WaveForM: Graph Enhanced Wavelet Learning for Long Sequence Forecasting of Multivariate Time Series

Fuhao Yang[1], Xin Li[*1], Min Wang[1], Hongyu Zang[1], Wei Pang[2], Mingzhong Wang[3]

[1] Beijing Institute of Technology
[2] Heriot-Watt University
[3] The University of the Sunshine Coast
{yfh, xinli, 3220225187, zanghyu}@bit.edu.cn, w.pang@hw.ac.uk, mwang@usc.edu.au

Abstract

Multivariate time series (MTS) analysis and forecasting are crucial in many real-world applications, such as smart traffic management and weather forecasting. However, most existing work either focuses on short sequence forecasting or makes predictions predominantly with time domain features, which is not effective at removing noises with irregular frequencies in MTS. Therefore, we propose WAVEFORM , an end-to-end graph enhanced Wavelet learning framework for long sequence FORecasting of MTS. WaveForM first utilizes Discrete Wavelet Transform (DWT) to represent MTS in the wavelet domain, which captures both frequency and time domain features with a sound theoretical basis. To enable the effective learning in the wavelet domain, we further propose a graph constructor, which learns a global graph to represent the relationships between MTS variables, and graph-enhanced prediction modules, which utilize dilated convolution and graph convolution to capture the correlations between time series and predict the wavelet coefficients at different levels. Extensive experiments on five real-world forecasting datasets show that our model can achieve considerable performance improvement over different prediction lengths against the most competitive baseline of each dataset.

Introduction

Multiple interconnected streams of data, also known as multivariate time series (MTS), have pervasive presence in real-world applications. Examples of MTS include the recorded traffic flows from various roadway sensors and the weather observations from multiple weather stations over time. Multivariate time series forecasting, which makes predictions based on historical MTS observations, has attracted extensive interest as it can integrate multiple sources of observations to provide a global view of applications and help make meaningful and accurate application-wide predictions. For example, to predict the future power consumption of a household, it is beneficial to consider and integrate the usage observations of multiple sectors, such as kitchen, laundry, and the average current intensity of the household.

Early solutions (Box and Jenkins 1970), which utilize statistical models, generally assume linear dependencies among variables, thus failing to capture complex non-linear patterns, which frequently occur in MTS. In recent work, researchers have proposed a series of graph neural network (GNN)-based models to capture interconnections and inter-dependencies (also known as spatial dependencies) among MTS due to GNN's strength in modeling complex structures of graph data. For example, STGCN (Yu, Yin, and Zhu 2017) skillfully utilizes both graph convolution and gated causal convolution to tackle the MTS prediction problems in the traffic domain. Graph Multi-attention Network (GMAN) (Zheng et al. 2020) extends STGCN with an encoder-decoder architecture and incorporates attention mechanisms to better capture spatial-temporal relations in traffic data. The use of GNN in STGCN and GMAN relies on the assumption that prior knowledge of stable relationships among variables is available, and such knowledge is represented in a pre-defined graph structure. MTGNN (Wu et al. 2020) focuses on learning and recovering the latent dependencies (graph structure) among variables for tasks without explicitly defined graph structures by using a graph learning module, leading to better interpretability and performance for MTS forecasting tasks.

However, the existing work still overlooks *long sequence forecasting* (LSF) of MTS, which uses a given length of MTS to predict longer future sequences. LSF of MST is crucial for facilitating long-term planning and offering early warning in various real-world applications. However, predicting long sequences is challenging as long-term MTS are often composed of more entangled temporal patterns than short-term ones, and overlooking this may lead to unreliable discoveries of temporal dependencies (Wu et al. 2021). Recently, transformer-based architectures have proven their effectiveness in modeling sequential data owing to the use of self-attention mechanisms (Zaheer et al. 2020), empowering MTS forecasting models for long-term prediction (Wen et al. 2022). However, these models frequently suffer from high computational cost in LSF. The existing transformer-based LSF approaches mainly focus on developing sparse self-attention schema to improve model efficiency, inevitably sacrificing the rate of information utilization and resulting in a bottleneck for MTS LSF.

In comparison, this paper proposed a novel solution for effective long sequence forecasting in MTS.

In practice, MTS can be analyzed in the time domain,

[*]Corresponding author.

which studies how signals change over time, and/or frequency domain, which studies signals from the perspective of their frequencies. However, we noticed that most existing work with deep learning models centers on extracting and utilizing time-domain features from MTS, leaving frequency domain analysis generally unattended. Although some models, such as Autoformer (Wu et al. 2021) and FEDFormer (Zhou et al. 2022), utilize time-frequency transformations, such as Fourier transform, they mainly aim to reduce the time complexity of the Transformer models rather than fully exploit the rich features in the frequency domain. Autoformer (Wu et al. 2021) and FEDFormer (Zhou et al. 2022) demonstrated their effectiveness in leveraging additional frequency domain features with Fourier transform or discrete wavelet transform. However, they generally simply utilize the extracted frequency domain features as a complement to the representations in the time domain. They feed the combined/concatenated features to deep learning models for forecasting. We argue that such a simple combination of features from two different domains cannot provide clear and sufficient information to the deep learning models and diminish the effect of features in frequency domain. It lacks a theoretical basis and may even introduce noises to the models and lead to sub-optimal performance.

Therefore, we propose to model MTS in the "*wavelet domain*" to effectively capture and exploit wavelet domain features, leveraging the capability of Discrete Wavelet Transform (DWT) (Shensa et al. 1992) which captures both the frequency-domain and time-domain features of MTS in a theoretically guaranteed framework. More specifically, we utilize DWT to decompose MTS into different frequency bands (wavelets) with different resolutions, which are represented as wavelet coefficients, to enable more sophisticated time series feature extraction. Thereafter, we propose to adapt a Graph-enhanced Prediction module (GP) to model the changes of the wavelet coefficients with the same resolution over time. The graph convolution in GP is used to tackle the inter-dependencies among variables. Thus, the inter-dependency relationship can be captured at different resolutions in the wavelet domain. More importantly, we inject the same/global graph structure across all GP modules, indicating that variables in different views in the wavelet domain share the same basic message-passing behavior and avoid model overfitting. Once we obtain the predicted wavelet coefficients, we utilize Inverse Discrete Wavelet Transform (IDWT) to enable supervised learning in the training set. Note that the global graph in the framework is learned end-to-end from data, which leads to a better interpretation of the inter-dependencies among variables.

The novel contributions of this research is as follows:

1. We propose a DWT-based end-to-end framework that transforms MTS into a wavelet domain for MTS long sequence prediction tasks. Owing to the features of DWT, our model is capable of fully exploiting the inherent features of MTS in both frequency and time domains.
2. We propose a global graph constructor to extract global information on the interrelationship among variables in the wavelet domain, preventing the framework training from overfitting.
3. We conducted comprehensive experiments on long sequence forecasting tasks in MTS and the results show our model consistently/effectively outperforms the state-of-the-art models for LSF tasks by a large margin.

Related Work

MTS forecasting can be considered a typical seq2seq task and various deep sequence models have been proposed. DeepAR (Salinas et al. 2020) combines the idea of autoregression with recurrent neural networks (RNNs) to model the probability distribution of sequences. Besides RNN models, convolutional neural networks (CNNs) are also used for MTS forecasting. For example, Graph WaveNet (Wu et al. 2019) utilizes the dilated causal convolution to force the model to focus only on historical information and expands the perspective field to obtain a broader range of periodic and tendency patterns. However, most of the existing models are not designed for LSF of MTS.

Transformer-based models for MTS predictions have received increasing attention (Wen et al. 2022) with two strands of research along this line. One strand, such as LogTrans (Li et al. 2019) and Autoformer (Wu et al. 2021), focuses on developing sparse attention mechanisms to replace the original attention mechanism which has been recognized as the computational bottleneck for long sequence MTS predictions due to its $O(L^2)$ complexity in both time and space. Another strand, such as Informer (Zhou et al. 2021) and Pyraformer (Liu et al. 2021), focuses on reducing the computational complexity by improving the attention mechanism at the decomposed structural level by introducing different resolution representations for the original sequences through convolution operators and/or Fourier transform to obtain the time dependence of the original sequences at different scales. However, such resolutions are solely or mostly performed in the time domain. Their purpose is to reduce the sequence length and thus improve computational efficiency. Therefore, the frequency domain information is not fully exploited as it is used as a supplementary or a means of reducing computational complexity.

Spatial-temporal GNNs have also been proposed for MTS forecasting tasks. They model each variate in MTS as a graph node and then represent the interdependencies between nodes with a latent graph. The features of each node are obtained by mainly considering the temporal dependency among each time series. Specifically, Graph WaveNet (Wu et al. 2019) designs a self-adaptive matrix to reveal the spatial dependencies with node embeddings. MTGNN (Wu et al. 2020) and GTS (Shang, Chen, and Bi 2021) extend Graph WaveNet by jointly learning the latent graph and spatial-temporal GNN in an end-to-end framework with more sophisticated designs.

Autoformer (Wu et al. 2021) and FEDFormer (Zhou et al. 2022) utilized Fourier Transform to extract frequency domain features, which are then simply concatenated with time domain features for further processing in deep learning models. However, as we have argued before, the simple mixture of features from completely different domains may intro-

duce additional noises or phantom dependencies as there is no general theoretical guide for the cross-domain concatenation/combination in deep learning. Thus, this paper proposes to center the analysis on the wavelet domain, which theoretically reflects both time and frequency features, to better exploit the complex patterns in MTS.

Methodology

This section explains the details of the proposed WAVEFORM , the overview of which is illustrated in Fig. 1. WAVEFORM is a multi-resolution analysis (MRA) model based on discrete wavelet transforms, and it forecasts MTS in the wavelet domain. WAVEFORM consists of three main components: discrete wavelet transform (DWT) module, global graph constructor (GGC), and graph-enhanced prediction (GP) modules.

As an MRA model, WAVEFORM relies on the scaling and translation of the DWT module to obtain the detail coefficients ($\mathbf{cD}_i$) and approximate coefficients ($\mathbf{cA}_i$) of different levels ($i = 1, 2, \dots$) in the wavelet domain. The GP modules utilize dilated convolution and graph convolution to capture the correlations between time series and predict the wavelet coefficients at different levels, and all these modules share the same graph that is learned from GGC. With the use of an inverse DWT module, the framework is trained end-to-end.

The technical details of each component are presented in the rest of this section.

Problem Definition

An MTS is denoted as $\mathbf{X} = [\boldsymbol{x}_1^\intercal; \boldsymbol{x}_2^\intercal; \cdots; \boldsymbol{x}_N^\intercal]$, where $\mathbf{X} \in \mathbb{R}^{N \times T}$ represents N-variate time series. $\boldsymbol{x}_i \in \mathbb{R}^T$ represents the time series of the i-th variable, which consists of sequential recordings at T timestamps.

For an MTS forecasting task, we set an observation window H for historical time series and a forecasting window P for prediction. Accordingly, for each time step t, its historical series $\boldsymbol{H}_t = \mathbf{X}_{t-H+1:t}$ and forecasting series $\boldsymbol{P}_t = \mathbf{X}_{t+1:t+P}$ are defined as follows:

$$\boldsymbol{H}_t = [\boldsymbol{x}_{1,t-H+1:t}^\intercal; \boldsymbol{x}_{2,t-H+1:t}^\intercal; \cdots; \boldsymbol{x}_{N,t-H+1:t}^\intercal], \quad (1)$$

$$\boldsymbol{P}_t = [\boldsymbol{x}_{1,t+1:t+P}^\intercal; \boldsymbol{x}_{2,t+1:t+P}^\intercal; \dots; \boldsymbol{x}_{N,t+1:t+P}^\intercal]. \quad (2)$$

Specifically, to be considered as a long sequence MTS forecasting task, $H \ll P$.

Given a historical series $\boldsymbol{H}_t$, the goal is to learn a mapping function f that is capable of predicting the next P time steps $\hat{\boldsymbol{P}}_t = f(\boldsymbol{H}_t, \boldsymbol{\Theta})$ accurately, where $\boldsymbol{\Theta}$ is the learnable parameter set.

Discrete Wavelet Transform Module and Its Inverse Version

DWT module transforms an input MTS into its corresponding multi-scale frequency representations with DWT.

DWT is generally used to decompose input signals into a set of wavelets, which captures both the frequency and time features of the original signals and enables the following prediction modules to make predictions in parallel.

As depicted in Fig. 1, DWT can be performed multiple times, and each DWT uses a high-pass filter $\boldsymbol{h}$ and a low-pass filter $\boldsymbol{g}$ to decompose a time series signal $\boldsymbol{x}$ into different resolutions. The outputs of the high-pass and low-pass filters at layer l are denoted as ($\mathbf{cD}_l$) and ($\mathbf{cA}_l$), respectively: $\mathbf{cD}_l, \mathbf{cA}_l = DWT(\mathbf{cA}_{l-1})$, where l indicates the l-th decomposition and $\mathbf{cA_0} = \mathbf{x}$. Specifically, we have

$$\begin{aligned}\mathbf{cD}_l &= \boldsymbol{h} \star \mathbf{cA}_{l-1} \\ &= \sum_{m=1}^{M} \boldsymbol{h}[2s-m]\mathbf{cA}_{l-1}[m], s = 1, 2, \cdots, \frac{M}{2}, \end{aligned} \quad (3)$$

$$\begin{aligned}\mathbf{cA}_l &= \boldsymbol{g} \star \mathbf{cA}_{l-1} \\ &= \sum_{m=1}^{M} \boldsymbol{g}[2s-m]\mathbf{cA}_{l-1}[m], s = 1, 2, \cdots, \frac{M}{2}, \end{aligned} \quad (4)$$

where M represents the length of $\mathbf{cA}_{l-1}$ after decomposing $(l-1)$ times and s represents the scale. One feature of DWT is that after passing through $\boldsymbol{h}$ and $\boldsymbol{g}$ (namely $\boldsymbol{h}$ and $\boldsymbol{g}$ perform convolution operation ($\star$) with $\mathbf{cA}_{l-1}$, respectively), only half the number of samples characterizes the original signal $\mathbf{cA}_{l-1}$ owing to the double scale. Therefore, according to Nyquist's rule, we can remove half of the samples with downsampling while keeping the original information. Besides, the selection of $\boldsymbol{h}$ and $\boldsymbol{g}$ depends on the form of the wavelet basis. In theory, once the wavelet basis is determined[1], the form of $\boldsymbol{h}$ and $\boldsymbol{g}$ are determined as well. The detail coefficients $\mathbf{cD}$ depict the short-term trend of the series and carry the signal nuances, while the approximate coefficients $\mathbf{cA}$ describe the signal's long-term trend which characterizes its identity. In addition, the frequency resolution of the original signal increases as the decomposition goes deeper.

After l layers of decomposition, for each $\boldsymbol{x}_i$, the DWT module outputs a set of $l+1$ coefficients $\boldsymbol{p}^{(i)} = \{\mathbf{cD}_1^{(i)}, \mathbf{cD}_2^{(i)}, ..., \mathbf{cD}_l^{(i)}, \mathbf{cA}_l^{(i)}\}$. Different levels of DWT represent different resolutions of the original signal.

Let $\boldsymbol{\mathcal{C}} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_l, \mathbf{C}_{l+1}\}$ represent the layered wavelet coefficients where each layer contains each variable's corresponding coefficients, denoted as follows:

$$\mathbf{C}_j = [\mathbf{cD}_j^{(1)}; \mathbf{cD}_j^{(2)}; \dots; \mathbf{cD}_j^{(N)}] \in \mathbb{R}^{N \times H_j}, \ j \in [1, l] \quad (5)$$

$$\mathbf{C}_{l+1} = [\mathbf{cA}_l^{(1)}; \mathbf{cA}_l^{(2)}; \dots; \mathbf{cA}_l^{(N)}] \in \mathbb{R}^{N \times H_l}, \quad (6)$$

where $H_j = \frac{H}{2^j}$, H is the length of the input MTS, and N denotes the number of variables.

Note that after the following graph-enhanced modules output the i-th variable's coefficients for future P time steps, denoted as $\hat{\boldsymbol{p}}^{(i)} = \{\hat{\mathbf{cD}}_1^{(i)}, \hat{\mathbf{cD}}_2^{(i)}, \dots, \hat{\mathbf{cD}}_l^{(i)}, \hat{\mathbf{cA}}_l^{(i)}\}$, we apply the Inverse Discrete Wavelet Transform (IDWT) to reconstruct their corresponding sequence in the time domain. The process can be formulated as follows:

$$\begin{aligned}\hat{\mathbf{cA}}_{l-1} &= IDWT(\hat{\mathbf{cD}}_l, \hat{\mathbf{cA}}_l) \\ &= \boldsymbol{h}' \star \hat{\mathbf{cD}}_l + \boldsymbol{g}' \star \hat{\mathbf{cA}}_l \end{aligned} \quad (7)$$

[1]This paper utilizes the Haar wavelet (Pattanaik and Bouatouch 1995) for simplicity and $l = 3$

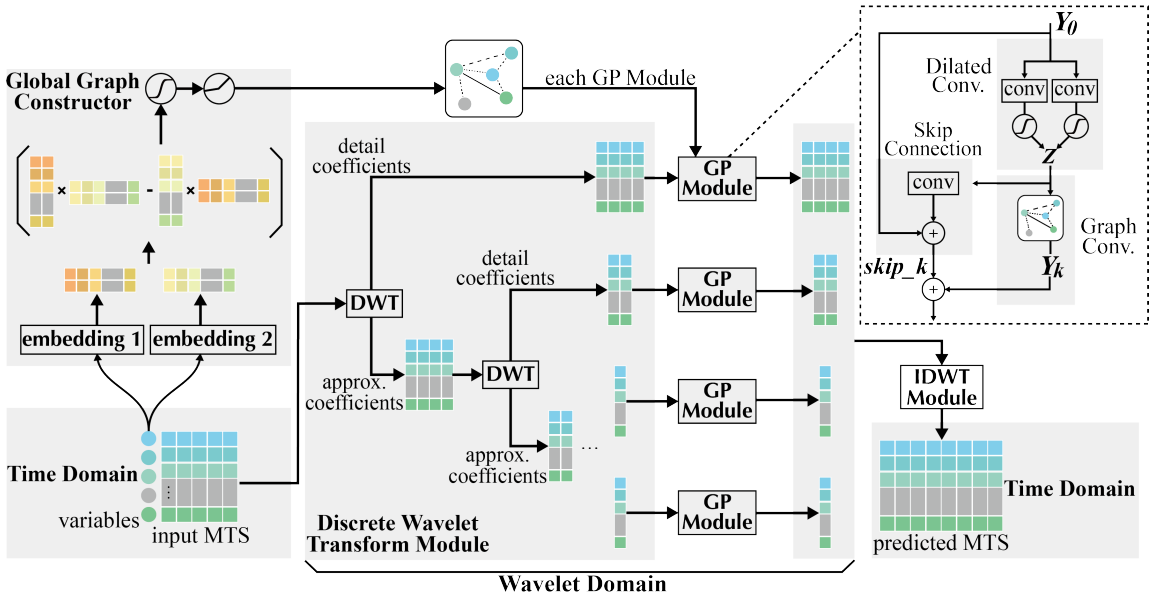

Figure 1: The WAVEFORM framework. The input MTS is decomposed into different coefficients in the wavelet domain and then fed into separate GP modules sharing the same global graph to make predictions. The learnable global graph is generated by two embedding layers and shared by all GP modules. The outputs of GP modules are reconstructed into time domain with the Inverse DWT (IDWT) module.

where $\boldsymbol{h}^{'}$ and $\boldsymbol{g}^{'}$ are the synthesis version of $\boldsymbol{h}$ and $\boldsymbol{g}$. When using the Haar wavelet (Pattanaik and Bouatouch 1995), $\boldsymbol{h}^{'} = -\boldsymbol{h}$ and $\boldsymbol{g}^{'} = \boldsymbol{g}$. Then, $\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \mathbf{c}\hat{\mathbf{A}}_0$ is the reconstructed time series of the i-th variable in the time domain.

Global Graph Constructor (GGC)

After obtaining the wavelet coefficients at different scales, the model intends to forecast the coefficient changes over time in the wavelet domain. Although the wavelet coefficients at different layers reflect the time series at different frequency subbands, we assume the variables share the same basic interaction structure at different resolutions without the loss of generality. Using a global graph rather than learning graphs in each GP module also avoids overfitting and saves memory. The GGC module learns a global graph to represent the relationships between variables.

For most real-world tasks, as we do not have the proper prior knowledge of what the graph looks like, we propose to utilize a graph constructor to learn the global graph, which in turn guides the graph-enhanced prediction modules for more sophisticated feature extraction. Following (Wu et al. 2020), we use two independent and learnable embedding layers, $\mathcal{E}_1$ and $\mathcal{E}_2$, to learn two embedding representations for each node after assigning each node/variable as an integer scalar. $\mathcal{N} = \{1, 2, \ldots, N\}$ denotes the index set of the nodes/variables, given as follows: $\boldsymbol{E}_1 = \mathcal{E}_1(\mathcal{N}) \in \mathbb{R}^{N \times d}$, $\boldsymbol{E}_2 = \mathcal{E}_2(\mathcal{N}) \in \mathbb{R}^{N \times d}$, where $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ denote variable representations obtained from two different layers. Then, the adjacency matrix $\boldsymbol{A}$ can be defined as follows:

$$\boldsymbol{A} = ReLU(tanh(\alpha(\boldsymbol{E}_1\boldsymbol{E}_2^{\intercal} - \boldsymbol{E}_2\boldsymbol{E}_1^{\intercal}))), \qquad (8)$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ and α is the hyper-parameter for the activation function. It is worth noting that Eq.(8) regularizes the adjacency matrix A to a uni-directional acyclic graph so that the influence between the nodes is uni-directional. This is more consistent with the hypothesis widely adopted in MTS analysis that the influences between variables are not mutual. To further reduce the computational cost of the following graph convolution, we can simply set up a threshold to filter out the linkages with weights smaller than the threshold to make the graph sparse.

Graph-Enhanced Prediction Modules

Given the learnable adjacency matrix of the variables, we build Graph-enhanced Prediction (GP) modules to methodically exploit the graphical information for predictions. A GP module consists of three main components: a) learning the multi-scale representation that incorporates the wavelet information via dilated convolution, b) aggregating neighborhood messages via graph convolution, and c) generating the final representations by combining skip connection layers.

Dilated Convolution Component Following MTGNN (Wu et al. 2020), we pass the input through stacked 1D dilated convolutions, which filter the wavelet coefficients to incorporate the wavelet information. In general, the standard convolution layer is ill-suited for dealing with long sequence forecasting since they require many layers or large filters to increase the receptive field, while both of them result in a substantial increase in model complexity. Alternatively, the dilated convolution (Yu and Koltun 2016), which is known to be stemmed from the wavelet decomposition, can capture long-term information/more complex patterns without sacrificing computational efficiency.

To better predict the changes of the time series signal in the wavelet domain, with an assumption that the wavelet coefficients contain latent patterns and are by no means the best raw signal for the following graph convolution component, we further utilize multiple dilated convolution filters with different kernel sizes to capture respective features for the wavelet coefficients at each level of resolutions. The output representations of these filters are activated by a sigmoid function and then concatenated to obtain the final representations of the stacked dilated convolution module. Given an input $\boldsymbol{z}$, and G filters $\boldsymbol{f}_1, \boldsymbol{f}_1, \cdots, \boldsymbol{f}_G$, the dilated convolution module has the following form:

$$\boldsymbol{z} = \text{concatenate}\big(\sigma(\boldsymbol{f}_1 \bar{\otimes} \boldsymbol{z}), \sigma(\boldsymbol{f}_2 \bar{\otimes} \boldsymbol{z}), \cdots, \sigma(\boldsymbol{f}_G \bar{\otimes} \boldsymbol{z})\big), \tag{9}$$

where $\bar{\otimes}$ denotes the dilated convolution operator.

Graph Convolution Component The purpose of the graph convolution module is to aggregate the node information with its neighbors' information to capture the global dependencies among different variables. It is widely known that vanilla GCNs are susceptible to over-smoothing issues due to the simplification of convolution as a neighborhood averaging operator, resulting in limited distinguishable representations of the nodes (Li, Han, and Wu 2018; Abu-El-Haija et al. 2019; Huang et al. 2020). To mitigate the issue, we utilize the MixHop layer proposed by (Abu-El-Haija et al. 2019; Wu et al. 2020) to capture complex relationships of neighbors at various hops instead of simply aggregating information from immediate neighbors. Concretely, the graph convolution includes two main steps: i) the message propagating step (Eq.(10)), and ii) the message aggregating step (Eq.(11)). These two steps recursively pass the local information to the nodes in the global graph structure. Given the adjacency matrix $\boldsymbol{A}$, the process of $\mathcal{K}$-layer MixHop can be formulated as follows:

$$\text{MixHop}(\mathbf{H}_{\text{in}}, \mathbf{A}) = \sum_{k=1}^{\mathcal{K}} \mathbf{H}_k \mathbf{W}_k, \tag{10}$$

$$\mathbf{H}_k = \beta(\mathbf{H}_{k-1} \cdot \tilde{\mathbf{A}}) + (1-\beta)\mathbf{H}_{\text{in}}, \tag{11}$$

where $\mathbf{H}_1 = \mathbf{H}_{\text{in}}$, $\mathbf{H}_{\text{in}}$ is the representations outputted from the previous layer, $\tilde{\mathbf{A}} = \mathbf{D}^{-1} \cdot (\mathbf{A} + \mathbf{I})$ is the normalized adjacency matrix, $\mathbf{D}_{ii} = 1 + \sum_j \mathbf{A}_{ij}$, and β is a hyperparameter that controls the proportion of information maintained from the previous representation, which helps to alleviate the over-smoothing problem. Following (Wu et al. 2020), we use two MixHop layers to obtain exhaustive information by processing inflow and outflow information passed through nodes separately.

Eventually, given the dilated convolution component's output $\mathbf{Z}$, the process of graph convolution component can be described as $\mathbf{H}_{\text{out}} = \text{MixHop}_1(\mathbf{Z}, \tilde{\mathbf{A}}) + \text{MixHop}_2(\mathbf{Z}, \tilde{\mathbf{A}}^\intercal)$.

Skip Connection and Output A naive combination of the dilated convolution component and the graph convolution component is shown to be prone to gradient vanishing issues. The proposed GP module uses skip-connections to improve its representational capability by preserving original information. Given the wavelet coefficients $\mathbf{C} \in \mathbb{R}^{N \times L}$, we first initialize two factors:

$$\mathbf{Y}_0 = \boldsymbol{W}_0 \bar{\otimes} \mathbf{C}, \tag{12}$$

$$\mathbf{Y}_{\text{skip}_0} = \boldsymbol{W}_{\text{skip}_0} \bar{\otimes} \mathbf{C}, \tag{13}$$

where $\boldsymbol{W}_0$ is a 1×1 convolution kernel for the convolution module in GP, and $\boldsymbol{W}_{\text{skip}_0}$ is a $1 \times L$ convolution kernel for a skip connection layer. Then we take the adjacency matrix $\boldsymbol{A}$ and these two factors as the input to pass through a K-layer stacked GP modules:

$$\mathbf{Y}_k, \mathbf{Y}_{\text{skip}_k} = \text{GP}_k(\mathbf{Y}_{k-1}, \mathbf{Y}_{\text{skip}_{k-1}}, \boldsymbol{A}), \text{for} \quad k \in \{1, \cdots, K\}. \tag{14}$$

In this process, the skip-output of the previous GP module, represented as $\mathbf{Y}_{\text{skip}_{k-1}}$, joins the output of the current dilated convolution module, represented as $\mathbf{Z}_k$, forming $\mathbf{Y}_{\text{skip}_k}$:

$$\mathbf{Y}_{\text{skip}_k} = \tau \mathbf{W}_{\text{skip}_k} \bar{\otimes} \mathbf{Z}_k + (1-\tau)\mathbf{Y}_{\text{skip}_{k-1}}, \tag{15}$$

where τ is a hyperparameter to control the balance. Similarly, the other output factor of the previous GP module, $\mathbf{Y}_{k-1}$, joints the skip-output of the current GP module, $\mathbf{Y}_{\text{skip}_k}$, to form $\mathbf{Y}_k$:

$$\mathbf{Y}_k = \tau \mathbf{W}_k \bar{\otimes} \mathbf{Y}_{\text{skip}_k} + (1-\tau)\mathbf{Y}_{k-1}. \tag{16}$$

After passing through all K-layer stacked GP modules, we can obtain the final output representation as the prediction of the wavelet coefficients. It is worth noting that in this process, wavelet coefficients at different scales are predicted separately by different GP modules while sharing the same global graph adjacency matrix.

Experiments

This section reports the experiments on WAVEFORM and state-of-the-art (SOTA) baselines with five public datasets.

Datasets and Settings

We applied widely used datasets in the experiments: Electricity (Wu et al. 2021), Traffic (Lai et al. 2018), Weather (Wu et al. 2021), and Solar-Energy (Lai et al. 2018). Each dataset was split in chronological order with 70% for training, 20% for validation, and 10% for testing. Following (Wu et al. 2019, 2020), we set the length of input sequence (I) as 96 to predict the next 96, 192, 336, and 720 future steps (O), and utilized mean absolute error (MAE) and mean squared error (MSE) to assess the long sequence forecasting performance of WAVEFORM and baselines. More detailed descriptions of the datasets, evaluation metrics, and experimental settings are provided in the Appendix. Code is available at https://github.com/alanyoungCN/WaveForM.

Comparison Models

We compared WAVEFORM with the general sequence modeling approaches, including LSTM (Hochreiter and Schmidhuber 1997) and Transformer (Vaswani et al. 2017), and SOTA MTS forecasting models, including Graph WaveNet (Wu et al. 2019), Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), and MTGNN (Wu et al. 2020). The details of the baseline models can be found in Introduction & Related Work.

Models	Metrics	Electricity(I=96)				Solar-Energy(I=96)				Weather(I=96)				Traffic(I=96)			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
LSTM	MSE	0.457	0.466	0.485	0.689	0.220	0.221	0.539	0.690	0.356	0.400	0.350	0.512	0.873	0.920	1.150	1.486
	MAE	0.485	0.493	0.493	0.634	0.256	0.258	0.546	0.709	0.396	0.436	0.386	0.479	0.476	0.500	0.632	0.796
Transformer	MSE	0.262	0.264	0.278	0.311	0.194	0.229	0.217	0.211	0.345	0.546	0.655	0.927	0.638	0.643	0.677	0.704
	MAE	0.359	0.366	0.374	0.393	0.208	0.247	0.251	0.285	0.396	0.504	0.577	0.706	0.350	0.351	0.365	0.381
Graph WaveNet	MSE	0.329	0.333	0.358	0.399	0.224	0.257	0.265	0.264	0.190	0.229	0.287	0.371	0.984	0.973	0.984	1.016
	MAE	0.409	0.416	0.436	0.464	0.290	0.320	0.324	0.319	0.246	0.284	0.348	0.410	0.543	0.540	0.542	0.554
MTGNN	MSE	0.208	0.217	0.221	0.308	0.179	0.195	0.206	0.238	0.174	0.216	0.284	0.375	0.754	0.752	0.760	0.788
	MAE	0.308	0.315	0.319	0.385	0.231	0.263	0.264	0.291	0.243	0.285	0.339	0.401	0.436	0.434	0.435	0.450
Informer	MSE	0.274	0.296	0.300	0.373	0.205	0.227	0.252	0.246	0.300	0.598	0.578	1.059	0.719	0.696	0.777	0.864
	MAE	0.368	0.386	0.394	0.439	0.228	0.242	0.265	0.302	0.384	0.544	0.523	0.741	0.391	0.379	0.420	0.472
Autoformer	MSE	0.199	0.216	0.231	0.295	0.540	0.571	0.778	0.889	0.261	0.316	0.357	0.416	0.604	0.614	0.643	0.668
	MAE	0.315	0.328	0.339	0.382	0.521	0.570	0.665	0.709	0.336	0.374	0.389	0.427	0.373	0.388	0.403	0.411
WaveForM	MSE	**0.153**	**0.172**	**0.197**	**0.242**	**0.165**	**0.189**	**0.195**	**0.204**	**0.160**	**0.209**	**0.278**	**0.364**	**0.567**	**0.565**	**0.581**	**0.623**
	MAE	**0.257**	**0.272**	**0.294**	**0.332**	**0.203**	**0.233**	**0.239**	**0.265**	**0.227**	**0.280**	**0.332**	**0.399**	**0.326**	**0.318**	**0.323**	**0.346**

Table 1: A comparison of baselines and our model on different datasets and prediction lengths. We used 96, 192, 336, and 720 for prediction lengths, and used 96 for the input length for all cases. We repeated every case three times and used the average as the final result. Lower MSE and MAE mean higher prediction accuracy. Bold texts indicate the best results.

Main Results

Table 1 shows the experimental results. For each method, we repeated three runs with different seeds and reported the averaged results. Our model consistently outperforms SOTA models on both MSE and MAE metrics (the lower, the better) for all datasets, whereas none of the existing models can consistently serve as the second-best model for all datasets. For each dataset, our model can roughly achieve 15-20% performance improvement over different prediction lengths against the most competitive baseline. We attribute such a significant improvement to the use of multi-level signals in the wavelet domain and the global graph.

In general, for the datasets with relatively small numbers of nodes/variables, such as Solar-Energy and Weather, the graph-based MST models, including Graph WaveNet, MTGNN, and our WaveForM, perform better than the transformer-based MST models, indicating the capability of graph-based modeling and GNNs in capturing the interdependence among variables. We also observe that for the datasets with a large number of nodes/variables, such as Traffic, graph-based models except WaveForM achieve inferior performance than other models.

Models	Metrics	Temperature(I=336)	
		720	1260
Autoformer	MSE	0.340 ± 0.013	0.984 ± 0.162
	MAE	0.457 ± 0.004	0.774 ± 0.071
MTGNN	MSE	1.005 ± 0.007	1.009 ± 0.005
	MAE	0.843 ± 0.006	0.845 ± 0.008
WaveForM	MSE	$\mathbf{0.307 \pm 0.002}$	$\mathbf{0.346 \pm 0.003}$
	MAE	$\mathbf{0.433 \pm 0.004}$	$\mathbf{0.461 \pm 0.005}$

Table 2: Performance on Temperature Dataset

Note that Traffic has the least number of records but the largest number of nodes, making it the most challenging task. We believe the existing graph-based approaches are somehow underfitting in the Traffic dataset for modeling such a large graph. Whereas WaveForM can discover more features from signal/data with a more sophisticated design in wavelet domain, thus enabling better training of the end-to-end model to capture the complex interdependence among a large number of variables.

Besides, our global graph modeling can be considered as further "fine-tuning" the interdependence of variables in multiple levels using multiple GP modules, thus leading to better performance. Although Autoformer also utilizes frequency domain features, as we have argued, its inferior performance to ours may be due to the improper mixed use of features from different domains.

We further experimented the Temperature dataset (Grigsby, Wang, and Qi 2021) to assess model performance for extra longer sequence forecasting, of which Table 2 presents the results. Only the two most competitive models from the previous experiments, Autoformer and MTGNN, are included for comparison. We can observe that Autoformer's performance decreases sharply when the prediction steps are extended from 720 to 1260, while MTGNN shows rather inferior performance under such a setting. We believe that as the Temperature dataset has an extremely small number (only six) of nodes but the largest number of records in all datasets, the graph used in MTGNN might overfit the interdependence among MTS. In comparison, WaveForM experiences only minor degradation when processing extra long sequence forecasting and outperforms other models with 300% performance improvement.

Ablation Study

We conducted an ablation study on the Electricity dataset to assess the effectiveness of different modules in WaveForM . The variants of WaveForM include:

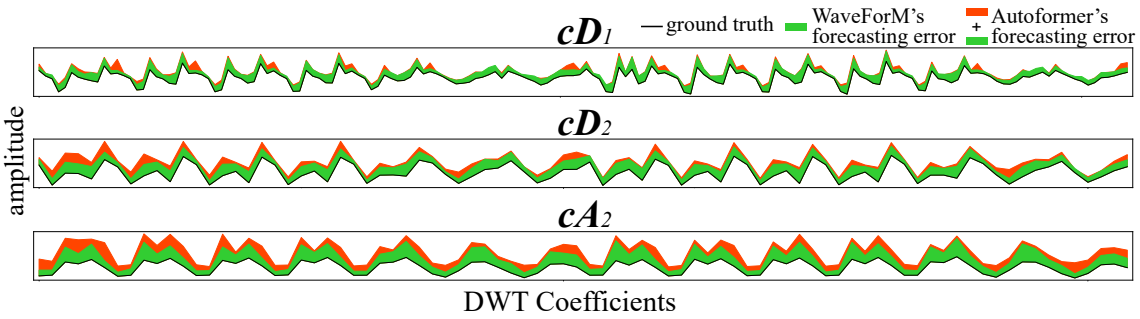

Figure 2: The comparison in the wavelet domain. With the Electricity dataset, the trained Autoformer and WAVEFORM were used for predictions, respectively. The prediction results were transformed by DWT, and the error against the ground truth was calculated separately. The green areas represent the errors of WAVEFORM , while the red areas represent the additional errors of Autoformer on top of WAVEFORM . The coefficients from the top to bottom reflect different frequency resolutions, and the lower means more precise.

- WAVEFORM w/o GGC: Remove the Global Graph Constructor module from WAVEFORM , and apply a separate Graph Constructor in each layer of the GP Module.
- WAVEFORM w/ single GP: After passing the last high-pass and low-pass filters, concatenate the wavelet coefficients into one single sequence in the same order of decomposition, and then use only one GP Module to make predictions. GP's output is then split manually into wavelet coefficients of different levels for IDWT.
- WAVEFORM w/o GP: Remove the Graph-enhanced Prediction (GP) module from WAVEFORM and use multiple affine transforms as a substitute.

Table 3 reports the experiment results. In comparison with the results in Table 1, WAVEFORM and all its variants outperform all other comparison models, proving the effectiveness of the skillful use of wavelet domain features.

Table 3 also shows that the GGC module plays an important role in effectively providing global information among different wavelet coefficients, which remarkably improves the prediction performance. In addition, the inferior performance of WAVEFORM w/ single GP demonstrates that the coefficients for different scales obtained with DWT benefit from being processed separately, so as to guarantee the validity of the use of IDWT.

Models	Metrics	Electricity(I=96) 96	192	336	720
WAVEFORM w/o GGC	MSE	0.179	0.197	0.216	0.259
	MAE	0.274	0.290	0.310	0.350
WAVEFORM w/ single GP	MSE	0.159	0.182	0.204	0.254
	MAE	0.265	0.283	0.299	0.337
WAVEFORM w/o GP	MSE	0.161	0.177	0.211	0.249
	MAE	0.268	0.281	0.308	0.342
WAVEFORM	MSE	**0.153**	**0.172**	**0.197**	**0.242**
	MAE	**0.257**	**0.272**	**0.294**	**0.332**

Table 3: Ablation Study on Electricity Dataset

Wavelet-Domain Observations

This section utilizes DWT to explain the performance from the perspective of wavelet domain. We used a 2-layer DWT to transform the input to wavelet coefficients cD_1, cD_2, and cA_2, which represent the correlation between the input and the wavelet function over time. Specifically, we transformed the ground truth, the predictions of Autoformer, and the predictions of WAVEFORM on the Electricity dataset to the wavelet domain coefficients for comparison.

Fig. 2 shows that the prediction errors of Autoformer (red and green areas) and WAVEFORM (green areas) against the ground truth gradually become larger as the wavelet domain features are gradually refined with more layers (from top to bottom), which means that it is more difficult for Autoformer to discover fine-grained, low-frequency features (conveyed by deeper cD and cA). Meanwhile, the performance gap (red areas) between Autoformer and WAVEFORM also becomes larger as the decomposition goes deeper, which indicates that WAVEFORM is more capable of uncovering complex patterns of MTS.

Conclusion

We proposed WAVEFORM , a novel framework for long sequence multivariate time series forecasting. WAVEFORM uses DWT to transform the time-domain series into wavelet-domain coefficients at multiple resolutions and then applies a graph convolution module to model the relationships between multivariates. Experiments show that the transformed coefficients in the wavelet domain are more capable of describing the input series from multiple resolutions, thus allowing the model to learn fine-grained complex patterns. Experiments on widely used benchmark datasets show that our model significantly outperforms the SOTA models with remarkable margins for long sequence forecasting of MTS.

Acknowledgments

This work has been partially supported by NSFC under Grant No. 62276024, No. 92270125 and No. U19B2020.

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Ver Steeg, G.; and Galstyan, A. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, 21–29. PMLR.

Box, G. E.; and Jenkins, G. M. 1970. Time Series Analysis Forecasting and Control. Technical report, WISCONSIN UNIV MADISON DEPT OF STATISTICS.

Grigsby, J.; Wang, Z.; and Qi, Y. 2021. Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Huang, W.; Rong, Y.; Xu, T.; Sun, F.; and Huang, J. 2020. Tackling Over-Smoothing for General Graph Convolutional Networks. *CoRR*, abs/2008.09864.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.

Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32.

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*.

Pattanaik, S. N.; and Bouatouch, K. 1995. Haar wavelet: A solution to global illumination with general surface properties. In *Photorealistic Rendering Techniques*, 281–294. Springer.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191.

Shang, C.; Chen, J.; and Bi, J. 2021. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*.

Shensa, M. J.; et al. 1992. The discrete wavelet transform: wedding the a trous and Mallat algorithms. *IEEE Transactions on signal processing*, 40(10): 2464–2482.

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.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

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. *arXiv preprint arXiv:1906.00121*.

Yu, B.; Yin, H.; and Zhu, Z. 2017. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *CoRR*, abs/1709.04875.

Yu, F.; and Koltun, V. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33: 17283–17297.

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, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*.