# Rethinking Attention Mechanism for Spatio-Temporal Modeling: A Decoupling Perspective in Traffic Flow Prediction


Qi Yu
North China University of Technology
Beijing, China
yuqi_april@mail.ncut.edu.cn

Weilong Ding*
North China University of Technology
Beijing, China
dingweilong@ncut.edu.cn

Hao Zhang
North China University of Technology
Beijing, China
2023322030127@mail.ncut.edu.cn

Yang Yang
North China University of Technology
Beijing, China
2023312120113@mail.ncut.edu.cn

Tianpu Zhang
SINOPEC Beijing Research Institute of Chemical Industry
Beijing, China
zhangtianpu@hotmail.com



## Abstract

The attention mechanism has the advantage of handling long-term correlations, and has been widely adopted in multivariate time series (MTS) prediction. As an important application of MTS, traffic flow prediction has the most popular solution using transformer-based prediction models nowadays. Just with attention mechanism, those models can learn the spatio-temporal correlations from traffic data. However, the up-to-date linear prediction models have questioned the effectiveness of current transformer-based models in certain conditions, which provides new possibilities for more efficient work. We rethink the role of the attention mechanism during spatio-temporal modeling from a decoupling perspective, and propose DEC-Former for traffic flow prediction. Specifically, the trend and seasonal parts of the time series data, the geographical adjacency of the nodes in the road network, and the traditional encoder-decoder architecture, are respectively decoupled. Such decoupling leverages the attention mechanism's advantage to capture long-term and long-range correlations. From extensive experiments on four real-world datasets, our work proves better predictive performance and efficiency than state-of-the-art attention-based models. Two case studies further show the distinct real effects.



## CCS Concepts

• **Information systems** → **Spatial-temporal systems**.

## Keywords

Attention Mechanism, Spatio-Temporal Representation, Traffic Prediction



*Weilong Ding is the corresponding author.





*CIKM '24, October 21–25, 2024, Boise, ID, USA*





ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679571

**ACM Reference Format:**
Qi Yu, Weilong Ding, Hao Zhang, Yang Yang, and Tianpu Zhang. 2024. Rethinking Attention Mechanism for Spatio-Temporal Modeling: A Decoupling Perspective in Traffic Flow Prediction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679571


## 1 Introduction

The attention mechanism, due to its superior performance, has been widely adopted in spatio-temporal modeling. The Graph Attention Network (GAT) [26] calculates the attention coefficients of neighbor nodes to extract the spatial correlation among nodes in graph topology. With the popularity of Transformer [25], many works have employed multi-head attention mechanism to extract temporal correlations of different time steps on time series. In addition to those spatial or temporal attentions, recent Transformer variants [36, 37] have performed attention calculations in Fourier or wavelet spaces, and learn global features better for multivariate time series (MTS) prediction.

Traffic flow prediction, as an important application of MTS in real scenarios, has its own characteristic when modeling the temporal correlations within the time series and capturing the spatial dependencies among the time series. On the one hand, as shown in Fig. 1(a), various traffic flow patterns among nodes in a road network have to be considered. Graph Neural Networks (GNNs) models usually use the geographical adjacency among nodes of road network to capture the spatial correlations of time series. However, as shown in Fig. 1(b), for the neighbor nodes in different traffic flow patterns, such a correlation assumption via geographical adjacency would result in insufficient predictive effect. On the other hand, dynamic long-term temporal correlations from time series have to be considered as well. As intuitively shown in Fig. 1(c) and (d) by distinguishing the trend and seasonal parts of the traffic flow, the traffic flow patterns of a specific node on different days (e.g., weekdays, weekends, or holidays) show various trends, and the traffic flow has obvious periodicity from a long-term perspective.

However, current attention-based models for traffic flow prediction still have limitations when extracting spatio-temporal correlations from time series. Firstly, attention mechanism generally

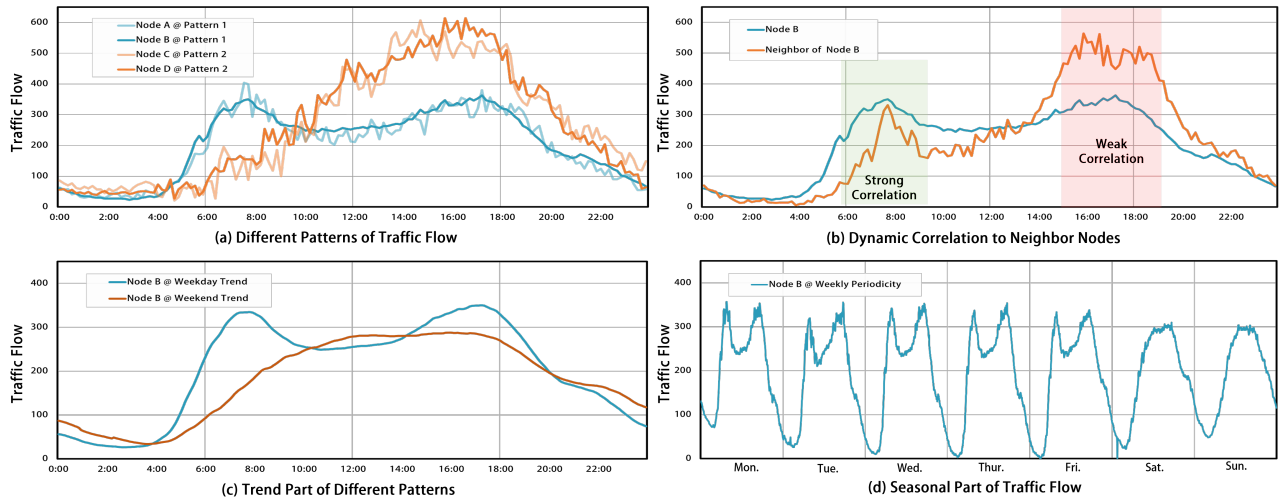


**Figure 1: Spatio-Temporal Correlations of Time Series (from Traffic Flow in PeMS07 dataset).**

exhibits poor generalization ability for time series data with trends. The up-to-date linear models [15, 22, 32] has questioned the effectiveness of those popular models. That necessitates a rethinking to leverage attention mechanism capturing long-term correlations. Secondly, the attention calculation is too heavy to emphasize its efficiency. Attention-based models have to extract spatial correlations either calculating attention scores for all global nodes or calculating for specific neighbor nodes in the geographical context. The former brings a lot of unnecessary computational overhead since not all those nodes contribute correlative traffic patterns. The latter neglects the long-range spatial correlations, i.e., the pattern correlations among nodes even geographically non-adjacent.

To address the above challenges, we propose a traffic flow prediction model from a DECoupling perspective, namely DEC-Former. For the temporal correlations within time series, we *decouple* the data into trend and seasonal parts, using Multi-layer Perceptron (MLP) and Fourier attention mechanism, respectively. For the spatial correlations among time series, we *decouple* the geographical adjacency of road network to obtain the pattern correlations among nodes based on historical traffic flow. In addition, we *decouple* the classical encoder-decoder architecture of Vanilla Transformer [25]: remove the cross-attention module, use the encoder to learn the features of input sequence and get the prediction results, and employ the decoder to optimize the prediction elements. In summary, the main contributions of this paper are as follows.

- Our DEC-Former could leverage attention better to capture long-term and long-range correlations from a decoupling perspective through two modules. The trend decomposition module decouples the time series data into trend and seasonal parts, modeling these two parts separately. The traffic pattern extraction module captures the historical traffic flow patterns among different nodes.
- We re-position the attention module ingeniously to improve its calculative efficiency. Attention is employed only for the seasonal part of time series, while a linear model is applied for the trend part. A dynamic spatial attention module is designed then to learn the pattern correlation among topological nodes that change over time.
- We evaluate our model on four real-world datasets. From the experimental results, our DEC-Former not only outperforms the state-of-the-art models on three predictive metrics, but also demonstrates better efficiency on two computational metrics.

## 2 Preliminaries

In this section, we first present the task definition , and then briefly review the key idea of attention mechanism.

### 2.1 Definition and Problem Statement

**Definition 1 (Traffic Topology Graph).** *A traffic topology graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ within certain road network. Here, $\mathcal{V}$ represents the nodes set ($|\mathcal{V}| = N$) and each node corresponds to a road sensor; $\mathcal{E}$ represents the edge set, indicating the physical connectivity of sensors; and $\mathcal{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of this graph, whose element is the connectivity between any node pair in the graph.*

**Definition 2 (Traffic Flow Tensor).** *The traffic flow tensor of $N$ nodes over $T$ time steps is presented as $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_t, \ldots, \mathbf{X}_T\} \in \mathbb{R}^{N \times C \times T}$. Here, $\mathbf{X}_t \in \mathbb{R}^{N \times C}$ represents the traffic flow of $N$ nodes in the road network at time step $t$, and $C$ is the number of traffic features (e.g.,$C = 3$ for features traffic flow, speed, and occupy.)*

The traffic prediction problem can be formalized as Formula 1. Given a series of observations from $N$ sensors over past $T$ time steps in a graph $\mathcal{G}$, our objective is to predict the traffic for next $P$ time steps through a mapping function $\mathcal{F}$.

$$[\mathbf{X}_{t+1}, \cdots, \mathbf{X}_{t+P}] = \mathcal{F}_\theta([\mathbf{X}_{t-T+1}, \cdots, \mathbf{X}_t; \mathcal{G}]) \tag{1}$$

### 2.2 Attention Mechanism

The attention mechanism is a fundamental operation in our model. Its core idea is to assign different weights to different parts of

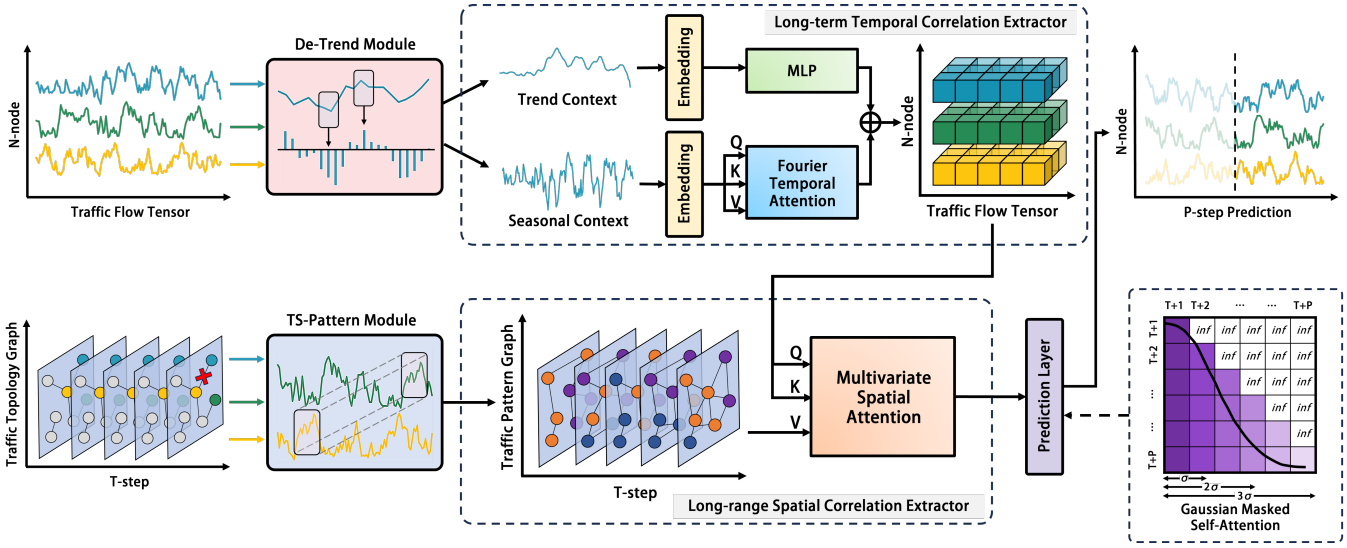


**Figure 2: The Framework of DEC-Former. The input traffic flow tensor is decomposed into trend and seasonal parts through Trend Decomposition (De-Trend) module, and those two parts are modeled separately to extract temporal correlation. The input traffic topology graph derives a pattern matrix through Traffic Pattern Extraction (TS-Pattern) module, and that pattern matrix assists in extracting spatial correlation.**

the input data, thereby highlighting important information and ignoring unimportant information [2]. The basic operation of the attention mechanism is to map a query and a set of key-value pairs to an output, where the query, key, value, and output are all vectors. The output is the weighted sum of the values, and the weight of each value is determined by the corresponding key and query together. Each weight represents the relationship strength between the query and each key-value pair.

Among the many attention functions, Scaled Dot-Product Attention [25] is a common type. Its weight is the dot product between the query and the value, so it has attractive features such as spatial and temporal efficiency. Its definition is shown in Formula 2, where $Q, K, V$, and $d_{model}$ are the query, key, value, and their dimensions, respectively.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_{model}}})V \tag{2}$$

## 3 Methodology

The framework of our DEC-Former is demonstrated in Fig. 2 and its technical details are discussed in this section. DEC-Former consists of several key modules. First, we preprocess the input data from both temporal and spatial dimensions. From a temporal perspective, we utilize the Trend Decomposition (De-Trend) module to decompose the input traffic flow tensor into trend and seasonal parts. From a spatial perspective, the input traffic topology graph is processed by the Traffic Pattern Extraction (TS-Pattern) module to derive a pattern matrix. Next, we embed the decomposed data in the spatio-temporal domain to capture static local spatio-temporal correlations. Then, we use MLP modules and Fourier temporal attention to extract long-term temporal correlations from trend and seasonal parts separately. We further employ the pattern matrix and a multivariate spatial attention module to capture long-range spatial correlations. Finally, a linear layer transforms the output from the spatio-temporal correlation extractor into prediction results.

Our DEC-Former model incorporates three independent attention computations. Fourier temporal attention is utilized to extract periodic correlations in the temporal data, as shown in Figure 3(a), while multivariate spatial attention captures pattern correlations among road network nodes, as shown in Figure 3(b). Additionally, we introduce a Gaussian masked self-attention at the prediction layer (as shown in Figure 2) to enhance accuracy, enabling the attention mechanisms to operate more effectively. These enhancements highlight how our model enables attention mechanisms to exert their maximum effectiveness in suitable contexts.

### 3.1 Trend Decomposition Module

Time series data typically contains long-term trends, seasonal variations, cyclical fluctuations, and irregular fluctuations [1]. We primarily focus on the trend and seasonal parts. The trends of traffic flow on a weekday typically exhibits a notable surge before the morning peak, while conversely, a decline is observed after the evening peak, as shown in Fig. 1 (c). The seasonal characteristics of traffic flow exhibit obvious daily and weekly periodicity, implying that traffic experiences daily regular fluctuations, like Fig. 1 (d). The time series decomposition, such as STL decomposition [3], has been applied to prediction models for time series like Autoformer [29] and FEDformer [37] to infer complex time patterns. Furthermore, TDformer [33] empirically demonstrates that attention models result in significant predictive errors to extrapolate linear trends, when learning historical context. Consequently, to improve accuracy, we decompose traffic flow data into trend and

seasonal parts, apply the attention mechanism only to the seasonal part, and use a linear model for the trend part.

Specifically, moving averages is adopted to extract long-term trend patterns, and the seasonal part is obtained by subtracting the trend from the original time series. As Formula 3, $\mathcal{X}_{tre}, \mathcal{X}_{sea} \in \mathbb{R}^{N\times C\times T}$, the trend and seasonal parts, are achieved respectively for input data $\mathcal{X}_{input} \in \mathbb{R}^{N\times C\times T}$ of length $T$. Here, $AvgPool(\cdot)$ is a moving average operation with window size $m$, and the operation *padding* keeps the sequence length unchanged.

$$\begin{aligned} \mathcal{X}_{tre} &= AvgPool(padding(\mathcal{X}_{input}), m), \\ \mathcal{X}_{sea} &= \mathcal{X}_{input} - \mathcal{X}_{tre}. \end{aligned} \tag{3}$$

## 3.2 Spatio-Temporal Embedding Layer

In the data embedding layer, a spatio-temporal embedding mechanism is designed to process traffic flow tensor $\mathcal{X}_*$, where $*$ is *tre* or *sea*. First, a fully connected network is utilized to map input into a high-dimensional representation $\mathcal{X}_* \in \mathbb{R}^{N\times D\times T}$, where $D$ is the dimension of the linear projection. Subsequently, through temporal feature embedding, we obtained temporal embedding $TE(\mathcal{X}_*)$ with both daily and weekly periodic patterns. Traffic flow experiences various fluctuations at different times in a day: high values during the daytime but stably low at night. Traffic flow is also affected by day types due to weekly periodicity, and traffic shows its similarity on an ordinal day of a week. We concatenate past $T_w$ steps traffic flow of the same ordinal day in previous week as $\mathcal{X}_{Tw} \in \mathbb{R}^{D\times Tw}$, and concatenate past $T_d$ steps traffic flow within a previous day as $\mathcal{X}_{Td} \in \mathbb{R}^{D\times Td}$. Thus, a temporal embedding $TE(\mathcal{X}_*) \in \mathbb{R}^{D\times(T+Tw+Td)}$ of the traffic flow tensor with periodic time series features is obtained. Then, by leveraging multiple layers of Graph Convolutional Network (GCN) to aggregate neighborhoods, we obtained a spatial embedding $SE(\mathcal{X}_*) \in \mathbb{R}^{N\times D}$ of the traffic flow tensor representing static spatial relationship. Additionally, the temporal position encoding $PE(\mathcal{X}_*)$ of the original transformer architecture [25] is retained to introduce the position of input sequence. Ultimately, the output $\mathcal{X}_{emb,*}$ of the data embedding layer can be calculated as Formula 4, after the input's spatio-temporal embedding.

$$\mathcal{X}_{emb,*} = \mathcal{X}_* + TE(\mathcal{X}_*) + SE(\mathcal{X}_*) + PE(\mathcal{X}_*). \tag{4}$$

## 3.3 Long-term Temporal Correlation Extractor

The trend decomposition module decomposes the input of $T$ time steps into trend and seasonal parts. The decomposed results first pass through the data embedding layer, adding static spatio-temporal features, to obtain high-dimensional traffic flow tensors. Next, MLP models the trend part and predicts future trends, and Fourier attention mechanism is used to predict the seasonal part. Finally, the outputs of the two parts are fused for a feature representation of the long-term correlation within the time series.

*3.3.1 MLP for Trend Context.* In the original Transformer architecture, the self-attention mechanism mainly focuses on capturing semantic correlations between paired elements, and its ability for temporal relationships modeling largely depends on the position encodings of input tokens. Despite many variant models targeting position encodings and token embedding subsequences to retain ordered information, the self-attention mechanism substantially interpolates context historical data, leading to the inevitable loss of temporal information. Inspired by [32], for the trend part, we employ a MLP with $l$ layers to predict future trends as Formula 5, where $W_L$ and $b_L$ are learnable parameters.

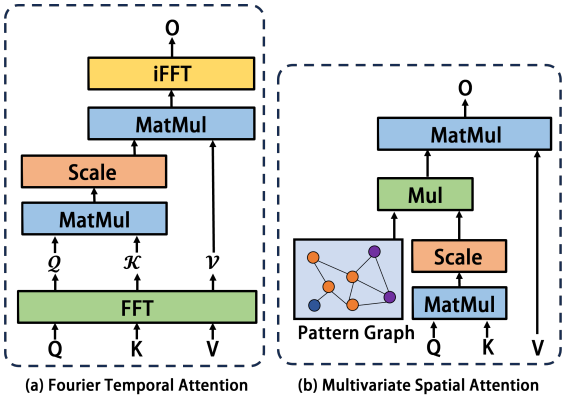


**Figure 3: The Process of Calculating Fourier Temporal Attention and Multivariate Spatial Attention.**

$$\hat{\mathcal{X}}_{tre} = ReLU(W_L^{(l)}\mathcal{X}_{tre}^{(l-1)} + b_L^{(l)}). \tag{5}$$

*3.3.2 Fourier Attention for Seasonal Context.* The seasonal part of time series explicitly indicates frequent modes. Softmax operation has the "*polarization*" effect with exponential terms, amplifying the difference between large and small values, and softmax attention further concentrates the scores on dominant frequency and captures seasonal information better [33]. Compared to the time domain, data modeling in the frequency domain can quickly identify the main frequent patterns, offering greater sampling efficiency. Thus, as Formula 6 and 7, Fourier attention first converts queries, keys, and values via Fourier transformation, calculates attention in the frequency domain, and finally converts the results back to the time domain via inverse Fourier transformation. Here, $W_{F,q}$, $W_{F,k}$ and $W_{F,v}$ are learnable parameters. With the help of Fast Fourier Transform ($FFT$), the computational complexity can be reduced from $O(n^2)$ to $O(nlogn)$.

$$\begin{cases} \mathcal{Q}_f = FFT(Q_f) = FFT(\mathcal{X}_{sea}W_{F,q}), \\ \mathcal{K}_f = FFT(K_f) = FFT(\mathcal{X}_{sea}W_{F,k}), \\ \mathcal{V}_f = FFT(V_f) = FFT(\mathcal{X}_{sea}W_{F,v}), \end{cases} \tag{6}$$

$$\hat{\mathcal{X}}_{sea} = iFFT(softmax(\mathcal{Q}_f\mathcal{K}_f^T)\mathcal{V}_f). \tag{7}$$

## 3.4 Long-range Spatial Correlation Extractor

We initially extract traffic patterns based on the historical traffic flow of the nodes to obtain a long-range pattern correlation matrix, regarding the spatial dependency among nodes. Subsequently, we employ a spatial attention mechanism to acquire the dynamic spatial correlation among time series.

*3.4.1 Traffic Pattern Extraction.* Traffic flow in road network exhibits various traffic patterns. For instance, the traffic flow near

schools and business districts on workdays, or that of around tourist sites on holidays, presents similarities. Their traffic flow patterns are referential, even though some nodes may be geographically non-adjacent and potentially quite distant. Therefore, a traffic pattern extraction module is designed to capture traffic patterns from the historical traffic flow of topological nodes.

Firstly, we decouple the geographical adjacency among nodes and conduct pattern clustering based on current traffic state features in time dimension. Specifically, we segment the historical data of length $N \times T$ into traffic flow sequences based on a time interval $\tau$. Kshape clustering algorithm [20] is employed to cluster these sequences, resulting in $K$ clusters $C = \{C_1, \cdots, C_K\}$. Each cluster $C_k$ as a traffic pattern is recognized by its centroid.

Next, upon the discussion on the daily and weekly periodicity of traffic flow in Section 3.2 and the clustering results, we consider the pattern frequency of nodes $V_m$ and $V_n$ at the same intervals $\tau$ on the ordinal day $d$ of a week. If the traffic patterns are identical, they are marked as 1 and the comparison results for each ordinal day are summed up to obtain the frequency $S^{(\tau)}_{m,n}$ as Formula 8. We assume $V_{m,\tau}$ and $V_{n,\tau}$ eligible for allocation in the same spatial cluster only if their frequencies exceed a predefined threshold $\varphi$. Then, the pattern correlation matrix $A_p$ is generated at each time interval, in which $(A^{(\tau)}_P)_{mn} = 1$.

$$S^{(\tau)}_{m,n} = \sum (C^{\tau_d}_m == C^{\tau_d}_n). \tag{8}$$

*3.4.2 Multivariate Spatial Attention.* In each time step, self-attention mechanism is employed to model interactions among the nodes in spatial dimension. Its advantage lies in the adaptability of spatial dependencies across different time steps, thereby to capture dynamic correlations among nodes. Here, we take the output $\hat{\mathcal{X}}$ of the temporal correlation extractor as input.

First, we use multi-head self-attention to project the query, key, and value into different subspaces as Formula 9, where $h$ represents the number of attention heads, $W^h_{M,q}$, $W^h_{M,k}$, $W^h_{M,v} \in \mathbb{R}^{D \times d_m}$ are learnable parameters, and $d_m$ is the dimension of subspace.

$$\mathcal{Q}^h_m = \hat{\mathcal{X}} W^h_{M,q}, \mathcal{K}^h_m = \hat{\mathcal{X}} W^h_{M,k}, \mathcal{V}^h_m = \hat{\mathcal{X}} W^h_{M,v}. \tag{9}$$

Next, we calculate the spatial correlation weight matrix $A^h_S$ among $N$ nodes as Formula 10. The pattern correlation is incorporated into the spatial weight matrix, explicitly modeling patterns through the propagation of spatial information as Formula 11.

$$A^h_S = softmax(\frac{\mathcal{Q}^h_m (\mathcal{K}^h_m)^T}{\sqrt{d_m}}) \in \mathbb{R}^{N \times N}, \tag{10}$$

$$\mathcal{L}_h = (A^h_S \odot A_P) \mathcal{V}^h_m. \tag{11}$$

Finally, the ultimate output of the spatial correlation extractor is projected as Formula 12, where the outputs from multiple attention heads are concatenated to form a comprehensive representation.

$$\hat{\mathcal{L}} = \oplus(\mathcal{L}_1, \cdots, \mathcal{L}_h) W_M. \tag{12}$$

## 3.5 Prediction Layer

For multi-step prediction, a linear layer is employed to directly transform the output from spatio-temporal correlation extractor into the prediction results $\hat{\mathcal{P}}$ within dimensions required. Unlike the original transformer architecture, we adopt a direct way instead of a recursive one, which enhances the efficiency of the model and reduces the accumulative and inherent errors by recursion.

To enhance the reliability of predictions, an additional attention mechanism is designed, whose input is the predictive results of the linear layer. Based on the principle [22] that the accuracy of predictions is higher at time steps closer to the present, we utilize a set of learnable Gaussian distributions [24] for each prediction sequence. During the computation of attention scores, more scores are allocated to earlier and therefore more reliable elements. In Formula 13, $\mathcal{Q}_p \mathcal{K}^T_p \in \mathbb{R}^{l \times l}$ is the attention score matrix, $l$ is the length of the prediction sequence, the center $\mu$ is fixed as zero, and the scale $\sigma$ is a learnable parameter.

$$\hat{\mathcal{P}} = \frac{\mathcal{Q}_p \mathcal{K}^T_p - \mu}{\sqrt{\sigma}} \cdot \mathcal{V}_p. \tag{13}$$

# 4 Experiment

## 4.1 Setting

*4.1.1 Datasets & Processing.* The performance of DEC-Former was evaluated on four real-world traffic flow datasets, PeMS03, PeMS04, PeMS07 and PeMS08. All the datasets were collected from California highway by the Caltrans Performance Measurement System (PeMS). Distinct in the sampling period and region, any dataset has different time spans and node scales, implying respective unique representativeness. As detailed in Table 1, the traffic flow is aggregated at 5-minute intervals, i.e., 12 sample points per hour.

In line with the most contemporary solution, all four datasets are split into training, validation, and test sets in a 6:2:2 ratio. Furthermore, we conduct multi-step predictions on the past one hour (i.e., 12 steps) of data to predict the traffic flow for the next hour (i.e., 12 steps).

**Table 1: Datasets Description.**

| Datasets | #Node | #Time step | Time Range |
|---|---|---|---|
| PeMS03 | 358 | 26202 | 09/01/2018-11/30/2018 |
| PeMS04 | 307 | 16992 | 01/01/2018-02/28/2018 |
| PeMS07 | 883 | 28224 | 05/01/2017-08/31/2017 |
| PeMS08 | 170 | 17856 | 07/01/2016-08/31/2016 |

*4.1.2 Baselines.* We evaluate DEC-Former with three classes of baselines, including time series models, GNN-based models, and attention-based models. These baselines are outlined below:

- **VAR** [12] is a statistical linear model for MTS prediction;
- **SVR** is a statistical model for regression prediction;
- **FC-LSTM** is a deep sequence-to-sequence model.
- **DCRNN** [11] combines GRU and Diffusion Convolutional Neural Networks to predict traffic speed;
- **STGCN** [31] establishes a complete convolutional graph structure to capture spatio-temporal correlation;
- **GWNet** [30] combines graph convolution and temporal convolution to capture long-term correlations;

**Table 2: Predictive Performance of DEC-Former and Baselines on PeMS Datasets. The optimal and suboptimal results are highlighted in bold and underlined respectively.**

| Model | PeMS03 | | | PeMS04 | | | PeMS07 | | | PeMS08 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) |
| VAR | 19.72 | 32.38 | 20.50 | 24.44 | 37.76 | 17.27 | 27.96 | 41.31 | 12.11 | 19.83 | 29.24 | 13.08 |
| SVR | 19.77 | 32.78 | 23.04 | 26.18 | 38.91 | 22.84 | 28.45 | 42.67 | 14.00 | 20.92 | 31.23 | 14.24 |
| FC-LSTM | 19.56 | 33.38 | 19.56 | 23.60 | 37.11 | 16.17 | 34.05 | 55.70 | 15.31 | 21.18 | 31.88 | 13.72 |
| DCRNN | 17.62 | 29.86 | 16.83 | 24.42 | 37.48 | 16.86 | 24.45 | 37.61 | 10.67 | 18.49 | 27.30 | 11.69 |
| STGCN | 19.76 | 33.87 | 17.33 | 23.90 | 36.43 | 13.67 | 26.22 | 39.18 | 10.74 | 18.79 | 28.20 | 10.55 |
| GWNet | 15.67 | 26.42 | 15.72 | 19.91 | 31.06 | 13.62 | 20.83 | 33.62 | 9.10 | 15.57 | 24.32 | 10.32 |
| STSGCN | 17.51 | 29.05 | 16.92 | 21.52 | 34.14 | 14.50 | 23.99 | 39.32 | 10.10 | 17.88 | 27.36 | 11.71 |
| ASTGCN | 18.67 | 30.71 | 19.85 | 22.90 | 33.59 | 16.75 | 28.13 | 43.67 | 13.31 | 18.72 | 28.99 | 12.53 |
| GMAN | 15.52 | 26.53 | 15.19 | 19.25 | 30.85 | 13.00 | 20.68 | 33.56 | 9.31 | 14.87 | 24.06 | 9.77 |
| DSTAGNN | 15.57 | 27.21 | 14.68 | 19.30 | 31.46 | 12.70 | 21.42 | 34.51 | 9.01 | 15.67 | 24.77 | 9.94 |
| STGSA | 15.36 | 27.89 | 14.45 | 19.32 | 31.30 | 12.90 | 20.80 | 34.30 | 8.86 | 15.26 | 24.28 | 9.81 |
| ISTNet | 15.12 | 25.14 | 15.43 | 18.54 | 30.46 | 12.52 | 19.79 | 33.06 | 8.77 | 14.13 | 23.39 | 9.43 |
| PDFormer | 14.73 | 24.54 | 15.42 | 18.51 | 30.24 | 12.38 | 20.65 | 34.36 | 8.68 | 14.34 | 23.68 | 9.88 |
| DEC-Former | **14.33** | **23.55** | **14.27** | **18.23** | **29.24** | **12.04** | **19.48** | **33.04** | **8.54** | **13.23** | **23.06** | **9.12** |

- **STSGCN** [23] captures heterogeneity in spatio-temporal graphs through synchronous graph convolution.
- **ASTGCN** [5] captures dynamic correlations of traffic data through a spatio-temporal attention mechanism;
- **GMAN** [35] alleviates error propagation through a transformation attention layer;
- **DSTAGNN** [10] represents dynamic spatial correlations through an improved multi-head attention mechanism;
- **STGSA** [28] fuses local and long-term spatio-temporal correlations through a multi-head attention mechanism;
- **ISTNet** [27] captures local correlations through CNN to supplement into transformer model;
- **PDFormer** [8] designs a self-attention mechanism through a graph mask and a delay-aware module.

*4.1.3 Evaluative Metrics.* Three common metrics are adopted for evaluation: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). All the experiments are repeated five times to report the average results.

*4.1.4 Implementation Details.* The implementation environment of DEC-Former is Ubuntu 20.10, Python 3.7, and PyTorch 1.10.1. The evaluation environment is a server equipped with an eight-core Intel(R) Xeon(R) Platium 8163 2.50GHz CPU and two NVIDIA Tesla T4 16GB GPU. To train our model, Adam optimization is used with an initial learning rate of 0.001, a batch size of 16, and a maximum of 80 epochs. The hyperparameters and their optimums are determined on validation set: the dimension $D$ is 64, the number of attention heads is 8, and the number of layers of encoder is [3,4,4] respectively.

## 4.2 Predictive Performance

Table 2 displays the results, in which the bold values are the optimal, and the underlined ones are the suboptimal.

Based on Table 2, we find the following observations. 1) From the experimental results, DEC-Former outperforms all the baselines by any metric on four datasets. Compared with the suboptimal models, DEC-Former improves MAE/RMSE/MAPE by an average of 3.04%, 2.20%, and 2.23%, respectively. 2) Traditional statistical models perform poorly, because they only consider temporal correlations and ignore spatial dependencies. 3) In GNN-based models, GWNet and STSGCN lead to competitive performance because they consider the dynamics of spatial correlation in temporal propagation, compared to static graph embedding. Our DEC-Former, combining spatial attention mechanism and pattern extraction module, can fully consider the dynamic long-range correlations among nodes, thereby achieving better performance. 4) Most of the attention-based models show a significant improvement compared to time series and GNN-based models. Among them, STGSA, ISTNet, and PDFormer have shown superior performance to other baselines in the metrics of different datasets. Compared with them, our DEC-Former has achieved better performance. We believe, through the trend decomposition module, temporal attention can fully exploit attention mechanism on the seasonal part to extract long-term correlations better. 5) Specifically, the most distinct improvement is on PeMS08 dataset, with over 6% in MAE. We believe that the reasons come from that dataset: fewer nodes makes temporal correlations more important than spatial correlations; the larger range of daily traffic flow per node implies that the trend part dominates the data. In brief, our decoupling design of DEC-former presents the noticeable improvement.

## 4.3 Ablation Study

To further evaluate the effects of modules of DEC-Former, DEC-Former is compared with the following variants.

- **DEC-NTA** removes the trend decomposition module and uses MLP to extract all temporal correlations.
- **DEC-NTR** removes the trend decomposition module and uses the attention mechanism to extract all temporal correlations.
- **DEC-NFA** replaces the Fourier attention with a time-domain attention.
- **DEC-NSA** replaces the multivariate spatial attention module with static graph convolution.
- **DEC-NP** removes the pattern extraction module and directly uses the predefined geographical adjacency matrix to calculate spatial correlations.

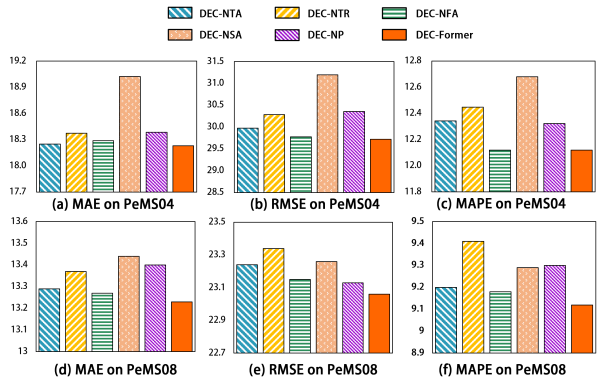


**Figure 4: Ablation Study on PeMS04 and PeMS08.**

Fig. 4 shows the comparison of these variants on PeMS04 and PeMS08 datasets. From the results, we get the following conclusions. 1) DEC-NTR performs worse than DEC-NTA, which indicates the poor generalization ability of the attention mechanism on the trend part of time series. 2) DEC-NFA outperforms DEC-NTA, which proves time series decomposition necessary. It also proves the fact that the attention mechanism can play better on the seasonal part of time series. 3) DEC-Former further improves performance than DEC-NFA, which indicates that Fourier attention helps the model capture seasonal information better. 4) In most cases, DEC-NSA has poorest performance, which proves the attention mechanism important to capture dynamic spatial correlations among nodes. 5) DEC-Former brings better performance than DEC-NP, which highlights the value of pattern correlations. In summary, attention mechanism is enhanced by our decoupling design, and its shortcomings is relieved distinctly by the modules.

## 4.4 Computational Performance

The computational performance of DEC-Former is compared with several major baselines on the PeMS04 dataset, and the results are reported in Table 3. Here, the metrics are average training time and inference time per epoch. We find that DEC-Former achieves competitive computational efficiency in both metrics. Firstly, DEC-Former reduces the training and inference time over 55% and 90% compared to GNN-Based STSGCN, due to its effective attention module. Secondly, DSTAGNN performs worst due to the time-consuming encoder-decoder architecture; in contrast ISTNet and DEC-Former, replacing decoder by the prediction layer, play better. Finally, compared to two suboptimal baselines in performance on PeMS04(i.e., ISTNet and PDFormer), DEC-Former reduced the inference time by over 30% and 18% respectively. That is due to reduced complexity by the linear MLP after trend decomposition module for the trend part and by Fourier attention mechanism in the module of Temporal Correlation Extractor.

**Table 3: The Computation Cost on the PeMS04 Dataset.**

| Model | Training Time (s/epoch) | Inference Time on Testset (s) |
|---|---|---|
| STSGCN | 848.08 | 532.80 |
| DSTAGNN | 1064.61 | 588.22 |
| ISTNet | 443.28 | 64.78 |
| PDFormer | 430.24 | 56.00 |
| DEC-Former | 371.74 | 45.56 |

## 4.5 Parameter Analysis

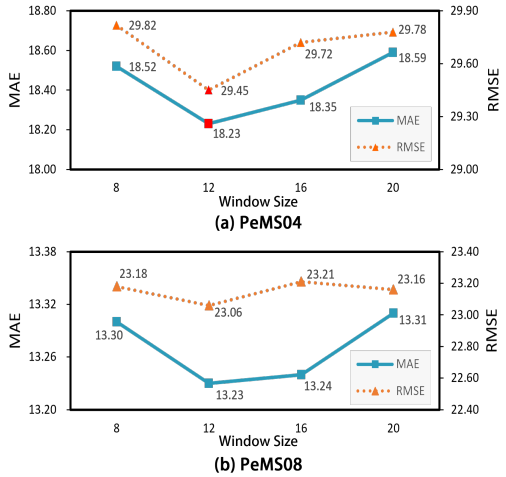


**Figure 5: Parameter Analysis for Window Size $m$ on PeMS04 and PeMS08 datasets.**

*4.5.1 The Impact of Window Size m.* One major advantage of our DEC-Former model is its capability to decompose raw time series data and model different parts separately, thereby guarantees the robustness and accuracy of the model. Within the trend decomposition module, the parameter $m$ represents the window size for the moving average operation. In our experiments, we set the values of $m$ as 8, 12, 16, and 20 respectively, on PeMS04 and PeMS08 datasets. The specific experimental outcomes are illustrated in Figure 5. Notably, when analyzing traffic flow data with a 5-minute time step, we found that the trend extraction was most effective when $m = 12$

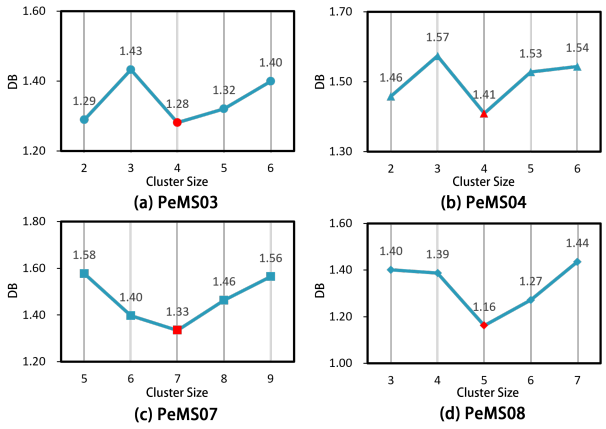


**Figure 6: Parameter Analysis for Clusters size $k$ on Four Datasets.**

in a 1-hour periodicity. On one hand, when the window size $m$ is smaller than 12, the model is affected by transient influences and noise of the data, and would overfit on minor, short-term variations possibly neglecting more stable, long-term trends. On the other hand, when the window size $m$ is larger than 12, the model may excessively smooth the data, and diminish the sensitivity to rapid fluctuations in traffic flow data. Therefore, the setting $m = 12$ provides a balance that captures the subtleties of short-term variations without losing long-term trends, and is proved the most effective in our analysis.

*4.5.2 The Impact of the Cluster Size k.* To better capture the pattern correlations between road network nodes, the traffic pattern extraction module performs clustering to learn traffic patterns. For that clustering effect, the Davies-Bouldin index (DB) [4] is adopted to measure the tightness and separability of clusters. The calculation is shown in Formula 14, where $k$ is the cluster size, $\sigma_i$ is the average distance from samples in the $i$-th cluster to the cluster center, $c_i$ is the center of the $i$-th cluster, and $d(c_i, c_j)$ is the distance between centers of clusters $i$ and $j$. A smaller DB value indicates better clustering performance.

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d\left(c_i, c_j\right)}\right). \tag{14}$$

In order to evaluate the clustering performance, we analyzed the varying cluster size $k$ on different datasets, when extracting patterns from traffic flow data. We categorized traffic flow data into three types: workdays, weekends, and holidays. Considering the different data collection periods of dataset, we first labeled holiday data in different datasets, then averaged the weekly data to obtain $7 \times 24$ averages of traffic flow data at intervals of 5-minute. Finally, we selected the data from the same ordinal day in a week to evaluate clustering performance. The experimental results are illustrated in Figure 6. We found that most datasets achieved optimal clustering performance with 4 to 5 clusters, which may imply urban functional zones such as commercial districts, residential areas, administrative districts, educational campuses, and scenic resorts. Noticeably, on the largest PeMS07 dataset, the best clustering performance was observed when $k = 7$. The larger the data is, the more complex patterns the data would own. Such complexity could be attributed to factors like a denser network of traffic nodes, a broader spectrum of traffic behaviors, and distinct geographical or infrastructural features. All those may require a finer-grained clustering approach from intricate traffic patterns. Our observations emphasize that the optimal cluster size is related with the specific characteristics of each dataset.

## 4.6 Case Study

*4.6.1 Comparison of Predictive Result Curves.* We randomly selected two nodes, Node 002 and Node 101 here, from the PeMS08 dataset to draw the prediction curves of PDFormer and DEC-Former. PDFormer is particularly emphasized for this comparison, because it also considers pattern correlation in modeling spatial relationships like our DEC-Former. The better result by DEC-Former proves the trend decomposition module effective. In the shaded area of Fig. 7 (a) when the traffic flow data undergoes a sudden change, i.e., the data exhibits a strong trend, DEC-Former is able to capture the temporal context more effectively. That is attributed to MLP module, which extracts features from the trend part of data to address the limitations of the attention mechanism in capturing trends. In Fig. 7 (b), we see that the traffic flow fluctuates within a lower range of values, i.e., the data exhibits a strong seasonality. Our DEC-Former captures the fluctuation of the context from the beginning, making a better fitting curve. That is attributed to the attention re-placement, mentioned in Section 3.3, by which attention would be more effective.

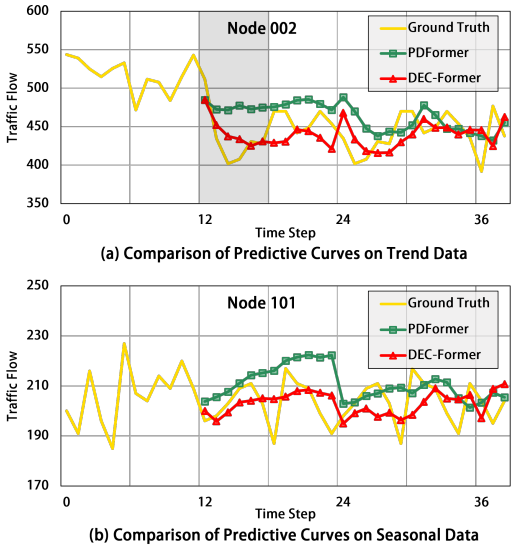


**Figure 7: A Snapshot of Prediction Curves Comparison between PDFormer and DEC-Former on Test Data of PeMS08.**

*4.6.2 Visualization of Spatial Attention Maps.* The pattern extraction module of DEC-Former is further analyzed by visualizing the spatial attention maps, which demonstrates the effectiveness of the

long-range pattern correlation among nodes. The attention maps are visually compared under two scenarios: the attention matrix based on geographical adjacency, and that of based on pattern correlation. In Fig. 8 (b), when pattern correlation is not considered, the global attention distribution is relatively uniform. It means that the nodes have to refer to almost global information of the dispersed attention. That makes the nodes unable to extract truly useful correlation. When the pattern correlation of the nodes is emphasized, as discussed in Section 3.4.1 and illustrated in Fig. 8 (c), the attention would focus more on the positions in the similar patterns. In fact, Nodes A and B are geographically adjacent, but they have different traffic patterns of the daily average trends, from Fig. 8 (a). It means that the geographical adjacency brings erroneous reference to Node A, and further would make poor performance by traditional attention mechanism.

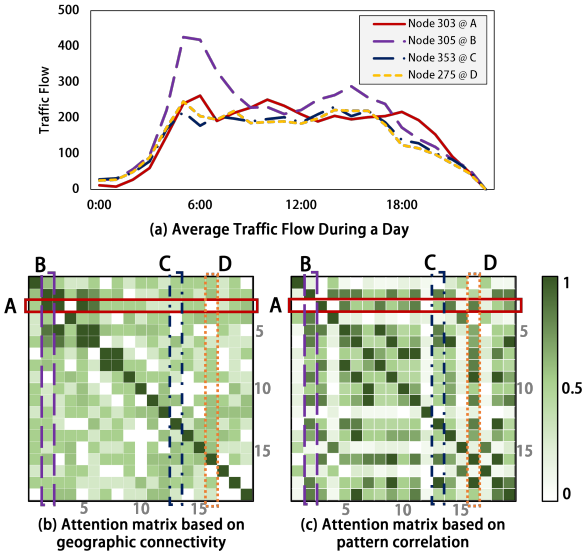


**Figure 8: Case Study of Attention Map (from Traffic Flow on PeMS03 dataset).**

## 5 Related Work

### 5.1 Traffic Flow Prediction.

Traffic flow prediction, as one typical MTS prediction, has become important in real-world applications. Deep learning techniques are adopted evolutionally to discover spatio-temporal correlations on traffic data. Early models based on Recurrent Neural Networks (RNNs) [17, 21] showed advantages in extracting temporal correlations. However, RNN-based models simultaneously implies limitations in capturing long-term dependencies. Later, Graph Neural Networks (GNNs) were popular to model spatial correlations due to their powerful graph data modeling capabilities[7, 13, 18]. Many of these models, such as STGCN [31] and DCRNN [11], define the traffic graph as a static graph based on geographical adjacency, overlooking the dynamic correlations between temporal data. Accordingly, to further capture dynamic correlations, some studies employ the automatic graph learning by the inter-series correlations, such as self-supervised learning [6] and node similarity [8]. Furthermore, the attention mechanism, which proves effective to capture long-term correlations and global spatial dependencies, had gained popularity in traffic flow prediction[9, 14, 27]. Unlike previous works, our DEC-Former proposes a trend decomposition module and a traffic pattern extraction module, better leveraging the attention mechanism in spatio-temporal modeling to improve the predictive performance.

### 5.2 Transformer with Attention Mechanism.

The transformer architecture [25], entirely based on the attention mechanism, has been proved successful in natural language processing and computer vision. Numerous transformer variants have been designed for time series prediction in two main categories. One category involves modifying the original modules, such as adjusting the attention module to enhance the extraction of complex relationships [36, 37], and altering the data embedding module to focus more on the processing of the time series itself [16, 19]. The other category involves structural modifications. For instance, Crossformer [34] captures the cross-dimensional dependence to enhance long sequence prediction accuracy, and GBT [22] designs a two-stage transformer predictor to improve the predictive accuracy of non-stationary time series. The up-to-date linear models [15, 32] have challenged traditional views and opened new possibilities. TDformer [33] has demonstrated that compared with trend data, attention-based models perform better on seasonal data. Our DEC-Former allocates attention exclusively to the seasonal part of the time series, using a linear model for the trend component. Additionally, a dynamic spatial attention module captures evolving pattern correlations among topological nodes over time, enhancing predictive accuracy.

## 6 Conclusion

In this paper, to prove our understanding of how the attention mechanism functions within temporal sequence and spatial nodes, we design a novel traffic flow prediction model from a decoupling perspective capturing spatio-temporal correlation. To address the poor generalization ability of attention-based models for trend data observed in current popular models, a trend decomposition module is presented to separately model the trend and seasonal parts of time series, better leveraging the attention mechanism in long-term correlations. Furthermore, a traffic pattern extraction module is designed to consider pattern correlations among nodes when calculating spatial attention and capturing long-range correlations. Extensive experiments are conducted on four real-world datasets to demonstrate the superiority of our model. Compared to the current state-of-the-art baselines, the predictive performance of DEC-Former has improved by an average of 3.04%, 2.20%, and 2.23% in terms of MAE, RMSE, and MAPE, respectively, and the computational efficiency has been enhanced by at least 14%. Two visualized case studies show interpretable effects as well. In the future, we will extend our work to other spatio-temporal tasks besides traffic flow prediction, and address the few-shot problem among data-imbalanced nodes.

## References


[1] Oliver D Anderson. 1976. Time-Series. 2nd edn.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. 1990. STL: A seasonal-trend decomposition. *J. Off. Stat* 6, 1 (1990), 3–73.

[4] David L. Davies and Donald W. Bouldin. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1, 2 (1979), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

[5] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* (Sep 2019), 922–929. https://doi.org/10.1609/aaai.v33i01.3301922

[6] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4356–4364.

[7] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. 2022. STDEN:Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4048–4056.

[8] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 4 (Jun. 2023), 4365–4373. https://doi.org/10.1609/aaai.v37i4.25556

[9] Di Jin, Jiayi Shi, Rui Wang, Yawen Li, Yuxiao Huang, and Yu-Bin Yang. 2023. Trafformer: Unify Time and Space in Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 7 (Jun. 2023), 8114–8122. https://doi.org/10.1609/aaai.v37i7.25980

[10] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. DSTAGNN: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 11906–11917. https://proceedings.mlr.press/v162/lan22a.html

[11] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJiHXGWAZ

[12] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 2013. Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (2013), 871–882. https://doi.org/10.1109/TITS.2013.2247040

[13] Dachuan Liu, Jin Wang, Shuo Shang, and Peng Han. 2022. MSDR: Multi-Step Dependency Relation Networks for Spatial Temporal Forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1042–1050. https://doi.org/10.1145/3534678.3539397

[14] Yan Liu, Bin Guo, Jingxiang Meng, Daqing Zhang, and Zhiwen Yu. 2024. Spatio-Temporal Memory Augmented Multi-Level Attention Network for Traffic Prediction. *IEEE Transactions on Knowledge and Data Engineering* 36, 6 (2024), 2643–2658. https://doi.org/10.1109/TKDE.2023.3322405

[15] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=JePfAI8fah

[16] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 9881–9893. https://proceedings.neurips.cc/paper_files/paper/2022/file/4054556fcaa934b0bf76da52cf4f92cb-Paper-Conference.pdf

[17] Changxi Ma, Guowen Dai, and Jibiao Zhou. 2022. Short-Term Traffic Flow Prediction for Urban Road Sections Based on Time Series Analysis and LSTM_BILSTM Method. *IEEE Transactions on Intelligent Transportation Systems* 23, 6 (2022), 5615–5624. https://doi.org/10.1109/TITS.2021.3055258

[18] Ying Ma, Haijie Lou, Ming Yan, Fanghui Sun, and Guoqi Li. 2024. Spatio-temporal fusion graph convolutional network for traffic flow forecasting. *Information Fusion* 104 (2024), 102196. https://doi.org/10.1016/j.inffus.2023.102196

[19] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=Jbdc0vTOcol

[20] John Paparrizos and Luis Gravano. 2016. K-Shape: Efficient and Accurate Clustering of Time Series. 45, 1 (jun 2016), 69–76. https://doi.org/10.1145/2949741.2949758

[21] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

[22] Li Shen, Yuning Wei, and Yangzhu Wang. 2023. GBT: Two-stage transformer framework for non-stationary time series forecasting. *Neural Networks* 165 (2023), 953–970. https://doi.org/10.1016/j.neunet.2023.06.044

[23] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* (Jun 2020), 914–921. https://doi.org/10.1609/aaai.v34i01.5438

[24] Souhaib Ben Taieb and Amir F Atiya. 2015. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems* 27, 1 (2015), 62–76.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ

[27] Chu Wang, Jia Hu, Ran Tian, Xin Gao, and Zhongyu Ma. 2023. ISTNet: Inception Spatial Temporal Transformer for Traffic Prediction. In *Database Systems for Advanced Applications*, Xin Wang, Maria Luisa Sapino, Wook-Shin Han, Amr El Abbadi, Gill Dobbie, Zhiyong Feng, Yingxiao Shao, and Hongzhi Yin (Eds.). Springer Nature Switzerland, Cham, 414–430.

[28] Zebing Wei, Hongxia Zhao, Zhishuai Li, Xiaojie Bu, Yuanyuan Chen, Xiqiao Zhang, Yisheng Lv, and Fei-Yue Wang. 2023. STGSA: A Novel Spatial-Temporal Graph Synchronous Aggregation Model for Traffic Prediction. *IEEE/CAA Journal of Automatica Sinica* 10, 1 (2023), 226–238.

[29] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems* 34 (2021), 22419–22430.

[30] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling *(IJCAI'19)*. AAAI Press, 1907–1913.

[31] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. 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*. International Joint Conferences on Artificial Intelligence Organization, 3634–3640. https://doi.org/10.24963/ijcai.2018/505

[32] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 9 (Jun. 2023), 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317

[33] Xiyuan Zhang, Xiaoyong Jin, Karthick Gopalswamy, Gaurav Gupta, Youngsuk Park, Xingjian Shi, Hao Wang, Danielle C. Maddix, and Bernie Wang. 2022. First De-Trend then Attend: Rethinking Attention for Time-Series Forecasting. In *NeurIPS '22 Workshop on All Things Attention: Bridging Different Perspectives on Attention*. https://openreview.net/forum?id=GLc8Rhney0e

[34] Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=vSVLM2j9eie

[35] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* (Jun 2020), 1234–1241. https://doi.org/10.1609/aaai.v34i01.5477

[36] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 12 (May 2021), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[37] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 27268–27286. https://proceedings.mlr.press/v162/zhou22g.html