W are obtained by broadcasting these embeddings and subsequently summed to form the periodic embedding $E_T \in \mathbb{R}^{P \times N \times d}$ of the time series:

$$E_T = T^D + T^W \tag{2}$$

Next, we transform the input into a high-dimensional representation to facilitate the exploration of more complex spatio-temporal features, obtaining $\mathcal{X}_{data}$ in series with the periodic embedding E_T to get $\mathcal{X}_{emb} \in {}^{P \times N \times 2d}$:

$$\mathcal{X}_{emb} = \mathcal{X}_{data} \parallel E_T \tag{3}$$

where d denotes the feature dimension. For convenience, we use $\mathcal{X}$ instead of $\mathcal{X}_{emb}$in the sequel.

4.2 Series Decomposition

Sequence decomposition [3] has been applied in time series forecasting models, such as Autoformer [17] and FEDformer [26], to capture complex temporal patterns. To enhance prediction accuracy, the traffic series is decomposed into trend and seasonal components, representing the long-term trend and underlying periodicity, respectively. Specifically, a moving average method is applied to smooth out short-term fluctuations, and the seasonal component is derived by subtracting the trend from the original time series. As shown in Eq. 4, the trend and seasonal components $\mathcal{X}_{tre}, \mathcal{X}_{sea} \in \mathbb{R}^{P \times N \times D}$ can thus be obtained for input data $\mathcal{X} \in \mathbb{R}^{P \times N \times D}$ of length P.

$$\begin{aligned} \mathcal{X}_{tre} &= AvgPool(padding(\mathcal{X}), w) \\ \mathcal{X}_{sea} &= \mathcal{X} - \mathcal{X}_{tre} \end{aligned} \tag{4}$$

where $AvgPool(\cdot)$ is a moving average operation with a window size of w and uses a padding operation to keep the length of the sequence constant.

4.3 Temporal Convolution

The Temporal Convolution (TConv) module captures temporal correlations in the time series. A larger kernel $(1, k_1)$ in the first layer models long-term dependencies, while smaller kernels $(1, k_2)$ in the second layer emphasize short-term

features. The formulation is as follows:

$$H_t = Conv2d(Conv2d(ReplicationPad(H))) \tag{5}$$

where $H \in \mathbb{R}^{D\times N\times P}$ and $H_t \in \mathbb{R}^{D\times N\times P}$ are the input and output of the TConv module.

4.4 Dynamic Spatial Learning

The current graph structure for modeling spatial correlation suffers from several limitations. First, it is difficult to effectively capture the dynamic nature of traffic patterns. In addition, most graph structures primarily focus on latent spatial features among nodes while overlooking the unique characteristics of individual nodes. To address these challenges, the Dynamic Spatial Learning (DSL) module has been proposed. This module employs a dynamic graph generator to learn real-time spatio-temporal dependencies, offering a data-driven representation of node relationships. Furthermore, latent and unique node-specific information is concurrently extracted through diffusion-based graph convolution. The detailed operation of DSL is summarized below.

Dynamic Graph Generator First, the core spatial features of the transportation network are extracted through the dynamic adjustment of node connections using an adaptive graph structure. The initialized node embedding matrices $E_1 \in \mathbb{R}^{N\times d_e}$ and $E_2 \in \mathbb{R}^{N\times d_e}$ are utilized to generate an adaptive adjacency matrix that characterizes the fundamental spatial correlations, where the embedding dimension is denoted as d_e, and the correlation matrix is computed as follows:

$$A_{adp} = E_1 E_2^{\mathrm{T}} \tag{6}$$

This adjacency matrix acts as the foundational structure for subsequent spatial modeling and enables the dynamic capture of latent spatial dependencies. A data-driven strategy for dynamic graph construction is subsequently employed. The representation $H_t \in \mathbb{R}^{D\times N\times P}$, which encodes temporal dependencies through the TConv module, is initially transformed through a nonlinear mapping via a fully connected layer to yield a hidden representation $\hat{H}_t \in \mathbb{R}^{D\times N}$, in which the temporal dimension has been removed, as defined below:

$$\hat{H}_t = FC(\sum_{t=1}^{P} H_t[..., t]) \tag{7}$$

Next, the spatial similarity of $\hat{H}_t$ with itself is computed, i.e., the degree of correlation between nodes is measured. This computation yields a dynamic adjacency matrix $A_H \in \mathbb{R}^{N\times N}$. The procedure for generating this matrix is defined as follows:

$$A_H = \frac{\exp(\hat{H}_t^{\mathrm{T}}[..., i]\hat{H}_t[..., j])}{\sum_{j=1}^{N} \exp(\hat{H}_t^{\mathrm{T}}[..., i]\hat{H}_t[..., j])} \tag{8}$$

The matrix A_H is capable of adapting to real-time traffic data, thereby more accurately capturing the dynamic correlations among traffic states. The matrix A_{adp} is used to extract potential spatial dependencies. These two dynamic neighbor matrices reflect spatial correlations between nodes from complementary perspectives. The two matrices are put through a cascade operation. The fused dynamic adjacency matrix $A_{adp} \in \mathbb{R}^{N \times N}$ is then obtained through a fully connected layer:

$$A_{dyn} = FC(A_{adp} \parallel A_H) \tag{9}$$

Spatial Information Aggregation The aggregated adjacency matrix A_{dyn} is then fed into the diffusion-based GCN for dynamic spatial information aggregation:

$$H_s = \sum_{k=0}^{K} A_{dyn}^k H_t W \tag{10}$$

where H_s represents the output of the DSL module, K indicates the number of diffusion iterations, and W refers to the learnable weights of the graph convolution layer. This process facilitates the effective capture of both local and global spatial correlations, thereby enabling a deeper exploration of dynamic spatial associations.

4.5 Seasonal-Trend Dynamic Interactive Integration

Modeling of Seasonal Components Given the input $\mathcal{X}_{sea} \in \mathbb{R}^{P \times N \times D}$, the Temporal Convolution (TConv) module is first applied to extract temporal features from the frequently fluctuating data. Subsequently, the Dynamic Spatial Learning (DSL) module is employed to capture spatial features. The specific operations can be summarized as follows:

$$\hat{\mathcal{X}}_{sea} = DSL(TConv(\mathcal{X}_{sea})) \tag{11}$$

Modeling of Trend Components Given the input $\mathcal{X}_{tre} \in \mathbb{R}^{P \times N \times D}$, we introduce a multilayer perceptron (MLP) to predict future trends:

$$\hat{\mathcal{X}}_{tre} = ReLU(W\mathcal{X}_{tre} + b)) \tag{12}$$

where W and b are learnable parameters.

Dynamic Interactive Integration To integrate the seasonal and trend components, A Seasonal-Trend Dynamic Interactive Integration (STDII) module is proposed, as illustrated in Fig. 2(b). In contrast to the generally smooth and slow-varying trend series, seasonal series typically exhibit periodic fluctuations with dynamically varying amplitudes. Although these periodic variations are rich in information, they can hinder the stability of long-term trend modeling. To address this issue, a Fusion Unit is introduced, as depicted in Fig. 2(c), to

preserve meaningful trend and seasonal features and integrate them effectively, thereby enhancing the stability of forecasting.

The core of this module lies in the design of the Query and Key-Value selection strategy. In this study, interaction-based fusion is formulated under two distinct scenarios:

- Scenario 1: Trend-Guided Interaction. The trend sequence is designated as the Query, while the seasonal sequence serves as the Key and Value. This configuration enables long-term trends to guide the extraction of seasonal features, thereby mitigating potential prediction biases caused by excessive seasonal fluctuations.
- Scenario 2: Seasonal-Guided Interaction. The seasonal sequence is utilized as the Query, and the trend sequence is employed as both the Key and Value. This approach acknowledges that seasonal fluctuations may influence long-term trends, thereby enhancing the model's capacity for dynamic adaptation and preventing performance degradation due to over-smoothing.

To support bidirectional dynamic fusion between trend and seasonal features, the fusion unit is designed to function as a "relay station." The initial interaction process is defined as follows:

$$\begin{aligned} \mathcal{X}'_{sea} &= FU(\tanh(\hat{\mathcal{X}}_{tre}), \mathcal{X}_{sea}, \mathcal{X}_{sea}) \\ \mathcal{X}'_{tre} &= FU(\tanh(\hat{\mathcal{X}}_{sea}), \mathcal{X}_{tre}, \mathcal{X}_{tre}) \end{aligned} \tag{13}$$

where $\mathcal{X}'_{sea}$ and $\mathcal{X}'_{tre}$ represent the trend-guided seasonal feature representation and the seasonal-guided trend feature representation, respectively, while $FU\,(Q, K, V)$ denotes the fusion unit function. To mitigate potential information loss caused by the initial fusion process, a second round of interactive fusion is conducted. This additional step aims to further enhance the synergistic modeling capability between the trend and seasonal components.

$$\begin{aligned} \mathcal{X}''_{sea} &= \hat{\mathcal{X}}'_{tre} + \mathcal{X}'_{sea} \\ \mathcal{X}''_{tre} &= \hat{\mathcal{X}}'_{sea} + \mathcal{X}'_{tre} \end{aligned} \tag{14}$$

where $\mathcal{X}''_{sea}$ and $\mathcal{X}''_{tre}$ represent the outputs of the STDII module. The output from the current STDII module is used as the input to the subsequent STDII module. By stacking multiple STDII modules with residual connections, the model's capacity to capture both broad and deep spatio-temporal features is significantly enhanced.

4.6 Output Layer

The obtained outputs are concatenated to form H_f, which are then passed through a Gated Linear Unit (GLU). The GLU is employed to fuse and refine the global spatio-temporal representations while also filtering and correcting the learned features. Finally, a fully connected regression layer is applied to the GLU output to produce the final prediction. The overall process is defined as follows:

Table 1: Hyperparameter settings.

Dataset	Batch Size	Learning Rate	Weight Decay	d	n
NYCBike	64	0.001	0.0001	64	3
NYCTaxi	64	0.001	0.0001	64	3
PeMS08	64	0.001	0.0001	96	2

$$H_f = \text{Concat}(\mathcal{X}''_{sea}, \mathcal{X}''_{tre}) \tag{15}$$

$$\mathcal{Y} = FC(ReLU(FC(H_f) \odot \sigma(FC(H_f)))) \tag{16}$$

where σ denotes the Sigmoid function and $\mathcal{Y}$ denotes the predicted result.

5 Experiment

5.1 Settings

Datasets & Processing We utilize two real-world transportation demand datasets [23], NYCBike and NYCTaxi, as well as a highway traffic flow dataset [13], PeMS08. The NYCBike dataset records daily shared bicycle demand at NYC docking stations, and NYCTaxi captures taxi trip data, both at 30-minute intervals. The PeMS08 dataset, from California freeways, provides traffic flow data at 5-minute intervals, with 12 samples per hour. All datasets are normalized using Z-score normalization to improve training efficiency.

In line with the most contemporary solution, all datasets are divided into training, validation, and test sets in the ratio of 6:2:2. In addition, we perform multistep prediction on the past 12 steps of data to predict the 12 steps of traffic data.

Implementation Details The proposed model is implemented using PyTorch 1.10.1 and trained on an NVIDIA RTX 4060 GPU. Model training is conducted using the Adam optimizer with an initial learning rate of 0.001, a batch size of 64, and a maximum of 200 epochs. The convolution kernel sizes k_1, and k_2 of TConv are set to 5,3. The feature dimension d is selected from the set {32, 64, 96, 128}, and the number n of stacked STDII modules is chosen from {1, 2, 3, 4}. The optimal configuration is determined based on validation set performance. Table 1 summarizes the optimal key hyperparameter settings for STDIINet across different datasets.

Evaluation Metrics We evaluate the model performance using three commonly adopted metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Among these, MAE is employed as the loss function during training.

Table 2: Performance comparison of different models on NYCBike and NYCTaix.

Model	NYCBike Drop-off			NYCBike Pick-up			NYCTaix Drop-off			NYCTaix Pick-up		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DCRNN	1.96	2.94	51.42%	2.09	3.30	54.22%	5.19	9.63	37.78%	5.40	9.71	35.09%
STGCN	2.01	3.07	50.45%	2.08	3.31	53.12%	5.38	9.60	39.12%	5.71	10.22	36.51%
STSGCN	2.73	4.50	57.89%	2.36	3.73	58.17%	5.62	10.21	37.92%	6.19	11.14	39.67%
GWNet	1.95	2.98	50.40%	2.04	3.20	53.08%	5.03	8.78	35.63%	5.43	9.39	37.79%
AGCRN	2.06	3.19	51.91%	2.16	3.46	56.35%	5.45	9.56	40.67%	5.79	10.11	40.40%
MTGNN	1.94	2.91	50.47%	2.03	3.19	53.73%	5.02	8.76	37.62%	5.39	9.41	37.21%
GMAN	2.09	3.00	54.82%	2.20	3.35	57.34%	5.09	8.95	35.00%	5.43	9.47	34.39%
DSTAGNN	2.33	3.54	58.62%	2.34	3.62	58.73%	5.01	8.74	35.81%	5.32	9.12	35.51%
STG-NCDE	2.28	3.42	60.96%	2.15	3.97	55.49%	5.38	9.74	40.45%	6.24	11.25	43.20%
MegaCRN	2.18	3.30	61.42%	2.31	3.59	67.07%	5.07	9.11	35.08%	5.47	9.96	35.13%
DGCRN	1.96	2.93	51.99%	2.06	3.21	54.06%	5.14	9.39	35.09%	5.44	9.82	35.78%
PDFormer	2.18	3.29	56.83%	2.29	3.58	59.23%	5.12	8.76	37.45%	5.47	9.33	37.83%
STDIINet	**1.89**	**2.81**	**50.05%**	**2.01**	**3.14**	**52.67%**	**4.91**	**8.70**	**33.98%**	**5.16**	**8.96**	**34.01%**

Table 3: Performance comparison of different models on PeMS08.

Dataset	Metric	DCRNN	STGCN	STSGCN	GWNet	AGCRN	MTGNN	GMAN	DSTAGNN	STG-NCDE	MegaCRN	DGCRN	PDFormer	STDIINet
PeMS08	MAE	18.49	18.79	17.88	15.57	15.95	15.12	14.87	15.67	15.45	14.68	14.60	14.34	**13.70**
	RMSE	27.30	28.20	27.36	24.32	25.22	24.23	24.06	24.77	24.81	23.68	24.16	23.68	**22.85**
	MAPE	11.69%	10.55%	11.71%	10.32%	10.09%	9.65%	9.77%	9.94%	9.92%	9.53%	9.33%	9.88%	**9.09%**

Baselines The baselines are DCRNN [9], STGCN [24], STSGCN [13], GWNet [19], AGCRN [1], MTGNN [18], GMAN [25], DSTAGNN [7], STG-NCDE [2], MegaCRN [6], DGCRN [8], and PDFormer [5].

5.2 Performance Comparison

Based on the results presented in Table 2 and 3, several key observations can be made. First, our proposed model, STDIINet, consistently outperforms all baseline algorithms across all evaluation metrics on the four datasets. Traditional methods such as STGCN, DCRNN, and AGCRN rely on predefined graph structures, which constrain their ability to capture dynamic spatio-temporal dependencies from varying input samples. In contrast, models like DSTAGNN and DGCRN, which employ dynamically generated graphs, demonstrate improved performance, highlighting the importance of modeling dynamic spatial relationships. While attention-based models such as GMAN and PDFormer exhibit strong representational power, they still face inherent limitations due to the performance bottlenecks of the attention mechanism itself.

In contrast, STDIINet introduces a novel approach by explicitly decomposing time series into seasonal and trend components, enabling bidirectional informa-

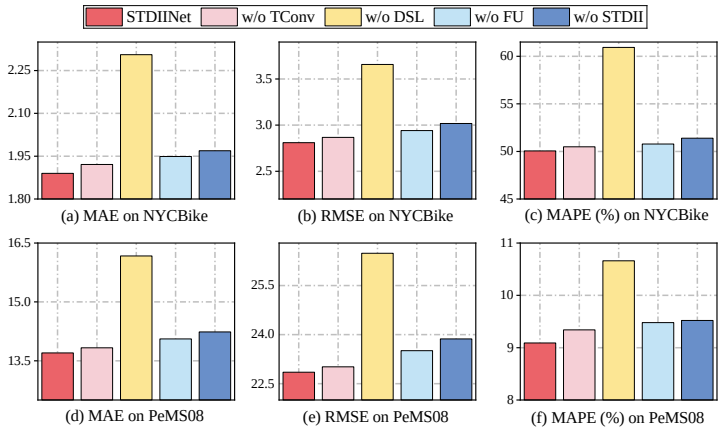

Fig. 3: Ablation Study on NYCBike and PeMS08.

tion exchange between the trend and seasonal components. In this mechanism, the trend component provides a global temporal context to guide seasonal pattern learning, while the seasonal component offers local feedback to refine trend modeling. These integrated innovations collectively enhance the model's capability to learn complex spatio-temporal dependencies and significantly improve forecasting performance.

5.3 Ablation Study

To further evaluate the effectiveness of each component in STDIINet, we conducted ablation studies on the NYCBike and PeMS08 datasets. For the NYCBike dataset, we report results for drop-off demand only. We design four model variants to assess the contribution of key modules:

- **w/o TConv** removes the TConv module to assess its impact on temporal feature extraction.
- **w/o DSL** removes the DSL module to evaluate its role in capturing spatial dependencies.
- **w/o FU** replaces the Fusion Unit with the traditional matrix product.
- **w/o STDII** replaces the STDII with a sequential modeling strategy, eliminating the interaction and integration between seasonal and trend components.

The results are shown in Fig. 3, we can conclude the following: w/o TConv leads to a significant decline in model performance, underscoring its critical role in capturing seasonal temporal patterns. Even more notably, w/o DSL results in a marked performance drop, confirming the effectiveness of our proposed spatial learning approach in modeling dynamic spatial dependencies. Furthermore, w/o FU weakens the bidirectional interaction between trend and seasonal features,

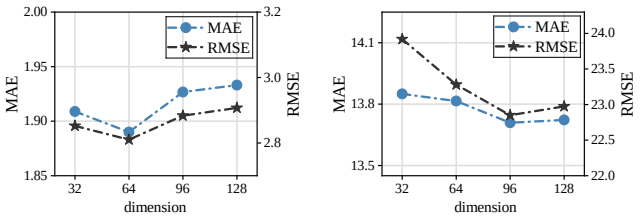

Fig. 4: Analysis on dimension.

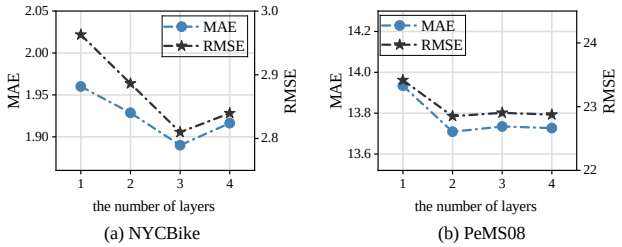

Fig. 5: Analysis on the number of layers.

resulting in suboptimal information integration and reduced overall representational capacity. Similarly, w/o STDII also leads to noticeable performance degradation, demonstrating the advantages of our proposed interaction framework in achieving feature complementarity and coordinated evolution.

5.4 Parameter Sensitivity Analysis

To further examine parameter sensitivity, we conducted a hyperparameter study on the PeMS08 and NYCBike datasets, focusing on two key hyperparameters: the feature dimension d and the number of stacked STDII modules n. For NYCBike, we report only drop-off demand results.

The experimental results are presented in Fig. 4 and 5. First, we observe that the performance of STDIINet gradually improves as the feature dimension increases, since richer features better capture complex spatio-temporal patterns. However, beyond a certain threshold, performance gains plateau or even decline. Next, we evaluate model depth by varying the number of stacked STDII modules. As noted in Section 4.5, STDII fuses trend and seasonal features. Results show that three layers yield the best performance, while deeper models increase training time and overfitting, degrading results.

6 Conclusion

In this paper, we propose STDIINet, a novel model for time series decomposition and interactive integration. STDIINet first decomposes traffic time series into

seasonal and trend components. These components are then processed within a dynamic interactive integration framework that captures spatio-temporal dependencies while enabling two-way information exchange between the trend and seasonal features. Specifically, the trend component provides a global evolutionary context to guide seasonal fluctuations, while seasonal fluctuations offer localized feedback corrections to the trend, enhancing the model's capability to capture complex spatio-temporal dynamics. Experimental results on several real-world traffic datasets demonstrate that STDIINet significantly outperforms existing baseline models, validating the effectiveness of the proposed approach. In future work, we plan to extend STDIINet to other spatio-temporal prediction tasks, such as marine environment forecasting and wind speed prediction.

Acknowledgments. This work was funded by the National Natural Science Foundation of China, Grant Number 82374621.

References

1. Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems **33**, 17804–17815 (2020)
2. Choi, J., Choi, H., Hwang, J., Park, N.: Graph neural controlled differential equations for traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 6367–6374 (2022)
3. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I., et al.: Stl: A seasonal-trend decomposition. J. off. Stat **6**(1), 3–73 (1990)
4. Dan, T., Pan, X., Zheng, B., Meng, X.: Double hierarchical labeling shortest distance querying in time-dependent road networks. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). pp. 2077–2089. IEEE (2023)
5. Jiang, J., Han, C., Zhao, W.X., Wang, J.: Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 4365–4373 (2023)
6. Jiang, R., Wang, Z., Yong, J., Jeph, P., Chen, Q., Kobayashi, Y., Song, X., Fukushima, S., Suzumura, T.: Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 8078–8086 (2023)
7. Lan, S., Ma, Y., Huang, W., Wang, W., Yang, H., Li, P.: Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In: International conference on machine learning. pp. 11906–11917. PMLR (2022)
8. Li, F., Feng, J., Yan, H., Jin, G., Yang, F., Sun, F., Jin, D., Li, Y.: Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. ACM Transactions on Knowledge Discovery from Data **17**(1), 1–21 (2023)
9. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
10. Lippi, M., Bertini, M., Frasconi, P.: 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)
11. Ouyang, K., Liang, Y., Liu, Y., Tong, Z., Ruan, S., Zheng, Y., Rosenblum, D.S.: Fine-grained urban flow inference. IEEE transactions on knowledge and data engineering **34**(6), 2755–2770 (2020)

12. Ren, T., Zhou, X., Li, K., Gao, Y., Zhang, J., Li, K.: Efficient cross dynamic task assignment in spatial crowdsourcing. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). pp. 1420–1432. IEEE (2023)
13. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 914–921 (2020)
14. Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F.M., Qin, A.K.: A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. IEEE Transactions on Knowledge and Data Engineering **34**(4), 1544–1561 (2020)
15. Tian, W., Shi, J., Luo, S., Li, H., Xie, X., Zou, Y.: Effective and efficient route planning using historical trajectories on road networks. Proceedings of the VLDB Endowment **16**(10), 2512–2524 (2023)
16. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. Journal of transportation engineering **129**(6), 664–672 (2003)
17. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems **34**, 22419–22430 (2021)
18. Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 753–763 (2020)
19. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121 (2019)
20. Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.: Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 5668–5675 (2019)
21. Ye, J., Zhao, J., Ye, K., Xu, C.: How to build a graph-based deep learning architecture in traffic domain: A survey. IEEE Transactions on Intelligent Transportation Systems **23**(5), 3904–3924 (2020)
22. Ye, J., Sun, L., Du, B., Fu, Y., Tong, X., Xiong, H.: Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 305–313 (2019)
23. Ye, J., Sun, L., Du, B., Fu, Y., Xiong, H.: Coupled layer-wise graph convolution for transportation demand prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 4617–4625 (2021)
24. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
25. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 1234–1241 (2020)
26. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning. pp. 27268–27286. PMLR (2022)