

WaveRoRA: Wavelet Rotary Route Attention for Multivariate Time Series Forecasting

Aobo Liang, Yan Sun and Nadra Guizani

Abstract—In recent years, Transformer-based models (Transformers) have achieved significant success in multivariate time series forecasting (MTSF). However, previous works focus on extracting features either from the time domain or the frequency domain, which inadequately captures the trends and periodic characteristics. To address this issue, we propose a wavelet learning framework to model complex temporal dependencies of the time series data. The wavelet domain integrates both time and frequency information, allowing for the analysis of local characteristics of signals at different scales. Additionally, the Softmax self-attention mechanism used by Transformers has quadratic complexity, which leads to excessive computational costs when capturing long-term dependencies. Therefore, we propose a novel attention mechanism: Rotary Route Attention (RoRA). Unlike Softmax attention, RoRA utilizes rotary position embeddings to inject relative positional information to sequence tokens and introduces a small number of routing tokens r to aggregate information from the KV matrices and redistribute it to the Q matrix, offering linear complexity. We further propose WaveRoRA, which leverages RoRA to capture inter-series dependencies in the wavelet domain. We conduct extensive experiments on eight real-world datasets. The results indicate that WaveRoRA outperforms existing state-of-the-art models while maintaining lower computational costs. Our code is available at <https://github.com/Leopold2333/WaveRoRA>.

Index Terms—Time series forecasting, Multivariate time series, Attention mechanism, Rotary position embeddings, Wavelet transform.

I. INTRODUCTION

MULTIVARIATE time series (MTS) are prevalent in many real world scenarios, including weather [1], energy management [2], transportation [3] and network flow [4]. Through analyzing historical observations, multivariate time series forecasting (MTSF) aims to extract the underlying patterns of time series data and provides future trends prediction. In the past few years, deep neural networks especially Transformer-based models (Transformers) have achieved great success in MTSF. Transformers rely on the Softmax self-attention (SA) mechanism [5] to effectively model long-term dependencies in sequences. Models such as Informer [6] and PatchTST [7] make better predictions than previous works that are based on CNNs [8], [9] and RNNs [10]. However, these approaches neglect the correlations between time series, thereby limiting their predictive performance. Some recent

works have identified this issue. Models like Sageformer [11] and iTransformer [12] focus on explicitly capturing inter-series dependencies and get superior performance. Nonetheless, SA has quadratic computational complexity and may hinder these models' scalability. Specifically, when modeling intra- or inter-series dependencies, SA incurs excessive computational costs as the sequence length or the number of variables increases, thus limiting its application in large-scale or high-dimensional scenarios [13], [14]. This motivates us to develop a lightweight attention mechanism that simultaneously achieves the expressivity of SA. In this paper, we design a novel **Rotary Route Attention** mechanism (RoRA) for MTSF. We first introduce a set of routing tokens r which has much fewer elements than that of the input sequence. The routing tokens aggregate information from the KV matrices and redistribute it to the Q matrix, offering linear complexity. Subsequently, we inject relative positional information through rotary positional embeddings (RoPE) [35] to enhance the feature diversity.

In addition, existing works mainly focus on analyzing the underlying patterns of MTS data in either time or frequency domains. Features of time domain and frequency domain can reflect different characteristics of the time series. For example, time domain features are conducive to extracting the trends of time series, while frequency domain features reveal the amplitude and phase information of each frequency component of the time series through discrete Fourier transform (DFT) [15]. Compared with time domain features, frequency domain features intuitively reflect the periodicity or seasonality of time series [16]. However, DFT faces the challenges in accurately fitting discontinuous signals, such as abrupt or non-triangular waveforms, which leads to the Gibbs phenomenon [17], [30]. In addition, DFT has limitations in capturing the changes in the frequency domain over time. As shown in Figure 1. Although the three signals exhibit significant differences in the time domain, their frequency spectra are notably similar. Relying exclusively on frequency domain features may result in misleading information and negatively impact the predicting performance of the model. Recent works such as Autoformer [18] and FEDformer [19] introduce frequency-domain features into time-domain representations to facilitate in extracting richer information. However, directly concatenating features of different domains may introduce noise to the model [20], resulting in suboptimal results.

Therefore, our intuition is that extracting features in the wavelet domain could be beneficial for more accurate predictions. Wavelet transform inherits and develops the localization concept of short-time Fourier transform [21], while overcoming its drawbacks such as the fixed window size

A. Liang and Y. Sun are with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: liangaobo@bupt.edu.cn; sunyan@bupt.edu.cn.

N. Guizani is with the school of computer science and engineering, University of Texas at Arlington, Arlington, Texas, USA, 76019. E-mail: nadra.guizani@uta.edu.

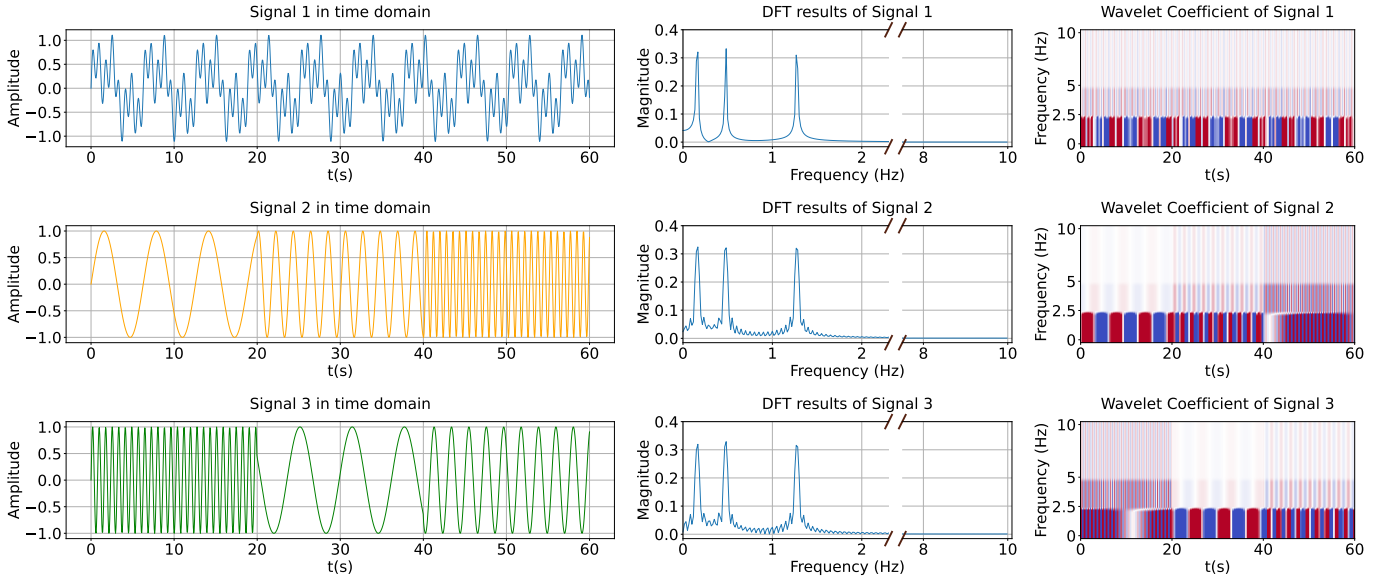


Fig. 1: Three signals consisting of the same three functions: $\sin x$, $\sin 3x$ and $\sin 8x$ with the sampling rate 20Hz. Signal 1 is defined as $y = \sin x + \sin 3x + \sin 8x$, Signal 2 is evenly divided into 3 segments in the order of $\sin x$, $\sin 3x$ and $\sin 8x$, while signal 3 adjusts the order to $\sin 8x$, $\sin x$ and $\sin 3x$. These signals are different in the time domain but share similar patterns in the frequency domain. The rightmost column shows the wavelet coefficients obtained from a three-level DWT. Compared to the DFT results, the multi-layer wavelet coefficients preserve different periodic characteristics while better revealing the intervals where different periodic patterns are dominant.

that does not vary with different frequencies [23]. Wavelet transform extracts the time-frequency characteristics of a time series by scaling and translating a set of localized wavelet basis functions that decay within a short temporal window. For discrete time series, the Discrete Wavelet Transform (DWT) [22], [24] is typically employed, involving a multi-level wavelet decomposition. As illustrated in Fig. 1, the wavelet coefficients produced from the multi-layer decomposition reflect the different periodic characteristics of the time series and accurately reveal the intervals where different periodic patterns are dominant. Our intuition is that extracting features in the wavelet domain allows us to simultaneously leverage the strengths of both time and frequency domains.

We decompose time series into multi-scale high-pass and low-pass components by DWT and design a multi-scale wavelet learning framework to facilitate the mutual conversion between wavelet coefficients and unified-size embeddings, aiming to capture temporal patterns within the time series. We then propose WaveRoRA to capture inter-series dependencies in the wavelet domain. Compared with SA, RoRA uses the relative positional information to facilitate in identifying the interactions between variables. Meanwhile, the routing tokens effectively alleviate the inherent redundancy between the attention weights and extract critical inter-series dependencies among different variables. Our contributions can be summarized as follows:

- We propose modeling MTS data in the wavelet domain. To capture temporal patterns within the time series, we design a multi-scale wavelet learning framework that leverages both high-pass and low-pass components derived from DWT.

- We design a novel RoRA mechanism with linear complexity, introducing a small number of routing tokens r to aggregate critical information from the KV matrices and redistribute it to the Q matrix. We then use RoPE to inject relative position information.
- Based on the wavelet learning framework and RoRA, we further propose WaveRoRA, which captures inter-series dependencies in the wavelet domain. We conduct extensive experiments on eight real-world datasets. WaveRoRA outperforms existing SOTA models while maintaining low computational costs.

II. RELATED WORK

In this section, we summarize the typical methods of MTSF and provide a simple review of deep models combined with wavelet transform.

A. Multivariate Time Series Forecasting

MTSF aims to forecast future values based on a specific period of historical observations [25]. Recently, Transformers achieve great success in MTSF with the Softmax self-attention mechanism [5]. However, the quadratic complexity of sequence length L limits these models' application on MTSF. Researches in recent years thus focus on balancing the predicting performance and computational efficiency of Transformers. LogTrans [26] introduces a local convolutional module in self-attention layers and propose *LogSparse* attention, which reduces the complexity of $O(L(\log L)^2)$. Informer [6] proposes a *ProbSparse* attention using a distilling technique and selecting top-k elements of the attention weight

matrix to achieve $O(L \log L)$ complexity. Autoformer [18] designs a decomposition architecture with an *Auto-Correlation* mechanism. It discovers the sub-series similarity based on the series periodicity and reduce the complexity to $O(L \log L)$. Despite improving forecasting performance and reducing computational costs, these models still rely on point-wise tokens, which limit the receptive field of the input sequence and may lead to overfitting in long-term forecasting. To address this issue, PatchTST [7] generates patch-wise tokens of each univariate series independently and capture intra-series dependencies. It reduces the computational complexity to $O((\frac{L}{P})^2)$, where P denotes the patch length. The patch-wise tokens facilitates in expanding the receptive field of the input series, leading to better predicting performance. However, PatchTST overlooks the complex interactions between different variables. A recent model named iTransformer [12] generates series-wise tokens and inverts the attention layers to model the inter-series dependencies directly. Although iTransformer achieves superior performance, it still faces high computational costs of $O(M^2)$ when the number of variables M is large.

B. Wavelet Transform

MTS data can be analyzed from various perspectives, such as the time domain, frequency domain, or wavelet domain, each offering unique insights of the underlying patterns of the time series [28]. Generally, the time domain and frequency domain provide intuitive features related to trends and periodic characteristics respectively [27]. Some existing works have explored combining features from these two domains. Autoformer [18] extracts series periods from frequency domain to assist its proposed *Auto-correlation* in capturing dependencies between periodic segments. FEDformer [19] randomly selects a fixed number of Fourier components to better represent a time series. Through combining Transformer with frequency analysis, FEDformer reduces the computational complexity while leading to performance improvement. However, the Fourier transform struggles to accurately fit discontinuous signals, such as abrupt changes or non-triangular waves. These types of signals are commonly found in real-world scenarios and can cause the Gibbs phenomenon [17], [30].

Analyzing time series in the wavelet domain naturally benefits from its ability to capture both temporal trends in the time domain and periodic patterns in the frequency domain. Wavelet transform decomposes a signal into different frequency and time scales by using scaled and shifted versions of a mother wavelet [22]. In recent years, several researches have been done to explore the effectiveness of modeling time series in the wavelet domain. TCDformer [31] initially encoding abrupt changes using the local linear scaling approximation (LLSA) module and adopt Wavelet Attention to capture temporal dependencies. However, the model uses DWT to directly process point-wise tokens, which causes large computational costs. WaveForM [20], on the contrary, utilizes DWT to map the input series to latent representations in the wavelet domain. It uses dilated convolution and graph convolution modules to further capture the inter-series dependencies [29]. However, WaveForM overlooks the correlation of wavelet coefficients at

different levels, which may adversely affect the modeling of the periodic characteristics of time series data.

III. PRELIMINARIES

In this section, we first introduce the concept of multivariate time series forecasting, then summarize the definition of DWT and several attention mechanisms.

A. Multivariate Time Series Forecasting

Given a multivariate historical observations $\mathbf{X}_{in} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\} \in \mathbb{R}^{L \times M}$ with L time steps and M variables, MTSF aims to predict $\mathbf{X}_{out} = \{\mathbf{x}_{L+1}, \mathbf{x}_{L+2}, \dots, \mathbf{x}_{L+H}\}$, where H denotes the prediction length.

B. Discrete Wavelet Transform

DWT decomposes a given signal X multiple times to generate wavelet coefficients at different resolutions through a high-pass filter \mathbf{h} and a low-pass filter \mathbf{g} . Typically, \mathbf{h} and \mathbf{g} correspond to a wavelet function $\psi(t) = a^{-\frac{1}{2}}\psi(\frac{t-b}{a})$ and a scaling function $\phi(t) = a^{-\frac{1}{2}}\phi(\frac{t-b}{a})$, which collectively form an orthogonal basis. The parameter a is the scaling factor that controls the dilation of the wavelet while b represents the translation factor that shifts the wavelet in time. Specifically, the process of DWT can be defined as Eq 1 and Eq 2, where j indicates the j -th decomposition, $y_h^{(j)}, y_l^{(j)}$ are the high-pass and low-pass components and $y_l^{(0)} = X$, S denotes the length of filters and $L^{(j)} = \lfloor \frac{L^{(j-1)} + S - 1}{2} \rfloor$ refers to the length of $y_l^{(j)}$. \mathbf{h} and \mathbf{g} are performed convolution operations with the low-pass component of the former layer. After applying the convolutional filters, the frequency range of the signal is compressed to half of its original size. Therefore, the results can be downsampled by a factor of 2 according to the Nyquist theorem. It's worth noting that the scaling factor a and translation factor b of \mathbf{h} and \mathbf{g} require no adjustment because (a) the downsampling process simultaneously changes the resolution, allowing for multi-scale analysis; (b) the DWT process inherently encompasses translation as part of the wavelet decomposition.

$$y_h^{(j)}(n) = \sum_{s=1}^S \mathbf{h}(s) y_l^{(j-1)}(2n - s), \quad n = 1, 2, \dots, L^{(j-1)} \quad (1)$$

$$y_l^{(j)}(n) = \sum_{s=1}^S \mathbf{g}(s) y_l^{(j-1)}(2n - s), \quad n = 1, 2, \dots, L^{(j-1)} \quad (2)$$

There will be $J + 1$ wavelet coefficient components $\mathcal{C} = \{y_h^{(1)}, y_h^{(2)}, \dots, y_h^{(J)}, y_l^{(J)}\}$ after J -level decomposition. Given a multivariate input $X \in \mathbb{R}^{L \times M}$, each output component contains the wavelet coefficient of each variable, denoted as $y_h^{(j)} = [y_{h,1}^{(j)}, y_{h,2}^{(j)}, \dots, y_{h,M}^{(j)}] \in \mathbb{R}^{L^{(j)} \times M}$ and $y_l^{(j)} = [y_{l,1}^{(j)}, y_{l,2}^{(j)}, \dots, y_{l,M}^{(j)}] \in \mathbb{R}^{L^{(j)} \times M}$.

To reconstruct the time series, the Inverse Discrete Wavelet Transform (IDWT) uses the synthesis versions of \mathbf{h} and \mathbf{g} ,

denoted as \mathbf{h}' and \mathbf{g}' . The detailed process of IDWT can be defined in Eq 3:

$$\hat{y}_l^{(j-1)}(n) = \sum_{s=1}^S \mathbf{h}'(s) \hat{y}_h^{(j)}(n-2s) + \sum_{s=1}^S \mathbf{g}'(s) \hat{y}_l^{(j)}(n-2s), \quad (3)$$

$$n=1, 2, \dots, 2L^{(j)}$$

$$\hat{y}_h^{(j)}(n) = \begin{cases} y_h^{(j)}(\frac{n}{2}), & \text{if } n \text{ is even} \\ 0, & \text{if } n \text{ is odd} \end{cases} \quad (4)$$

$$\hat{y}_l^{(j)}(n) = \begin{cases} \hat{y}_l^{(j)}(\frac{n}{2}), & \text{if } n \text{ is even} \\ 0, & \text{if } n \text{ is odd} \end{cases}$$

where $\hat{y}_h^{(j)}$, $\hat{y}_l^{(j)}$ are the upsampled results defined in Eq 4, $\hat{y}_l^{(j)}$ is the reconstructed low-pass component of the j -th layer and $\hat{y}_l^{(J)} = y_l^{(J)}$. \mathbf{h}' and \mathbf{g}' are performed convolution operations respectively with the high-pass and low-pass components of the current layer. The output of $\hat{y}_l^{(0)} \in \mathbb{R}^{L \times M}$ is the final reconstructed sequence in the time domain.

C. Attention Mechanisms

Given an input sequence $x \in \mathbb{R}^{N \times d}$ where N denotes as the number of tokens and d represents the token dimension, the attention mechanism can be defined as Eq 5:

$$\text{Attention}(Q_i, K_j, V_j) = \sum_{j=1}^N \frac{\text{Sim}(Q_i, K_j)}{\sum_{j=1}^N \text{Sim}(Q_i, K_j)} V_j, \quad (5)$$

$$Q = xW_Q, K = xW_K, V = xW_V$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ represent the projection matrices and Q, K, V denotes query, key and value matrices respectively. The self-attention [5] uses $\text{Sim}(Q_i, K_j) = \exp(Q_i K_j^T / \sqrt{d})$ to produce scaled Softmax attention (SA) scores. Such calculation processes lead to $O(N^2 d)$ complexity.

To address the quadratic computational complexity, a variation of the self-attention mechanism is proposed, namely linear attention (LA) [32]. linear attention is based on the assumption that there exists a kernel function Φ that can approximate the results of $\text{Sim}(Q, K)$. Specifically, the function Φ aims to decouple the parameters of $\text{Sim}(Q, K)$ and reformulate it as shown in Eq 6:

$$\text{LA}(Q_i, K_j, V_j) = \sum_{j=1}^N \frac{\Phi(Q_i) \Phi(K_j)^T}{\sum_{j=1}^N \Phi(Q_i) \Phi(K_j)^T} V_j \quad (6)$$

$$= \frac{\Phi(Q_i) \sum_{j=1}^N \Phi(K_j)^T V_j}{\Phi(Q_i) \sum_{j=1}^N \Phi(K_j)^T},$$

where $\Phi(x) = \text{Elu}(x) + 1$ typically. The order of matrix multiplication changes from Q - K pairs to K - V pairs, leading to $O(Nd^2)$ complexity. However, linear attention still faces the challenges with excessive computational complexity when dealing with high-dimensional input data.

IV. METHODOLOGY

In this section, we overview the model and then detail the operations of each component in the model.

A. Overview

As shown in Fig 2, given the input MTS data $\mathbf{X}_{in} \in \mathbb{R}^{L \times M}$, we first apply instance normalization to stabilize the time series. We then apply J -level DWT decomposition and get J high-pass coefficients $y_h^{(j)} \in \mathbb{R}^{M \times L^{(j)}}$, $j \in \{1, 2, \dots, J\}$ and 1 low-pass coefficients $y_l^{(J)} \in \mathbb{R}^{M \times L^{(J)}}$. Due to the downsampling during the multi-layer decomposition of DWT, the lengths of the wavelet coefficients of each layer are inconsistent. Therefore, we introduce a set of wave-embedding layers to map the multi-layer coefficients to wavelet-wise tokens with a uniformed size D , denoted as $\mathbb{E} \in \mathbb{R}^{(J+1) \times M \times D}$. After the projection, we transpose \mathbb{E} into series-wise tokens $\mathbb{S} \in \mathbb{R}^{M \times ((J+1) \cdot D)}$. We then use N -layer WaveRoRA Encoders to capture the inter-series dependencies in the wavelet domain. For the obtained features, we transpose them again to wavelet-wise tokens $\mathbb{O} \in \mathbb{R}^{(J+1) \times M \times D}$. Our objective is that each element in \mathbb{O} approximates the high-pass and low-pass coefficients obtained from J -level wavelet decomposition of the ground truth. Therefore, we use a set of wave-predictors to map them to their corresponding lengths. Finally, we apply IDWT to get the predicted values.

B. Instance Normalization

Real-world series usually present non-stationarity [33], resulting in the distribution drift problem [34]. Therefore, we adopt the instance normalization and denormalization technique that is used in Stationary Transformers [33]. Specifically, we record the mean value μ_{in} and variance value σ_{in} of the input series and use them to normalize the input series. When obtaining the output, we apply the same μ_{in} and σ_{in} to denormalize the output series.

C. Multi-scale Wave Embedding

The input MTS data is first decomposed to J high-pass coefficients $y_h^{(j)} \in \mathbb{R}^{M \times L^{(j)}}$ and 1 low-pass coefficients $y_l^{(J)} \in \mathbb{R}^{M \times L^{(J)}}$ with J -level DWT, where $j \in \{1, 2, \dots, J\}$. Each coefficient series is passed through a corresponding wave-embedding layer defined in Eq 7, where $\mathcal{C} = \{y_h^{(1)}, y_h^{(2)}, \dots, y_h^{(J)}, y_l^{(J)}\}$ and WaveEmbedding is a linear projector. Specifically, WaveEmbedding $_j$ contains a weight matrix W_j and a bias matrix b_j . Note that $W_j \in \mathbb{R}^{D \times L^{(j)}}$ for $j \in \{1, 2, \dots, J\}$ and $W_{J+1} \in \mathbb{R}^{D \times L^{(J)}}$, while $b_j \in \mathbb{R}^{D \times 1}$. The multi-scale wave-embedding layers unify the feature dimensions of wavelet coefficients obtained from the multi-layer DWT. What's more, the embedding process facilitates in capturing the temporal patterns within the time series.

$$\mathbb{E}_j = \text{WaveEmbedding}_j(\mathcal{C}_j) \quad (7)$$

$$= \mathcal{C}_j W_j^T + b_j^T$$

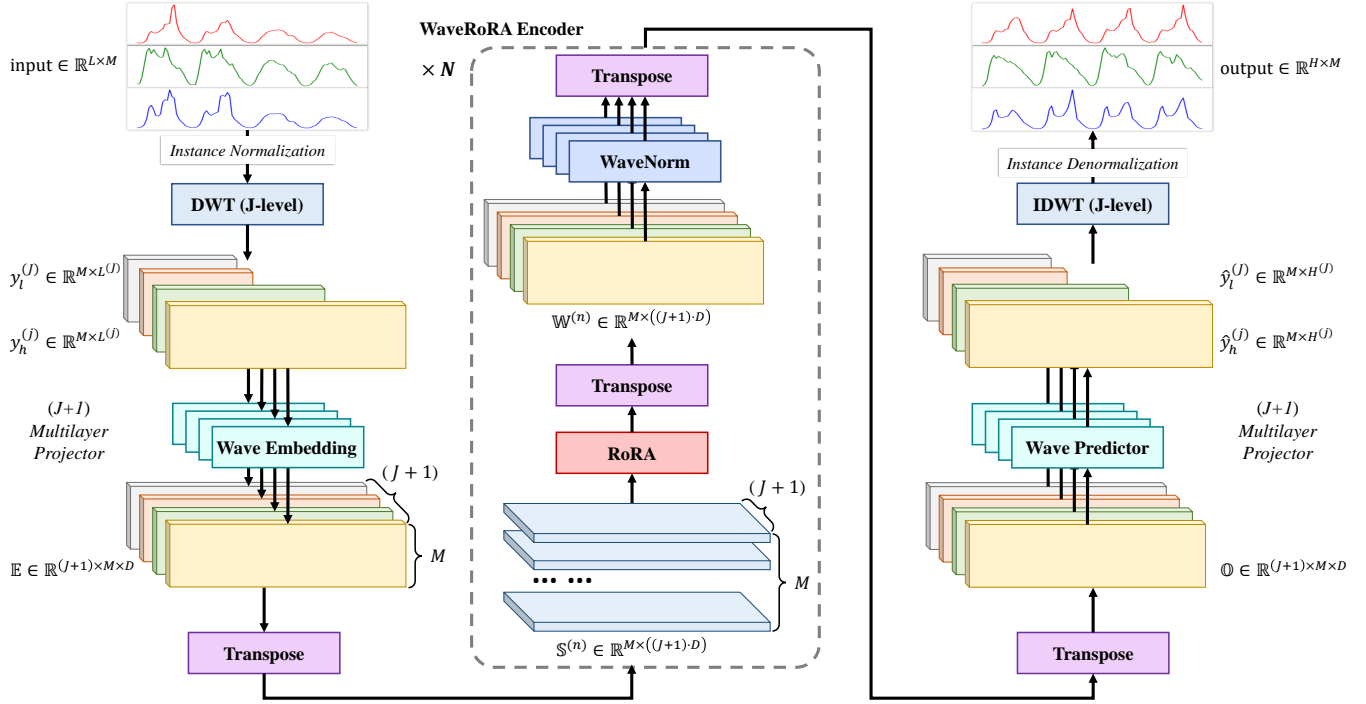


Fig. 2: The architecture of WaveRoRA. The input MTS data is first stabilized by instance normalization and then transformed to multi-scale wavelet coefficients through J -level DWT. Each series of coefficients is passed through a corresponding wave embedding layer to generate uniformed wavelet-wise tokens. These embeddings are then transposed to series-wise tokens and fed into N -layer WaveRoRA Encoders to capture inter-series dependencies. Subsequently, the outputs are input to a set of wave predictors to obtain the predicted wavelet coefficients, which are then supplied to the IDWT to generate the final predictions.

$$\mathcal{M}_{\Theta,n}^d = \begin{pmatrix} \cos n\theta_1 & -\sin n\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin n\theta_1 & \cos n\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos n\theta_2 & -\sin n\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin n\theta_2 & \cos n\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos n\theta_{\frac{d}{2}} & -\sin n\theta_{\frac{d}{2}} \\ 0 & 0 & 0 & 0 & \cdots & \sin n\theta_{\frac{d}{2}} & \cos n\theta_{\frac{d}{2}} \end{pmatrix} \quad (8)$$

D. WaveRoRA Encoder

RoRA is proposed to capture inter-series dependencies in the wavelet domain. We first transpose $\mathbb{E} \in \mathbb{R}^{(J+1) \times M \times D}$ into series-wise tokens $\mathbb{S} \in \mathbb{R}^{M \times ((J+1) \cdot D)}$ where each token in \mathbb{S} represents the concatenation of wavelet embeddings across all decomposition levels for a single variable. To simplify the representation, we use D' instead of $(J+1) \cdot D$ and $I \in \mathbb{R}^{M \times D'}$ instead of $\mathbb{S}^{(n)}$ of each WaveRoRA Encoder.

1) *Rotary Route Attention*: A set of linear projectors are applied to generate query, key and value matrices, represented as $Q \in \mathbb{R}^{M \times D'}$, $K \in \mathbb{R}^{M \times D'}$ and $V \in \mathbb{R}^{M \times D'}$. Subsequently, a small number of routing tokens $R \in \mathbb{R}^{r \times D'}$ is randomly initialized and the multi-head version of RoRA can be represented in Alg 1, where \odot refers to Hadamard product and the activation function σ is set as SiLU [37], generating $G \in \mathbb{R}^{M \times D'}$ which serves as a gating function. The routing tokens R is initially viewed as the query for K and V , aggregating routing features from all keys and values. Afterward, R and V_R will serve as the key and value

for Q respectively, enabling the transfer of information from the routing features to each query token in Q . In our implementations, R is expected to prune redundant relationships between low-correlation variables while preserving interaction information among high-correlation variables. In addition, we set $r \ll M$ as a fixed number to achieve linear complexity of $O(Mrd)$ relative to the variable number M .

RoRA could be simplified to a form of generalized linear attention as shown in Eq 10, which may lead to insufficient feature diversity [39]. Therefore, we try to solve this issue by injecting relative positional information through rotary mechanism. Typically, a rotary matrix \mathcal{M}_{Θ}^d defined in Eq 8 is multiplied with Q and K in SA or LA, where Θ represents a set of predefined rotational angles and d is the dimension of each token. \mathcal{M}_{Θ}^d adaptively adjusts the angles based on each token's position, thereby incorporating the relative positional information $(m-n)\Theta$ for any indexes m, n in Q and K . However, such mechanism relies on the assumption that Q and K are directly multiplied to generate the attention scores. In

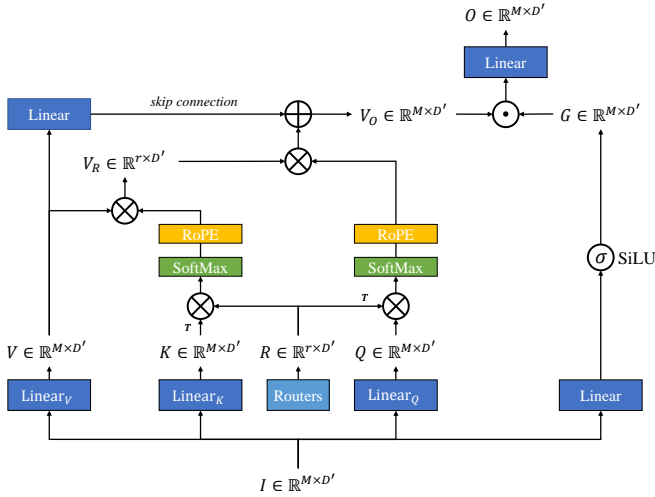


Fig. 3: The architecture of RoRA.

our proposed RoRA, the routing tokens R may disrupt the rotational angles applied to Q and K . Assuming that we follow the typical use of RoPE as shown in Eq 9, the relative positional information would not be correctly computed. Specifically, the hypothetical rotary attention can be expressed as Eq 10, where Ψ could be a certain kernel function. Assuming that Ψ satisfies linear properties, we can then get an approximate solution $\Psi(Q\mathcal{M}_{\Theta}^{D'}R^TR(\mathcal{M}_{\Theta}^{D'})^TK^T)$. Furthermore, it can easily be demonstrated that for any position indexes m and n , the relative positional embeddings with the rotational angle $(m-n)\Theta$ could only be correctly obtained when R^TR equals to an identity matrix. However, the routing tokens are randomly initialized and Ψ could also be non-linear functions like Softmax. Therefore, applying rotary matrix to Q and K may not properly model the relative positional information.

$$V'_R = \text{Attention}(R, K\mathcal{M}_{\Theta}^{D'}, V) \quad (9)$$

$$V'_O = \Psi(Q\mathcal{M}_{\Theta}^{D'}R^T)\Psi(R(K\mathcal{M}_{\Theta}^{D'})^T)V \approx \Psi(Q\mathcal{M}_{\Theta}^{D'}R^TR(\mathcal{M}_{\Theta}^{D'})^TK^T)V \quad (10)$$

Based on the conclusions, we apply \mathcal{M}_{Θ}^r to the attention scores of Q - R and R - K pairs, as shown in Alg 1. The basic formula of RoRA can then be written as Eq 11, where Ψ here is the Softmax function. In this case, the rotary matrix properly reflect the relative positional differences between elements. Although the corresponding dimensionality is reduced to r , RoRA could take advantages of the nonlinearity of Softmax and the mapping mechanism of R . In addition, we introduce a linear projector as the skip-connection module to further enhance the feature diversity of Q and K .

$$V_O = \Psi(QR^T)\mathcal{M}_{\Theta}^r(\mathcal{M}_{\Theta}^r)^T\Psi(RK^T)^TV + \text{Linear}(V) \quad (11)$$

2) *Wavelet-wise Normalization*: Considering that the wavelet coefficients of different decomposition layers correspond to different frequency intervals, we transpose the output

Algorithm 1 The implementation of multi-head RoRA

Input: Input sequence tokens $I \in \mathbb{R}^{M \times D'}$

Output: Output representations $O \in \mathbb{R}^{M \times D'}$

- 1: $G \in \mathbb{R}^{N \times D'} \leftarrow \sigma(\text{Linear}(I))$
- 2: $R \in \mathbb{R}^{r \times D'} \leftarrow$ Randomly initialized
- 3: **for** $h = 1$ to H **do**
- 4: $Q^h \in \mathbb{R}^{M \times (D'/H)} \leftarrow \text{Linear}(I)$
- 5: $K^h \in \mathbb{R}^{M \times (D'/H)} \leftarrow \text{Linear}(I)$
- 6: $V^h \in \mathbb{R}^{M \times (D'/H)} \leftarrow \text{Linear}(I)$
- 7: $R^h \in \mathbb{R}^{r \times (D'/H)} \leftarrow \text{Linear}(R)$
- 8: $A_{R-K}^h \in \mathbb{R}^{r \times M} \leftarrow (\Psi(RK^T)\mathcal{M}_{\Theta}^r)^T$
- 9: $V_R^h \in \mathbb{R}^{r \times (D'/H)} \leftarrow A_{R-K}^h V^h$
- 10: $A_{Q-R}^h \in \mathbb{R}^{M \times r} \leftarrow \Psi(QR^T)\mathcal{M}_{\Theta}^r$
- 11: $V_O^h \in \mathbb{R}^{M \times (D'/H)} \leftarrow A_{Q-R}^h V_R^h + \text{Linear}(V^h)$
- 12: **end for**
- 13: $O \in \mathbb{R}^{M \times D'} \leftarrow \text{Linear}(\text{Concat}(V_O^1, V_O^2, \dots, V_O^H) \odot G)$
- 14: **return** O

of RoRA to wavelet-wise tokens again and apply wavelet-wise normalization. The process is defined in Eq 12, where RoRA corresponds to Alg 1, $\mathbb{S}^{(n)} \in \mathbb{R}^{M \times ((J+1) \cdot D)}$ is the series-wise tokens, $\mathbb{W}^{(n)} \in \mathbb{R}^{(J+1) \times M \times D}$ is the wavelet-wise tokens and n represents the number of WaveRoRA Encoder layer.

$$\begin{aligned} \mathbb{W}^{(n)} &= \text{Transpose}(\text{RoRA}(\mathbb{S}^{(n)})) \\ \mathbb{S}^{(n+1)} &= \text{Transpose}(\text{WaveNorm}(\mathbb{W}^{(n)})) \end{aligned} \quad (12)$$

E. Multi-scale Wavelet Predictor

In order to reconstruct the predicted series with length H , the length of high-pass and low-pass coefficients of each decomposition layers input to the IDWT should adhere to the same rules as DWT. Specifically, for $\hat{y}_h^{(j)} \in \mathbb{R}^{M \times H^{(j)}}$ and $\hat{y}_l^{(j)} \in \mathbb{R}^{M \times H^{(j)}}$, we set $H^{(j)} = \lfloor \frac{H^{(j-1)} + S - 1}{2} \rfloor$ and $H^{(0)} = H$. To map the embeddings to the specific $H^{(j)}$, we transpose the output of the last RoRA layer to wavelet embeddings $\mathbb{O} \in \mathbb{R}^{(J+1) \times M \times D}$ and pass them through a corresponding MLP individually, defined as Eq 13:

$$\begin{aligned} \hat{\mathcal{C}}_j &= \text{WavePredictor}_j(\mathbb{O}_j) \\ &= (\text{GELU}(\mathbb{O}_j))\hat{W}_j^T + \hat{b}_j^T \end{aligned} \quad (13)$$

where $\hat{\mathcal{C}} = \{\hat{y}_h^{(1)}, \hat{y}_h^{(2)}, \dots, \hat{y}_h^{(J)}, \hat{y}_l^{(J)}\}$ contains the predicted high-pass and low-pass coefficients, $\hat{W}_j \in \mathbb{R}^{H^{(j)} \times D}$ and $\hat{b}_j \in \mathbb{R}^{H^{(j)} \times 1}$ for $j \in \{1, 2, \dots, J\}$, while $\hat{W}_{J+1} \in \mathbb{R}^{H^{(J)} \times D}$ and $\hat{b}_{J+1} \in \mathbb{R}^{H^{(J)} \times 1}$. We set $\hat{\mathcal{C}}$ as the input of IDWT to get the final outputs $X_{out} \in \mathbb{R}^{H \times M}$.

V. EXPERIMENTS

A. Datasets

We evaluate our model on 8 real-world datasets: Weather, Electricity, Traffic, Solar and 4 ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) [12]. These datasets are extensively utilized, covering multiple fields including weather, energy management and transportation. The statistics of the datasets are shown in Table I.

TABLE I: Statistics of all datasets.

Dataset	Weather	Electricity	Traffic	ETTh1	ETTh2	ETTm1	ETTm2	Solar
Variables	21	321	862	7	7	7	7	137
Frequency	10 min	1 hour	1 hour	1 hour	1 hour	15 min	15 min	10 min

TABLE II: Experimental results of MTS time series forecasting task on 8 real-world datasets. The best results are in **bold**.

Models		WaveRoRA		iTransformer		PatchTST		Crossformer		Autoformer		DLinear		TimesNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.159	0.204	0.175	0.216	0.172	0.214	0.158	0.230	0.249	0.329	0.198	0.260	0.172	0.220
	192	0.208	0.250	0.228	0.261	0.218	0.256	0.206	0.277	0.325	0.370	0.240	0.301	0.219	0.261
	336	0.263	0.292	0.282	0.300	0.275	0.297	0.272	0.335	0.351	0.391	0.287	0.340	0.280	0.306
	720	0.345	0.346	0.360	0.350	0.352	0.347	0.398	0.418	0.415	0.426	0.353	0.395	0.365	0.359
Traffic	96	0.364	0.250	0.392	0.263	0.435	0.274	0.522	0.290	0.597	0.371	0.652	0.386	0.593	0.321
	192	0.376	0.258	0.414	0.272	0.444	0.280	0.530	0.293	0.607	0.382	0.601	0.373	0.617	0.336
	336	0.389	0.270	0.430	0.283	0.455	0.285	0.558	0.305	0.623	0.387	0.605	0.373	0.629	0.336
	720	0.425	0.285	0.452	0.297	0.488	0.304	0.589	0.328	0.639	0.395	0.649	0.399	0.640	0.350
Electricity	96	0.136	0.231	0.153	0.245	0.167	0.253	0.219	0.314	0.196	0.313	0.195	0.278	0.168	0.272
	192	0.153	0.247	0.167	0.256	0.175	0.261	0.231	0.322	0.211	0.324	0.195	0.281	0.184	0.289
	336	0.169	0.266	0.181	0.272	0.190	0.277	0.246	0.337	0.214	0.327	0.207	0.296	0.198	0.300
	720	0.196	0.291	0.219	0.305	0.236	0.314	0.280	0.363	0.236	0.342	0.242	0.328	0.220	0.320
ETTh1	96	0.381	0.402	0.394	0.410	0.380	0.399	0.423	0.448	0.435	0.446	0.391	0.403	0.384	0.402
	192	0.425	0.429	0.443	0.437	0.425	0.427	0.471	0.474	0.456	0.457	0.444	0.438	0.436	0.429
	336	0.466	0.447	0.485	0.460	0.464	0.445	0.570	0.546	0.486	0.487	0.480	0.463	0.491	0.469
	720	0.458	0.464	0.504	0.490	0.485	0.469	0.653	0.621	0.515	0.517	0.513	0.508	0.521	0.500
ETTh2	96	0.281	0.337	0.303	0.353	0.293	0.342	0.745	0.584	0.332	0.368	0.345	0.397	0.340	0.374
	192	0.356	0.387	0.385	0.401	0.383	0.396	0.877	0.656	0.426	0.434	0.473	0.472	0.402	0.414
	336	0.394	0.416	0.425	0.433	0.423	0.431	1.043	0.731	0.477	0.479	0.584	0.535	0.452	0.452
	720	0.403	0.433	0.434	0.449	0.424	0.442	1.104	0.763	0.453	0.490	0.795	0.642	0.462	0.468
ETTm1	96	0.320	0.357	0.339	0.374	0.326	0.363	0.404	0.426	0.510	0.492	0.344	0.372	0.338	0.375
	192	0.365	0.383	0.380	0.394	0.362	0.385	0.450	0.451	0.514	0.495	0.383	0.393	0.374	0.387
	336	0.396	0.404	0.417	0.418	0.393	0.409	0.532	0.515	0.521	0.497	0.414	0.416	0.410	0.411
	720	0.463	0.442	0.478	0.452	0.451	0.439	0.666	0.589	0.537	0.513	0.474	0.452	0.478	0.450
ETTm2	96	0.175	0.259	0.189	0.274	0.177	0.261	0.287	0.366	0.205	0.293	0.190	0.287	0.187	0.267
	192	0.241	0.302	0.254	0.313	0.247	0.309	0.414	0.492	0.278	0.336	0.274	0.349	0.249	0.309
	336	0.304	0.342	0.315	0.352	0.312	0.351	0.597	0.542	0.343	0.379	0.376	0.421	0.321	0.351
	720	0.405	0.398	0.413	0.405	0.424	0.416	1.073	1.042	0.414	0.419	0.545	0.519	0.408	0.403
Solar	96	0.197	0.222	0.207	0.241	0.202	0.240	0.310	0.331	0.864	0.707	0.284	0.372	0.250	0.292
	192	0.232	0.254	0.240	0.264	0.240	0.265	0.734	0.725	0.844	0.695	0.320	0.396	0.296	0.318
	336	0.250	0.275	0.251	0.274	0.249	0.276	0.750	0.735	0.751	0.700	0.354	0.421	0.319	0.330
	720	0.247	0.274	0.253	0.277	0.248	0.277	0.769	0.765	0.867	0.710	0.354	0.415	0.338	0.337

B. Baselines

We select 6 baseline methods for the comparative experiments, including 4 Transformer-based models: iTransformer [12], PatchTST [7], Crossformer [14], Auto-former [18]; 1 CNN-based method: TimesNet [9] and 1 MLP-based method: DLinear [36].

C. Experiment Settings

We set a uniform input length $L = 96$ for all models and conduct experiments with $H = \{96, 192, 336, 720\}$. By default, WaveRoRA employs a 4-level DWT using the Symlet3 [22] wavelet basis. We map the wavelet coefficients to embeddings with $D = 64$. For datasets with less variables, such as ETT datasets, we set RoRA layer number N to 2; while for those with more variables like Weather, Electricity and Solar, we set $N = 3$; For Traffic that has large number of variables, we set $N = 4$. For RoRA, we set the token number

of R to $r = \min(10, \frac{\log M + \sqrt{M}}{2})$. The attention heads number is set to 8 by default.

We replicate and run PatchTST, iTransformer and DLinear using the publicly available optimal parameters. For other baseline models, we adhere to their established implementations. We record the MSE and MAE to measure the performance of different models. These metrics are defined in Eq 14 and Eq 15, where Y and \hat{Y} represent ground truth and predicted values respectively. All experiments are conducted on one NVIDIA A800 GPU with 40GB memory to exclude the influence of platforms.

$$\text{MSE}(Y, \hat{Y}) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} |Y_i - \hat{Y}_i|^2 \quad (14)$$

$$\text{MAE}(Y, \hat{Y}) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} |Y_i - \hat{Y}_i| \quad (15)$$

TABLE III: Ablation results of MTS forecasting task on Weather, Traffic and Electricity. Each MSE and MAE score is the average value of $H = \{96, 192, 336, 720\}$. The best results are in **bold**.

Models	w/ SA		w/ LA		w/o Ro		w/o Gate		w/o skip		WaveRoRA		iTransformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	0.400	0.275	0.426	0.280	0.396	0.271	0.405	0.276	0.393	0.270	0.389	0.269	0.422	0.279
Electricity	0.168	0.265	0.174	0.267	0.171	0.266	0.174	0.269	0.168	0.264	0.165	0.261	0.180	0.270
ETTh1	0.454	0.448	0.452	0.446	0.444	0.446	0.450	0.445	0.455	0.447	0.433	0.436	0.457	0.449
ETTh2	0.367	0.402	0.364	0.398	0.365	0.396	0.369	0.398	0.371	0.400	0.359	0.393	0.387	0.409

D. Main Results

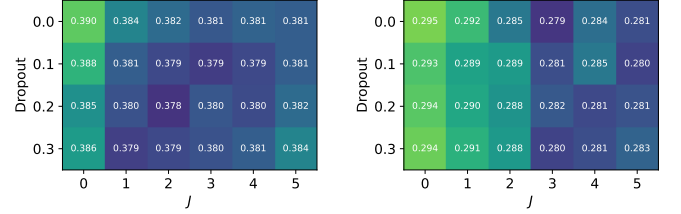
Table II presents all results of MTS forecasting. Compared with the current SOTA Transformer-based model, namely iTransformer, WaveRoRA reduces MSE by 5.91% and MAE by 3.50%. Among all the datasets, WaveRoRA achieves the most improvement on Traffic and Electricity, which contain large amount of variables and exhibit more pronounced periodic characteristics. In particular, the reductions of MSE and MAE come to 7.94% and 4.66% on Traffic and 9.17% and 3.99% on Electricity respectively. Such degree of improvement can be attributed to two aspects: (a) The wavelet transform can accurately model varying periodic patterns over time; (b) The proposed RoRA prunes redundant inter-series dependencies and retain key information from variables with genuine interactions.

For ETT datasets with less relevancy and non-significant periodicity, capturing inter-series dependencies may negatively impact the predicting accuracy. Models adopting the channel-independent strategy usually achieve better results, such as PatchTST and TimeNet. However, WaveRoRA also shows comparable performance on ETTh1 and ETTm1, while achieving more accurate predictions on ETTh2 and ETTm2. Compared to PatchTST, the average improvements of MSE and MAE are 2.36% and 1.39%. This indicates that wavelet transform can effectively preserve time-frequency characteristics while retaining the benefits of time-domain modeling. In addition, we believe that high-quality relative positional information facilitates enhancing interactions between highly correlated variables while weakening those between less correlated ones.

E. Ablation Study

To justify the effectiveness of our designs, we conduct additional experiments on Traffic, Electricity, ETTh1 and ETTh2 and accord the average MSE and MAE. The results are shown in Table III.

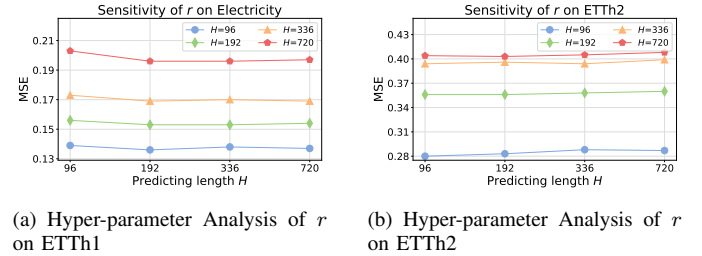
1) *Wavelet Learning Framework*: We replace RoRA with other existing backbones to capture inter-series dependencies, which corresponds to a) w/ SA which uses Softmax attention. Since the wavelet domain embeddings are also generated through an MLP structure, w/ SA can be viewed as replacing the temporal features of iTransformer with wavelet domain features. Compared with iTransformer, w/ SA reduces the MSE by 4.40% and MAE by 1.36% on average, which indicates that the wavelet learning framework could provide better predicting results.



(a) Hyper-parameter Analysis of J and dropout values on ETTh1

(b) Hyper-parameter Analysis of J and dropout values on ETTh2

Fig. 4: The MSE results with prediction length $H = 96$ on ETTh1 and ETTh2. Varying settings of DWT decomposition level J and dropout values are applied. The model's prediction accuracy improves and stabilizes with a larger J , while different dropout values have a minimal effect on performance.



(a) Hyper-parameter Analysis of r on ETTh1

(b) Hyper-parameter Analysis of r on ETTh2

Fig. 5: The MSE results with prediction length $H = \{96, 192, 336, 720\}$ on Electricity and ETTh2.

2) *RoRA structure*: To evaluate the effectiveness of the overall structure of RoRA, we further conduct b) w/ LA which uses linear attention to replace RoRA. SA and LA are typical representatives of the attention family, with various attention mechanisms essentially being derivatives of the two. It can be summarized from Table III that w/ SA and w/ LA achieve suboptimal results compared to WaveRoRA, indicating that RoRA has superior representational capacity.

We then conduct c) w/o Ro which removes the rotary positional embeddings and d) w/o skip which removes the skip connection design to evaluate the components of RoRA. After these modules are removed, the predictive performance of the model decreases to varying degrees. The performance decline is most significant in w/o Gate on the Traffic and Electricity datasets, where the MSE and MAE increase by 5.03% and 3.81% on average. However, for ETTh1 and ETTh2, removing

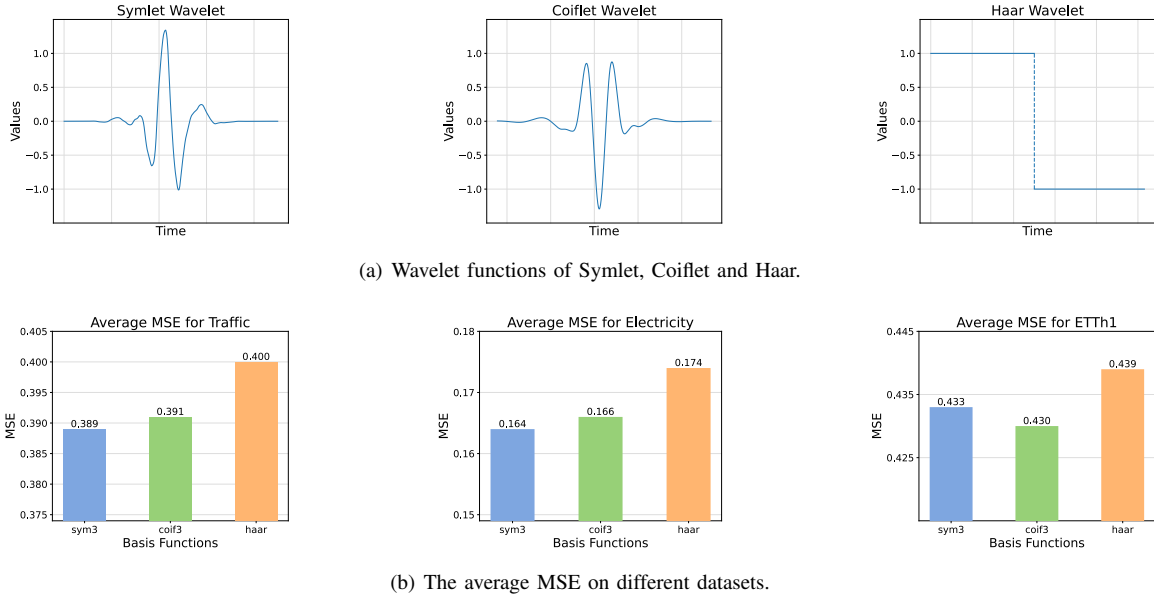


Fig. 6: The figures of (a) specific shape of Symlet, Coiflet and Haar wavelets and (b) average MSE results with prediction length $H = \{96, 192, 336, 720\}$ on Traffic, Electricity and ETTh2.

the skip connection layer causes more deterioration in the prediction accuracy, where the MSE and MAE increase by an average of 4.25% and 1.95%. For datasets with a larger number of variables, the feature diversity carried by series-wise tokens is generally guaranteed, making it beneficial to select high-quality information for more accurate predictions. In contrast, for datasets with fewer variables, enhancing the feature diversity often leads to improved performance.

F. Hyper-parameter Sensitivity Analysis

1) *DWT decomposition levels & Dropout*: DWT hierarchically decomposes a time series into low-pass and high-pass components, corresponding to coarse trends and finer periodical details at different scales. With the increase of decomposition level J , coarser periodic information is separated and the low-pass component will exhibit more significant trend characteristics. To obtain more comprehensive results, we change the value of dropout layers at the same time. Dropout layers improve the generalization ability of the model by randomly masking part of the input. A proper dropout value is conducive to avoiding overfitting. The dropout layer mainly works in each **WaveNorm** layer. We set $J = \{0, 1, 2, 3, 4, 5\}$ and $dropout = \{0.0, 0.1, 0.2, 0.3\}$. Note that when $J = 0$, the model does not make any DWT decomposition and RoRA is applied to time-domain features. We conduct the experiments on ETTh1 and ETTh2 and the final results are shown in Fig 4. Generally, different dropout values do not cause substantial changes of model performance. In contrast, whether to operate DWT to use wavelet-domain features could lead to significant variations in the results. Specifically, when $J = 0$, i.e. only time-domain features are used, the model provides suboptimal results. With the increase of J , the model's prediction accuracy improves and stabilizes. This indicates that key periodic

components is separated and the remaining trend components no longer contain extra periodic information.

2) *Number of routing tokens*: Typically, the routing tokens are kept with $r \ll M$. A larger r could lead to more comprehensive capturing of element-wise correlation but may also cause more information redundancy. To evaluate the effect of r , we manually set $r = \{2, 20, 40, 200\}$ and record the average MSE and MAE on Weather, Electricity and ETTh2. The settings roughly cover the range of variable numbers of these datasets. Although a larger number of router tokens may intuitively aggregate information from the KV matrices more comprehensively, the results in Fig 5 provide limited support for this assumption. For Electricity, the best r is 10 or 20, while $r = 2$ yields best performance on ETTh2. This may be due to the potentially sparse dependencies between variables. Specifically, while complex associations may exist among multivariate data, the patterns of the correlations are likely to be similar. Therefore, a small number of routing tokens could also filter out the redundant information, enhancing the model to focus on more critical inter-series dependencies. Nonetheless, the model performance the overall model performance does not exhibit substantial fluctuations with varying r , which demonstrates the robustness of our proposed model.

G. Effect of Varying Wavelet Basis

Unlike the Fourier transform, the results of wavelet transform vary greatly depending on the chosen wavelet function ψ and scale function ϕ . Therefore, to explore how different wavelet bases impact model prediction performance across various types of datasets, we selected three wavelet bases: Haar wavelet, Symlet wavelet and Coiflet wavelet [22]. The Haar wavelet, one of the earliest orthogonal wavelet functions with compact support, is the simplest wavelet function used in wavelet analysis. While computationally efficient, it has

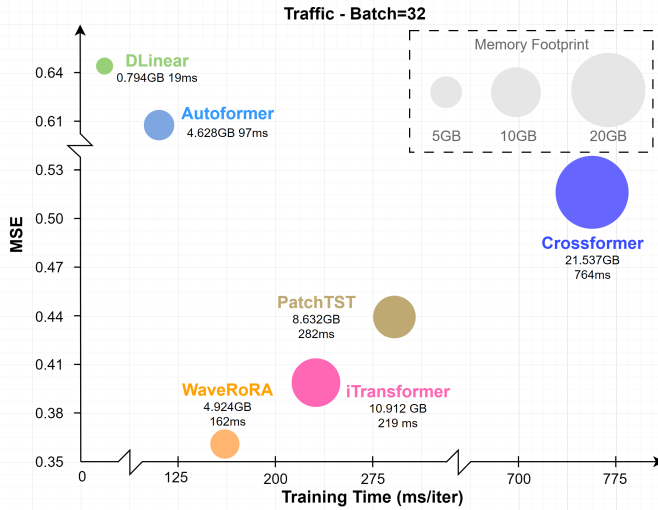


Fig. 7: The model efficiency comparison. All the models follow the hyper-parameter settings in section V-C and the batch size is set to 32.

limited performance as a wavelet basis due to its discontinuity in the time domain. The Symlet wavelet is a nearly symmetric wavelet function that improves the Daubechies function. It reduces phase distortion to some extent during signal analysis and reconstruction. Generally, the Symlet wavelet is more beneficial for signals with stronger periodic characteristics. In this paper, we primarily use the Sym3 wavelet for DWT. The Coiflet wavelet, which is known for its higher symmetry, could simultaneously maintain high vanishing moments on its wavelet function and scale function. It usually performs well in handling signals with abrupt features. In this experiment, we use Coif3 to evaluate the effectiveness of Coiflet wavelet. The results can be seen in Fig 6. For Traffic and Electricity that exhibit more pronounced periodic characteristics, the Sym3 wavelet leads to the best predicting performance, while for ETTh1, the best wavelet function changes to Coif3. This indicates that different wavelet functions can provide tokens with varying quality when handling data with different characteristics.

H. Model Efficiency

As discussed in section IV-D1, RoRA has linear complexity $O(MrD)$ relative to the variable number and the token dimension. To visually evaluate the efficiency of WaveRoRA and other baseline models, we set $r = 20$ and batch size $B = 32$ on Traffic and record the training speed and GPU memory usage for WaveRoRA, iTransformer, PatchTST, Autoformer, Crossformer and DLinear. The results are shown in Fig 7. WaveRoRA achieves a good balance among predicting performance, training speed and memory usage.

VI. CONCLUSION

In this paper, we investigate to extract the underlying patterns of MTS data in the wavelet domain. We propose a wavelet learning framework to map the input MTS to wavelet embeddings and capture the inter-series dependencies with a

novelty designed RoRA mechanism. The embeddings in the wavelet domain could leverage the strengths of both time and frequency domains, modeling the trends and periodic changing patterns of the time series. What's more, the proposed RoRA introduces a small number of routing tokens to aggregate information from the KV matrices and redistribute it to the Q matrix. Such designs retain the powerful expressivity of Softmax and achieve linear complexity relative to both sequence length and token dimension. We conduct extensive experiments and WaveRoRA achieves superior performance compared to SOTA methods. Currently, our wavelet learning framework simply fits for models capturing inter-series dependencies. What's more, different types of wavelet functions exhibit potentials for processing MTS data with varying patterns. Future researches will be done to (a) explore the combination of wavelet domain features with other types of backbones and (b) find an automatic approach to adaptively select the best wavelet function for specific data. In addition, we are going to try applying WaveRoRA on more kinds of applications such as Large Time Series Model [40] and AIOPS [38].

REFERENCES

- [1] G. Zhang, D. Yang, G. Galanis, and E. Androulakis, "Solar forecasting with hourly updated numerical weather prediction," *Renewable and Sustainable Energy Reviews*, vol. 154, p. 111768, 2022.
- [2] N. Uremović, M. Bizjak, P. Sukić, G. Štumberger, B. Žalik, and N. Lukač, "A new framework for multivariate time series forecasting in energy management system," *IEEE Transactions on Smart Grid*, 2022.
- [3] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 922–929, 2019.
- [4] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [6] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 11106–11115, 2021.
- [7] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, 2023.
- [8] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.
- [9] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *International Conference on Learning Representations*, 2023.
- [10] D. Liu, J. Wang, S. Shang, and P. Han, "Msdr: Multi-step dependency relation networks for spatial temporal forecasting," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1042–1050, 2022.
- [11] Z. Zhang, L. Meng, and Y. Gu, "Sageformer: Series-aware framework for long-term multivariate time series forecasting," *IEEE Internet of Things Journal*, 2024.
- [12] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *The Twelfth International Conference on Learning Representations*, 2023.
- [13] Y. Jia, Y. Lin, X. Hao, Y. Lin, S. Guo, and H. Wan, "Witrans: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [14] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

- [15] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [16] Z. Liu, Q. Ma, P. Ma, and L. Wang, "Temporal-frequency co-training for time series semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 8923–8931, 2023.
- [17] J. W. Gibbs, "Fourier's series," *Nature*, vol. 59, no. 1539, pp. 606–606, 1899.
- [18] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22419–22430, 2021.
- [19] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*, pp. 27268–27286, PMLR, 2022.
- [20] F. Yang, X. Li, M. Wang, H. Zang, W. Pang, and M. Wang, "Waveform: Graph enhanced wavelet learning for long sequence forecasting of multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 10754–10761, 2023.
- [21] M. Li, Y. Liu, S. Zhi, T. Wang, and F. Chu, "Short-time fourier transform using odd symmetric window function," *Journal of Dynamics, Monitoring and Diagnostics*, vol. 1, no. 1, pp. 37–45, 2022.
- [22] I. Daubechies, "Ten lectures on wavelets," *Society for industrial and applied mathematics*, 1992.
- [23] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE transactions on information theory*, vol. 36, no. 5, pp. 961–1005, 1990.
- [24] F. Cotter, *Uses of Complex Wavelets in Deep Convolutional Neural Networks*. PhD thesis, Apollo - University of Cambridge Repository, 2019.
- [25] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.
- [26] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.
- [27] Y. Bai, J. Wang, X. Zhang, X. Miao, and Y. Lin, "Crossfun: Multi-view joint cross fusion network for time series anomaly detection," *IEEE Transactions on Instrumentation and Measurement*, 2023.
- [28] X. Zhang, X. Jin, K. Gopalswamy, G. Gupta, Y. Park, X. Shi, H. Wang, D. C. Maddix, and Y. Wang, "First de-trend then attend: Rethinking attention for time-series forecasting," *arXiv preprint arXiv:2212.08151*, 2022.
- [29] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.
- [30] M. Jiang, P. Zeng, K. Wang, H. Liu, W. Chen, and H. Liu, "Fecam: Frequency enhanced channel attention mechanism for time series forecasting," *Advanced Engineering Informatics*, vol. 58, p. 102158, 2023.
- [31] J. Wan, N. Xia, Y. Yin, X. Pan, J. Hu, and J. Yi, "Tcdformer: A transformer framework for non-stationary time series forecasting based on trend and change-point detection," *Neural Networks*, vol. 173, p. 106196, 2024.
- [32] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning*, pp. 5156–5165, PMLR, 2020.
- [33] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.
- [34] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021.
- [35] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.
- [36] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 11121–11128, 2023.
- [37] S. Elfving, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural networks*, vol. 107, pp. 3–11, 2018.
- [38] J. Diaz-De-Arcaya, A. I. Torre-Bastida, G. Zárate, R. Miñón, and A. Almeida, "A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–30, 2023.
- [39] D. Han, X. Pan, Y. Han, S. Song, and G. Huang, "Flatten transformer: Vision transformer using focused linear attention," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5961–5971, 2023.
- [40] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: Transformers for time series analysis at scale," *arXiv preprint arXiv:2402.02368*, 2024.