FreEformer: Frequency Enhanced Transformer for Multivariate Time Series Forecasting

Wenzhen Yue[1], **Yong Liu**[2], **Xianghua Ying**[1*], **Bowei Xing**[1], **Ruohao Guo**[1] and **Ji Shi**[1]
[1]State Key Laboratory of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University
[2]School of Software, BNRist, Tsinghua University
yuewenzhen@stu.pku.edu.cn, liuyong21@mails.tsinghua.edu.cn, xhying@pku.edu.cn

Abstract

This paper presents **FreEformer**, a simple yet effective model that leverages a **Fre**quency **E**nhanced Trans**former** for multivariate time series forecasting. Our work is based on the assumption that the frequency spectrum provides a global perspective on the composition of series across various frequencies and is highly suitable for robust representation learning. Specifically, we first convert time series into the complex frequency domain using the Discrete Fourier Transform (DFT). The Transformer architecture is then applied to the frequency spectra to capture cross-variate dependencies, with the real and imaginary parts processed independently. However, we observe that the vanilla attention matrix exhibits a low-rank characteristic, thus limiting representation diversity. To address this, we enhance the vanilla attention mechanism by introducing an additional learnable matrix to the original attention matrix, followed by row-wise L1 normalization. Theoretical analysis demonstrates that this enhanced attention mechanism improves both feature diversity and gradient flow. Extensive experiments demonstrate that FreEformer consistently outperforms state-of-the-art models on eighteen real-world benchmarks covering electricity, traffic, weather, healthcare and finance. Notably, the enhanced attention mechanism also consistently improves the performance of state-of-the-art Transformer-based forecasters. Code is available at https://anonymous.4open.science/r/FreEformer.

1 Introduction

Multivariate time series forecasting holds significant importance in real-world domains such as weather [Wu *et al.*, 2023b], energy [Zhou *et al.*, 2021], transportation [He *et al.*, 2022] and finance [Chen *et al.*, 2023]. In recent years, various deep learning models have been proposed, significantly pushing the performance boundaries. Among these models, Recurrent Neural Networks (RNN) [Salinas *et al.*, 2020], Convolutional Neural Networks (CNN) [Bai *et al.*, 2018; Wu *et al.*, 2023a], LLM [Zhou *et al.*, 2023; Jin *et al.*, 2021], Multi-Layer Perceptrons (MLP) [Zeng *et al.*, 2023; Xu *et al.*, 2023], Transformers-based methods [Nie *et al.*, 2023; Liu *et al.*, 2024a; Wang *et al.*, 2024c] have demonstrated great potential due to their strong representation capabilities.

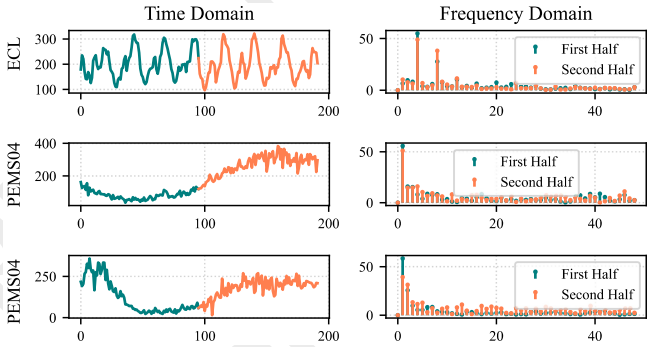

Figure 1: Time series and their corresponding frequency spectra. The series are normalized before applying the DFT, and the amplitudes of the frequency spectra are plotted. (1) The frequency spectra often exhibit strong consistency across adjacent temporal spans within the same time series, forming the basis for frequency-based forecasting. (2) Strong correlations between the two variables in PEMS04 (rows 2 and 3) are observed, suggesting that exploring such multivariate relationships could lead to more robust representations. (3) The frequency spectrum usually exhibits sparsity, with a few dominant frequencies.

In recent years, frequency-domain-based models have been proposed and have achieved great performance [Yi *et al.*, 2024c; Xu *et al.*, 2023], benefiting from the robust frequency domain modeling. As shown in Figure 1, frequency spectra exhibit strong consistency across different spans of the same series, making them suitable for forecasting. Most existing frequency-domain-based works [Yi *et al.*, 2024a] rely on linear layers to learn frequency-domain representations, resulting in a performance gap. Frequency-domain Transformer-based models remain under-explored. Recently, Fredformer [Piao *et al.*, 2024] applies the vanilla Transformer to patched frequency tokens to address the frequency bias issue. However, the patching technique introduces additional hyper-parameters and undermines the inherent global perspective [Yi *et al.*, 2024c] of frequency-domain modeling.

In this paper, we adopt a simple yet effective approach

*Corresponding Author

by applying the Transformer to frequency-domain variate tokens for representation learning. Specifically, we embed the entire frequency spectrum as variate tokens and capture cross-variate dependencies among them. This architecture offers four main advantages: 1) As shown in Section 3, simple frequency-domain operations can correspond to complex temporal operations [Yi *et al.*, 2024c]; 2) Multivariate correlations typically exists (Figure 1), and learning these dependencies facilitates forecasting; 3) Minimal correlations among frequency points [Wang *et al.*, 2024a] hinder the efficacy of cross-frequency dependency learning (Table 5); 4) The permutation invariance of the attention mechanism aligns naturally with the order insensitivity of variates.

Furthermore, we observe that for the frequency-domain representation, the attention matrix of vanilla attention often exhibits a low-rank characteristic, which reduces the diversity of representations. To address this issue, we propose a general solution: adding a learnable matrix to the original softmax attention matrix, followed by row-wise normalization. We term this approach **enhanced attention** and name the overall model **FreEformer**. Despite its simplicity, the enhanced attention mechanism is proven effective both theoretically and empirically. The main contributions of this work are summarized as follows:

- This paper presents a simple yet effective model, named FreEformer, for multivariate time series forecasting. FreEformer achieves robust cross-variate representation learning using the enhanced attention mechanism.
- Theoretical analysis and experimental results demonstrate that the enhanced attention mechanism increases the rank of the attention matrix and provides greater flexibility for gradient flow. As a plug-in module, it consistently enhances the performance of existing Transformer-based forecasters.
- Empirically, FreEformer consistently achieves state-of-the-art forecasting performance across 18 real-world benchmarks spanning diverse domains such as electricity, transportation, weather, healthcare and finance.

2 Related Works

2.1 Transformer-Based Forecasters

Classic works such as Autoformer [Wu *et al.*, 2021], Informer [Zhou *et al.*, 2021], Pyraformer [Liu *et al.*, 2022a], FEDformer [Zhou *et al.*, 2022b], and PatchTST [Nie *et al.*, 2023] represent early Transformer-based time series forecasters. iTransformer [Liu *et al.*, 2024a] introduces the inverted Transformer to capture multivariate dependencies, and achieves accurate forecasts. More recently, research has focused on jointly modeling cross-time and cross-variate dependencies [Zhang and Yan, 2023; Wang *et al.*, 2024c; Han *et al.*, 2024]. Leddam [Yu *et al.*, 2024] uses a dual-attention module for decomposed seasonal components and linear layers for trend components. Unlike previous models in the time domain, we shift our focus to the frequency domain to explore dependencies among the frequency spectra of multiple variables for more robust representations.

2.2 Frequency-Domain Forecasters

Frequency analysis is an important tool in time series forecasting [Yi *et al.*, 2023]. FEDformer [Zhou *et al.*, 2022b] performs DFT and sampling prior to Transformer. DEPTS [Fan *et al.*, 2022] uses the DFT to capture periodic patterns for better forecasts. FiLM [Zhou *et al.*, 2022a] applies Fourier analysis to preserve historical information while mitigating noise. FreTS [Yi *et al.*, 2024c] employs frequency-domain MLPs to model channel and temporal dependencies. FourierGNN [Yi *et al.*, 2024b] transfers GNN operations from the time domain to the frequency domain. FITS [Xu *et al.*, 2023] applies a low-pass filter and complex-valued linear projection in the frequency domain. DERITS [Fan *et al.*, 2024] introduces a Fourier derivative operator to address non-stationarity. Fredformer [Piao *et al.*, 2024] addresses frequency bias by dividing the frequency spectrum into patches. FAN [Ye *et al.*, 2024] introduces frequency adaptive normalization for non-stationary data. In this work, we adopt a simple yet effective Transformer-based model to capture multivariate correlations in the frequency domain, outperforming existing methods.

2.3 Transformer Variants

Numerous variants of the vanilla Transformer have been developed to enhance efficiency and performance. Informer [Zhou *et al.*, 2021] introduces a ProbSparse self-attention mechanism with $O(N\log N)$ complexity. Flowformer [Wu *et al.*, 2022] proposes Flow-Attention, achieving linear complexity based on flow network theory. Reformer [Kitaev *et al.*, 2020] reduces complexity by replacing dot-product attention with locality-sensitive hashing. Linear Transformers, such as FLatten [Han *et al.*, 2023] and LSoftmax [Yue *et al.*, 2024], achieve linear complexity by precomputing $\mathbf{K}^T\mathbf{V}$ and designing various mapping functions. FlashAttention [Dao *et al.*, 2022] accelerates computations by tiling to minimize GPU memory operations. LASER [Surya Duvvuri and Dhillon, 2024] mitigates the gradient vanishing issue using exponential transformations. In this work, we focus on the low-rank issue and adopt a simple yet effective strategy by adding a learnable matrix to the attention matrix. This improves both matrix rank and gradient flow with minimal modifications to the vanilla attention mechanism.

3 Preliminaries

The discrete Fourier transform (DFT) [Palani, 2022] converts a signal $\mathbf{x} \in \mathbb{R}^N$ into its frequency spectrum $\mathcal{F} \in \mathbb{C}^N$. For $k = 0, 1, \cdots, N-1$, we have

$$\mathcal{F}[k] = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}nk}\mathbf{x}[n]. \tag{1}$$

Here, j denotes the imaginary unit. For a real-valued vector $\mathbf{x}$, $\mathcal{F}[k]$ is complex-valued and satisfies the property of Hermitian symmetry [Palani, 2022]: $\mathcal{F}[k] = (\mathcal{F}[N-k])^*$ for $k = 1, \cdots, N-1$, where $(\cdot)^*$ denotes the complex conjugate. The DFT is a linear and reversible transform, with the inverse discrete Fourier transform (IDFT) being:

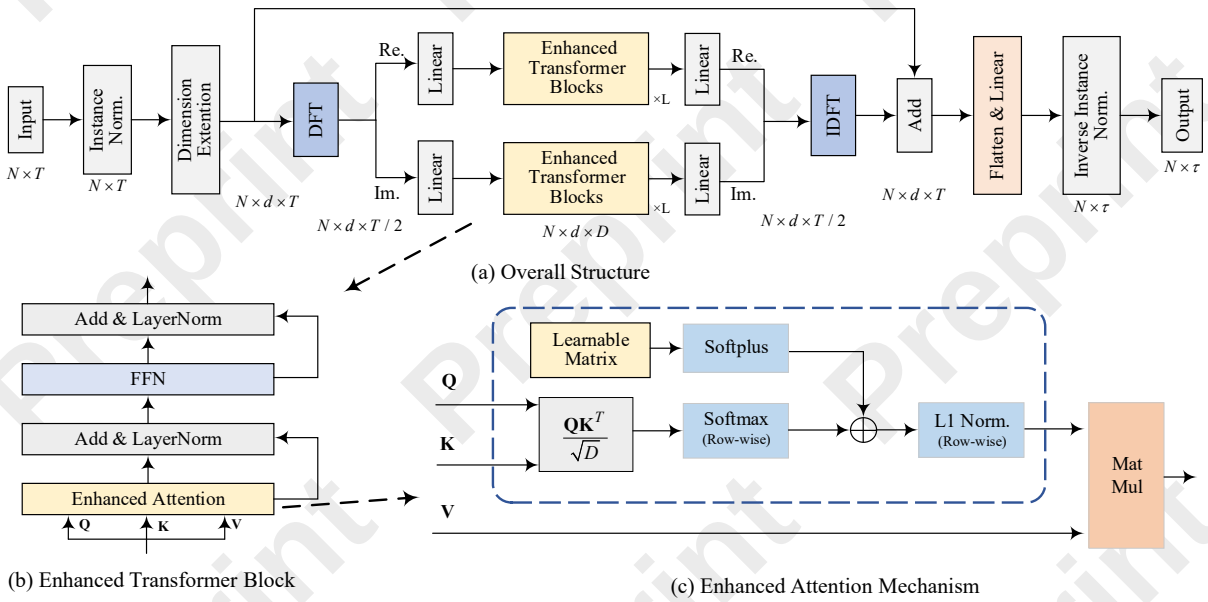

Figure 2: Overall structure of the FreEformer. We leverage the frequency spectrum to capture temporal patterns and employ an enhanced Transformer to model dependencies among multivariate spectra. The enhanced Transformer introduces a learnable matrix to the attention mechanism, which, as shown through theoretical analysis, addresses potential low-rank issues and improves gradient flow.

$$\mathbf{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{F}[k] \cdot e^{j\cdot 2\pi \frac{k}{N} n},\ k = 0, 1, \cdots, N-1. \quad (2)$$

Linear projections in the frequency domain are widely employed in works such as FreTS [Yi *et al.*, 2024c] and FITS [Xu *et al.*, 2023]. The following theorem establishes their equivalent operations in the time domain.

Theorem 1 (Frequency-domain linear projection and time–domain convolutions)**.** *Given the time series $\mathbf{x} \in \mathbb{R}^N$ and its corresponding frequency spectrum $\mathcal{F} \in \mathbb{C}^N$. Let $\mathbf{W} \in \mathbb{C}^{N\times N}$ denote a weight matrix and $\mathbf{b} \in \mathbb{C}^N$ a bias vector. Under these definitions, the following DFT pair holds:*

$$\tilde{\mathcal{F}} = \mathbf{W}\mathcal{F} + \mathbf{b} \leftrightarrow \sum_{i=0}^{N-1} \Omega_i \circledast \mathcal{M}_i(\mathbf{x}) + \mathrm{IDFT}(\mathbf{b}), \quad (3)$$

where

$$\begin{aligned} w_i &= [\mathrm{diag}(\mathbf{W}, i), \mathrm{diag}(\mathbf{W}, i-N)] \in \mathbb{C}^N, \\ \Omega_i &= \mathrm{IDFT}\left(w_i\right) \in \mathbb{C}^N, \\ \mathcal{M}_i(\mathbf{x}) &= \mathbf{x} \odot \left[\mathrm{e}^{-j\frac{2\pi}{N} ik}\right]_{k=0,1,\cdots N-1} \in \mathbb{C}^N. \end{aligned} \quad (4)$$

Here, $\circledast$ denotes the circular convolution, and $\odot$ represents the Hadamard (element-wise) product. The notation $[\cdot,\cdot]$ indicates the concatenation of two vectors. $\mathrm{diag}(\mathbf{W}, i) \in \mathbb{C}^{N-|i|}$ *extracts the i-th diagonal of $\mathbf{W}$. $\mathcal{M}_i(\mathbf{x})$ represents the i-th modulated version of $\mathbf{x}$, with $\mathcal{M}_0(\mathbf{x})$ being $\mathbf{x}$ itself.*

We provide the proof of this theorem in Section A of the appendix. **Theorem 1** extends Theorem 2 from FreTS [Yi *et al.*, 2024c] and demonstrates that a linear transformation in the frequency domain is equivalent to the sum of circular convolution operations applied to the series and its modulated versions. This equivalence highlights the computational simplicity of performing such operations in the frequency domain compared to the time domain.

4 Method

In multivariate time series forecasting, we consider historical series within a lookback window of T, each timestamp with N variates: $\mathbf{x} = \{\mathrm{x}_1, \cdots, \mathrm{x}_T\} \in \mathbb{R}^{N\times T}$. Our task is to predict future τ timestamps to closely approximate the ground truth $\mathbf{y} = \{\mathrm{x}_{T+1}, \cdots, \mathrm{x}_{T+\tau}\} \in \mathbb{R}^{N\times\tau}$.

4.1 Overall Architecture

As shown in Figure 2, FreEformer employs a simple architecture. First, an instance normalization layer, specifically RevIN [Kim *et al.*, 2021], is used to normalize the input data and de-normalize the results at the final stage to mitigate non-stationarity. The constant mean component, represented by the zero-frequency point in the frequency domain, is set to zero during normalization. Subsequently, a dimension extension module is employed to enhance the model's representation capabilities. Specifically, the input $\mathbf{x}$ is expanded by a learnable weight vector $\phi_d \in \mathbb{R}^d$, yielding higher-dimensional and more expressive series data: $\bar{\mathbf{x}} = \mathbf{x} \times \phi_d \in \mathbb{R}^{N\times d\times T}$. We refer to d as the embedding dimension.

Frequency-Domain Operations Next, we apply the Discrete Fourier Transform (DFT) to convert the time series $\bar{\mathbf{x}}$ into its frequency spectrum along the temporal dimension:

$$\mathcal{F} = \mathrm{DFT}(\bar{\mathbf{x}}) = \mathrm{Re}(\mathcal{F}) + j \cdot \mathrm{Im}(\mathcal{F}) \in \mathbb{C}^{N\times d\times T}, \quad (5)$$

Dataset	ECL	Weather	Traffic	COVID-19	NASDAQ	COVID-19	NASDAQ
$T-\tau$	96-{96,192,336,720}			36-{24,36,48,60}		12-{3,6,9,12}	
Concat.	**0.162**	0.243	0.443	8.705	0.190	1.928	**0.055**
S.W.	0.165	0.240	0.440	8.520	0.189	1.895	**0.055**
N.S.W.	**0.162**	**0.239**	**0.435**	**8.435**	**0.185**	**1.892**	**0.055**

Table 1: Comparison of different processes for real and imaginary parts. Average MSEs are reported in this table. 'S.W.' and 'N.S.W.' denote 'Shared Weights' and 'Non-Shared Weights', respectively. 'Concat.' denotes the concatenation method.

where $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ represent the real and imaginary parts, respectively. Due to the conjugate symmetry property of the frequency spectrum of a real-valued signal, only the first $\lceil (T+1)/2 \rceil$ elements of the real and imaginary parts need to be retained. Here, $\lceil \cdot \rceil$ denotes the ceiling operation.

To process the real and imaginary parts, common strategies include employing complex-valued layers [Yi *et al.*, 2024c; Xu *et al.*, 2023], or concatenating the real and imaginary parts into a real-valued vector and subsequently projecting the results back [Piao *et al.*, 2024]. In this work, we adopt a simple yet effective scheme: processing these two parts independently. As shown in Table 1, this dual-branch scheme yields better performance for FreEformer.

After flattening the last two dimensions of the real and imaginary parts and projecting them into the hidden dimension D, we construct the frequency-domain variate tokens $\tilde{\mathrm{Re}}, \tilde{\mathrm{Im}} \in \mathbb{R}^{N \times D}$. These tokens are then fed into L stacked Transformer blocks to capture multivariate dependencies among the spectra. Subsequently, the tokens are projected back to the lookback length. The real and imaginary parts are then regrouped to reconstruct the frequency spectrum. Then, the time-domain signal $\tilde{\mathbf{x}}$ is recovered using the IDFT. The entire process is summarized as follows:

$$
\begin{aligned}
&\tilde{\mathcal{F}} = \mathcal{F}[:,:,0:\lceil (T+1)/2 \rceil] \in \mathbb{C}^{N \times d \times \lceil (T+1)/2 \rceil}, \\
&\tilde{\mathrm{Re}}^0 = \mathrm{Linear}(\mathrm{Flatten}(\mathrm{Re}(\tilde{\mathcal{F}}))) \in \mathbb{R}^{N \times D} \\
&\tilde{\mathrm{Re}}^{l+1} = \mathrm{TrmBlock}(\tilde{\mathrm{Re}}^l) \in \mathbb{R}^{N \times D}, l = 0, \cdots, L-1, \\
&\tilde{\mathrm{Re}} = \mathrm{Linear}(\mathrm{Reshape}(\tilde{\mathrm{Re}}^L)) \in \mathbb{R}^{N \times d \times \lceil (T+1)/2 \rceil}, \\
&\tilde{\mathrm{Im}}^0 = \mathrm{Linear}(\mathrm{Flatten}(\mathrm{Im}(\tilde{\mathcal{F}}))) \in \mathbb{R}^{N \times D} \\
&\tilde{\mathrm{Im}}^{l+1} = \mathrm{TrmBlock}(\tilde{\mathrm{Im}}^l) \in \mathbb{R}^{N \times D}, l = 0, \cdots, L-1, \\
&\tilde{\mathrm{Im}} = \mathrm{Linear}(\mathrm{Reshape}(\tilde{\mathrm{Im}}^L)) \in \mathbb{R}^{N \times d \times \lceil (T+1)/2 \rceil}, \\
&\tilde{\mathbf{x}} = \mathrm{IDFT}(\tilde{\mathrm{Re}} + j \cdot \tilde{\mathrm{Im}}) \in \mathbb{R}^{N \times d \times T}.
\end{aligned} \tag{6}
$$

In the above equation, the final step is implemented via the irfft function in PyTorch to ensure real-valued outputs.

Prediction Head A shortcut connection is applied to sum $\tilde{\mathbf{x}}$ with the original $\bar{\mathbf{x}}$. Finally, a flatten layer and a linear head are used to ensure the output matches the desired size. The final result is obtained through a de-normalization step:

$$
\hat{\mathbf{y}} = \mathrm{DeNorm}\left(\mathrm{Linear}(\mathrm{Flatten}(\tilde{\mathbf{x}} + \bar{\mathbf{x}}))\right) \in \mathbb{R}^{N \times \tau}. \tag{7}
$$

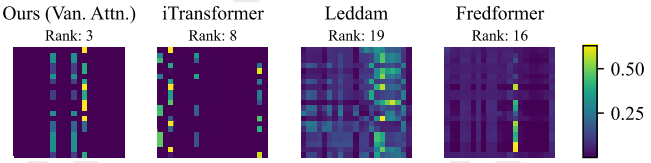

Figure 3: Attention matrices from state-of-the-art forecasters on the Weather dataset. The FreEformer with vanilla attention typically exhibits a low rank, likely due to the inherent sparsity of the frequency spectrum and the strong-value-focused nature of the Softmax function in vanilla attention.

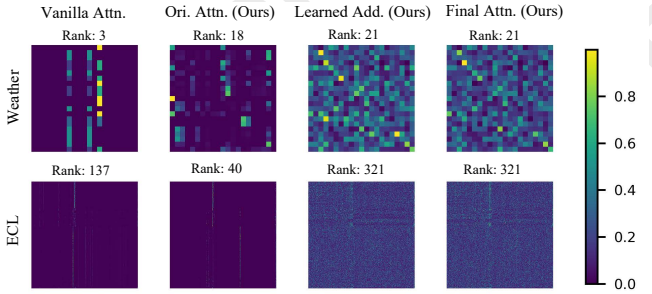

Figure 4: Attention matrices from vanilla and enhanced attention. The left column shows the low-rank attention matrix from the vanilla attention (Weather: 3, ECL: 137), with most entries near zero. The right three columns show the original attention matrix ($\mathbf{A}$), the learned addition matrix (Softplus($\mathbf{B}$)), and the final attention matrix (Norm ($\mathbf{A}$ + Softplus($\mathbf{B}$))). The final matrix exhibits more prominent values and higher ranks (Weather: 21, ECL: 321).

4.2 Enhanced Attention

In the Transformer block, as shown in Figure 2(b), we first employ the attention mechanism to capture cross-variate dependencies. Then the LayerNorm and FFN are used to update frequency representations in a variate-independent manner. According to Theorem 1, the FFN corresponds to a series of convolution operations in the time domain for series representations. The vanilla attention mechanism is defined as:

$$
\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \underbrace{\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}\right)}_{\text{Attention Matrix} \triangleq \mathbf{A}} \mathbf{V}. \tag{8}
$$

Here, $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times D}$ are the query, key and value matrix, respectively, obtained through linear projections. We denote D as the feature dimension and refer to Softmax $\left(\mathbf{Q}\mathbf{K}^T/\sqrt{D}\right)$, represented as $\mathbf{A}$, as the attention matrix.

However, as shown in Figure 3, compared to other state-of-the-art forecasters, FreEformer with the vanilla attention mechanism usually exhibits an attention matrix with a lower rank. This could arise from the inherent sparsity of the frequency spectrum [Palani, 2022] and the strong-value-focused properties of the vanilla attention mechanism [Surya Duvvuri and Dhillon, 2024; Xiong *et al.*, 2021]. While patching adjacent frequency points can mitigate sparsity (as in Fredformer), we address the underlying low-rank issue within the attention mechanism itself, offering a more general solution.

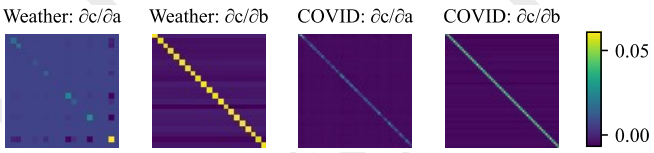

Figure 5: Illustration of the Jacobian matrices of $\mathbf{c}$ with respect to $\mathbf{a}$ and $\mathbf{b}$ for the Weather and COVID-19 datasets.

In this work, we adopt a straightforward yet effective solution: introducing a learnable matrix $\mathbf{B}$ to the attention matrix. The enhanced attention mechanism, denoted as EnhAttn $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, is defined as [1]:

$$\text{Norm}\left(\text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}\right) + \text{Softplus}(\mathbf{B})\right)\mathbf{V}, \tag{9}$$

where $\text{Norm}(\cdot)$ denotes row-wise L1 normalization. The $\text{Softplus}(\cdot)$ function ensures positive entries, thereby preventing potential division-by-zero errors in $\text{Norm}(\cdot)$.

Theoretical Analysis

Feature Diversity According to Equation (9), feature diversity is directly influenced by the rank of the final attention matrix $\text{Norm}\left(\mathbf{A} + \bar{\mathbf{B}}\right)$, where $\bar{\mathbf{B}} \triangleq \text{Softplus}(\mathbf{B})$. Since row-wise L1 normalization does not alter the rank of a matrix, we have: $\text{rank}\left(\text{Norm}\left(\mathbf{A} + \bar{\mathbf{B}}\right)\right) = \text{rank}\left(\mathbf{A} + \bar{\mathbf{B}}\right)$. For further analysis, we present the following theorem:

Theorem 2. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two matrices of the same size* $N \times N$*. The rank of their sum satisfies the following bounds:*

$$|\text{rank}(\mathbf{A}) - \text{rank}(\mathbf{B})| \leq \text{rank}(\mathbf{A} + \mathbf{B}) \leq \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B}) \tag{10}$$

The proof is provided in Section B of the appendix. As illustrated in Figure 4, the original attention matrix $\mathbf{A}$ often exhibits a low rank, whereas the learned matrix $\bar{\mathbf{B}}$ is nearly full-rank. According to Theorem 2, the combined matrix $\mathbf{A} + \bar{\mathbf{B}}$ generally achieves a higher rank. This observation aligns with the results shown in Figure 4.

Gradient Flow Let $\mathbf{a} \in \mathbb{R}^N$ denote a row in $\mathbf{Q}\mathbf{K}^T/\sqrt{D}$. For vanilla attention, the transformation is $\tilde{\mathbf{a}} \triangleq \text{Softmax}(\mathbf{a})$. Then the Jacobian matrix of $\tilde{\mathbf{a}}$ regarding $\mathbf{a}$ can be derived as:

$$\frac{\partial \tilde{\mathbf{a}}}{\partial \mathbf{a}} = \text{Diag}(\tilde{\mathbf{a}}) - \tilde{\mathbf{a}}\tilde{\mathbf{a}}^T, \tag{11}$$

where $\text{Diag}(\tilde{\mathbf{a}})$ is a diagonal matrix with $\tilde{\mathbf{a}}$ as its diagonal.

For the enhanced attention, the transformation is given by:

$$\mathbf{c} = \text{Norm}\left(\text{Softmax}(\mathbf{a}) + \mathbf{b}\right), \tag{12}$$

where $\mathbf{b}$ represents a row of $\text{Softplus}(\mathbf{B})$. The Jacobian matrices of $\mathbf{c}$ with respect to $\mathbf{a}$ and $\mathbf{b}$ can be derived as:

$$\begin{aligned}\frac{\partial \mathbf{c}}{\partial \mathbf{a}} &= \frac{1}{\left\|\tilde{\mathbf{b}}\right\|_1}\left(\text{Diag}(\tilde{\mathbf{a}}) - \tilde{\mathbf{a}}\tilde{\mathbf{a}}^T\right), \\ \frac{\partial \mathbf{c}}{\partial \mathbf{b}} &= \frac{1}{\left\|\tilde{\mathbf{b}}\right\|_1^2}\left(\left\|\tilde{\mathbf{b}}\right\|_1 \cdot \mathbf{I} - \tilde{\mathbf{b}}\mathbf{1}^T\right),\end{aligned} \tag{13}$$

[1] The variants are discussed in Section C.3 of the appendix.

where $\tilde{\mathbf{b}} = \tilde{\mathbf{a}} + \mathbf{b} = \text{Softmax}(\mathbf{a}) + \mathbf{b}$, $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the unit matrix; $\mathbf{1} = [1, \cdots, 1]^T \in \mathbb{R}^N$. The detailed proofs of Equations (11) and (13) are provided in Section C of the appendix.

We can see that $\partial \mathbf{c}/\partial \mathbf{a}$ in Equation (13) shares the same structure as that of vanilla attention in Equation (11), except for the scaling factor $1/\left\|\tilde{\mathbf{b}}\right\|_1$. Since $\left\|\tilde{\mathbf{b}}\right\|_1 = 1 + \|\mathbf{b}\|_1 > 1$ and $\tilde{\mathbf{b}}$ is learnable, the gradient is scaled down by a learnable factor, providing extra flexibility in gradient control.

Moreover, as shown in Figure 5, $\partial \mathbf{c}/\partial \mathbf{b}$ exhibits a pronounced diagonal than $\partial \mathbf{c}/\partial \mathbf{a}$, suggesting a stronger dependence of $\mathbf{c}$ on $\mathbf{b}$ than $\mathbf{a}$. This aligns with the design, as $\mathbf{b}$ directly modulates the attention weights.

Combination of MLP and Vanilla Attention We now provide a new perspective on the enhanced attention. In Equation (9), the attention matrix is decomposed into two components: the input-independent, dataset-specific term $\bar{\mathbf{B}}$, and the input-dependent term $\mathbf{A}$. If $\mathbf{A}$ is zero, the enhanced attention reduces to a linear transformation of $\mathbf{V}$, effectively functioning as an MLP along the variate dimension. By jointly optimizing $\mathbf{A}$ and $\bar{\mathbf{B}}$, the enhanced attention can be interpreted as an adaptive combination of MLP and vanilla attention.

5 Experiments

Datasets and Implementation Details We extensively evaluate the FreEformer using eighteen real-world datasets: **ETT** (four subsets), **Weather**, **ECL**, **Traffic**, **Exchange**, **Solar-Energy**, **PEMS** (four subsets), **ILI**, **COVID-19**, **METR-LA**, **NASDAQ** and **Wiki**. During training, we adopt the L1 loss function from CARD [Wang *et al.*, 2024c]. The embedding dimension d is fixed at 16, and the dimension D is selected from $\{128, 256, 512\}$. The dataset description and implementation details are provided in the appendix.

5.1 Forecasting Performance

We choose 10 well-acknowledged deep forecasters as our baselines, including (1) Transformer-based models: Leddam [Yu *et al.*, 2024], CARD [Wang *et al.*, 2024c], Fredformer [Piao *et al.*, 2024], iTransformer [Liu *et al.*, 2024a], PatchTST [Nie *et al.*, 2023], Crossformer [Zhang and Yan, 2023]; (2) Linear-based models: TimeMixer [Wang *et al.*, 2024b], FreTS [Yi *et al.*, 2024c] and DLinear [Zeng *et al.*, 2023]; (3) TCN-based model: TimesNet [Wu *et al.*, 2023a].

Comprehensive results for long- and short-term forecasting are presented in Tables 2 and 3, respectively, with the best results highlighted in bold and the second-best underlined. FreEformer consistently outperforms state-of-the-art models across various prediction lengths and real-world domains. Compared with sophisticated time-domain-based models, such as Leddam and CARD, FreEformer achieves superior performance with a simpler architecture, benefiting from the global-level property of the frequency domain. Furthermore, its performance advantage over Fredformer, another Transformer- and frequency-based model, suggests that the deliberate patching of band-limited frequency spectra may introduce noise, hindering forecasting accuracy.

Notably, in Table 4, we compare FreEformer with additional frequency-based models, where it also demonstrates a clear performance advantage. The visualization results of

Model	FreEformer (Ours)		Leddam [2024]		CARD [2024c]		Fredformer [2024a]		iTrans. [2024a]		TimeMixer [2024b]		PatchTST [2023]		Crossfm. [2023]		TimesNet [2023a]		FreTS [2024c]		DLinear [2023]	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.379**	**0.381**	0.386	0.397	0.383	0.384	0.384	0.395	0.407	0.410	0.381	0.395	0.387	0.400	0.513	0.496	0.400	0.406	0.407	0.415	0.403	0.407
ETTm2	**0.272**	**0.313**	0.281	0.325	**0.272**	0.317	0.279	0.324	0.288	0.332	0.275	0.323	0.281	0.326	0.757	0.610	0.291	0.333	0.335	0.379	0.350	0.401
ETTh1	0.433	0.431	**0.431**	0.429	0.442	0.429	0.435	**0.426**	0.454	0.447	0.447	0.440	0.469	0.454	0.529	0.522	0.458	0.450	0.488	0.474	0.456	0.452
ETTh2	0.372	0.393	0.373	0.399	0.368	**0.390**	0.365	0.393	0.383	0.407	**0.364**	0.395	0.384	0.405	0.942	0.684	0.414	0.427	0.550	0.515	0.559	0.515
ECL	**0.162**	**0.251**	0.169	0.263	0.168	0.258	0.176	0.269	0.178	0.270	0.182	0.272	0.208	0.295	0.244	0.334	0.192	0.295	0.202	0.290	0.212	0.300
Exchange	0.354	0.399	0.354	0.402	0.362	0.402	**0.333**	**0.391**	0.360	0.403	0.387	0.416	0.367	0.404	0.940	0.707	0.416	0.443	0.416	0.439	0.354	0.414
Traffic	0.435	**0.251**	0.467	0.294	0.453	0.282	0.433	0.291	**0.428**	0.282	0.484	0.297	0.531	0.343	0.550	0.304	0.620	0.336	0.538	0.328	0.625	0.383
Weather	**0.239**	**0.260**	0.242	0.272	**0.239**	0.265	0.246	0.272	0.258	0.279	0.240	0.271	0.259	0.281	0.259	0.315	0.259	0.287	0.255	0.298	0.265	0.317
Solar	0.217	**0.219**	0.230	0.264	0.237	0.237	0.226	0.262	0.233	0.262	**0.216**	0.280	0.270	0.307	0.641	0.639	0.301	0.319	0.226	0.254	0.330	0.401
PEMS03	**0.102**	**0.206**	0.107	0.210	0.174	0.275	0.135	0.243	0.113	0.221	0.167	0.267	0.180	0.291	0.169	0.281	0.147	0.248	0.169	0.278	0.278	0.375
PEMS04	**0.094**	**0.196**	0.103	0.210	0.206	0.299	0.162	0.261	0.111	0.221	0.185	0.287	0.195	0.307	0.209	0.314	0.129	0.241	0.188	0.294	0.295	0.388
PEMS07	**0.080**	**0.167**	0.084	0.180	0.149	0.247	0.121	0.222	0.101	0.204	0.181	0.271	0.211	0.303	0.235	0.315	0.124	0.225	0.185	0.282	0.329	0.395
PEMS08	**0.110**	**0.194**	0.122	0.211	0.201	0.280	0.161	0.250	0.150	0.226	0.226	0.299	0.280	0.321	0.268	0.307	0.193	0.271	0.212	0.297	0.379	0.416

Table 2: Long-term time series forecasting results for $T = 96$ and $\tau \in \{96, 192, 336, 720\}$. For PEMS, $\tau \in \{12, 24, 48, 96\}$. Results are averaged across these prediction lengths. These settings are used throughout the following tables.

Model		FreEformer (Ours)		Leddam [2024]		CARD [2024c]		Fredformer [2024a]		iTrans. [2024a]		TimeMixer [2024b]		PatchTST [2023]		TimesNet [2023a]		DLinear [2023]		FreTS [2024c]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ILI	S1	**1.140**	**0.585**	1.468	0.679	1.658	0.707	1.518	0.696	1.437	0.659	1.707	0.734	1.681	0.723	1.480	0.684	2.400	1.034	1.839	0.782
	S2	**1.906**	**0.835**	1.982	0.875	2.260	0.938	1.947	0.899	1.993	0.887	2.020	0.878	2.128	0.885	2.139	0.931	3.083	1.217	3.036	1.174
COVID-19	S1	**1.892**	**0.673**	2.064	0.779	2.059	0.767	1.902	0.765	2.096	0.795	2.234	0.782	2.221	0.820	2.569	0.861	3.483	1.102	2.516	0.862
	S2	**8.435**	**1.764**	8.439	1.792	9.013	1.862	8.656	1.808	8.506	1.792	9.604	1.918	9.451	1.905	9.644	1.877	13.075	2.099	11.345	1.958
METR-LA	S1	0.336	**0.221**	0.327	0.243	0.349	0.233	0.336	0.242	0.338	0.244	0.334	0.245	0.335	0.243	0.344	0.253	0.341	0.294	**0.324**	0.279
	S2	0.840	**0.406**	0.878	0.490	0.929	0.466	0.898	0.495	0.916	0.501	0.881	0.499	0.893	0.502	0.890	0.488	0.819	0.550	**0.804**	0.543
NASDAQ	S1	**0.055**	**0.126**	0.059	0.135	0.057	0.130	0.059	0.135	0.060	0.137	**0.055**	**0.126**	0.058	0.132	0.068	0.151	0.072	0.170	0.080	0.184
	S2	**0.185**	**0.277**	0.196	0.286	0.193	0.284	0.194	0.285	0.207	0.297	0.186	0.281	0.198	0.286	0.255	0.343	0.228	0.331	0.263	0.361
Wiki	S1	6.524	**0.391**	6.547	0.404	6.553	0.400	6.705	0.406	6.569	0.405	6.572	0.409	6.523	0.404	7.956	0.520	6.634	0.481	**6.521**	0.448
	S2	6.259	**0.442**	6.286	0.463	6.285	0.453	**5.931**	0.453	6.275	0.458	6.315	0.468	6.212	0.444	7.310	0.623	6.205	0.539	6.147	0.505

Table 3: Short-term time series forecasting results under two settings: S1 (Input-12, Predict-$\{3, 6, 9, 12\}$) and S2 (Input-36, Predict-$\{24, 36, 48, 60\}$). Average results are reported across four prediction lengths. S1 is the default setting in the following experiments.

Models	FreEformer (Ours)		FITS [2023]		FAN [2024]		FilterNet [2024a]		FreDF [2024a]	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT(Avg)	**0.364**	**0.380**	0.408	0.405	0.405	0.427	0.367	0.384	0.369	0.384
ECL	**0.162**	**0.251**	0.384	0.434	0.208	0.298	0.201	0.285	0.170	0.259
Traffic	0.435	**0.251**	0.615	0.370	0.526	0.357	0.521	0.340	**0.421**	0.279
Weather	**0.239**	**0.260**	0.273	0.292	0.247	0.292	0.248	0.274	0.254	0.274

Table 4: Comparison with additional state-of-the-art frequency-based models. Average results are reported across four prediction lengths. 'Avg' refers to averages further computed over subsets.

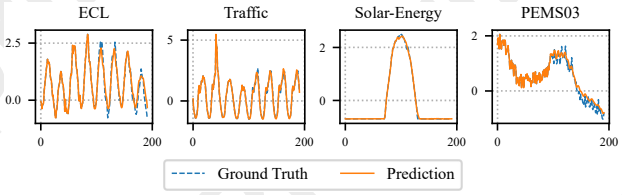

Figure 6: Visualization of the forecasting results under the 'Input-96-Predict-96' setting, demonstrating accurate approximations.

FreEformer are presented in Figure 6. Furthermore, as shown in Table 10 of the appendix, FreEformer exhibits state-of-the-art performance with variable lookback lengths.

5.2 Model Analysis

Architecture Ablations The FreEformer utilizes an enhanced Transformer architecture to capture cross-variate dependencies in the frequency domain. Table 5 presents a comparison of several FreEformer variants, evaluating the impact of linear and enhanced Transformer layers, different dimensional configurations, and patching along the frequency dimension. To ensure a fair comparison, the enhanced Transformer is used for all Transformer-based settings. The results indicate that: 1) Enhanced Transformer blocks outperform linear layers due to their superior representation capabilities; 2) Multivariate dependency learning generally outperforms inter-frequency learning, aligning with the claim in FreDF [Wang *et al.*, 2024a] that correlations among frequency points are minimal; 3) Furthermore, patching does not improve FreEformer, likely because patching frequencies creates localized receptive fields, thereby limiting access to global information.

Frequency-Domain vs. Temporal Representation To construct the time-domain variant of FreEformer, we remove the DFT and IDFT steps, as well as the imaginary branch.

Layer	Dim.	Patch.	ETTm1	Weather	ECL	Traffic	COVID-19
Linear	Var.	✗	0.385	0.245	0.189	0.488	2.040
Linear	Fre.	✗	0.386	0.246	0.184	0.482	2.086
Trans.	Fre.	✔	0.381	0.244	0.183	0.504	2.100
Trans.	Fre.	✗	0.383	0.245	0.181	0.489	2.116
Trans.	Var.	✔	0.385	0.241	**0.162**	0.443	2.029
Trans.	Var.	✗	**0.379**	**0.239**	**0.162**	**0.435**	**1.892**

Table 5: Architecture ablations on layers, dimensions, and patching settings. Layers include linear and enhanced Transformer layers, while dimensions refer to frequency and variate dimensions. 'Patch.' indicates patching along the frequency dimension, with patch length and stride set to 6 and 3 for COVID-19, and 16 and 8 for other datasets. Average MSEs are reported across four prediction horizons. The final row corresponds to the FreEformer configuration.

Attn.	Domain	Traffic	PEMS03	Weather	Solar	ILI
Ours	Fre.	**0.435**	**0.102**	**0.239**	**0.217**	**1.140**
	Time	0.443	0.122	0.243	0.228	1.375
Vanilla	Fre.	0.451	**0.113**	**0.245**	**0.220**	**1.510**
	Time	**0.441**	0.146	0.248	0.226	2.140

Table 6: Performance comparison of the frequency-domain and time-domain representation learning under two attention settings. Average MSEs are reported across four prediction lengths.

As shown in Table 6, the frequency-domain representation achieves an average improvement of 8.4% and 10.7% in MSE compared to the time-domain version under the enhanced and vanilla attention settings, respectively. Additionally, we show in Section I.1 of the appendix that Fourier bases generally outperform Wavelet and polynomial bases for our model.

Head	ETTm1	Weather	ECL	Traffic	Solar	NASDAQ	COVID-19
Fre.	**0.379**	0.245	**0.160**	0.441	**0.216**	**0.055**	1.930
Time	**0.379**	**0.239**	0.162	**0.435**	0.217	**0.055**	**1.892**

Table 7: Performance comparison of frequency-domain and temporal prediction heads. Average MSEs are reported.

Prediction Head In our model, after performing frequency domain representation learning, we apply a temporal prediction head to generate the final predictions. In contrast, some frequency-based forecasters (e.g., FITS and Fredformer) directly predict the future frequency spectrum and transform it back to the time domain as the final step. In FreEformer, the frequency prediction head is formulated as:

$$\hat{\mathbf{y}} = \text{DeNorm}\left(\text{IDFT}\left(\text{FlatLin}(\tilde{\text{Re}}) + j \cdot \text{FlatLin}(\tilde{\text{Im}})\right)\right), \quad (14)$$

where $\tilde{\text{Re}}$ and $\tilde{\text{Im}}$ are defined in Equation (6). As shown in Table 7, the temporal head slightly outperforms the frequency-domain head, highlighting the challenges of accurately forecasting the frequency spectrum. Additionally, Equation (14) incurs higher computational costs in the IDFT step when $\tau > T$, as in long-term forecasting scenarios.

5.3 Enhanced Attention Analysis

Dataset	Ours	Trans. [2017]	Flowfm. [2022]	Flashfm. [2022]	Flatten [2023]	Mamba [2023]	LASER [2024]	Lin.Attn. [2024]
Traffic	**0.435**	0.451	0.453	0.448	0.453	0.443	0.451	0.452
PEMS03	**0.102**	0.113	0.113	0.114	0.114	0.115	0.111	0.115
Weather	**0.239**	0.245	0.242	0.245	0.248	0.243	0.244	0.245
Solar	**0.217**	0.220	0.224	0.221	0.229	0.228	0.219	0.230
ILI	**1.140**	1.510	1.288	1.547	1.842	1.508	1.596	1.453

Table 8: Comparison of the enhanced Transformer with state-of-the-art attention models and Mamba. Average MSEs are reported across four prediction lengths. Results outperforming state-of-the-art forecasters Leddam and CARD are highlighted in red.

We compare the enhanced Transformer with vanilla Transformer, state-of-the-art Transformer variants and Mamba [Gu and Dao, 2023] in Table 8. The enhanced Transformer consistently outperforms other models, verifying the effectiveness of the enhanced attention mechanism.

Dataset	iTrans.		PatchTST		Leddam		Fredformer	
	Van.	E.A.	Van.	E.A.	Van.	E.A.	Van.	E.A.
ETTm1	0.407	**0.389**	0.387	**0.381**	0.386	**0.384**	**0.384**	0.385
ECL	0.178	**0.165**	0.208	**0.181**	0.169	**0.167**	0.176	**0.169**
PEMS07	0.101	**0.086**	0.211	**0.156**	0.084	**0.080**	0.121	**0.103**
Solar	0.233	**0.226**	0.270	**0.232**	0.230	**0.228**	0.226	**0.222**
Weather	0.258	**0.249**	0.259	**0.245**	**0.242**	**0.242**	0.246	**0.242**
METR-LA	0.338	**0.329**	**0.335**	**0.335**	0.327	**0.321**	0.336	**0.334**

Table 9: Comparison of state-of-the-art models using vanilla (Van.) and enhanced attention (E.A.). Only the attention mechanism is updated, with other components and the loss function kept unchanged. Average MSEs across four prediction lengths are reported.

We further apply the enhanced attention mechanism to state-of-the-art forecasters, as shown in Table 9. This yields average MSE improvements of 5.9% for iTransformer, 9.9% for PatchTST, 1.4% for Leddam (with updates only to the 'cross-channel attention' module), and 3.8% for FreEformer. These results demonstrate the versatility and effectiveness of the enhanced attention mechanism. Moreover, comparing Tables 2 and 9, FreEformer consistently outperforms these improved forecasters, underscoring its architectural advantages.

In the appendix, we further demonstrate FreEformer's performance superiority on more metrics (e.g., MASE, correlation coefficient). Remarkably, FreEformer, trained from scratch, achieves superior or comparable performance to a pre-trained model fine-tuned on the same training data.

6 Conclusion

In this work, we present a simple yet effective multivariate time series forecasting model based on a frequency-domain enhanced Transformer. The enhanced attention mechanism is demonstrated to be effective both theoretically and empirically. It can consistently bring performance improvements for state-of-the-art Transformer-based forecasters. We hope that FreEformer will serve as a strong baseline for the time series forecasting community.

Acknowledgments

This work was supported by Beijing Natural Science Foundation under Grant No. L247029, and the National Natural Science Foundation of China (NSFC) under Grant No. 62371009.

References

[Ansari *et al.*, 2024] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[Bai *et al.*, 2018] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[Chen *et al.*, 2023] Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97:101819, 2023.

[Dao *et al.*, 2022] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 35:16344–16359, 2022.

[Fan *et al.*, 2022] Wei Fan, Shun Zheng, Xiaohan Yi, Wei Cao, Yanjie Fu, Jiang Bian, and Tie-Yan Liu. Depts: Deep expansion learning for periodic time series forecasting. *arXiv preprint arXiv:2203.07681*, 2022.

[Fan *et al.*, 2024] Wei Fan, Kun Yi, Hangting Ye, Zhiyuan Ning, Qi Zhang, and Ning An. Deep frequency derivative learning for non-stationary time series forecasting. *arXiv preprint arXiv:2407.00502*, 2024.

[Goswami *et al.*, 2024] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *ICML*, 2024.

[Gu and Dao, 2023] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[Han *et al.*, 2023] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *ICCV*, pages 5961–5971, 2023.

[Han *et al.*, 2024] Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[He *et al.*, 2022] Hui He, Qi Zhang, Simeng Bai, Kun Yi, and Zhendong Niu. Catn: Cross attentive tree-aware network for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4030–4038, 2022.

[Jin *et al.*, 2021] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *ICLR*, 2021.

[Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2021.

[Kitaev *et al.*, 2020] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[Kollovieh *et al.*, 2024] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *NeurIPS*, volume 36, 2024.

[Li *et al.*, 2023] Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Ching-Feng Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[Liu *et al.*, 2022a] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *ICLR*, 2022.

[Liu *et al.*, 2022b] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *NeurIPS*, 35:9881–9893, 2022.

[Liu *et al.*, 2024a] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.

[Liu *et al.*, 2024b] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Transformers for time series analysis at scale. In *ICML*, 2024.

[Nie *et al.*, 2023] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *ICLR*, 2023.

[Palani, 2022] Sankaran Palani. *Signals and systems*. Springer, 2022.

[Piao *et al.*, 2024] Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2400–2410, 2024.

[Salinas *et al.*, 2020] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

[Surya Duvvuri and Dhillon, 2024] Sai Surya Duvvuri and Inderjit S Dhillon. Laser: Attention with exponential transformation. *arXiv e-prints*, pages arXiv–2411, 2024.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017.

[Wang *et al.*, 2024a] Hao Wang, Licheng Pan, Zhichao Chen, Degui Yang, Sen Zhang, Yifei Yang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. Fredf: Learning to forecast in frequency domain. *arXiv preprint arXiv:2402.02399*, 2024.

[Wang *et al.*, 2024b] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *ICLR*, 2024.

[Wang *et al.*, 2024c] Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. In *ICLR*, 2024.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *NeurIPS*, 34:22419–22430, 2021.

[Wu *et al.*, 2022] Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. *arXiv preprint arXiv:2202.06258*, 2022.

[Wu *et al.*, 2023a] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.

[Wu *et al.*, 2023b] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 5(6):602–611, 2023.

[Xiong *et al.*, 2021] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14138–14148, 2021.

[Xu *et al.*, 2023] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

[Ye *et al.*, 2024] Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *arXiv preprint arXiv:2409.20371*, 2024.

[Yi *et al.*, 2023] Kun Yi, Qi Zhang, Longbing Cao, Shoujin Wang, Guodong Long, Liang Hu, Hui He, Zhendong Niu, Wei Fan, and Hui Xiong. A survey on deep learning based time series analysis with frequency transformation. *arXiv preprint arXiv:2302.02173*, 2023.

[Yi *et al.*, 2024a] Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. *arXiv preprint arXiv:2411.01623*, 2024.

[Yi *et al.*, 2024b] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

[Yi *et al.*, 2024c] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

[Yu *et al.*, 2024] Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In *ICML*, 2024.

[Yue *et al.*, 2024] Wenzhen Yue, Xianghua Ying, Ruohao Guo, Dongdong Chen, Yuqing Zhu, Ji Shi, Bowei Xing, and Taiyan Chen. Sub-adjacent transformer: Improving time series anomaly detection with reconstruction error from sub-adjacent neighborhoods. In *IJCAI*, 2024.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI*, volume 37, pages 11121–11128, 2023.

[Zhang and Yan, 2023] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *ICLR*, 2023.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022a] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022.

[Zhou *et al.*, 2022b] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, pages 27268–27286. PMLR, 2022.

[Zhou *et al.*, 2023] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. *NeurIPS*, 36:43322–43355, 2023.