WaveletMixer: A Multi-Resolution Wavelets Based MLP-Mixer for Multivariate Long-Term Time Series Forecasting

Zichi Zhang*, Tuan Dung Pham*, Yimeng An*, Ngoc Phu Doan*, Majed Alsharari, Viet-Hung Tran, Anh-Tuan Hoang, Hans Vandierendonck, Son T. Mai

Queen's University Belfast, UK
{zzhang54, h.vandierendonck, thaison.mai}@qub.ac.uk

Abstract

Time Series Forecasting (TSF) aims at predicting future values for a time series data and plays a crucial role in many real-world applications, e.g., finance, disease spread, or weather predictions. However, it is also a very challenging task due to complex temporal dependencies in the data, especially for long-term forecasting. In this paper, we introduce WaveletMixer, an iterative multi-levels, multi-resolutions and multi-phases approach to effectively capture long-term dependencies of multivariate time series in both global and local perspectives for improving forecasting performance. WaveletMixer fundamentally differs from existing works in the following key aspects. First, it exploits multi-levels properties of Wavelet transformation to create multiple forecasting models for different frequency domains at various levels of resolutions. Second, the relationships among different frequency domains are exploited to iteratively adjust all prediction models at all levels simultaneously in both local and global perspectives to reduce prediction errors and biases, thus significantly improving the final accuracy. Third, while WaveletMixer is a general framework that can be used to boost the performance of any deep-learning architecture (e.g., MLP, LSTM or Transformer), we additionally introduce TS-Learner, an MLP-based model to further enhance the performance in long-term forecasting. Extensive experiments have been conducted on nine real-world datasets to demonstrate the outstanding performance of WaveletMixer compared to SOTA methods and to reveal its important characteristics.

1 Introduction

Time series forecasting (TSF), which aims at predicting future values using observed data in a time series, is a fundamental research problem in Machine Learning. It has been widely used in many real-world applications, e.g., weather prediction (Jain and Mallick 2016), disease spread modelling (Datilo, Ismail, and Dare 2019; Rodríguez et al. 2021), financial forecasting (Sezer, Gudelek, and Ozbayoglu 2020), or streamflow prediction (Li, Xu, and Anastasiu 2024). However, the intrinsic temporal variations and dependencies among different time points (e.g., increasing, decreasing, fluctuations) make TSF a very challenging task, especially when predicting many future values (i.e. long-term forecasting (LTSF)). TSF, particularly LTSF, remains an open research despite tremendous research efforts, e.g., (Box et al. 2015; Zhou et al. 2021; Wang et al. 2024b).

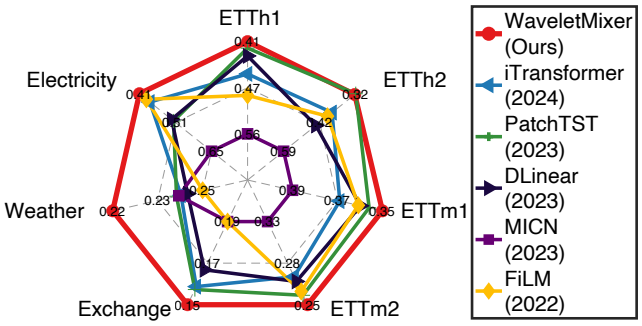

Figure 1: Model performance comparisons using MSE as a metric for different datasets (the more outward, the better).

With rapid developments of computer performance and AI in recent years, deep learning models, built upon diverse backbones like multilayer perceptron (MLP) (Chen et al. 2023; Zeng et al. 2023; Li et al. 2023), recurrent neural network (RNN) (Lai et al. 2018), convolution neural network (CNN) (Koprinska, Wu, and Wang 2018; Wu et al. 2023; Wang et al. 2023) and Transformer (Vaswani et al. 2017; Zhou et al. 2021; Nie et al. 2023; Liu et al. 2023), have shown great potential in capturing the periodicity, trends and long-sequence dependencies of time series for LTFS. Many of them, e.g., DLinear (Zeng et al. 2023), RLinear (Li et al. 2023), TimesNet (Wu et al. 2023), MICN (Wang et al. 2023), Informer (Zhou et al. 2021), PatchTST (Nie et al. 2023) and iTransformer (Liu et al. 2023), achieve state-of-the-art results on various real-world benchmarks. Most of these techniques focus on disentangling time series in the temporal dimension to capture the intrinsic periodic information for improving performance, e.g., decomposing time series into trend and seasonal components (Wu et al. 2021; Zeng et al. 2023; Zhou et al. 2022b; Wang et al. 2023) or breaking data into chunks with different period lengths (Wu et al. 2023; Zhou et al. 2022a). A few techniques, e.g. MICN (Wang et al. 2023) and Pathformer (Chen et al. 2024), aggregate time series into multi-scale to capture different

*These authors contributed equally.

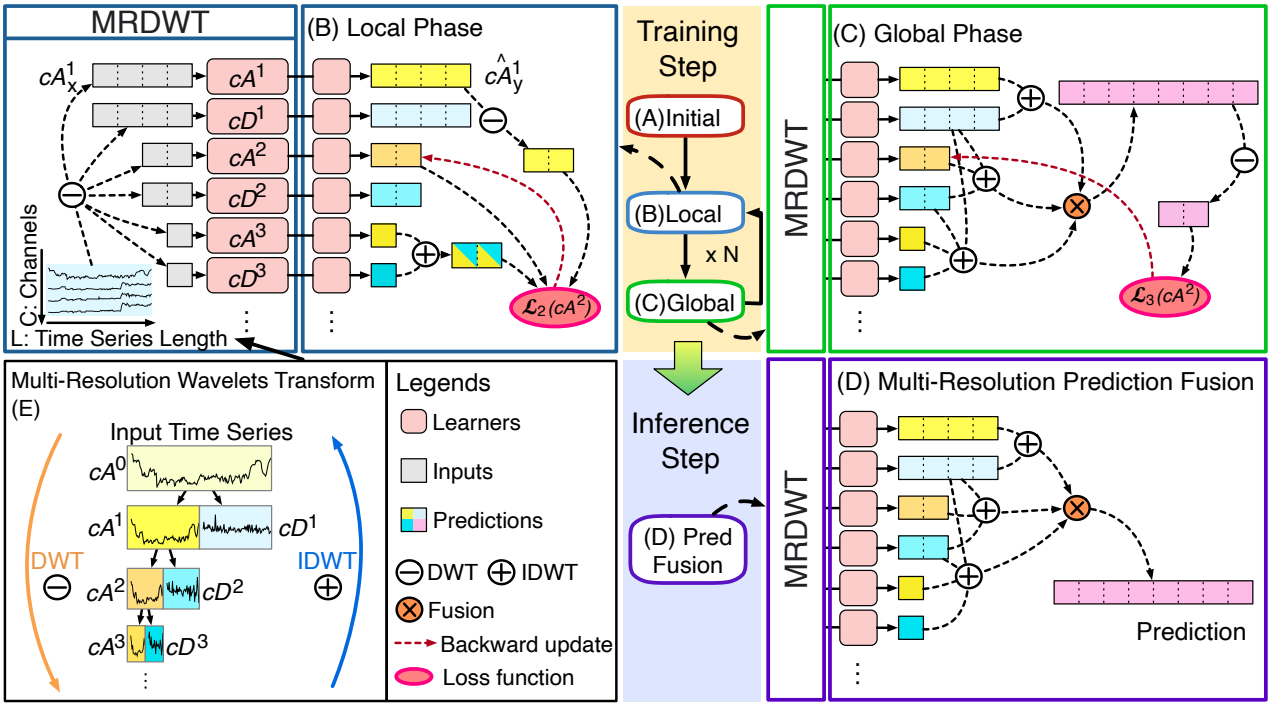

Figure 2: The overall process of WaveletMixer for both Training and Inference steps. Prior to the Training step, the input X and output Y are decomposed into different resolution levels (E). The training process is divided into 3 different phases. In the Initial phase (A), for each level ℓ, we build an independent model f_{cA^ℓ} to learn $\hat{cA}_y^\ell$ from cA_x^ℓ (i.e. $\hat{cA}_y^\ell = f_{cA_i}(cA_x^\ell)$). Similarly, we build f_{cD^ℓ} to learn $\hat{cD}_y^\ell$ from cD_x^ℓ. In the Local phase (B), all models f_{cA^ℓ} and f_{cD^ℓ} will be updated to maximize their agreements via the local error propagation scheme. E.g., f_{cA^2} can be updated by obtaining $IDWT(\hat{cA}_y^3, \hat{cD}_y^3)$ from f_{cA^3} and $DWT(\hat{cA}_y^1)$ from f_{cA^1} to calculate the loss function $\mathcal{L}_2(cA^2)$ stated in Eq. 10 wrt. the ground truth cA_y^2 and predicted value $\hat{cA}_y^2$. In the Global phase (C), we first reconstruct the final value $\hat{Y}$ using by combining prediction outputs of all f_{cA^ℓ} and f_{cD^ℓ} in a weighted ensemble (fusion) scheme with learnable weights. E.g., by doing IDWT with $(\hat{cA}_y^1, \hat{cD}_y^1)$, $(\hat{cA}_y^2, \hat{cD}_y^2, \hat{cD}_y^1)$, and $(\hat{cA}_y^3, \hat{cD}_y^3, \hat{cD}_y^2, \hat{cD}_y^1)$, we obtain 3 different prediction values $\hat{Y}^1$, $\hat{Y}^2$, and $\hat{Y}^3$ to form $\hat{Y}$ via Eq. 12 . After that, we decompose $\hat{Y}$ again into $cA_{\hat{y}}^2$ and $cD_{\hat{y}}^2$ to update f_{cA^2} in the global error propagation scheme via the loss function $\mathcal{L}_3(cA^2)$ stated in Eq. 13. The Local and Global phases are iteratively performed until the training process converge. In the Inference step, we perform Prediction Fusion (D) like in the Global phase for obtaining $\hat{Y}$ as the final prediction outcome. Here, we only demonstrate the error propagations for a specific level cA^2 as an example in the local and global phases. Best viewed in colors.

characteristics spanning accross different scales. However, they ignore frequency domains, which can provide important frequency features for the analysis of time series. Other methods, e.g. FEDformer (Zhou et al. 2022b) and FiLM (Zhou et al. 2022a), aim to use Fourier transformation for effectively forecasting long-term values by exploiting sparse representation of time series in frequency domains. However, these models still lack of global summarization ability, which can be particularly useful when forecasting long-term values. Moreover, most of these current approaches build a single prediction model for LTSF, thus limiting diverse perspectives on the prediction outcomes.

Contributions. In this paper, we propose WaveletMixer, an iterative multi-levels, multi-resolutions and multi-phases ensemble approach for LTSF. Compared to above mentioned works, WaveletMixer fundamentally differs as following.

First, WaveletMixer exploits the Multi-Resolution Discrete Wavelet Transform (MRDWT) (Harti 1993) to decompose input time series into a hierarchy of multiple levels of low-frequency and high-frequency components. Compared to Fourier transform, Wavelet can represent time series in both the time and frequency domains. Moreover, its multi-resolution property allows us to effectively capture irregular change patterns in the data (Stanković and Falkowski 2003). Hence, MRDWT provides deep characteristics of the input at both local and global perspectives, which can be exploited to enhance the forecasting accuracy as described below.

Second, rather than building a single model to predict future values using a chosen set of input features as above existing works, we construct multiple independent models, in which each model takes one frequency components of the

input to predict the same frequency components of the output at the same level of resolutions (c.f. Figure 2 for an illustration). The final prediction output will be constructed from the hierarchy of frequency component outputs via inverted Wavelet transform (Stanković and Falkowski 2003) via an ensemble (fusion) scheme with learnable weights. This approach allows each model to fully focus on learning a single component representing a characteristic of data at a specific level of resolution without being distracted by too many inputs from other components, thus being more effective.

Third, though all components are learned independently, their intrinsic relationships should be exploited to further boost the performance. Here we introduce a *unique* multi-phases approach to *iteratively* update all models from their initial stages via two proposed *local* and *global error propagation* schema. The key idea is to let each model adjusts each other locally via their decomposition/reconstruction relationships. While in the global phase, the final reconstructed signal is used to adjust each model via a global decomposition. By this way, we maximize the agreements from all models in both local and global perspectives. This helps to reduce the prediction errors and biases, and improve the overall accuracy, which is studied in details in Table 3.

Fourth, while most existing works rely on a fixed backbone like MLP, CNN or Transformer, WaveletMixer is a *general framework* that can be used with *any* existing backbones to boost their performance. Here, we additionally introduce TS-Learner, a special MLP-based module that is specifically designed for accurately capturing the features from channel and temporal dimensions by combining advanced components including Patching, Channel Mixing and Temporal Mixing, to use with our WaveletMixer framework (c.f. Full results table in the Supplementary Materials for comparative studies among TS-Learner, Linear MLP and LSTM wrt. WaveletMixer).

Summarization. WaveletMixer is a unique approach build upon the Wavelet transformation for LSTF. It provides a *unified* perspective of multi-resolution, temporal and frequency-domain analysis, and ensemble forecasting for enhancing prediction accuracy and can be used with any deep learning backbones. Extensive experiments are conducted over nine real-world benchmarks with different prediction length settings to demonstrate that WaveletMixer significantly outperforms SOTA models (c.f. Figure 1 for performance summary) and to reveal its characteristics.

2 Background

Discrete Wavelet Transform (DWT) (Sundararajan 2016) is a wavelet transformation where the wavelets are sampled discretely. The DWT of a discrete signal sequence $x[n]$, where n is an integer, is computed using a pair of filters: a low-pass filter $g[n]$ and a high-pass filter $h[n]$. The transformation can be formulated as:

$$cA[n] = \sum_{k=-\infty}^{\infty} x[2n-k]g[k] \tag{1}$$

$$cD[n] = \sum_{k=-\infty}^{\infty} x[2n-k]h[k] \tag{2}$$

or simplified

$$cA[n], cD[n] = DWT(x[n]) \tag{3}$$

In this paper, we utilize *Haar Wavelets* (Stanković and Falkowski 2003) as mother wavelets, where $g[n] = [\sqrt{2}/2, \sqrt{2}/2]$ and $h[n] = [-\sqrt{2}/2, \sqrt{2}/2]$.

Inverse Discrete Wavelet Transform (IDWT) upsampling the coefficients and applying the inverse filters reconstructs the signal from its approximation and detail coefficients.

$$x[n] = IDWT(cA[n], cD[n]) \tag{4}$$

Multi-Resolution Discrete Wavelet Transform (MRDWT) refers to apply multiple DWTs to the multi-levels approximation coefficients of the same original signal:

$$cA^{i+1}[n], cD^{i+1}[n] = DWT(cA^{i}[n]) \tag{5}$$

where $i \in \mathbb{Z}$ is the resolution level, and cA^i and cD^i denote approximation (lower frequency) and detail (higher frequency) coefficients at level i, respectively ($cA^0[n] = x[n]$ at level 0). Similarly, the signal at the previous level that can be recovered using IDWT from decomposed coefficients:

$$cA^{i}[n] = IDWT(cA^{i+1}[n], cD^{i+1}[n]) \tag{6}$$

3 Our Proposed Algorithm WaveletMixer

Given the time series $X = [x_0, x_1, \ldots, x_{L-1}] \in \mathbb{R}^{C \times L}$ with look-back window length L and C channels, where $x_t \in \mathbb{R}^C$ denotes the multivariate values of C dimension channels series with timestamp t. Also, let us introduce some common notions including the maximum wavelets decomposition levels l_w, the sub-sequence approximation coefficients (lower-frequency component) $cA_x^\ell \in \mathbb{R}^{C \times \lceil L/2^\ell \rceil}$ and the detail coefficients (higher-frequency component) $cD_x^\ell \in \mathbb{R}^{C \times \lceil L/2^\ell \rceil}$ of ℓ levels DWT decomposition of X.

Ground truth of future time series is denoted by $Y = [y_0, y_1, \ldots, y_{H-1}] \in \mathbb{R}^{C \times H}$ where H is the length of the prediction horizon. The decomposition components by DWT of Y at level ℓ are denoted as cA_y^ℓ and $cD_y^\ell \in \mathbb{R}^{C \times \lceil H/2^\ell \rceil}$ (or simply cZ_y^ℓ when it is clear as cA_y^ℓ or cD_y^ℓ).

Our target is to build an effective model f to predict the future Y based on the historical X, i.e. $\hat{Y} = f(X)$ where $\hat{Y} = [\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_{H-1}] \in \mathbb{R}^{C \times H}$ is the prediction of future time series. The reconstructed prediction of the future time series from level ℓ after IDWT are denoted by $\hat{Y}^\ell \in \mathbb{R}^{C \times H}$, where $\hat{Y}^\ell$ and Y have the same length.

Definition 1. *(Prediction MRDWT). Given the predicted approximate coefficients $\hat{cA}_y^i$ and detail coefficients $\hat{cD}_y^i$ sequences at level i, let $\hat{cA}_y^{i \to \ell}$ and $\hat{cD}_y^{i \to \ell}$ be the coefficients of the sequences $\hat{cA}_y^i$ and $\hat{cD}_y^i$ after $m \in \mathbb{Z}_+$ times transformation (DWT^m or $IDWT^m$) from level i to the target level ℓ, respectively. We have:*

$$\hat{cA}_y^{i \to \ell} = \begin{cases} DWT^{\ell-i}(\hat{cA}_y^i) & \text{if } i < \ell \\ IDWT^{i-\ell}(\hat{cA}_y^i, \hat{cD}_y^i, \ldots, \hat{cD}_y^{\ell-1}) & \text{if } i > \ell \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\hat{cD}_y^{i \to \ell} = \begin{cases} DWT^{\ell-i}(\hat{cA}_y^i) & \text{if } i < \ell \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

3.1 Multi-Resolution Wavelets LTSF Model

As mentioned in the Introduction section, the main key ideas of WaveletMixer are: (i) exploiting DWT to represent time series at multiple level of resolutions that capture both temporal and frequency aspects of the data for information richness; (ii) constructing multiple models to predict each wavelet component at each level independently for diversity; (iii) exploiting intrinsic relationships among different Wavelet levels to adjust the prediction outcomes for each model locally and globally; and (iv) providing final prediction outcome by fusing multiple reconstructed outputs from all models to reduce biases and prediction errors.

Figure 2 illustrates the overall pipeline of WaveletMixer for both Training and Inference steps. Before the Training step, we apply MRDWT decomposition to obtain multiple sub-sequence approximate and detail coefficients cA_x^ℓ and cD_x^ℓ from the input time series X and cA_y^ℓ and cD_y^ℓ from the output time series Y at each level ℓ where $\ell \leq l_w$. Thus, we have a hierarchy structure capturing relationships of of different Wavelet components as illustrated in Figure 2 (E). The Training step starts with an Initial phase (Figure 2 (A)) to train multiple prediction models for each Wavelet component independently before iteratively updating them in the Local and Global phases to reduce errors and biases. The Local phase (Figure 2 (B)) updates models so that their prediction results are consistent across different levels of resolutions via their local decomposition/reconstruction relationships. Hence, any component hard to predict can be adjusted via other components at higher or lower levels to enhance accuracy. The Global phase (Figure 2 (C)) creates final prediction output $\hat{Y}$ by reconstructing signals from all levels and combine them in a weighted ensemble (fusion) scheme with learnable weights. This $\hat{Y}$ contains global errors from the whole output and will be decomposed into Wavelet components to be used to adjust existing models at all levels to ensure better final prediction accuracy. The Inference step (Figure 2 (D)) works exactly similar to the Global phase without the decomposition part. The detailed descriptions for these phases can be found below while their pseudocodes can be found in the Supplementary Materials. Unless otherwise specified, we use Mean Square Error (MSE) to calculate the loss functions.

Phase 1: Initial Training Phase (ITP). In this phase, for each level ℓ, we build an independent model f_{cA^ℓ} to learn $\hat{cA}_y^\ell$ from cA_x^ℓ (i.e. $\hat{cA}_y^\ell = f_{cA^\ell}(cA_x^\ell)$) and a model f_{cD^ℓ} to learn $\hat{cD}_y^\ell$ from cD_x^ℓ (i.e. $\hat{cD}_y^\ell = f_{cD^\ell}(cD_x^\ell)$). Here, any backbone models (e.g., MLP or LSTM) can be employed for f, thus making our approach a very generic framework for boosting performance of any existing models. This phase ensures that each learner has good initialization weights for effective error propagations in the next phases. Since each model only learns the feature information of the corresponding resolution, there is no interaction between different levels. So, the loss function $\mathcal{L}_1$ of each learner will be:

$$\mathcal{L}_1(cZ_y^\ell) = MSE(\hat{cZ}_y^\ell, cZ_y^\ell) \tag{9}$$

Phase 2: Local Error Propagation Phase (LEPP). Learners f_{cY^ℓ} can learn time series features at each resolution level but can not ensemble information together to correct prediction errors si nce there is no direct interaction among them. Hence, LEPP performs DWT/IDWT to each level to decompose/reconstruct the learners' predictions to the same length as the target level's and to force learners to make consistent predictions across different levels. The multi-resolution information between each level propagates by constructing separate loss functions. Specifically, we treat two kind of different learners: *input learners* and a *target learner*, where input learners provide the prediction into DWT or IDWT to obtain the correction series $\hat{cZ}_y^{l\to\ell}$ as same length as the prediction $\hat{cZ}_y^{\ell}$ of the target learner at level ℓ which we want to correct the errors and the superscript $l \to \ell$ denotes from level l transform to level ℓ. If the target level $\ell > l$, using DWT to decompose the prediction of input learners to level ℓ, otherwise using IDWT. Then we calculate the MSE of these predictions applied to the target loss function as the following (c.f. Figure 2 as an example).

Definition 2. *(Local Error Propagation Loss). Given the predicted approximate coefficients $\hat{cA}_y^i$ and detail coefficients $\hat{cD}_y^i$ sequences at level i, we define loss functions as:*

$$\mathcal{L}_2(cA_y^\ell) = \alpha\mathcal{L}_1(cA_y^\ell) + (1-\alpha)MSE(\hat{cA}_y^\ell, \tfrac{1}{l_w-1}\sum_{i=0}^{l_w-1}\hat{cA}_y^{i\to\ell}) \tag{10}$$

$$\mathcal{L}_2(cD_y^\ell) = \alpha\mathcal{L}_1(cD_y^\ell) + (1-\alpha)MSE(\hat{cD}_y^\ell, \tfrac{1}{l_w-\ell}\sum_{i=0}^{l_w-\ell}\hat{cD}_y^{i\to\ell}) \tag{11}$$

Phase 3: Global Error Propagation Phase (GEPP). LEPP provides good guarantees for consistency between predictions generated at each level, however, our final objective is the prediction of future time series instead of the prediction of the decomposition components, and this deviation can cause the model to lose accuracy. GEPP is designed to solve this problem by utilizing IDWT to reconstruct the time series $\hat{Y}^\ell$ as same length as the future time series Y at level ℓ from $\hat{cA}_y^\ell$ and $\hat{cD}_y^s$ where $1 \leq s \leq \ell$. All these reconstructed output $\hat{Y}^\ell$ will be ensembled (fused) into the final prediction values $\hat{Y}$. Since some levels ℓ may be more accurate than the others depending on the data nature, we propose to combine them using a weighted scheme where the weights are learnable via a Linear Layer. We call it a multi-resolution prediction fusion scheme.

Definition 3. *(Multi-Resolution Prediction Fusion). Given multiple prediction sequences $\{\hat{Y}^0, \hat{Y}^1, \cdots, \hat{Y}^{l_w}\}$ from different levels, let F be the fusion model contains l_w+1 learnable parameters $\{p^0, \ldots, p^{l_w}\}$. We have the fusion module:*

$$\hat{Y} = F(\hat{Y}^0, \ldots, \hat{Y}^{l_w}) = \sum_{\ell=0}^{l_w} p^\ell \hat{Y}^\ell \tag{12}$$

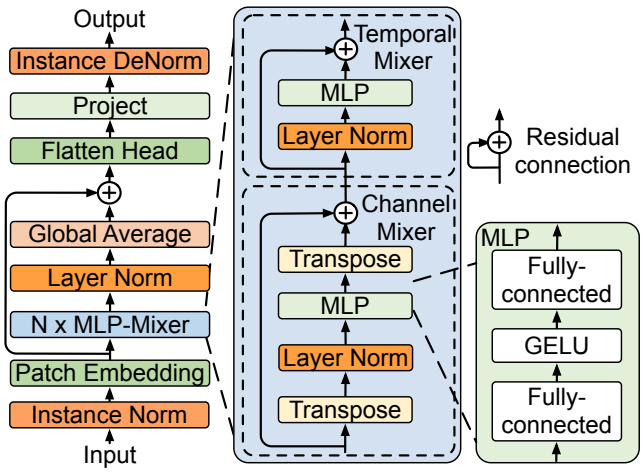

Figure 3: The architecture of the TS-Learner. It mainly consists of Normalization Layers, a Patch Embedding Layer, MLP-Mixer Layers and a projection Layer.

Then, for each level ℓ, we apply the level-specific DWT decomposition to gain the $cA_{\hat{y}}^{\ell}$ and $cD_{\hat{y}}^{\ell}$.

The loss function of each target learner in GEPP can be formulated as follows:

Definition 4. *(Global Error Propagation Loss). Given the predicted approximate coefficients $\hat{cA}_y^i$ and detail coefficients $\hat{cD}_y^i$ sequences at level i, let F be the fusion model. We define the loss function $\mathcal{L}_3(cZ_y^{\ell})$ at GEPP as follows:*

$$\mathcal{L}_3(cZ_y^{\ell}) = \beta\mathcal{L}_1(cZ_y^{\ell}) + (1-\beta)MSE(\hat{cZ}_y^{\ell}, cZ_{\hat{y}}^{\ell}) \quad (13)$$

where cZ means the cA or cD subject to target learner.

3.2 Time Series Learner

Though WaveletMixer can be used with any backbones, we additionally introduce Time Series Learner (TS-Learner), which is used in this paper to improve performance. The architecture of the TS-Learner is shown in Figure 3, which is built upon the concept of MLP-Mixer (Tolstikhin et al. 2021). MLP-Mixer layer processes channel mixing and temporal mixing by transposing inputs. Each MLP consists of two fully connected layers and a GELU nonlinearity. Other components include instance norm, residual connections, dropout, and layer norm on the channels.

Normalization. To eliminate the non-stationary factors of time series, we apply RevIN (Kim et al. 2022), a reversible instance normalization layer to standardizes the distribution of time series by subtracting mean and dividing the standard deviation. And we apply the layer normalization (Ba, Kiros, and Hinton 2016) inside the MLP-Mixer layer.

Patching. Patching the input time series into overlapped or non-overlapped patches allows to reduce the computing complexity and improve the performance (Nie et al. 2023). Our model based on MRDWT has different granularities input tokens naturally. In our experiments, we normally use the finest grain length as the patch length denoted as P and let stride $S = P$. Due to the lengths between different granularities are dyadic-related, which allows that patches to be non-overlap and eliminates the impact of padding.

MLP-Mixer. The MLP-Mixer (Tolstikhin et al. 2021) is a competitive architecture which does not use convolutions or self-attention. We perform Channel Mixer and Temporal Mixer for the time series to learn correlations across channel and temporal dimensions, each Mixer contains an MLP and residual connection inside.

4 Experiments

Benchmarks. We extensively employ 9 real-world multivariate benchmark datasets for evaluating the performance of our approach including ETT datasets (with 4 subsets: ETTh1, ETTh2, ETTm1, ETTm2), Weather, Electricity, Traffic, Exchange and ILI.

Baselines. We compare our WaveletMixer approach to 14 SOTA baselines including iTransformer (Liu et al. 2023), TimeMixer (Wang et al. 2024a), TSMixer (Chen et al. 2023), PatchTST (Nie et al. 2023), TimesNet (Wu et al. 2023), Crossformer (Zhang and Yan 2023), DLinear (Zeng et al. 2023), MICN (Wang et al. 2023), FiLM (Zhou et al. 2022a), NSformer (Liu et al. 2022b), FEDformer (Zhou et al. 2022b), Pyraformer (Liu et al. 2022a), Autoformer (Wu et al. 2021), and Informer (Zhou et al. 2021).

Implementation. WaveletMixer is implemented in PyTorch (Paszke et al. 2019). Experiments are conducted on a single Nvidia L4 24GB GPU. The batch size is uniformly set to 32 (8 for Traffic) and the number of initial and local phases epochs are fixed to 1. We utilize ADAM (Kingma and Ba 2015) as the optimizer and L2 loss for the model optimization. We fix the input length $L = 512$ and commonly used prediction horizon $H \in \{96, 192, 336, 720\}$ ($L = 36$ and $H \in \{24, 36, 48, 60\}$ for ILI) for all baselines, and find the best hyper-parameters for each dataset, model hidden dimension size in [16, 512], MLP-Mixer layers in [1, 4], initial learning rate in $\{1\times10^{-5}, 1\times10^{-4}, 1\times10^{-3}\}$. We use Mean Square Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics. We use PyWavelets (Lee et al. 2019) and ptwt (Wolter et al. 2024) to implement the multi-resolution discrete wavelets transform.

4.1 Main Results

Due to space limitation, Table 1 shows the averaged results from 4 prediction horizon settings in 6 selected datasets (the full results can be found in Supplementary). The results demonstrate that out of a total of 90 forecasting cases, WaveletMixer achieves 56 best results and surpasses the SOTA Transformer-based method iTransformer with an overall 7.71% reduction in MSE and 5.22% reduction in MAE averaged from all datasets. Notably, when compared to the SOTA MLP-based model DLinear, WaveletMixer achieves a 15.15%/11.64% reduction in MSE/MAE overall. In particular, WaveletMixer also outperforms on large datasets with a large number of channels, reducing

Method	**WaveletMixer (Ours)**		iTransformer (2024)		PatchTST (2023)		TimesNet (2023)		Crossformer (2023)		DLinear (2023)		MICN (2023)		FiLM (2022)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.405**	**0.423**	0.448	0.460	0.413	0.434	0.458	0.450	0.600	0.557	0.423	0.437	0.558	0.535	0.482	0.475
ETTm1	**0.347**	**0.377**	0.367	0.397	0.353	0.382	0.353	0.382	0.514	0.510	0.357	0.379	0.392	0.413	0.358	0.380
ETTm2	**0.250**	**0.313**	0.271	0.331	0.256	0.317	0.291	0.333	0.621	0.510	0.267	0.332	0.328	0.382	0.259	0.319
Exchange	**0.414**	**0.432**	0.437	0.449	0.500	0.468	0.861	0.634	1.125	0.782	0.494	0.491	0.650	0.602	0.429	0.541
Electricity	**0.155**	0.254	0.160	0.256	0.159	**0.253**	0.192	0.295	0.186	0.283	0.166	0.264	0.186	0.295	0.186	0.285
Weather	**0.219**	**0.260**	0.243	0.277	0.241	0.264	0.251	0.294	0.406	0.442	0.246	0.300	0.242	0.299	0.253	0.309

Table 1: Comprehensive experiments in 6 selected datasets of multivariate long-term time series forecasting in MSE and MAE (the lower the better). All the results are averaged from 4 different prediction length settings $H \in \{96, 192, 336, 720\}$. The best results are in **bold** and the second best are underlined. The full results can be found in Supplementary Materials.

Case	① $\clubsuit \rightarrow (\spadesuit \rightarrow \blacklozenge)^N$		② $\clubsuit \rightarrow \spadesuit^N$		③ $\clubsuit \rightarrow \blacklozenge^N$		④ $(\blacklozenge \rightarrow \spadesuit)^N$		⑤ $\clubsuit$		⑥ $\spadesuit$		⑦ $\blacklozenge$	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.4102**	**0.4282**	0.4303	0.4454	0.4143	0.4338	0.4445	0.4560	0.4154	0.4334	0.4546	0.4597	0.4142	0.4342
ETTh2	**0.3560**	**0.3960**	0.3638	0.4034	0.3574	0.3979	0.3669	0.4071	0.3565	0.3982	0.3753	0.4158	0.3652	0.4076
ETTm1	**0.3692**	**0.3941**	0.3728	0.3952	0.3721	0.3968	0.3882	0.4066	0.3715	0.3952	0.3752	0.3993	0.3759	0.3988
ETTm2	**0.2618**	**0.3216**	0.2710	0.3284	0.2629	0.3224	0.2809	0.3369	0.2626	0.3221	0.2869	0.3397	0.2746	0.3310
Weather	**0.2354**	**0.2793**	0.2454	0.2868	0.2423	0.2839	0.2782	0.3144	0.2414	0.2832	0.2673	0.3046	0.2543	0.2944

Table 2: Ablation studies of train phase, where ♣ denotes Initial Phase, ♠ denotes Local Phase, ♦ denotes Global Phase, and the superscript N denotes the number of iterations repeated. We fix $N = 3$ here for all cases. The best results are in **bold**.

10.97%/13.33% on Weather and 6.62%/3.78% on Electricity in MSE/MAE, where Weather has 21 and Electricity has 321 channels. And WaveletMixer outperforms mr-Diff (Shen, Chen, and Kwok 2024), better MAEs in 5/6 datasets.

Type		Fusion		Average		Concat	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	**0.268**	**0.334**	0.273	0.338	0.301	0.366
	192	0.325	**0.372**	**0.321**	0.373	0.441	0.461
	336	**0.334**	**0.384**	0.351	0.399	0.521	0.504
	720	**0.376**	**0.420**	0.432	0.459	0.871	0.650
	Avg	**0.326**	**0.378**	0.344	0.392	0.533	0.495

Table 3: Ablation study of Multi-Prediction Fusion Module.

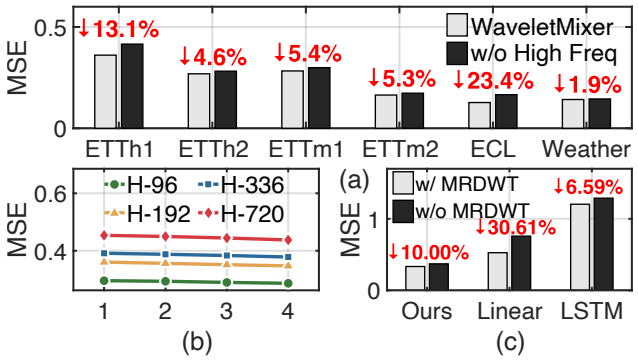

Figure 4: (a) The effectiveness of high frequency components; (b) Analysis of the number of Multi-resolution DWT levels on ETTh1; (c) The effectiveness of applying MRDWT to WaveletMixer, Linear and LSTM on ETTh2.

4.2 Ablation Studies

We study WaveletMixer to prove that its advantages come from efficient multi-phase training strategies, multi-resolution and high-frequency prediction.

Efficient Multi-Phase Training Strategies. We explore the ablation experiments of each training phase, and showcase the averaged results on 5 selected datasets in the Table 2. The ablation study clearly demonstrates that our proposed model (case ①) consistently achieves the best results across all benchmarks. This underscores the effectiveness of the full initial-local-global training strategy. Case ②, ③, ④, ⑤, ⑥ and ⑦ represent the model removing one or two phases, shows a higher MSE and MAE which suggests that all the phases contribute to the performance, likely helping model in capturing patterns in the data, which are essential for multi-grained accuracy. The Original model, which utilizes all three training phases, outperforms the ablated versions, proves WaveletMixer utilizing the most effective strategy for optimizing the model's performance.

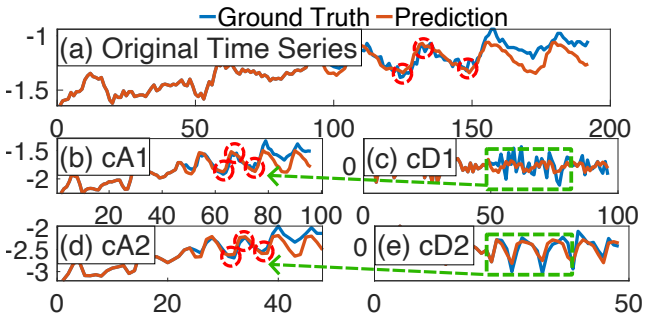

Figure 5: Visualization of prediction from different components under MRDWT on ETTh1.

High Frequency Components. Figure 4(a) shows that adding the high-frequency part (i.e., $c\hat{D}_y^{\ell} \neq 0$) helps to improve the MSE of up to 23.4% on 6 studied datasets.

Multi-Resolution Levels. Figure 4(b) shows that with the increasing of the number of Multi-resolution DWT levels l_w, the model achieves the better performance under four different prediction horizon lengths.

The Effectiveness of MRDWT Framework. In Figure 4(c), we apply our proposed MRDWT framework to two classical models (Linear and LSTM). The results show that these models also can be significant benefited from our WaveletMixer framework. Notably, Linear reduces 30.61% in MSE on ETTh2 with our MRDWT framework.

Multi-Prediction Fusion Module. We evaluate different types of the fusion module in Table 3. The results show that using our Fusion module can achieve the best performance.

Wavelets Bases. We run experiments using Haar, Bior and Rbio Wavelets basis on ETTh1. Haar acquires the best performance in terms of both MSEs and MAEs. Please refer to supplementary for full results.

4.3 Hyperparameter Sensitivity

We perform a comprehensive hyperparameter sensitivity analysis of WaveletMixer including the hidden dimension size, the number of MLP-Mixer layers, the look-back window length, learning rate, phases parameters alpha and beta. Full analysis can be found in Supplementary Materials.

4.4 Computational Complexity Analysis

WaveletMixer has fewer parameters and trains faster than most methods while acquiring better prediction accuracy, which makes it very practical. Eg., it takes 11.33s to run one epoch ($l_w = 1$) compared to 105.91s of PatchTST on ETTh1(H=96) (c.f. Supplementary for full details).

4.5 Visualization

We visualize the prediction of each component of multi-resolution DWT in Figure 5, where (a) shows the original time series, and (b)(c)(d)(e) show the approximation and detail coefficient of applying DWT to the original series. The red circles and green lines highlight how the high-frequency components add details to the final prediction and make the final prediction more accurate.

5 Related Work

Long-term Time Series Forecasting (LTSF). Recent transformer-based studies (Liu et al. 2023; Nie et al. 2023; Zhou et al. 2021) have been achieving promising results on real-world benchmarks. PatchTST (Nie et al. 2023), a Transformer-based model, introduces the concept of patch, which cuts the input time series data into patches and makes individual predictions for each channel to improve the overall performance. However, some studies have shown that MLP and CNN models (Zeng et al. 2023; Wu et al. 2023) can also achieve state-of-the-art results. DLinear (Zeng et al. 2023) unravels the trend and seasonality items of the time series and improves forecasting performance by predicting each component separately. TimesNet (Wu et al. 2023), a CNN-based model, divides periodicity into intraperiod- and interperiod variations and capture the multi-periodicities by transforming the 1D time series into 2D space.

Frequency Domain Analysis. Analysing in frequency domain can efficiently disentangle time series data showing the hidden pattern and characteristic. FEDformer (Zhou et al. 2022b) recognizes the key frequency components by designing a frequency-enhanced block based on Fourier transform. FiLM (Zhou et al. 2022a) uses frequency information to improve prediction results and speed up the computational speed of the model. Autoformer (Wu et al. 2021) uses an Auto-Correlation mechanism to find the correlation in time series by applying the Fast Fourier Transform.

Multi-Resolution Analysis. Reality long-term time series usually have multi-granularity periodicity, which has prompted the development of multi-scale and multi-resolution path modeling. MICN (Wang et al. 2023) proposes to use isometric convolution to capture global correlations for a short sequence and achieve $O(LD^2)$ complexity. mWDN (Wang et al. 2018) finds that MRDWT can improve the interpretability of deep learning, and has achieved good results in time series prediction and time series classification. MRA-AWNN (Doucoure, Agbossou, and Cardenas 2016) uses MRDWT to optimize the complexity and performance for time series prediction system.

6 Conclusion

In this paper, we propose WaveletMixer, a novel deep learning model based on multi-resolution wavelet transform to the field of long-term time series forecasting. We introduce the prediction of the high-frequency part as a complement for the multi-resolution model, redesign the multi-resolution training strategy and fully ensemble multiple prediction results to make the final prediction outcomes. Experimentally, WaveletMixer has demonstrated superior performance on 9 benchmarks compared to many SOTA methods. Our future works aim to explore the performance of WaveletMixer with different backbones and other types of Wavelet transformations for time series forecasting tasks.

Acknowledgments

This research is part-funded by the European Union (Horizon Europe 2021-2027 Framework Program Grant Agreement number 10107245. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. The European Union cannot be held responsible for them) and by the Engineering and Physical Sciences Research Council under grant number EP/X029174/1. In addition, Zichi Zhang is supported by the China Scholarship Council (CSC) under Grant No. 202410080008. Yimeng An is supported by Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX23_0404).

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. arXiv:1607.06450.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Chen, P.; Zhang, Y.; Cheng, Y.; Shu, Y.; Wang, Y.; Wen, Q.; Yang, B.; and Guo, C. 2024. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. *CoRR*, abs/2402.05956.

Chen, S.-A.; Li, C.-L.; Yoder, N.; Arik, S. O.; and Pfister, T. 2023. TSMixer: An All-MLP Architecture for Time Series Forecasting. arXiv:2303.06053.

Datilo, P. M.; Ismail, Z.; and Dare, J. 2019. A review of epidemic forecasting using artificial neural networks. *Epidemiol. Health System J.*, 6(3): 132–143.

Doucoure, B.; Agbossou, K.; and Cardenas, A. 2016. Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. *J. Renew. Energy*, 92: 202–211.

Harti, A. 1993. Discrete multi-resolution analysis and generalized wavelets. *APPL NUMER MATH*, 12(1-3): 153–192.

Jain, G.; and Mallick, B. 2016. A review on weather forecasting techniques. *IJARCCE*, 5(12): 177–180.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.; and Choo, J. 2022. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *ICLR*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *ICLR*.

Koprinska, I.; Wu, D.; and Wang, Z. 2018. Convolutional Neural Networks for Energy Time Series Forecasting. In *IJCNN*, 1–8.

Lai, G.; Chang, W.; Yang, Y.; and Liu, H. 2018. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *SIGIR*, 95–104.

Lee, G. R.; Gommers, R.; Wasilewski, F.; Wohlfahrt, K.; and O'Leary, A. 2019. PyWavelets: A Python package for wavelet analysis. *J. Open Source Softw.*, 4(36): 1237.

Li, Y.; Xu, J.; and Anastasiu, D. C. 2024. Learning from Polar Representation: An Extreme-Adaptive Model for Long-Term Time Series Forecasting. In *AAAI*, 171–179.

Li, Z.; Qi, S.; Li, Y.; and Xu, Z. 2023. Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping. *CoRR*, abs/2305.10721.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2022a. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *ICLR*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. *arXiv preprint arXiv:2310.06625*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-stationary Transformers: Rethinking the Stationarity in Time Series Forecasting. *CoRR*, abs/2205.14415.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR*, abs/1912.01703.

Rodríguez, A.; Muralidhar, N.; Adhikari, B.; Tabassum, A.; Ramakrishnan, N.; and Prakash, B. A. 2021. Steering a Historical Disease Forecasting Model Under a Pandemic: Case of Flu and COVID-19. In *AAAI*, 4855–4863.

Sezer, O. B.; Gudelek, M. U.; and Ozbayoglu, A. M. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.*, 90: 106181.

Shen, L.; Chen, W.; and Kwok, J. 2024. Multi-Resolution Diffusion Models for Time Series Forecasting. In *ICLR*.

Stanković, R. S.; and Falkowski, B. J. 2003. The Haar wavelet transform: its status and achievements. *Comput. Electr. Eng.*, 29(1): 25–44.

Sundararajan, D. 2016. *Discrete wavelet transform: a signal processing approach*. John Wiley & Sons.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; Lucic, M.; and Dosovitskiy, A. 2021. MLP-Mixer: An all-MLP Architecture for Vision. In *NeurIPS*, 24261–24272.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting. In *ICLR*.

Wang, J.; Wang, Z.; Li, J.; and Wu, J. 2018. Multilevel Wavelet Decomposition Network for Interpretable Time Series Analysis. In *SIGKDD*.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2024a. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *ICLR*.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Long, M.; and Wang, J. 2024b. Deep Time Series Models: A Comprehensive Survey and Benchmark. arXiv:2407.13278.

Wolter, M.; Blanke, F.; Garcke, J.; and Hoyt, C. T. 2024. ptwt - The PyTorch Wavelet Toolbox. *JMLR*, 25(80): 1–7.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *NeurIPS*, 22419–22430.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In Williams, B.; Chen, Y.; and Neville, J., eds., *AAAI*, 11121–11128.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *ICLR*.

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 *AAAI*, 11106–11115.

Zhou, T.; Ma, Z.; Wang, X.; Wen, Q.; Sun, L.; Yao, T.; Yin, W.; and Jin, R. 2022a. FiLM: Frequency improved Legendre Memory Model for Long-term Time Series Forecasting. In *NeurIPS*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022b. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *ICML*, volume 162, 27268–27286.