# ModWaveMLP: MLP-based Mode Decomposition and Wavelet Denoising Model to Defeat Complex Structures in Traffic Forecasting

**Ke Sun[1], Pei Liu[1], Pengfei Li[2], Zhifang Liao[1]***

[1]School of Computer Science and Engineering, Central South University, Changsha, China
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
224712262, 224712142@csu.edu.cn, li-pf22@mails.tsinghua.edu.cn, zfliao@csu.edu.cn

## Abstract

Traffic prediction is the core issue of Intelligent Transportation Systems. Recently, researchers have tended to use complex structures, such as transformer-based structures, for tasks such as traffic prediction. Notably, traffic data is simpler to process compared to text and images, which raises questions about the necessity of these structures. Additionally, when handling traffic data, researchers tend to manually design the model structure based on the data features, which makes the structure of traffic prediction redundant and the model generalizability limited. To address the above, we introduce the 'ModWaveMLP'—A multilayer perceptron (MLP) based model designed according to mode decomposition and wavelet noise reduction information learning concepts. The model is based on simple MLP structure, which achieves the separation and prediction of different traffic modes and does not depend on additional features introduced such as the topology of the traffic network. By performing experiments on real-world datasets METR-LA and PEMS-BAY, our model achieves SOTA, outperforms GNN and transformer-based models, and outperforms those that introduce additional feature data with better generalizability, and we further demonstrate the effectiveness of the various parts of the model through ablation experiments. This offers new insights to subsequent researchers involved in traffic model design. The code is available at: https://github.com/Kqingzheng/ModWaveMLP.

## Introduction

Time series prediction, a cornerstone of time series analysis, entails forecasting future values using historical sequential data patterns and trends. In the specific context of traffic prediction(James 2022), which includes forecasting traffic flow, speed, and demand, its applications span route planning, vehicle scheduling, and congestion management(Lee et al. 2021; Fang et al. 2021; Li et al. 2023a).

Neural networks have been extensively used in time series forecasting due to their powerful function-fitting capabilities(Zhou et al. 2022; Liu et al. 2021). Early on, convolutional neural network (CNN) captured spatial dependencies in grid-based traffic data, while recurrent neural network (RNN) learned temporal dynamics(Wu et al. 2018; Zhang et al. 2018). Presently, graph neural network (GNN) dominates this field, excelling in modeling complex spatio-temporal correlations(Jiang and Luo 2022; Jin et al. 2022). The transformer(Vaswani et al. 2017), a prevailing sequential data architecture, also showcases remarkable performance in traffic prediction tasks(Jiang et al. 2023a). Recently, influenced by the 'Large Model' trend, researchers favor Large Model for intricate tasks like traffic prediction.

On the one hand,unlike high-dimensional image(Han et al. 2022) and natural language(Vaswani et al. 2017) data, traffic timing data is simpler—sequential numerical information recorded at distinct time points(Tran et al. 2020). Nonetheless, its periodic features and spatial dependencies lead to complex temporal extraction designs. Recent approaches employ intricate models such as graph neural networks and transformer(Yan, Ma, and Pu 2021; Chen et al. 2022) variants to capture these traffic features. However, this complexity entails larger parameters and greater hardware demands for training and inference, prompting the query: Are such elaborate models essential for this data? Doubts about transformers' effectiveness in time series prediction have surfaced. Works like MTS-Mixer(Li et al. 2023b) and the MLP-based traffic prediction model proposed by Oreshkin(Oreshkin et al. 2021; Shao et al. 2022a) challenge transformer and graph neural network-based models. These studies demonstrate the prowess of simpler MLP models in time series prediction(Zeng et al. 2023). Given MLP's simplicity, training efficiency, and inference efficiency, it might be a promising direction for traffic prediction, offering strong performance without the complexity of transformers and graph neural networks.

On the other hand, in order to improve the performance of traffic prediction, new structures based on complex architectures have been designed to apply to unique characteristics, which further brings about model complexity and redundancy, such as new traffic features based on urban zones and delayed diffusion influenced by road factors(Jiang et al. 2023a). However, these features might not universally apply and the original data might lack corresponding information for these features(Zhang, Zheng, and Qi 2017). Therefore, manually selected features for prediction may lead to poor generalization of the model or even partial feature selection that increases the complexity of the model with no improvement in performance. In addition, previous researchers have ignored noise in traffic data, traffic data is sensor-collected,

*Corresponding author

potentially tainted by noise(Tang et al. 2019; Chen et al. 2021). Extracting modes and reducing noise is also crucial. Therefore, how to design a unified structure to process and learn from all the information is a key issue.

In this study, we term these features collectively as "traffic modes"—a fusion of single or multiple features. We employ residual separation to decompose these modes within the model, replacing manual feature design. We propose an MLP-based structure designed according to the mathematical idea of "differential". The model does not rely on the complex structure such as GNN and transformer to extract traffic modes, but uses the idea of residual separation to design the MLP structure to decompose and capture the different traffic modes and noise in the traffic sequence. The basic MLP module of the model has two branches, one for prediction and the other for subtracting the traffic mode information extracted by the current module, which is superimposed to decompose different traffic modes and extract the predicted values of different traffic modes. The model adds a noise reduction information module of wavelet decomposition, which gives the model the ability to decompose the traffic noise and anti-noise ability by learning the information of wavelet decomposition. In summary, the main contributions of this paper are summarized as follows:

- We propose an mlp-based traffic prediction model called ModWaveMLP, which utilizes the residual separation idea to decompose and capture different traffic modes and noise in a traffic sequence for traffic prediction.
- We design a mode decomposition learning module.Through the stacking of fully connected decomposition structure, this module can effectively decompose and learn the original traffic information and wavelet decomposition information.
- Our model excels on real-world data, achieves state-of-the-art results, effectively isolating various traffic modes and demonstrating strong noise resistance. Compared to GNN and transformer-based models, our method approach excels in traffic prediction, offering fresh insights for researchers exploring traffic forecasting in the era of "Large Model".

## Problem statement

**Problem Formalization:***Traffic prediction aims to predict the traffic data of a traffic system based on historical observations.Formally, given the traffic data tensor $X$ observed on a traffic system, our goal is to learn a mapping function $f$ from the previous $T$ steps' data observation value to predict future $T$' steps' traffic data* $\left[x_{(t-T+1)}, \cdots, x_t\right] \xrightarrow{f} \left[x_{(t+1)}, \cdots, x_{(t+T')}\right]$

## Methods

ModWaveMLP's framework, depicted in Fig. 1, involves inputting the original traffic sequence data and noise-reduced wavelet-transformed data into the Information Coding Module. The module includes the Time Gate, Information Aggregation, and Mode Decomposition Learning Block (MDL block). Wavelet Decomposition Learning Module contains the wavelet decomposition and the three modules above. Cohesive information in traffic time steps is considered by inputting 1/4 and 1/2 sub-time mode segments. ModWaveMLP supports stacking for learning specific traffic modes. Specific details of the modules are as follows:

### Time Gate Module

We design time gate module to avoid periodic disturbances while maintaining the periodicity feature. It captures time-periodic traits using three embeddings: $T_{week(t)}, T_{day(t)} \in \mathbb{R}^{N \times T}$ , and $T_{dynamic(t)} \in \mathbb{R}^{N \times E}$.These embeddings encompass weekly, daily, and dynamic periodicities. $T_{week(t)}$ and $T_{day(t)}$ transform time $t$ into weekly and daily embeddings, respectively. Embedding values are normalized based on week indices (1 to 7) and minute indices (1 to 1440) of $t$, yielding normalized weekly and daily embeddings between 0 and 1. Dynamic node time information $T_{dynamic(t)}$is learned dynamically in the network through node embedding information.

Next, we use a multiplicative gate model to decode the input temporal data into effects that need to be removed or reduced later for restoration through a fully connected module. In contrast to the static modeling of daily and weekly cycles, we dynamically model hidden temporal cycles by splicing node embeddings and temporal information separately.

$$T_{day(t)} = Concat(T_{day(t)}, T_{dynamic(t)}) \tag{1}$$

$$T_{week(t)} = Concat(T_{week(t)}, T_{dynamic(t)}) \tag{2}$$

As shown on the left in Fig. 1, the input $T$ is first mapped through a fully connected layer and activated by the relu function.

$$FC() = RELU(Linear\,()) \tag{3}$$

$$T_H = FC\,(T) \tag{4}$$

The processed results are output through two fully-connected layer branches,

$$T_{backcast} = FC_{backcast}\,(T_H) \tag{5}$$

$$T_{forecast} = FC_{forecast}\,(T_H) \tag{6}$$

The output of the backcast branch $T_{backcast}$ is the information to remove the time effect, and the information of the forecast branch $T_{forecast}$ is the time information to be restored, and the information of the output of the backcast/-forecast branch is divided/multiplied with the information of the other modules for the removal and restoration of the information of the time period mode.

$$X_{inputs} = X_{inputs} / T_{backcast} \tag{7}$$

$$X_{outputs} = X_{outputs} \otimes T_{forecast} \tag{8}$$

### Information Aggregation Module

We design a Information Aggregation Module, which models the node history information in the traffic network, to get enough information for the MDL block to perform mode decomposition. To model the dynamic history information we use node embedding $X_{embedding}^{N \times E}$ and obtain the dynamic

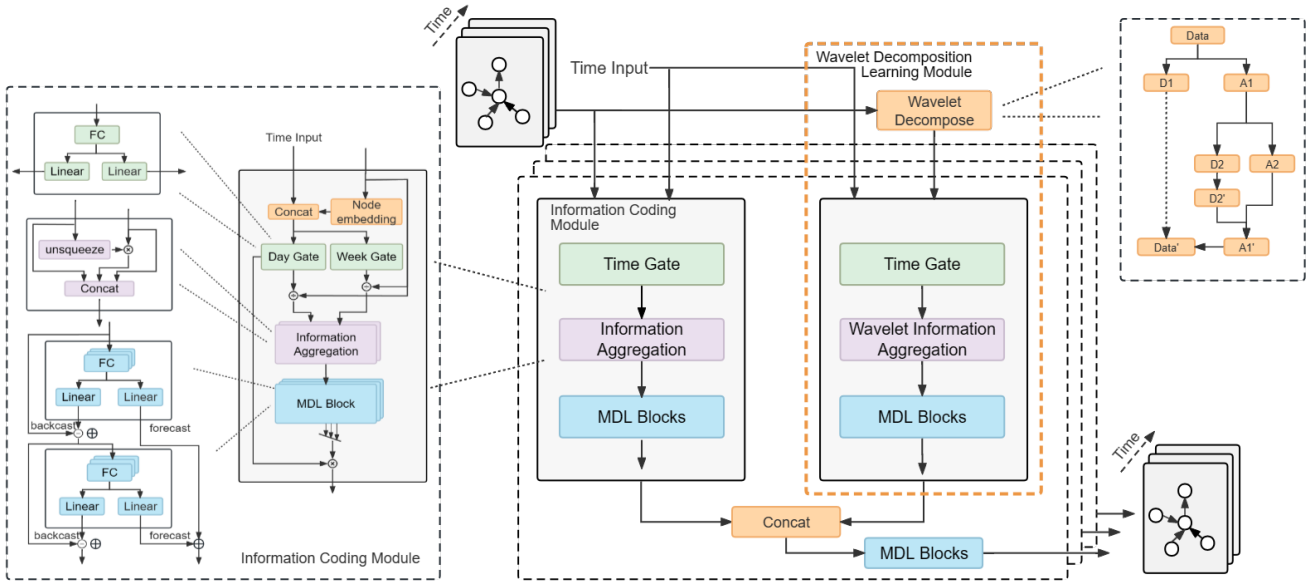


Figure 1: The overall architecture of the proposed ModWaveMLP

graph node information tensor $X_{dynamic}^{N\times N}$ by extrapolating the embedding information $X_{embedding}^{N\times E}$ learned by the node with its own transpose.

$$X_{dynamic}^{N\times N} = X_{embedding} \otimes X_{embedding}^{T} \tag{9}$$

$$X_{unsqueeze}^{N\times N\times T} = unsqueeze(X_{inputs}^{N\times T}) \tag{10}$$

$$X_{node\_history}^{N\times N\times T} = RELU(X_{dynamic}^{N\times N} \otimes X^{N\times N\times T}) \tag{11}$$

The dynamic graph node information tensor is multiplied with the ascending traffic history information $X_{unsqueene}^{N\times N\times T}$ after $T$ time steps to obtain the historical similarity information between nodes, which $X_{node_history}^{N\times N\times T}$contains all the node history information learned by the current node. After RELU removes useless associations between nodes, nodes with similar historical information can learn the same traffic mode. Then we convert the information dimension to $X_{node\_history}^{N\times NT}$, which means we get all the historical information of the current node, along with other nodes. Finally we concatenate this information$X_{node_history}^{N\times NT}$ with the original information $X^{N\times T}$ as well as the node embedding information $X_{embedding}^{N\times E}$ to get the final node aggregation information $X_{final\_information} \in \mathbb{R}^{N\times((N+1)T+E)}$.

## Mode Decomposition Learning Block

Traffic data comprises diverse modes influenced by factors such as node interactions and city zoning. These modes often overlap, requiring distinct feature extraction modules. To maintain generality and eliminate redundancy, we employ top-to-bottom mode decomposition and prediction, avoiding the need for multiple specialized modules. We design a MDL Block, which is the basic module in the model through which all the input information is decomposed and output.

The MDL Block contains $M$ fully connected decomposition structures for information separation and integration. The individual fully connected decomposition structure is very simple and contains one fully connected structure with $L$ hidden layers and two fully connected prediction branches for information decomposition and information output. In order to gradually learn the information in the input data $Y$, the fully connected decomposition structure first feeds the input information into $Y_0$ the hidden layer mapping it to $l \in [1, L)$ . The output of each hidden layer is transformed by the RELU activation function, and the final hidden layer $H_L$ is output to the information decomposition layer and the information output layer, where the information $\hat{Y}^{m-1}m \in [1, M)$ of the information decomposition layer is the information learned from the module decomposition. We subtract this information from the input $Y^{m-1}$ of the structure to get the remaining information $Y^m$ after module learning, which can be further input to the next fully connected decomposition structure for the learning of the remaining information. At the same time, the output information of each structure is summed to get the final output information. By stacking $M$ fully connected decomposition structure into MDL module, we have accomplished the separation and learning of different modes in traffic data.Fully connected decomposition structure works as follows:

$$H^{m-1,1}=FC_{m-1,1}(Y^{m-1}),...,H^{m-1,L}=FC_{m-1,L}(H^{m-1,L-1}) \tag{12}$$

$$\hat{Y}^{m-1}=FC_{m-1,backcast}\left(H^{m-1,L}\right) \tag{13}$$

$$Z^{m-1}=FC_{m-1,forecast}\left(H^{m-1,L}\right) \tag{14}$$

$$Y^{m}=\text{RELU}\left(Y^{m-1}-\hat{Y}^{m-1}\right),\ Z=\sum_{m}^{M-1} Z^{m} \tag{15}$$

If the MDLblock is located in the information encoding module, the final output is multiplied with the time-gated

forecast branch to restore the previously removed time effects. Besides step-by-step information decomposition, we enrich the information. Each decomposed $\hat{Y}_{m-1}$ is added to the original $Y^{m-1}$using a shortcut, progressively enhancing separated traffic modes. Stacking modules enriches each piece of information while predicting traffic modes with this progressively enriched information as output.

### Wavelet Decomposition Learning Module

To remove noise from the raw data and provide more information for the MDL block, We design this module. Wavelet transform can detect traffic data at different scales and has the ability to detect features of traffic data (e.g., abrupt changes, spikes, and periodic cycles) and can be used to filter noise from traffic data to improve data quality. The wavelet decomposition is shown in Eq. where $t$ is time point, $f(t)$ is the original data, $a$ is the scale and $b$ is the translation.

$$WT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) * \psi(\frac{t-b}{a}) dt \tag{16}$$

Noise is implied in the high-frequency ($A_i$) components after wavelet decomposition, and the noise reduction process, i.e., thresholding the high-frequency vectors($A_i'$), ultimately reconstructs them($D_i \& A_i'$) to be close to the original traffic data. We choose to use different wavelet basis functions for denoising and aggregate the data after different noise reduction methods, which enables the prediction model to learn the information after multiple methods of noise reduction. Based on previous wavelet transform studies, since different wavelet basis functions are suitable for prediction at different moments, we choose a combination of $n$ wavelet basis functions, set the number of decomposition layers, and select the soft threshold reconstruction method. The original 2D data $f(t)^{N \times T}$ are decomposed and reconstructed several times using wavelets, and the multi-reconstructed 3D data $f(t)^{N \times nT}$(n is the number of wavelet basis functions) are integrated as model input. More details on the 3D traffic tensor construction process and the selection of the wavelet basis functions can be found in Appendix Section A&D.

## Experiments

### Datasets

The ModWaveMLP model is assessed using two traffic datasets: METR-LA and PEMS-BAY(Li et al. 2018). These datasets consist of traffic speed readings collected from loop detectors, aggregated in 5-minute intervals. METR-LA contains 34,272 time steps from 207 sensors deployed in Los Angeles County over 4 months. PEMS-BAY comprises 52,116 time steps from 325 sensors in the Bay Area over 6 months. Following prior research (Li et al., 2018), the datasets are divided into 70% for training, 10% for validation, and 20% for testing.

### Baselines

We have selected a diverse range of baselines. For time series analysis, we included HA, VAR, SVR, and FC-LSTM (Sutskever, Vinyals, and Le 2014). HA, VAR, and SVR utilize statistical and machine learning approaches for prediction. In the realm of graphical neural networks, we chose DCRNN, STGCN, Graph Wavenet, MTGNN, and MegaCRN. DCRNN (Li et al. 2018) employs diffusion convolution, while STGCN, Graph Wavenet, MTGNN and MegaCRN (Yu, Yin, and Zhu 2018; Wu et al. 2019, 2020; Jiang et al. 2023b) integrate spatio-temporal graph convolution. D$^2$STGNN (Shao et al. 2022b) builds upon this foundation. In the context of attention or transformer-based techniques, we considered GMAN, ASTGCN, PDFormer, and STGRAT. GMAN and ASTGCN (Zheng et al. 2020; Guo et al. 2019) leverage spatio-temporal attention, and PDFormer and STGRAT (Jiang et al. 2023a; Park et al. 2020) extend this concept with spatio-temporal transformer structures. We also examined FC-GAGA and STID (Oreshkin et al. 2021; Shao et al. 2022a), which employ MLP-like strategies through fully connected and embedding layers. The hardware environment for the experiments and baselines is detailed in Appendix Section B.

### ModWaveMLP architecture details and training setup

ModWaveMLP is stacked with 4 layers, embedding dimensionality of 96, and 128 hidden layer width for all fully connected layers. Wavelet decomposition comprises 4 layers with 'db1', 'db2', 'db3', 'db4' Daubechies wavelet functions and 'soft' thresholds(shown in Appendix Section D). A 3-layer fully connected decomposition structure with 2 mode decomposition learning model blocks is employed. Fully-connected layers have weight decay of 1e-5.The model uses Adam optimizer, starting with a learning rate of 0.001 for 80 epochs. Learning rate anneals by a factor of 2 every 8 epochs starting from epoch 49. Each epoch has 800 batches of size 4, considering a history of 12 points and predicting 12 points (60 minutes) ahead. Training batches randomly select 4 time points from the training set, collecting histories for all nodes at each time point. For METR-LA ($N$=207 nodes) and PEMS-BAY ($N$=325 nodes), respectively. The training loss is Mean Absolute Error (MAE) averaged over nodes and forecasts within the horizon $H$=12. $Loss = \frac{1}{NH} \sum_{i}^{N} \sum_{j}^{H} |y_{i,T+j} - \hat{y}_{i,T+j}|$. We use three metrics in the experiments: (1) Mean Absolute Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Squared Error (RMSE). Missing values are excluded when calculating these metrics.

### Performance Study

Comparison results with baselines on METR-LA and PEMS-BAY datasets are summarized in Tab. 1. Bold results indicate superiority. Key observations from these tables are: (1)Deep learning models outperform traditional methods like VAR, capturing hidden nonlinear information and spatial dependencies in traffic data. (2)ModWaveMLP achieves the best performance in all metrics, outperforming many GNN and transformer-based models. In the initial 15-minute and 30-minute predictions of PEMS-BAY, ModWaveMLP improves the MAE, RMSE, and MAPE by 20.97%, 27.31%, 25.58%, 9.35%, 18.47%

Table 1: Traffic forecasting on the METR-LA and PEMS-BAY datasets( average over last time step of horizon, input window length 12). ModWaveMLP† indicates that the number of layers decomposed by the wavelet decomposition learning module is 5. The best results are bolded, suboptimal results are underlined(excluding ModWaveMLP variants results). Numbers marked with * indicate that the improvement is statistically significant compared with the best baseline (t-test with p-value $< 0.05$).

| Datasets | Methods | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| **METR-LA** | HA | 4.79 | 10.00 | 11.70% | 5.47 | 11.45 | 13.50% | 6.99 | 13.89 | 17.54% | 5.75 | 11.78 | 14.25% |
| | VAR | 4.42 | 7.80 | 13.00% | 5.41 | 9.13 | 12.70% | 6.52 | 10.11 | 15.80% | 5.45 | 9.01 | 13.83% |
| | SVR | 3.39 | 8.45 | 9.30% | 5.05 | 10.87 | 12.10% | 6.72 | 13.76 | 16.70% | 5.05 | 11.03 | 12.70% |
| | FC-LSTM | 3.44 | 6.30 | 9.60% | 3.77 | 7.23 | 10.09% | 4.37 | 8.69 | 14.00% | 3.86 | 7.41 | 11.23% |
| | DCRNN | 2.77 | 5.38 | 7.30% | 3.15 | 6.45 | 8.80% | 3.60 | 7.60 | 10.50% | 3.17 | 6.48 | 8.87% |
| | STGCN | 2.88 | 5.74 | 7.62% | 3.47 | 7.24 | 9.57% | 4.59 | 9.40 | 12.70% | 3.65 | 7.46 | 9.96% |
| | Graph WaveNet | 2.69 | 5.15 | 6.90% | 3.07 | 6.22 | 8.37% | 3.53 | 7.37 | 10.01% | 3.10 | 6.25 | 8.43% |
| | MTGNN | 2.69 | 5.18 | 6.88% | 3.05 | 6.17 | 8.19% | 3.49 | 7.23 | 9.87% | 3.08 | 6.19 | 8.31% |
| | MegaCRN | 2.52 | 4.94 | 6.44% | 2.93 | 6.06 | 7.96% | 3.38 | 7.23 | 9.72% | 2.94 | 6.08 | 8.04% |
| | $D^2$STGNN | 2.56 | 4.88 | 6.48% | 2.90 | 5.89 | 7.78% | 3.35 | 7.03 | 9.40% | 2.94 | 5.93 | 7.89% |
| | GMAN | 2.80 | 5.55 | 7.41% | 3.12 | 6.49 | 8.73% | 3.44 | 7.35 | 10.07% | 3.12 | 6.46 | 8.74% |
| | ASTGCN | 4.86 | 9.27 | 9.21% | 5.43 | 10.61 | 10.13% | 6.51 | 12.52 | 11.64% | 5.60 | 10.80 | 10.33% |
| | PDFormer | 3.00 | 6.68 | 7.23% | 3.44 | 7.95 | 8.48% | 4.06 | 9.53 | 10.24% | 3.50 | 8.05 | 8.65% |
| | STGRAT | 2.60 | 5.07 | 6.61% | 3.01 | 6.21 | 8.15% | 3.49 | 7.42 | 10.01% | 3.03 | 6.23 | 8.26% |
| | FC-GAGA | 2.75 | 5.38 | 7.26% | 3.11 | 6.35 | 8.58% | 3.51 | 7.34 | 10.12% | 3.12 | 6.36 | 8.65% |
| | STID | 2.80 | 5.53 | 7.70% | 3.18 | 6.60 | 9.40% | 3.55 | 7.54 | 10.95% | 3.18 | 6.56 | 9.35% |
| | ModWaveMLP | **2.20*** | **4.19*** | **5.65%*** | 2.63* | 5.16* | 6.91%* | 3.21* | 6.63* | 9.12%* | 2.67* | 5.33* | 7.23%* |
| | ModWaveMLP† | 2.29* | 4.22* | 5.69%* | **2.59*** | **5.07*** | **6.81%*** | **3.05*** | **6.24*** | **8.59%*** | **2.64*** | **5.18*** | **7.03%*** |
| **PEMS-BAY** | HA | 1.89 | 4.30 | 4.16% | 2.50 | 5.82 | 5.62% | 3.31 | 7.54 | 7.65% | 2.57 | 5.89 | 5.81% |
| | VAR | 1.74 | 3.16 | 3.60% | 2.32 | 4.25 | 5.00% | 2.93 | 5.44 | 6.50% | 2.33 | 4.28 | 5.03% |
| | SVR | 1.85 | 3.59 | 3.80% | 2.48 | 5.18 | 5.50% | 3.28 | 7.08 | 8.00% | 2.54 | 5.28 | 5.77% |
| | FC-LSTM | 2.05 | 4.19 | 4.80% | 2.20 | 4.55 | 5.20% | 2.37 | 4.96 | 5.70% | 2.21 | 4.57 | 5.23% |
| | DCRNN | 1.38 | 2.95 | 2.90% | 1.74 | 3.97 | 3.90% | 2.07 | 4.74 | 4.90% | 1.73 | 3.89 | 3.90% |
| | STGCN | 1.36 | 2.96 | 2.90% | 1.81 | 4.27 | 4.17% | 2.49 | 5.69 | 5.79% | 1.89 | 4.31 | 4.29% |
| | Graph WaveNet | 1.30 | 2.74 | 2.73% | 1.63 | 3.70 | 3.67% | 1.95 | 4.52 | 4.63% | 1.63 | 3.65 | 3.68% |
| | MTGNN | 1.32 | 2.79 | 2.77% | 1.65 | 3.74 | 3.69% | 1.94 | 4.49 | 4.53% | 1.64 | 3.67 | 3.66% |
| | MegaCRN | 1.28 | 2.72 | 2.67% | 1.60 | 3.68 | 3.57% | 1.88 | 4.42 | 4.41% | 1.59 | 3.61 | 3.55% |
| | $D^2$STGNN | 1.24 | 2.60 | 2.58% | 1.55 | 3.52 | 3.49% | 1.85 | 4.30 | 4.37% | 1.55 | 3.47 | 3.48% |
| | GMAN | 1.34 | 2.91 | 2.86% | 1.63 | 3.76 | 3.68% | 1.86 | 4.32 | 4.37% | 1.61 | 3.66 | 3.64% |
| | ASTGCN | 1.52 | 3.13 | 3.22% | 2.01 | 4.27 | 4.48% | 2.61 | 5.42 | 6.00% | 2.05 | 4.27 | 4.57% |
| | PDFormer | 1.36 | 2.88 | 2.89% | 1.70 | 3.87 | 3.89% | 2.00 | 4.53 | 4.71% | 1.69 | 3.76 | 3.83% |
| | STGRAT | 1.29 | 2.71 | 2.67% | 1.61 | 3.63 | 3.69% | 1.95 | 4.64 | 4.54% | 1.62 | 3.66 | 3.63% |
| | FC-GAGA | 1.35 | 2.86 | 2.83% | 1.68 | 3.80 | 3.78% | 1.99 | 4.52 | 4.67% | 1.67 | 3.73 | 3.76% |
| | STID | 1.30 | 2.81 | 2.73% | 1.62 | 3.72 | 3.68% | 1.89 | 4.40 | 4.47% | 1.60 | 3.64 | 3.63% |
| | ModWaveMLP | **0.86*** | **1.80*** | **1.76%*** | **1.22*** | 2.89* | **2.72%*** | 1.80* | 4.18* | 4.20%* | **1.29*** | 2.96* | 2.89%* |
| | ModWaveMLP† | 0.98* | 1.89* | 1.92%* | 1.25* | **2.87*** | 2.74%* | **1.63*** | **3.95*** | **3.88%*** | **1.29*** | **2.90*** | **2.85%*** |

Table 2: Ablation study on METR-LA

| Methods | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| MoDWaveMLP | 2.18 | 4.02 | 5.45% | 2.67 | 5.24 | 7.03% | 3.28 | 6.37 | 9.39% | 2.71 | 5.38 | 7.29% |
| w/o Day Gate | 2.20 | 4.20 | 5.68% | 2.73 | 5.49 | 7.42% | 3.40 | 7.10 | 9.82% | 2.78 | 5.60 | 7.64% |
| w/o Week Gate | 2.19 | 4.04 | 5.46% | 2.67 | 5.24 | 7.04% | 3.32 | 6.81 | 9.41% | 2.73 | 5.36 | 7.30% |
| w/o -Add | 2.24 | 4.18 | 5.66% | 2.77 | 5.52 | 7.44% | 3.37 | 6.99 | 9.68% | 2.79 | 5.56 | 7.59% |
| w/o Wavelet | 2.72 | 5.29 | 7.11% | 3.05 | 6.25 | 8.43% | 3.42 | 7.20 | 9.99% | 3.06 | 6.25 | 8.51% |
| w/ MDLBlock | 2.44 | 4.54 | 6.23% | 2.94 | 5.86 | 7.99% | 3.50 | 7.26 | 10.18% | 2.96 | 5.89 | 8.13% |

and 21.49%, respectively, which greatly outperforms the next best, D$^2$STGNN, which demonstrates that our mode decomposition module reduces the noise has a more important impact on short-term prediction. As our model uses MLP structure, it demonstrates quicker training and inference (refer to Appendix Section E). (3)GNN-based traffic models excel, combining spatial and temporal dependence modeling. Among GNN-based models, D$^2$STGNN excels. Yet, ModWaveMLP leverages embedding and node history dynamically, achieving better performance. (4)For self-attention and transformer models, STGRAT is the optimal baseline, capturing long sequence features and spatial correlations. ModWaveMLP's mode decomposition through stacking yields competitive performance. (5)Compared to FC-GAGA and STID (both MLP-based), ModWaveMLP's wavelet-based denoising and mode decomposition achieve optimal MLP-based model performance.

## Ablation Study

To further investigate the effectiveness of different parts in ModWaveMLP, we compare ModWaveMLP with the following variants. (1)w/o Day Gate: Removes day cycle gating and encoding. (2)w/o Week Gate: Eliminates weekly cycle gating and encoding. (3)w/o MDLBlock_Add: Omits information enrichment from mode decomposition. (4)w/o Wavelet Decomposition: Excludes wavelet decomposition and coding. (5)w/ MDLBlocks: Replaces MDLblocks with MLPResidual, a base module in TiDE transformer model(Das et al. 2023).

Tab.2 shows the comparison of these variants on the METR-LA datasets. Based on the results, we can conclude the followings: (1) Day and week gate module removal degrades performance. Cycles within a day impact traffic data significantly.(2)MDLBlock_Add enhances mode decomposition, improving information encoding.(3)w/o Wavelet Decomposition performs worse for short predictions (15 and 30 minutes) due to ignoring noise's impact on data.(4)w/ MDLBlock's performance is worse, possibly due to inapplicability of NLP-related normalization and dropout, leading to gradient issues.Further we remove the normalization and dropout operations and the final performance is still lower than ModWaveMLP.

## Robustness Study

In this section, we introduce Gaussian noise to the training data to simulate data noise interference. Noise is added with varying means (-4 to 4) and variances (1 to 9), resulting in 81 noise groups. This aims to test the proposed model's robustness. We compare our model with FC-GAGA, another MLP-based model, in the same noisy environment. Figs. 2 a, b, and c. Fig. 2 e, f, and g display MAE, RMSE, and MAPE changes under mixed noise conditions for both models.

The results in Figs. 2 a, d indicate a maximum MAE change of 0.6 for ModWaveMLP, compared to FC-GAGA's 1.2. Figs. 2 c, f showcase RMSE changes under interference, with ModWaveMLP's maximum RMSE at 7.0 and a deviation range of about 1.5, while FC-GAGA exhibits a maximum deviation of 12 and an overall deviation of 6. ModWaveMLP's error distribution is relatively smooth, adapting

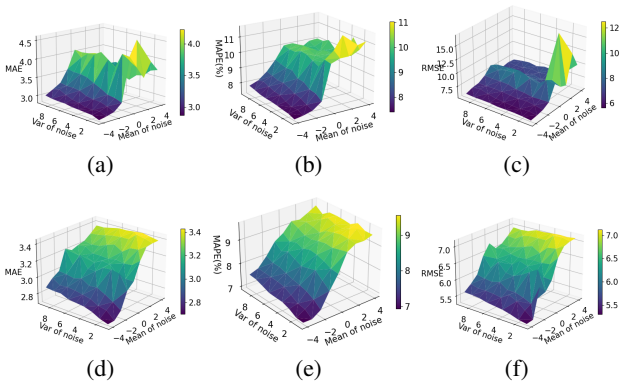


Figure 2: Variation of Individual Metrics under Noise Interference

well to noise-induced disturbances. As noise mean and variance increase, ModWaveMLP maintains a smoother error distribution, while FC-GAGA's model error surges under local noise disturbance, emphasizing ModWaveMLP's robustness and its ability to handle diverse scenarios and noise attacks.The reliability of ModWaveMLP starts declining with higher noise variations but remains insensitive to variance changes. Model accuracy primarily responds to mean value alterations, indicating ModWaveMLP's resistance to local abrupt data changes due to its wavelet-based information learning.

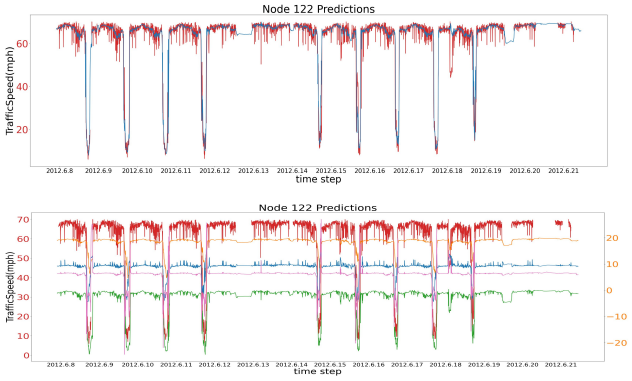


Figure 3: ModWaveMLP 15 min ahead forecasts for node 122 in METR-LA dataset.where the true value (red), the output of prediction value ((a)blue),the output of layer 0 ((b)blue), the output of layer 1 (green), the output of layer 2 (orange), and the output curve of layer 3 (pink), with the red-green y-axis to the left and the rest to the right

## Case Study

In this section, we analyze the correlation between traffic modes and dynamic nodes in ModWaveMLP's layers. We validate mode decomposition learning by visualizing model output and correlated nodes. We will take node 122 in the METR-LA dataset as an example, which is situated on the key Hollywood Freeway Fig. 4a. Fig. 3a shows

ModWaveMLP's prediction curve alongside real values for node 122. Clear daily cycles are seen in traffic data during weekdays, with 5 cycles of peaks and valleys indicating traffic speed changes. Also influenced by the fact that people are resting on weekends, the traffic data on weekends have different traffic patterns than on weekdays. Due to the time gating mechanism module designed in MoDWaveMLP, our model predictions fit this trend well.The final ModWaveMLP prediction is the average of layer predictions. Fig. 3b displays layer contributions for a 4-layer stack (scaled by ¼). Unlike typical stacks, all ModWaveMLP layers contribute. Layer 2 establishes a baseline, and Layer 1 enhances local changes. Layer 3's input is smoother, learning inter-node delays. Layer 4 refines baselines with recent data, countering abrupt changes like on 2012.6.15 and 2012.6.16. Ablation experiments confirm wavelet noise reduction's impact on short-term prediction correction.Fig. 4 b-e depicts node weight distribution learned by embedding in the Node Information Aggregation Module across layers 1 to 4 for node 122. Each layer captures distinct node relationships, reflecting diverse information aggregation. Fig. 4 f-i details this in 3D plots for each layer. In Fig. 4b, layer 1 nodes encircle node 122 and Cahuenga Hill, suggesting data integration from nearby nodes. Fig. 4c shows layer 2 nodes along Hollywood Freeway, crucial for stable baseline prediction. Layers 3 to 4 (Fig. 4d,e) cluster nodes tightly around node 122, especially with east-west nodes from Ventura Freeway. This indicates iterative baseline updates focusing on closer nodes, refining neighboring information to correct noisy predictions. This study confirms our mode decomposition learning block's mode separation capability in

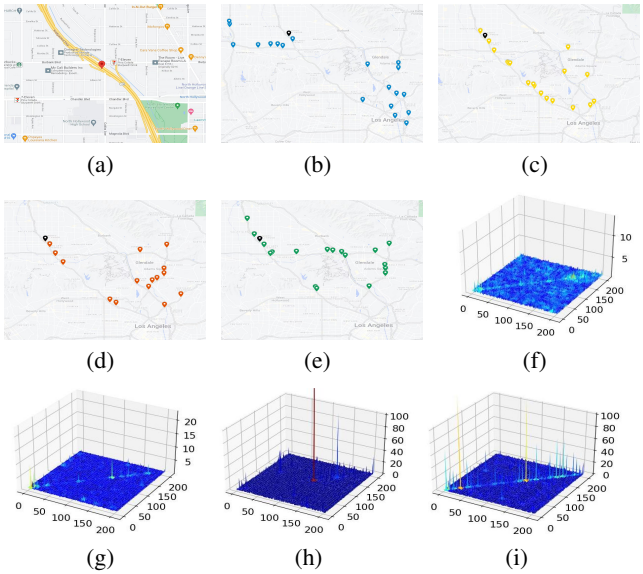


Figure 4: Location of point 122 on the map, Maps of highest correlation of 122 node in each layer of embedding for layers 1, 2, 3 and 4. Black star is the forecasted node, and ModWaveMLP weight visualization of the 4 layers of embedding

the original data. Layers collaborate to enhance and refine predictions. Ablation experiments stress component importance, particularly wavelet decomposition and the time gate module, which significantly affect model performance when absent.

## Related Work

Recently, deep learning has gained traction in addressing traffic prediction challenges. Initially, temporal prediction was the focus, centering on time-related aspects(Sutskever, Vinyals, and Le 2014). Convolutional Neural Networks (CNNs) were then applied to grid-based traffic data to capture spatial dependencies(Zhang, Zheng, and Qi 2017; Lin et al. 2020). Graph Neural Networks (GNNs) gained prominence for traffic prediction, leveraging their ability to model graph data(James 2022; Yu, Yin, and Zhu 2018; Song et al. 2020; Wu et al. 2020; Shao et al. 2022b; Li et al. 2022; Shang, Chen, and Bi 2021; Choi et al. 2022; Jiang et al. 2023b). The attention mechanism gained popularity for dynamic dependency modeling(Guo et al. 2019; Zheng et al. 2020). The success of the transformer in different tasks such as text and images has motivated researchers to design new structures, which are based on the new features of transportation that have been mined(Jiang et al. 2023a; Park et al. 2020; Ye et al. 2022).

Unlike transformers' position coding for text generation, temporal prediction models usually learn periodicity without this coding(Vaswani et al. 2017; Sutskever, Vinyals, and Le 2014). Researchers found that simple linear layers could outperform transformer in long time prediction and uncovered the problem of channel independence in multivariate time series(Zeng et al. 2023; Li et al. 2023b). Further, researchers replaced multi-attention structures with MLP and introduced temporal and spatial embedding structures as alternatives to GNNs in traffic prediction(Das et al. 2023; Shao et al. 2022a; Oreshkin et al. 2021).

## Conclusion

We propose a model based on the MLP structure designed according to the ideas of mode decomposition and wavelet noise reduction learning. Compared with previous researchers who manually design the corresponding model structure based on the characteristics of the traffic data, our model decomposes the modes in the traffic data through constant mode decomposition.Through the learning of wavelet noise reduction information,the model can remove the effect of noise on the traffic data. Comparison experiments on real-world dataset and baseline demonstrate the effectiveness of our model, which has better performance than the one based on GNN and transformer structure. Amidst the growing complexity of traffic prediction models, our study highlights the effectiveness of a straightforward MLP-based approach. While Large Model offer novel insights, they also introduce challenges like test data leakage. Our work demonstrates achieving accurate traffic prediction without complex structures, pointing to new future directions in large model design and traffic prediction research.

## References


Chen, C.; Liu, Y.; Chen, L.; and Zhang, C. 2022. Bidirectional spatial-temporal adaptive transformer for Urban traffic flow forecasting. *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, X.; Chen, H.; Yang, Y.; Wu, H.; Zhang, W.; Zhao, J.; and Xiong, Y. 2021. Traffic flow prediction by an ensemble framework with data denoising and deep learning model. *Physica A: Statistical Mechanics and Its Applications*, 565: 125574.

Choi, J.; Choi, H.; Hwang, J.; and Park, N. 2022. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6367–6374.

Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *arXiv preprint arXiv:2304.08424*.

Fang, Z.; Pan, L.; Chen, L.; Du, Y.; and Gao, Y. 2021. MDTP: A multi-source deep traffic prediction framework over spatio-temporal trajectory data. *Proceedings of the VLDB Endowment*, 14(8): 1289–1297.

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

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 87–110.

James, J. 2022. Graph construction for traffic prediction: A data-driven approach. *IEEE Transactions on Intelligent Transportation Systems*, 23(9): 15015–15027.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023a. PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. In *AAAI*. AAAI Press.

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

Jiang, W.; and Luo, J. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207: 117921.

Jin, G.; Li, F.; Zhang, J.; Wang, M.; and Huang, J. 2022. Automated dilated spatio-temporal synchronous graph modeling for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Lee, H.; Park, C.; Jin, S.; Chu, H.; Choo, J.; and Ko, S. 2021. An empirical experiment on deep learning models for predicting traffic data. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1817–1822. IEEE.

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

Li, F.; Yan, H.; Jin, G.; Liu, Y.; Li, Y.; and Jin, D. 2022. Automated spatio-temporal synchronous modeling with multiple graphs for traffic prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1084–1093.

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

Li, Z.; Rao, Z.; Pan, L.; and Xu, Z. 2023b. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*.

Lin, H.; Bai, R.; Jia, W.; Yang, X.; and You, Y. 2020. Preserving dynamic attention for long-term spatial-temporal prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 36–46.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

Oreshkin, B. N.; Amini, A.; Coyle, L.; and Coates, M. J. 2021. FC-GAGA: Fully Connected Gated Graph Architecture for Spatio-Temporal Traffic Forecasting. In *AAAI*.

Park, C.; Lee, C.; Bahng, H.; Tae, Y.; Jin, S.; Kim, K.; Ko, S.; and Choo, J. 2020. ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 1215–1224.

Shang, C.; Chen, J.; and Bi, J. 2021. Discrete Graph Structure Learning for Forecasting Multiple Time Series. In *International Conference on Learning Representations*.

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

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

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

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Tang, J.; Chen, X.; Hu, Z.; Zong, F.; Han, C.; and Li, L. 2019. Traffic flow prediction based on combination of support vector machine and data denoising schemes. *Physica A: Statistical Mechanics and its Applications*, 534: 120642.

Tran, L.; Mun, M. Y.; Lim, M.; Yamato, J.; Huh, N.; and Shahabi, C. 2020. DeepTRANS: a deep learning system for

public bus travel time estimation using traffic forecasting. *Proceedings of the VLDB Endowment*, 13(12): 2957–2960.

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

Wu, Y.; Tan, H.; Qin, L.; Ran, B.; and Jiang, Z. 2018. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90: 166–180.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. 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*, 753–763.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 1907–1913.

Yan, H.; Ma, X.; and Pu, Z. 2021. Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems*, 23(11): 22386–22399.

Ye, X.; Fang, S.; Sun, F.; Zhang, C.; and Xiang, S. 2022. Meta graph transformer: A novel framework for spatial–temporal traffic prediction. *Neurocomputing*, 491: 544–563.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 3634–3640.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, J.; Shi, X.; Xie, J.; Ma, H.; King, I.; and Yeung, D. Y. 2018. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*.

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

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

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.

# Appendix for *ModWaveMLP: MLP-based Mode Decomposition and Wavelet Denoising Model to Defeat Complex Structures in Traffic Forecasting*

## A Wavelet Decomposition 3d Tensor Construction

Algorithm 1 showing how we reconstruct the wavelet tensor

Algorithm 1: 3D tensor reconstruction algorithm

**Input**: Traffic data $Data$; Decomposition layers $L$; Basis function $wt_i \in [wt_1, ..., wt_n]$

**Output**: Wavelet Decomposition Information Enhancement data $wtData$

$wtData$= []
**for** $wt_i$ in $[wt_1, ..., wt_n]$ **do**
    $A_0 = Data$
    **for** i in range($L$) **do**
        $D_i, A_i$= decompose($A_{i-1}, wt$)
    **end for**
    $X = D_1 + D_2 + ... + D_L + A_L$
    **for** i in range($L$) **do**
        **if** mode = 'soft' **then**
            threshold($D_i$,'soft')
        **else if** mode = 'hard' **then**
            threshold($D_i$,'hard')
        **end if**
    **end for**
    data'=reconstruct($X, wt$)
    $wtData$.append(data')
**end for**
**return** $wtData$

## B Baselines Details

All experiments are conducted on a machine with the NVIDIA GeForce 4090 GPU and 512GB memory.To eliminate experimental environment variations, we adopt the best results published in papers for our baselines. Some classical models use results from the paper D$^2$STGNN accepted by VLDB22.

- HA: The Historical Average model represents traffic flows as a periodic process and leverages weighted averages from past periods to make predictions for future periods.
- VAR: Vector Auto-Regression assumes stationarity in the time series data and establishes relationships between the time series and their lag values. It considers multiple variables simultaneously and models their interdependencies over time.
- SVR: Support Vector Regression (SVR) utilizes a linear support vector machine approach for classical time series regression tasks. SVR is a variant of Support Vector Machines (SVM) adapted for regression problems.
- FC-LSTM: LSTM is a type of recurrent neural network (RNN) that incorporates memory cells, allowing it to retain information for extended periods. The fully connected hidden units refer to the connections between all neurons in consecutive layers.
- DCRNN: The Diffusion Convolutional Recurrent Neural Network (DCRNN) models traffic flow as a diffusion process. It introduces a diffusion convolutional layer in place of the fully connected layer within a Gated Recurrent Unit (GRU), resulting in a new architecture called Diffusion Convolutional Gated Recurrent Unit (DCGRU) .
- STGCN: Spatiotemporal graph convolutional networks that fuse graph convolution with gated spatiotemporal convolution to predict the output of future time steps.
- Graph WaveNet: Graph WaveNet employs a combination of Gated Temporal Convolutional Networks (Gated TCN) and Graph Convolutional Networks (GCN) in a stacked manner.
- MTGNN: MTGNN extends Graph WaveNet through the mix-hop propagation layer in the spatial module, the dilated inception layer in the temporal module, and a more delicate graph learning layer.
- MegaCRN: Meta-Graph Convolutional Recurrent Network (MegaCRN) plugs the Meta-Graph Learner powered by a MetaNode Bank into GCRN encoder-decoder.
- D$^2$STGNN: D$^2$STGNN propose a novel Decoupled Spatial-Temporal Framework (DSTF) that separates the diffusion and inherent traffic information in a data-driven manner, which encompasses a unique estimation gate and a residual decomposition mechanism. The separated signals can be handled subsequently by the diffusion and inherent modules separately. D$^2$STGNN captures spatial-temporal correlations and also features a dynamic graph learning module that targets the learning of the dynamic characteristics of traffic networks.
- GMAN: GMAN is an attention-based model that integrates spatial, temporal, and transform attentions.
- ASTGCN: ASTGCN leverages a spatial-temporal attention mechanism to simultaneously capture the dynamic spatial and temporal characteristics of traffic data.
- PDFormer: PDFormer introduces several key components to enhance traffic prediction performance. It includes a spatial self-attention module, which allows the model to capture dynamic spatial dependencies in the traffic data. And it's designed a delay-aware feature transformation module to integrate historical traffic modes into spatial self-attention and explicitly model the time delay of spatial information propagation.
- STGRAT: Spatio-Temporal Graph Attention Network for Traffic Forecasting, an encoder-decoder model using the positional encoding method of the Transformer to capture features of long sequences and node attention to capture spatial correlation) .
- FC-GAGA: FC-GAGA is a novel model that combines elements from the fully-connected time series model called N-BEATS and a hard graph gate mechanism. It incorporates a learnable graph weight mechanism to assign weights to historical observations from all other nodes in the graph.
- STID: STID based on an intuitive idea of attaching spatial-temporal identity information. It utilizes a spatial

embedding matrix, and two temporal embedding matrices to indicate the spatial and temporal identities.

## C Evaluation Metrics

We use three metrics in the experiments: (1) Mean Absolute Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Squared Error (RMSE). Missing values are excluded when calculating these metrics. The formulas are as follows:

$$\mathrm{MAE}(x, \hat{x}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} |x_i - \hat{x}_i|$$

$$\mathrm{MAPE}(x, \hat{x}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \frac{|x_i - \hat{x}_i|}{x_i}$$

$$\mathrm{RMSE}(x, \hat{x}) = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} (x_i - \hat{x}_i)^2}$$

## D Wavelet Decomposition Hyperparameter Setting Experiment

The aim of the experiments in this section is to select the best wavelet reconstruction parameters as well as to prove their effectiveness. Experiments are conducted in three aspects, namely, selection of wavelet basis functions, wavelet threshold reconstruction method and number of decomposition layers, to explore the most suitable wavelet reconstruction method. Since the traffic flow data collected by each sensor node is discrete one-dimensional data, the ModWaveMLP model structure is introduced at the stage of selecting a single wavelet basis function for the experiments of wavelet basis functions and subsequent experiments comparing the soft and hard thresholds and the number of decomposition layers. Since the original traffic flow data is a discrete type of data, the wavelet basis functions are selected for comparison with discrete wavelets, which are subdivided into a single discrete wavelet and a discrete wavelet family, which will contain multiple functions of the same type, e.g., (the sym family contains sym2, sym3, etc.).

The first function in the function group of the wavelet family and a single wavelet basis function were selected for the data noise reduction comparison experiments, and the noise reduction effect of the one-dimensional wavelet basis function is shown in Tab.1.Among the seven wavelet functions compared that can handle discrete data, the prediction accuracy of the first 30 minutes of coif , sym, and dmey is higher because these types of wavelet functions are more effective in dealing with data with symmetric features, and the intrinsic trend of traffic data consists of periodicities with symmetric patterns in a macroscopic sense. In addition, the predictions of different wavelets at different moments show their own lead, which inspires us to use wavelet combinations for data denoising and reconstruction.

Based on the above experimental results, this section selects the first 4 functions as combinations in the three wavelet families of db, sym, and coif, and 3 wavelet function combinations are formulated for the comparison test, and the Tab.2 shows the experimental results. Compare the prediction effects of different combinations of wavelet functions (e.g., "db1", "db2", "db3", "db4") in the same series, the results show that the db model performed best in the short-term prediction (30 minutes) and the sym model performed best at 60 minutes

The number of layers is a key kernel parameter that determines the complexity of the feature engineering to some extent, so the number of decomposition layers needs to be determined experimentally. Since the number of layers of wavelet decomposition has a great impact on the final prediction results, this section compares the effect of 1-4 layers of wavelet decomposition layers on the noise reduction effect.

The experimental results are shown in the table, it can be seen that the wavelet decomposition can indeed eliminate the noise of the data, with the increase of the number of decomposition layers the model performance is improving. The prediction result of decomposition 2 layers has a great improvement compared with decomposition 1 layer in a short time, with the number of layers decomposition model's long time prediction (60min) is more accurate but the short time prediction (15min) performance has declined. In practical use, considering that the more layers, the more complex the structure of the wavelet transform module, in the final experiment we chose to decompose four layers of the model, according to the requirements of the application scenario, the researchers can choose more than 2 layers of the appropriate number of decomposition layers.

The appropriate thresholding method is also one of the important influencing factors in the wavelet noise reduction process, so this section compares the two schemes of soft thresholding and hard thresholding, and the results are shown in Tab.4, which show that soft thresholding is more suitable for processing the traffic flow data than hard thresholding, because hard thresholding retains the spiky features in the data, and soft thresholding makes the time series become smoother, which is more conducive to the extraction of the significant changes of the traffic flow at a certain moment of each day moment in time.

The difference between this method and the existing methods is that we use the reconstructed data as input to the prediction model instead of using the components of the wavelet decomposition and consider the effect of the wavelet basis function and the thresholding method on the reconstruction. Since each dimension contains the intrinsic trend in the original data, it is equivalent to increasing the weight of the intrinsic trend in the original data, highlighting the importance of the intrinsic trend in the transportation data.

## E Model Efficiency Study

In this section we explore the running efficiency of ModWaveMLP, we choose the transformer-based PDFormer and the graph neural network-based $D^2$STGNN for comparison, all the models are run on a single RTX 4090 GPU, Tab. 5 shows the training time of the three models on the METR-LA and PEMS-BAY datasets for training one epoch on the METR-LA and PEMS-BAY datasets. It can be seen that the MLP-based ModWaveMLP model spends the least

Table 1: Noise reduction effect of one-dimensional wavelet basis function

| Methods | Wavelet basis function | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| **ModWaveMLP** | haar | 2.52 | 4.91 | 6.51% | 2.95 | 6.01 | 8.02% | 3.43 | 7.19 | 9.89% |
| | db1 | 2.51 | 4.91 | 6.52% | 2.93 | 5.99 | 8.05% | 3.40 | 7.18 | 9.92% |
| | coif1 | 2.42 | 4.59 | 6.23% | 2.89 | 5.85 | 8.0% | 3.42 | 7.17 | 10.04% |
| | bior1.1 | 2.47 | 4.72 | 6.36% | 2.91 | 6.08 | 8.09% | 3.42 | 7.3 | 10.07% |
| | rbio1.1 | 2.52 | 4.89 | 6.55% | 2.94 | 6.0 | 8.11% | 3.41 | 7.16 | 9.93% |
| | dmey | 2.38 | 4.42 | 6.03% | 2.90 | 5.85 | 7.9% | 3.42 | 7.17 | 9.97% |
| | sym2 | 2.39 | 4.52 | 6.13% | 2.91 | 5.88 | 7.97% | 3.43 | 7.17 | 10.08% |

Table 2: Combination effect of wavelet basis functions

| Methods | Combination of wavelet function | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| **ModWaveMLP** | 'sym2', 'sym3', 'sym4', 'sym5' | 2.24 | 4.11 | 5.68% | 2.77 | 5.42 | 7.31% | **3.33** | **6.8** | **9.49%** |
| | 'coif1', 'coif2', 'coif3', 'coif4' | 2.37 | 4.47 | 6.02% | 2.87 | 6.01 | 7.73% | 3.40 | 7.03 | 9.81% |
| | 'db1', 'db2', 'db3', 'db4' | **2.21** | **4.07** | **5.51%** | **2.72** | **5.35** | **7.23%** | 3.37 | 6.98 | 9.56% |

Table 3: Experiments on wavelet decomposition layers

| Methods | Level | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| **ModWaveMLP(db)** | 1 | 2.68 | 5.15 | 6.95% | 3.06 | 6.25 | 8.39% | 3.46 | 7.28 | 10.02% |
| | 2 | 2.32 | 4.34 | 5.85% | 2.94 | 5.91 | 8.01% | 3.42 | 7.18 | 9.89% |
| | 3 | 2.21 | **4.07** | **5.51%** | 2.72 | 5.35 | 7.23% | 3.37 | 6.98 | 9.56% |
| | 4 | **2.20** | 4.19 | 5.65% | 2.61 | 5.16 | 6.91% | 3.21 | 6.63 | 9.12% |
| | 5 | 2.29 | 4.22 | 5.69% | **2.59** | **5.07** | **6.81%** | **3.05** | **6.24** | **8.59%** |

Table 4: Comparison of hard soft and thresholds

| Methods | Threshold | Level | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| **ModWaveMLP(db)** | soft | 2 | 2.32 | 4.32 | 5.88% | 2.93 | 5.87 | 7.09% | 3.39 | 7.09 | 9.59% |
| | | 3 | 2.21 | 4.0 | 5.44% | 2.72 | 5.20 | 7.0% | 3.29 | 6.80 | 9.39% |
| | hard | 2 | 2.61 | 5.14 | 6.81% | 3.02 | 6.18 | 8.32% | 3.43 | 7.22 | 10.04% |
| | | 3 | 2.61 | 5.26 | 6.81% | 3.02 | 6.22 | 8.32% | 3.42 | 7.21 | 10.1% |

amount of training time, and the PDFormer spends the most amount of training time because it is based on the self-attention mechanism and the transformer architecture, which has more number of parameters and higher time complexity, and has a higher demand on hardware resources.

Table 5: The time spent by each model to train an epoch on different datasets in the same hardware environment.

| **Methods** | **Training (seconds/epoch)** | |
|---|---|---|
| | **METR-LA** | **PEMS-BAY** |
| MoDWaveMLP | 119 | 157 |
| $D^2$STGNN | 135 | 187 |
| PDFormer | 228 | 612 |