

Article

Not peer-reviewed version

---

# Exploring Time Series Analysis in Frequency Domain with Complex-Valued Spectral Attention and Bidirectional Variable Mamba

---

Qinghe Zhao , [Fang Lai](#) , [Xingye Mo](#) \*

Posted Date: 21 April 2025

doi: 10.20944/preprints202504.1614.v1

Keywords: time series analysis; state space model; complex-valued spectral attention



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

# Exploring Time Series Analysis in Frequency Domain with Complex-Valued Spectral Attention and Bidirectional Variable Mamba

Qinghe Zhao <sup>1</sup>, Fang Lai <sup>2</sup> and Xingye Mo <sup>3,\*</sup>

<sup>1</sup> Faculty of Environment, Science and Economy, University of Exeter, Exeter, EX4 4QJ, Devon, UK

<sup>2</sup> Department of Computer Science, University of Denver, Denver, 80208, Colorado, USA

<sup>3</sup> Department of Electrical and Computer Engineering, New York University, New York, 10012, NY, USA

\* Correspondence: xm2100@nyu.edu

**Abstract:** Recently, deep learning has made significant strides in multivariate time series forecasting (MTSF). While frequency-domain-based methods have shown promising results, existing models often struggle with frequency misalignment when handling diverse frequency combinations, leading to reduced forecasting accuracy. To address these issues, we propose the Spectral Attention Module (SAM), which integrates temporal and frequency domain information to effectively capture both local and global dependencies in time series data. Within the frequency-domain module, we introduce an Extended Discrete Fourier Transform (EDFT) to overcome frequency misalignment challenges and design a Complex-Valued Spectral Attention Mechanism (CV-SAM) to identify and exploit complex relationships among different frequency combinations. To further capture inter-variable correlations, we propose the Bidirectional Variable Mamba (BV-Mamba). It uses linear layers to encode timestamps for each variable and employs the Mamba layer to extract inter-variable correlations, supported by a feedforward network to learn temporal dependencies. By combining the SAM and the BV-Mamba, we construct the SpectroMamba, which demonstrates superior performance over state-of-the-art methods in long-term time series forecasting across multiple real-world datasets.

**Keywords:** time series analysis; state space model; complex-valued spectral attention

## 1. Introduction

In the real world, time series data exhibit diverse periodic patterns [1], manifesting not only across different segments of the same time series but also among various time series within the same dataset. While these periodic features are ubiquitous, their manifestations vary widely, reflecting the complexity and heterogeneity of time series data. Periodic time series often exhibit pronounced global dependencies, where data values at a given time are influenced by values at specific intervals in the past. In contrast, non-periodic time series predominantly display local dependencies [2], where data values at a given time are primarily determined by the values of nearby time points, lacking long-term consistency and predictability.

To gain a deeper understanding of the nature of time series data, time-domain and frequency-domain analyses have become two fundamental tools. Time-domain analysis focuses on the trends of data over time, making it particularly suitable for uncovering short-term fluctuations and instantaneous events. In contrast, frequency-domain analysis decomposes time series to identify their intrinsic periodic components [3], providing strong support for understanding long-term trends and global correlations. By combining the strengths of time-domain and frequency-domain analyses, it becomes possible to comprehensively capture the multi-scale characteristics

of time series data, which is crucial for addressing the complex challenges posed by real-world time series. Early research predominantly focused on time-domain analysis, as seen in methods like the Autoregressive Integrated Moving Average (ARIMA) [4] and Long Short-Term Memory (LSTM) [5] networks. In recent years, more advanced methods such as the Temporal Convolutional Network (TCN) [6] and Transformer-based architectures like the Time Series Transformer (TST) [7] have further improved the ability to capture temporal dependencies and model long-term relationships in time-series data. These approaches have laid a solid foundation for time-series forecasting, particularly in the time domain. With advancements in technology, increasing attention has shifted toward frequency-domain analysis, which has demonstrated immense potential, especially when integrated with deep learning techniques. For instance, the Frequency-domain Time-series Prediction (FITS) model [8] directly uses the frequency spectrum of time series as input, performs the prediction task in the frequency domain, and then restores the results to the time domain through a simple linear transformation. This approach not only improves prediction accuracy but also highlights the unique advantages of frequency-domain analysis. However, predicting directly in the frequency domain is not without challenges. One critical issue is efficiently extracting and integrating information about frequency combinations from the spectrum [9]. Harmonic sequences in the spectrum carry abundant information, but their complexity and diversity make this task highly challenging. Addressing these difficulties is key to unlocking the full potential of frequency-domain analysis in time-series forecasting. In addition to modeling temporal dependencies, another major challenge in multi-variate time series forecasting (MTSF) is capturing the correlations among variables [10]. Recent research has categorized MTSF models for modeling variable dependencies into two main types: variable-mixing models and variable-independent models [11]. Variable-mixing models extract features from all time series and project them into an embedding space to integrate information from multiple variables. In contrast, variable-independent models restrict their input to information from a single variable. Interestingly, recent studies have shown that variable-independent models outperform variable-mixing models on certain datasets [12]. However, this advantage comes at a cost: the omission of critical cross-variable information. This limitation can be detrimental, particularly when the variables are inherently correlated. Addressing this trade-off is essential to improve the robustness and generalizability of MTSF models across diverse real-world scenarios.

To address the challenges outlined above, we developed a novel framework named SpectroMamba for modeling both cross-variable relationships and temporal dependencies. Specifically, we propose an innovative Spectral Attention Module (SAM), which integrates both time-domain and frequency-domain information, effectively capturing local and global dependencies within time series data. In the Spectral Attention Module, we introduce the Extended Discrete Fourier Transform (EDFT) to overcome frequency misalignment issues and design a Complex-Valued Spectral Attention Mechanism (CV-SAM) to identify and leverage the complex relationships between different frequency combinations. To model cross-variable relationships, we propose the Bidirectional Variable Mamba (BV-Mamba) for variable-mixing methods, which is aimed at efficiently capturing and utilizing the intricate dependencies among multiple variables, thereby enhancing the performance of multivariate time series forecasting. In essence, the main contributions of our paper include:

- We introduce a novel Spectral Attention Module that integrates both time-domain and frequency-domain information to effectively capture local and global dependencies within time series data. It addresses frequency misalignment issues through the Extended Discrete Fourier Transform (EDFT) and utilizes a complex-valued spectral attention mechanism to identify and exploit intricate relationships among frequency combinations.
- We propose an innovative Bidirectional Variable Mamba that effectively captures and leverages complex relationships between multiple variables, enhancing the performance of multivariate time series forecasting.

- We present a comprehensive framework, called SpectroMamba that integrates the Spectral Attention and Variable Mamba modules. This unified approach is designed to tackle the challenges of multivariate time series forecasting and demonstrates superior performance on real-world datasets.

## 2. Related Works

### 2.1. Multivariate Time Series Forecasting

In multivariate time series forecasting (MTSF), model design typically falls into two main approaches: variable-independent modeling [9, 12, 13] and variable-dependent modeling [14–16]. In the variable-independent approach, each variable is processed independently, often using a shared underlying architecture. This approach is simpler and easier to implement. For example, PatchTST [12] introduced slicing and variable independence strategies to significantly improve the performance of the Transformer architecture. However, the variable-independent method tends to incur high computational costs during both training and inference, leading researchers to explore more efficient modeling strategies. In contrast, the variable-dependent approach considers inter-channel dependencies when predicting future values, enabling more comprehensive capture of complex relationships within multivariate time series data. For instance, the RLinear [14] that the associations between channels comprehensively during forecasting. The Crossformer [10] used serial time and variable attention mechanisms to capture dependencies across time and variables, while Dsformer [17] applied these two attention mechanisms in parallel, enhancing both efficiency and accuracy. Furthermore, iTransformer [18] innovatively disrupted the traditional allocation of attention mechanisms and Feedforward networks in Transformers, focusing on capturing multivariate correlations and learning nonlinear representations, demonstrating exceptional performance in multivariate time series forecasting tasks.

Frequency-domain prediction methods [9, 19, 20, 20, 21] have gained increasing attention due to their ability to reveal the periodic and frequency components within time series data, offering a unique perspective for capturing global dependencies. Recent advancements have focused on harnessing these frequency-domain features for improved forecasting. For example, Autoformer [19] utilized Fast Fourier Transform (FFT) to identify dominant periods and replaces the traditional self-attention mechanism with an autocorrelation mechanism, improving accuracy in long-sequence predictions. TimesNet [9] leveraged the top-k frequency amplitudes extracted via FFT, transforming one-dimensional time series into two-dimensional features to better capture spatial dependencies. Film [20] used Legendre polynomials to project historical information and extracts frequency-domain features through FFT-based frequency enhancement layers. FreDo [21] introduced a simple yet efficient model, AverageTile, which uses Discrete Fourier Transform (DFT) to map time series into the frequency domain, enabling more accurate frequency feature extraction. These methods highlight the growing potential of frequency-domain approaches in improving time series forecasting by capturing long-term dependencies and enhancing feature extraction across multiple scales.

Moreover, complex number processing is crucial for frequency-domain analysis, as real-valued representations may fail to capture deeper features within the frequency domain. For instance, StemGNN [22] integrated Graph Fourier Transform (GFT) to model inter-sequence dependencies and uses DFT to capture temporal dependencies. However, its separation of real and imaginary parts of complex numbers may limit its ability to fully capture the complex interactions in the data. To address this, many methods optimize frequency-domain feature extraction using complex-valued neural networks. For example, FEDformer [15] introduced learnable complex weights for frequency-domain transformations and offers two versions based on DFT and Discrete Wavelet Transform (DWT), enabling better exploration of frequency-domain relationships. FITS [8] applied FFT to transform time series into the frequency domain,

followed by complex-valued linear layers and low-pass filters to attenuate high-frequency noise. FreTS [2] designed a real-valued MLP structure that mimics complex number operations, allowing for accurate handling of both real and imaginary parts of frequency coefficients.

## 2.2. State Space Model

Sequence modeling has been a crucial aspect of various applications, including natural language processing, computer vision, and video understanding. Traditional sequence-to-sequence models with soft attention have shown success in these domains but suffer from quadratic time and space complexity during decoding, limiting their real-time applicability. To address this issue, Chiu et al. [23] introduced Monotonic Chunkwise Attention (MoChA), a linear-time attention mechanism that performs soft attention over adaptively-located chunks of the input sequence. While models like the Structured State-Spaces for Long-Form Video Understanding (S4) have been effective, treating all image-tokens equally can impact efficiency and accuracy. Wang et al. [24] proposed a Selective S4 (S5) model that uses a lightweight mask generator to selectively choose informative image tokens, resulting in more efficient and accurate modeling of long-term spatiotemporal dependencies in videos. Gu et al. [25] identified the computational inefficiency of Transformers on long sequences and introduced Mamba, a linear-time sequence modeling approach with selective state spaces. The Mamba layer offers an efficient selective state space model that is effective in various domains, including NLP and computer vision. Additionally, incorporating Mamba in Multiple Instance Learning (MIL) for long sequence modeling has shown promise, especially with the Sequence Reordering Mamba (SR-Mamba) [26] that considers the order and distribution of instances. State space models like Mamba have also been explored in motion generation tasks, showcasing improvements in FID scores and speed compared to previous methods. Qiao et al. [27] proposed VL-Mamba, a multimodal large language model based on state space models, to address the computational overhead of attention mechanisms in Transformers. In the realm of computer vision, Lu et al. [28] tackled the challenge of designing an appropriate scanning method for 4D light fields by employing state space models on informative 2D slices to enhance spatial contextual information and structure information.

## 3. Preliminary

### 3.1. Problem Definition

Long-term time series forecasting can be viewed as a sequence-to-sequence problem. Given a multivariate time series, where  $L$  is the size of the lookback window and  $N$  is the number of variables, the goal is to forecast the next  $T$  time steps.

Formally, let the historical time series data for the past  $L$  time steps, each containing information from  $N$  variables, be represented as:

$$X = \{x_{t-L}, x_{t-L+1}, \dots, x_{t-1}\} \text{ where } x_t \in \mathbb{R}^N \quad (1)$$

The objective is to predict the future  $T$  steps:

$$Y = \{y_t, y_{t+1}, \dots, y_{t+T-1}\} \quad (2)$$

where  $X$  represents the historical data and  $Y$  represents the future predictions. The goal of the model is to infer  $Y$  from  $X$ , i.e., to predict future values based on past data.

### 3.2. Discrete Fourier Transform

The Discrete Fourier Transform (DFT) [29, 30] is a mathematical transformation that converts a time series into its frequency domain representation, revealing the frequency components of the data. Specifically, for an input sequence of length  $L$ , the DFT spectrum is computed as follows:

$$X_k = \sum_{n=0}^{L-1} x_n \cdot e^{-j \frac{2\pi}{L} kn} \quad \text{for } k = 0, 1, \dots, L-1 \quad (3)$$

where  $x_n$  is the input sequence of length  $L$ ,  $X_k$  is the complex frequency domain representation,  $k$  represents the index of the frequency component,  $j$  is the imaginary unit,  $e^{-j \frac{2\pi}{L} kn}$  is the complex exponential function that captures the oscillatory nature of the frequency components.

The Inverse Discrete Fourier Transform (IDFT) is the reverse operation of the DFT. It converts the frequency-domain signal obtained by the DFT back into the original time-domain signal. The IDFT is computed as follows:

$$x_n = \frac{1}{L} \sum_{k=0}^{L-1} X_k \cdot e^{j \frac{2\pi}{L} kn} \quad \text{for } n = 0, 1, \dots, L-1 \quad (4)$$

where  $X_k$  is the frequency-domain representation of the signal,  $x_n$  is the time-domain signal after applying the inverse transform,  $j$  is the imaginary unit,  $L$  is the length of the sequence.

### 3.3. State Space Models

State Space Models (SSMs) [31] are widely used for modeling cyclic processes with hidden states. They describe the evolution of system states using first-order differential equations and relate the hidden states to the output sequence through a second set of equations. The general form of the system is given by:

$$\dot{h}(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad (5)$$

$$y(t) = \mathbf{C}h(t) + \mathbf{D}x(t), \quad (6)$$

Here,  $\mathbf{A}$  represents the state matrix that governs how the current state influences its rate of change,  $\mathbf{B}$  is the input matrix, and  $\mathbf{C}$  and  $\mathbf{D}$  are the output and feed-through matrices, respectively.

In practical applications, this continuous system is discretized for computational efficiency, typically under the zero-order hold assumption [32]. The discretized version of the system is obtained by converting the continuous matrices  $\mathbf{A}$  and  $\mathbf{B}$  to their discrete counterparts over a time step  $\Delta \in \mathbb{R}^+$ :

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\Delta \mathbf{A}), \\ \bar{\mathbf{B}} &= (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B}. \end{aligned} \quad (7)$$

Thus, the discretized state space model can be written as:

$$h_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t, \quad y_t = \mathbf{C}h_t + \mathbf{D}x_t. \quad (8)$$

A key limitation of this approach is that the model parameters remain fixed regardless of the input sequence. To address this issue, the Mamba framework introduces a selective state-space model, where the parameters adapt according to the input data, enabling more effective selective information processing across different time sequences [33]. This adaptation is expressed by the following equations:

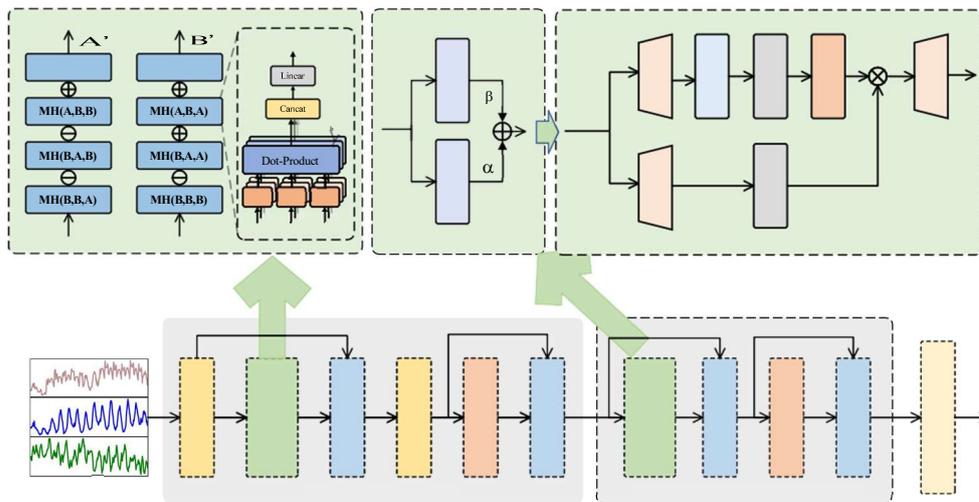
$$\bar{\mathbf{B}} = f_{\mathbf{B}}(x_t), \quad \bar{\mathbf{C}} = f_{\mathbf{C}}(x_t), \quad \Delta = \theta_{\mathbf{A}}(\mathbf{P} + f_{\mathbf{A}}(x_t)), \quad (9)$$

Here,  $f_B(x_t)$ ,  $f_C(x_t)$ , and  $f_A(x_t)$  are linear functions that adapt the model's parameters based on the input sequence, allowing for a more dynamic and responsive system.

## 4. Proposed Method

### 4.1. Overall Framework

Our proposed SpectroMamba architecture is shown in Figure 1. Given the input time series  $X = [x_1, x_2, \dots, x_L] \in \mathbb{R}^{R \times L}$ , we first apply Reverse Instance Normalization (RevIN) [34] to process the input, a method originally designed to address distributional shifts in the time domain. Next, we apply the extended Discrete Fourier Transform (DFT) to transform the data into the frequency domain, enabling the capture of global correlations in the time series. Subsequently, we use the Inverse Discrete



**Figure 1.** The framework overview of our proposed SpectroMamba.

Fourier Transform (IDFT) to map the frequency-domain data back to the time domain. On this basis, we use a Feed Forward Network (FFN) to capture local dependencies within the time domain. Furthermore, we introduce the Mamba layer to extract correlations among different variables, and combine it with the FFN to further learn temporal dependencies. The overall architecture can be formalized by the following equations:

$$x_0^f = \text{SAM}(X) \quad (10)$$

$$X = \text{BVMM}(x_0^f) \quad (11)$$

### 4.2. Spectral Attention Module

The SAM is built upon the Transformer encoder architecture and is specifically designed to process complex-valued data in the frequency domain. All computations within the module are conducted in the complex domain, ensuring the spectral characteristics of the input data are fully preserved.

The SAM takes as input the univariate spectrum  $F \in \mathbb{C}^{\hat{L}}$ , generated using the Extended Discrete Fourier Transform (DFT), where  $\hat{L} = \frac{L+T}{2} + 1$  represents the length of the spectrum. This design leverages the properties of complex-valued representations to model and capture intricate dependencies in the frequency domain effectively, as shown in Figure 1.

The traditional multi-head attention mechanism of Transformers is designed for real-valued data, making it challenging to directly apply to complex-valued data in the spectral domain. To address this limitation, we propose a Complex-Valued Spectra Attention Mechanism, which

treats the real and imaginary components of complex-valued data separately as inputs, enabling effective modeling in the complex domain through real-valued operations, as shown in Figure 1.

**Complex-Valued Spectra Attention Mechanism.** This CV-SAM leverages a fundamental property of complex-valued functions: for any complex-valued function  $f: \mathbf{C}^n \rightarrow \mathbf{C}$  and any complex vector  $x = a + ib$ , the function can be expressed as:

$$f(x) = f(a + ib) = u(a, b) + iv(a, b), \quad (12)$$

where  $u(a, b)$  and  $v(a, b)$  are real-valued functions representing the real and imaginary parts of the complex function, respectively.

Using this property, the attention mechanism decomposes the complex-valued inputs into their real and imaginary parts, applies real-valued transformations independently, and then recombines the results into a single complex-valued output. This approach preserves the integrity of the complex domain while maintaining compatibility with the underlying operations of the Transformer architecture. First, we represent the input spectrum as a complex tensor, where  $t$  is the time step and  $d$  is the feature dimension. Decompose the complex input into real and imaginary parts:

$$X = X_{\text{real}} + iX_{\text{imag}}, \quad X_{\text{real}}, X_{\text{imag}} \in \mathbf{R}^{L \times d} \quad (13)$$

Then, Generate complex query (Q), key (K), value (V) through real projection matrix  $W_Q, W_K, W_V \in \mathbf{C}^{d \times d}$ :

$$\begin{aligned} Q &= X \cdot W_Q = (X_{\text{real}}W_Q) + i(X_{\text{imag}}W_Q), \\ K &= X \cdot W_K = (X_{\text{real}}W_K) + i(X_{\text{imag}}W_K), \end{aligned} \quad (14)$$

$$V = X \cdot W_V = (X_{\text{real}}W_V) + i(X_{\text{imag}}W_V).$$

where  $W_Q, W_K, W_V$  are a real-valued projection matrix, and  $Q, K, V$  are all complex-valued matrices. The attention mechanism is calculated in the complex domain using the following formula:

$$QK^T = (X_{\text{real}}W_Q) + i(X_{\text{imag}}W_Q)(W_K^T X_{\text{real}}^T - iW_K^T X_{\text{imag}}^T) \quad (15)$$

Further decomposition into real and imaginary parts:

$$\begin{aligned} \text{Re}(QK^T) &= X_{\text{real}}W_QW_K^T X_{\text{real}}^T + X_{\text{imag}}W_QW_K^T X_{\text{imag}}^T, \\ \text{Im}(QK^T) &= X_{\text{real}}W_QW_K^T X_{\text{IMAG}}^T - X_{\text{imag}}W_QW_K^T X_{\text{real}}^T \end{aligned} \quad (16)$$

From the perspective of signal processing, this decomposition has clear physical meaning. The real part calculation captures the similarity between amplitude features and effectively reflects the degree of energy alignment of frequency components; while the imaginary part calculation focuses on capturing the interaction of phase differences and shows high sensitivity to frequency offset and phase modulation. Through this design, CV-SAM can simultaneously process amplitude and phase information in spectral data, overcoming the inherent limitations of traditional real-domain attention mechanisms in spectral analysis.

To solve the problem of probability normalization in the complex domain, we propose a solution to treat the real and imaginary parts independently. For complex attention scores  $A = A_{\text{real}} + iA_{\text{imag}}$ , we apply the Softmax function to the real and imaginary parts separately:

$$\text{Softmax}_c(A) = \text{Softmax}(A_{\text{real}}) + i\text{Softmax}(A_{\text{imag}}). \quad (17)$$

The theoretical basis of this normalization strategy is that the complex probability distribution theoretically needs to satisfy the condition of  $\sum |p_j| = 1$ , but directly normalizing

the complex amplitude will inevitably lead to the loss of phase information. By independently normalizing the real and imaginary parts, we ensure the validity of the probability distribution in the real subspace while retaining the statistical characteristics of the phase difference, so that the model can better utilize the rich information provided by the spectral phase.

Finally, the complex attention output is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}_C\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V. \quad (18)$$

Expand to calculate the real and imaginary parts:

$$\begin{aligned} \text{Re}(\text{Attention}) &= \text{Softmax}(\text{Re}(QK^T)) \cdot \text{Re}(V) - \text{Softmax}(\text{Im}(QK^T)) \cdot \text{Im}(V), \\ \text{Im}(\text{Attention}) &= \text{Softmax}(\text{Re}(QK^T)) \cdot \text{Im}(V) + \text{Softmax}(\text{Im}(QK^T)) \cdot \text{Re}(V). \end{aligned} \quad (19)$$

This calculation method combines the interaction between the real and imaginary parts. The real output enhances the signal strength by promoting the feature representation of consistent amplitude, while the imaginary output enhances the model's perception of frequency modulation and phase changes by capturing the dynamic changes of the phase. The synergy of this dual mechanism gives CV-SAM a significant advantage in processing complex spectral data.

#### 4.3. Bidirectional Variable Mamba

While the global attention mechanism in the Transformer architecture captures interactions among all variables, its computational complexity increases exponentially with the number of variables, limiting its applicability in large-scale scenarios. In contrast, the Mamba module efficiently evaluates variable importance through a selection mechanism, with computational overhead increasing linearly with the number of variables. However, due to the unidirectional nature of the selection mechanism, the Mamba module can only establish relationships based on preceding variables, making it challenging to fully capture global variable dependencies. Consequently, the standalone Mamba module exhibits limitations in handling complex multivariable scenarios: (1) It can only capture the dependencies between variables based on a single direction of the input sequence (e.g., from left to right). (2) This unidirectionality may lead to missing global dependencies between variables, especially in complex multivariate scenarios.

To overcome the aforementioned limitations, we propose a Bidirectional Variable Mamba by combining two Mamba modules, enabling global interaction modeling among variables, as shown in Figure 1. Specifically, one Mamba module is designed to capture forward dependencies among variables, while the other focuses on backward dependencies. This bidirectional design effectively addresses the shortcomings of unidirectional mechanisms, allowing the module to comprehensively account for mutual relationships among all variables. By weightedly combining the outputs of the forward and reverse modules, the global interactions between modeling variables are comprehensively modeled. The core of this design is to capture the interrelationships between variables, rather than simply treating them as "past" or "future" dependencies in the time series.

The core of the Mamba module is the state space model (SSM) based on the selection mechanism. Its basic calculation formula is as follows. State update formula: For the input sequence, the state update of the Mamba module can be expressed as:

$$h_t = \mathbf{A}h_{t-1} + \mathbf{B}x_t. \quad (20)$$

The output  $y_t$  of the module is obtained by mapping the hidden state  $h_t$  through the output matrix. In order to apply the continuous state space model on the discrete time series, the parameters and need to be converted through the discretization formula:

**Table 1.** Long-term forecasting dataset description.

Dataset	Dim	Series Length	Dataset Size	Sampling Frequency	Information
ETTM1, ETTM2	7	96	(34465, 11521, 11521)	15mins	Electricity
ETTh1, ETTh2	7	96	(8545, 2881, 2881)	1 hour	Electricity
Electricity	321	96	(18317, 2633, 5261)	1 hour	Electricity
Weather	21	96	(12185, 1757, 3509)	1 hour	Transportation
Traffic	862	96	(36792, 5271, 10540)	10 mins	Weather
Exchange	8	96	(5120, 665, 1422)	1 day	Exchange rate

$$\begin{aligned} \mathbf{A}_{\text{discrete}} &= e^{\Delta \mathbf{A}}, \\ \mathbf{B}_{\text{discrete}} &= (\Delta \mathbf{A})^{-1}(e^{\Delta \mathbf{A}} - \mathbf{I})\Delta \mathbf{B}. \end{aligned} \quad (21)$$

where  $\Delta$  is the time step,  $\mathbf{I}$  is the identity matrix.

In the Mamba module, the selection mechanism enables selective modeling of variable dependencies by allowing parameters  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  to be dynamically adjusted depending on the input  $x_t$ . This input dependency enables the model to dynamically filter irrelevant information and remember relevant information.

Here is how the bidirectional modeling can be expressed mathematically: For the forward Mamba module, capturing forward dependencies:

$$\mathbf{H}_{\text{forward}} = \text{Mamba}(\mathbf{H}_{\text{input}}) \quad (22)$$

For the backward Mamba module, capturing backward dependencies:

$$\mathbf{H}_{\text{backward}} = \text{Mamba}(\mathbf{H}_{\text{input}}^{\text{rev}}) \quad (23)$$

where  $\mathbf{H}_{\text{input}}^{\text{rev}}$  denotes the reversed input sequence to capture backward relationships.

The final output of the Bidirectional Mamba layer combines both directions as follows:

$$\mathbf{H}_{\text{output}} = \alpha \mathbf{H}_{\text{forward}} + \beta \mathbf{H}_{\text{backward}} \quad (24)$$

where  $\alpha$  and  $\beta$  are learnable weights to adaptively balance the contributions of forward and backward dependencies during training. This design ensures comprehensive modeling of all variable interactions.

## 5. Experiments

To evaluate the effectiveness of our proposed DWSL prediction framework, we conducted extensive experiments on eight publicly available datasets. These datasets cover real-world multivariate data, including five power energy datasets and one meteorological dataset. Detailed statistics for each dataset can be found in Table 1. Among these datasets, the Solar-Energy and Electricity datasets are particularly significant due to their inclusion of more variables, allowing for a more comprehensive assessment of the performance of our method. The datasets are described as follows:

1. ETT Dataset[7]: It originates from two power substations and spans from July 2016 to July 2018, recording seven variables, including load and oil temperature. The ETTM1 and ETTM2 datasets are recorded every 15 minutes, with a total of 69,680 time steps. The

ETTh1 and ETTh2 datasets are hourly equivalents of ETTm1 and ETTm2, each containing 17,420 time steps.

2. Electricity Dataset[19]: It contains hourly power consumption data from 321 users, covering a two-year period, with a total of 26,304 time steps.
3. Solar-Energy Dataset[35]: It records solar energy generation from 137 photo-voltaic stations in Alabama in 2006, with samples taken every 10 minutes.
4. Weather Dataset[19]: Collected by the Max Planck Institute for Biogeochemistry's weather stations in 2020, this dataset includes 21 meteorological factors, such as atmospheric pressure, temperature, and humidity, with data recorded every 10 minutes.
5. Traffic Dataset[19]: Covering the period from July 2016 to July 2018, this dataset records road occupancy data from 862 sensors deployed on highways in the San Francisco Bay Area. The data is collected every hour, providing detailed temporal resolution to analyze traffic patterns and dynamics over the two-year period.
6. Through experiments on these diverse real-world datasets, we demonstrate the performance of the SpectroMamba framework, validating its versatility and applicability across various scenarios.

### 5.1. Experimental Settings

In this study, based on previous research [7], we processed the datasets with a step size of 1 to construct diverse input-output pairs, aiming to enhance the model's learning efficiency and generalization capability. To ensure consistency and stability during the model training process, we applied zero-mean normalization preprocessing to the training, validation, and test sets using the mean and standard deviation of the training set. This standardization accelerates the model's convergence rate and improves its performance.

For model configuration, we set the input sequence length  $I$  to 96. We systematically explored various combinations of hyperparameters, including learning rates from  $10^{-4}$  to 0.05, encoder layers from 1 to 3, and model dimensions  $d_{\text{model}}$  ranging from 128 to 512, in an effort to find the optimal configuration that maximizes model performance.

Regarding evaluation metrics, we selected Mean Squared Error (MSE) as the loss function. To comprehensively assess the model's performance, follow [36], we also used both MSE and Mean Absolute Error (MAE), which reflect the model's predictive accuracy from different perspectives.

The Adam optimizer was chosen to train our model due to its efficiency in handling large-scale machine learning tasks. Additionally, we incorporated an early stopping strategy to effectively prevent overfitting and ensure the model's generalization ability on new data. All experiments were conducted on high-performance computing nodes equipped with an NVIDIA RTX 4090 GPU with 24GB of VRAM, and the software environment was developed and deployed using the PyTorch framework.

### 5.2. Compared to Sota Methods

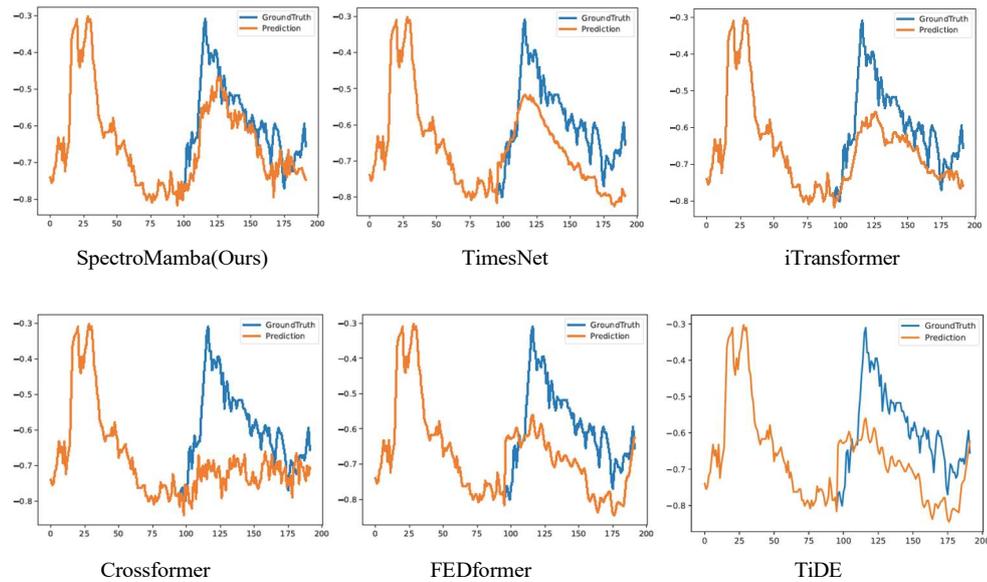
To comprehensively evaluate the effectiveness of the proposed algorithm, this study carefully selects a range of advanced models as comparison benchmarks. These models not only represent different strategies for handling variable interactions and independencies, but also reflect the latest developments in deep learning architectures for time series analysis. In particular, given the exceptional ability of the Transformer architecture to capture long-range dependencies and model complex temporal patterns, we prioritize it as one of the main focuses of our study. On this basis, we chose six representative Transformer-based models, including Autoformer [19], FEDformer [15], Stationary [37], Crossformer [10], and iTransformer [18].

Additionally, considering the growing attention and strong performance of Multi-Layer Perceptron (MLP) models in recent years due to their simplicity and effectiveness, we also included TiDE [38], a prominent and influential MLP variant, in our study. To further enrich the comparison framework, we introduce three CNN models—SCINet [39], TimesNet [9], and another RNN model-DeepAR [40] and GRU [41]. These CNN-based models, with their unique advantages in efficiently extracting multivariate time series features, provide crucial supplementary perspectives for our research.

As shown in Table 2, SpectroMamba demonstrates a significant advantage in both MAE and MSE metrics, thoroughly validating its exceptional performance and generalization capability. In experiments conducted across various public datasets and different prediction time points, SpectroMamba consistently outperforms most comparison models in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE). This is particularly evident in long-term sequence prediction tasks, where its accuracy advantage is most pronounced, highlighting its superior ability to capture complex long-term temporal dependencies.

**Table 2.** Full results of the long-term forecasting task. In accordance with the configuration of TimesNet, we compared widely competitive models at different prediction lengths. The input sequence length for all baseline models was set to 96. "Avg" represents the average results across four different prediction lengths. A lower MSE or MAE indicates better prediction. The best result for each dataset is highlighted in bold red, while the second-best result is in bold with an underline.

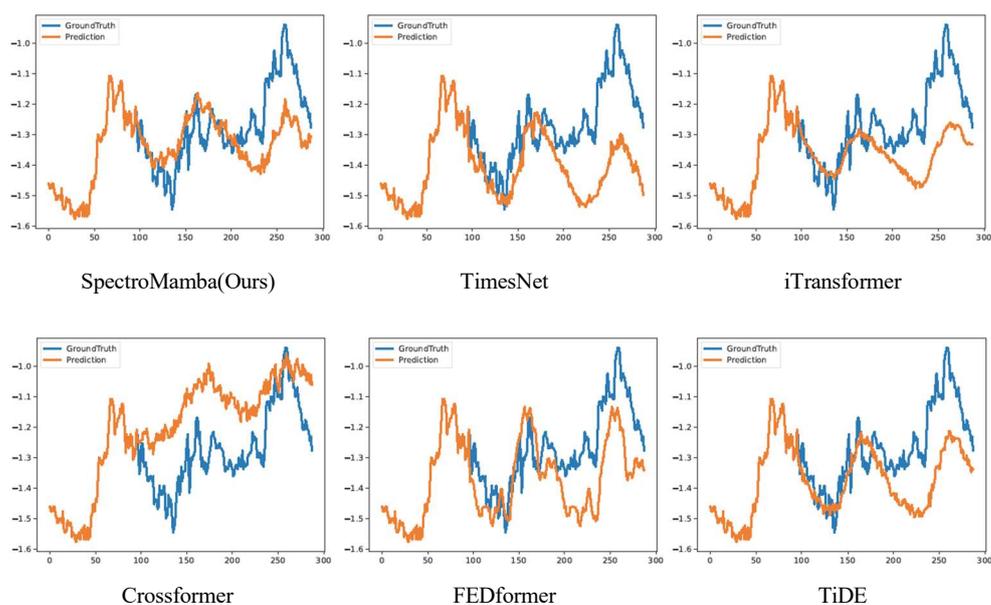
Models	Ours	DeepAR	GRU	iTransformer	TimesNet	DLinear	TiDE	PatchTST	Crossformer	FEDformer	Stationary	
Metric	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	
ETTm1	96	<b>0.330 0.361</b>	1.283 0.857	0.706 0.587	0.334 0.368	0.338 0.375	0.345 0.372	0.364 0.387	0.334 0.372	0.404 0.426	0.379 0.419	0.386 0.398
	192	<b>0.365 0.386</b>	1.181 0.839	0.946 0.738	0.377 0.391	0.374 0.387	0.380 0.389	0.398 0.404	0.378 0.394	0.450 0.451	0.426 0.441	0.459 0.444
	336	<b>0.415 0.409</b>	1.249 0.846	1.048 0.767	0.426 0.420	0.410 0.411	0.413 0.413	0.428 0.425	0.406 0.414	0.532 0.515	0.445 0.459	0.495 0.464
	720	<b>0.485 0.455</b>	1.075 0.770	1.076 0.765	0.491 0.459	0.478 0.450	0.474 0.453	0.487 0.461	0.462 0.445	0.666 0.589	0.543 0.490	0.585 0.516
	Avg	<b>0.399 0.403</b>	1.197 0.853	0.944 0.714	0.407 0.410	0.400 0.406	0.403 0.407	0.419 0.419	0.395 0.406	0.513 0.495	0.448 0.452	0.481 0.456
ETTm2	96	<b>0.177 0.259</b>	3.418 1.581	0.487 0.518	0.180 0.264	0.187 0.267	0.193 0.292	0.207 0.305	0.175 0.259	0.287 0.366	0.203 0.287	0.192 0.274
	192	<b>0.243 0.306</b>	3.894 1.635	1.725 0.984	0.250 0.309	0.249 0.309	0.284 0.362	0.290 0.364	0.240 <b>0.302</b>	0.414 0.492	0.269 0.328	0.280 0.339
	336	<b>0.308 0.341</b>	3.247 1.377	2.091 1.082	0.311 0.348	0.321 0.351	0.369 0.427	0.377 0.422	0.302 0.342	0.597 0.542	0.325 0.366	0.334 0.361
	720	<b>0.403 0.401</b>	2.588 1.313	2.301 1.144	0.412 0.407	0.408 0.403	0.554 0.522	0.558 0.524	0.399 <b>0.397</b>	1.730 1.042	0.421 0.415	0.417 0.413
	Avg	<b>0.283 0.327</b>	3.286 1.476	1.651 0.932	0.288 0.332	0.291 0.333	0.350 0.401	0.358 0.404	0.279 0.325	0.757 0.611	0.305 0.349	0.571 0.537
ETT1h	96	<b>0.377 0.397</b>	1.128 0.81	1.126 0.831	0.386 0.405	0.384 0.402	0.386 0.400	0.479 0.464	0.414 0.419	0.423 0.448	0.390 0.404	0.513 0.491
	192	<b>0.425 0.426</b>	1.077 0.776	1.133 0.790	0.441 0.436	0.436 0.429	0.437 0.432	0.525 0.492	0.460 0.445	0.471 0.474	<b>0.420 0.448</b>	0.534 0.504
	336	<b>0.468 0.452</b>	1.043 0.766	1.053 0.774	0.487 0.458	0.491 0.469	0.481 0.459	0.565 0.515	0.501 0.466	0.570 0.546	<b>0.459 0.465</b>	0.588 0.535
	720	<b>0.496 0.487</b>	3.134 1.352	3.138 1.343	0.503 0.491	0.521 0.500	0.519 0.516	0.594 0.558	0.500 0.488	0.653 0.621	<b>0.506 0.507</b>	0.643 0.616
	Avg	0.442 0.441	1.595 0.926	1.612 0.934	0.454 0.447	0.458 <b>0.450</b>	0.456 0.452	0.541 0.507	0.469 0.455	0.529 0.522	<b>0.440 0.460</b>	0.570 0.537
ETT2h	96	<b>0.249 0.326</b>	2.602 1.329	2.100 1.094	0.297 0.349	0.340 0.374	0.333 0.387	0.400 0.440	0.302 0.348	0.745 0.584	0.358 0.397	0.476 0.458
	192	<b>0.337 0.382</b>	3.198 1.368	2.548 1.249	0.380 0.400	0.402 0.414	0.477 0.476	0.528 0.509	0.388 0.400	0.877 0.656	0.429 0.439	0.512 0.493
	336	<b>0.405 0.420</b>	3.139 1.353	3.142 1.343	0.428 0.432	0.452 0.452	0.594 0.541	0.643 0.571	0.426 0.433	1.043 0.731	0.496 0.487	0.552 0.551
	720	<b>0.415 0.413</b>	3.134 1.352	3.138 1.343	0.427 <b>0.445</b>	0.462 0.468	0.831 0.657	0.874 0.679	0.431 0.446	1.104 0.763	0.463 0.474	0.562 0.560
	Avg	<b>0.352 0.385</b>	3.018 1.350	2.732 1.257	0.383 0.407	0.414 0.427	0.559 0.515	0.611 0.550	0.387 0.407	0.942 0.684	0.437 0.449	0.526 0.516
Electricity	96	<b>0.142 0.239</b>	0.575 0.535	0.404 0.434	<b>0.148 0.240</b>	0.168 0.272	0.197 0.282	0.237 0.329	0.195 0.285	0.219 0.314	0.193 0.308	0.169 0.273
	192	<b>0.158 0.253</b>	1.061 0.809	0.844 0.717	<b>0.162 0.253</b>	0.184 0.289	0.196 0.285	0.236 0.330	0.199 0.289	0.231 0.322	0.201 0.315	0.182 0.286
	336	<b>0.175 0.267</b>	1.04 0.795	1.015 0.775	<b>0.178 0.269</b>	0.198 0.300	0.209 0.301	0.249 0.344	0.215 0.305	0.246 0.337	0.214 0.329	0.200 0.304
	720	<b>0.209 0.303</b>	1.048 0.804	1.041 0.783	<b>0.225 0.317</b>	0.220 0.320	0.245 0.333	0.284 0.373	0.256 0.337	0.280 0.363	0.246 0.355	0.222 0.321
	Avg	<b>0.171 0.266</b>	0.931 0.735	0.826 0.677	<b>0.178 0.270</b>	0.193 0.295	0.212 0.300	0.252 0.344	0.216 0.304	0.244 0.334	0.214 0.327	0.193 0.296
Traffic	96	<b>0.199 0.235</b>	1.377 0.717	1.349 0.715	<b>0.395 0.268</b>	0.593 0.321	0.650 0.396	0.805 0.493	0.544 0.359	0.522 0.290	0.587 0.366	0.612 0.338
	192	<b>0.299 0.256</b>	1.442 0.75	1.42 0.75	<b>0.417 0.276</b>	0.617 0.336	0.598 0.370	0.756 0.474	0.540 0.354	0.530 0.293	0.604 0.373	0.613 0.340
	336	<b>0.244 0.271</b>	1.489 0.778	1.470 0.776	<b>0.433 0.283</b>	0.629 0.336	0.605 0.373	0.762 0.477	0.551 0.358	0.558 <b>0.305</b>	0.621 0.383	0.618 0.328
	720	<b>0.246 0.272</b>	1.526 0.793	1.524 0.794	<b>0.467 0.302</b>	0.640 0.350	0.645 0.394	0.719 0.449	0.586 0.375	0.589 0.328	0.626 0.382	0.653 0.355
	Avg	<b>0.247 0.259</b>	1.458 0.759	1.440 0.758	<b>0.428 0.282</b>	0.620 0.336	0.625 0.383	0.761 0.473	0.555 0.362	0.550 0.304	0.610 0.376	0.624 0.340
Weather	96	0.171 0.210	0.278 0.301	0.183 0.231	0.174 0.214	0.172 <b>0.220</b>	0.196 0.255	0.202 0.261	0.177 0.218	<b>0.158 0.230</b>	0.217 0.296	0.173 0.223
	192	<b>0.212 0.248</b>	0.376 0.369	0.299 0.336	0.221 0.254	<b>0.219 0.261</b>	0.237 0.296	0.242 0.298	0.225 0.259	0.260 0.277	0.276 0.336	0.245 0.285
	336	<b>0.262 0.287</b>	0.568 0.527	0.376 0.374	0.278 0.296	0.280 0.306	0.283 0.335	0.287 0.335	0.278 <b>0.297</b>	<b>0.272 0.335</b>	0.339 0.380	0.321 0.338
	720	0.331 0.336	0.571 0.533	0.459 0.433	0.358 0.349	0.365 0.359	0.345 0.381	0.351 0.386	0.354 <b>0.348</b>	0.398 0.418	0.403 0.428	0.414 0.414
	Avg	<b>0.244 0.27</b>	0.448 0.432	0.329 0.343	0.258 0.279	<b>0.259 0.287</b>	0.265 0.317	0.271 0.320	<b>0.259 0.281</b>	0.272 0.315	0.309 0.360	0.288 0.314



**Figure 2.** Visualization of input-96-predict-96 results on the ETTm1 dataset.

To enhance clarity, we visualized the prediction performance of SpectroMamba on the ETTm1 dataset for forecast horizons of 96, 192, 336, and 720 steps. Figures 2, 3, 4, and 5 illustrate the corresponding prediction curves, vividly demonstrating the model's remarkable capability to capture both short-term and long-term dependencies with precision.

Compared to advanced Transformer-based models such as iTransformer and Crossformer, SpectroMamba exhibits lower prediction errors. This is attributed to its effective extraction of multi-scale features through the spectral decomposition module, coupled with a dynamic self-attention mechanism that enhances the modeling of interactions between the time and variable dimensions. Additionally, when compared to methods like TimesNet and TiDE, SpectroMamba demonstrates stronger adaptability in handling different types of time series data. Whether dealing with the highly periodic solar energy data or the non-periodic electricity load data, its prediction results remain consistently stable and efficient.



**Figure 3.** Visualization of input-96-predict-192 results on the ETTm1 dataset.

Furthermore, SpectroMamba excels not only in single-domain tasks, such as weather data prediction, but also in cross-domain tasks, such as distributed photovoltaic power forecasting, showcasing its robustness and generalization ability. Its well-designed parameters and moderate model size help to improve prediction accuracy while effectively avoiding overfitting, resulting in high training efficiency.

Overall, by integrating multi-scale feature extraction and dynamic variable dependency modeling, SpectroMamba excels not only in capturing global features but also in uncovering fine-grained dependencies within the data, making it a powerful tool for a wide range of time series forecasting tasks.

## 6. Ablation Study

### Individual Component

To validate the effectiveness of the Spectral Attention Module and the Bidirectional Variable Mamba in the SpectroMamba model, we conducted detailed ablation experiments on the ETTh1 and ETTm1 datasets, analyzing the specific contribution of these two components to the model's performance. The experimental design included two sets of comparative models: 1) Removing the Spectral Attention Module from the SpectroMamba framework, labeled as "w/o SAM"; 2) Removing the Bidirectional Variable Mamba, labeled as "w/o BV-Mamba." 3) Replace SAM with traditional self-attention mechanism, labeled as "self-attention".

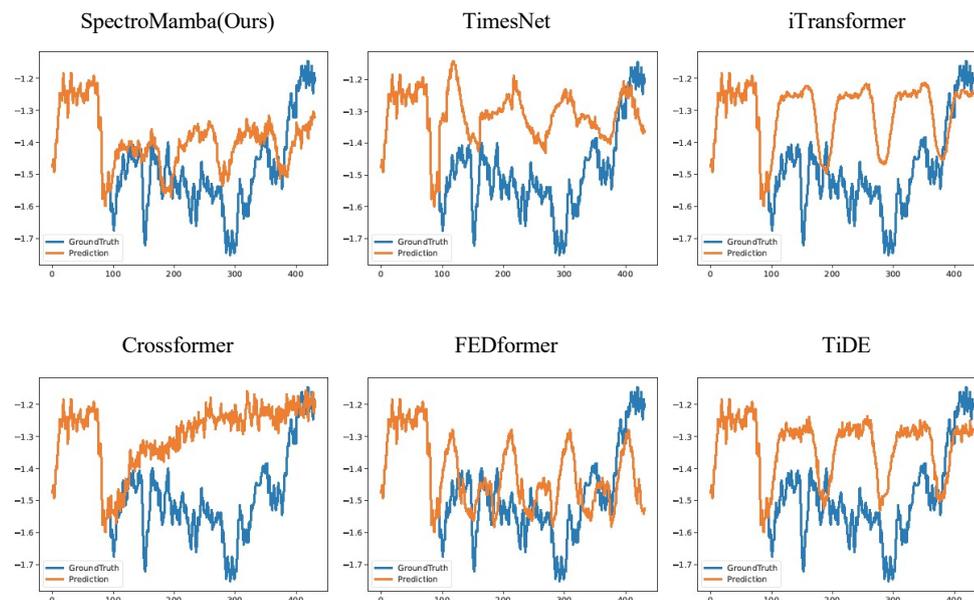


Figure 4. Visualization of input-96-predict-336 results on the ETTm1 dataset.

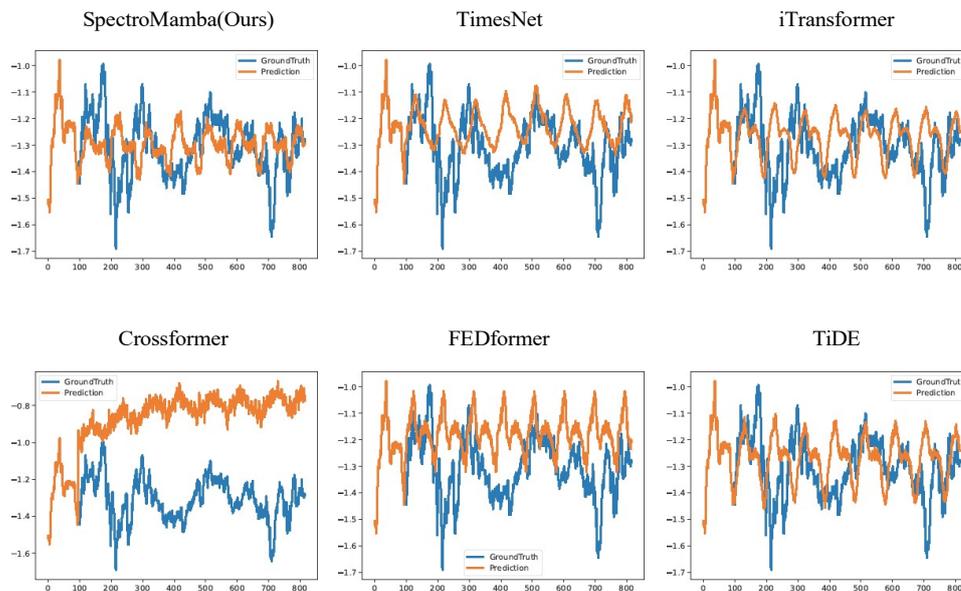
Table 3. Ablation study of individual component.

Models	Ours		w/o SAM	w/o BV-Mamba	Self-attention				
	MSE	MAE	MSE	MAE	MSE	MAE			
ETTh1	96	<b>0.330</b>	<b>0.361</b>	0.333	0.365	0.332	0.364	0.339	0.368
	192	<b>0.365</b>	<b>0.386</b>	0.370	0.389	0.369	0.389	0.373	0.393
	336	<b>0.415</b>	<b>0.409</b>	0.419	0.415	0.420	0.417	0.429	0.422
	720	<b>0.485</b>	<b>0.455</b>	0.491	0.459	0.490	0.458	0.450	0.468

	Avg	<b>0.399 0.403</b>	0.403 0.407	0.403 0.407	0.398 0.413
<b>ETTh1</b>	96	<b>0.377 0.397</b>	0.382 0.445	0.385 0.448	0.383 0.406
	192	<b>0.425 0.426</b>	0.429 0.432	0.432 0.435	0.434 0.429
	336	<b>0.468 0.452</b>	0.473 0.455	0.475 0.457	0.477 0.461
	720	<b>0.496 0.487</b>	0.501 0.490	0.500 0.490	0.516 0.501
	Avg	<b>0.442 0.441</b>	0.446 0.456	0.448 0.458	0.453 0.449

To ensure fairness and the comparability of results, all models were tested under the same dataset, experimental conditions, and hyperparameter settings, including the same learning rate, batch size, and number of iterations.

The ablation study presented in Table 3 provides a detailed analysis of the individual contributions of the spectral attention module (SAM) and the bidirectional variable Mamba (BV-Mamba) to the overall performance of the SpectroMamba model. The results clearly indicate that the full model, which incorporates both SAM and BV-Mamba, consistently outperforms the ablated versions across various datasets (ETTm1 and ETTh1) and different prediction horizons (96, 192, 336, and 720 time steps). In particular, when the SAM module is removed, the model shows a slight increase in MSE and MAE, indicating that SAM plays a critical role in effectively capturing multi-scale spectral features and enhancing the model's ability to learn complex temporal dependencies. Similarly, when the BV-Mamba module is omitted, the performance degradation is also noticeable, particularly for longer forecasting horizons, suggesting that BV-Mamba is vital for capturing the interdependencies between variables over extended time periods.

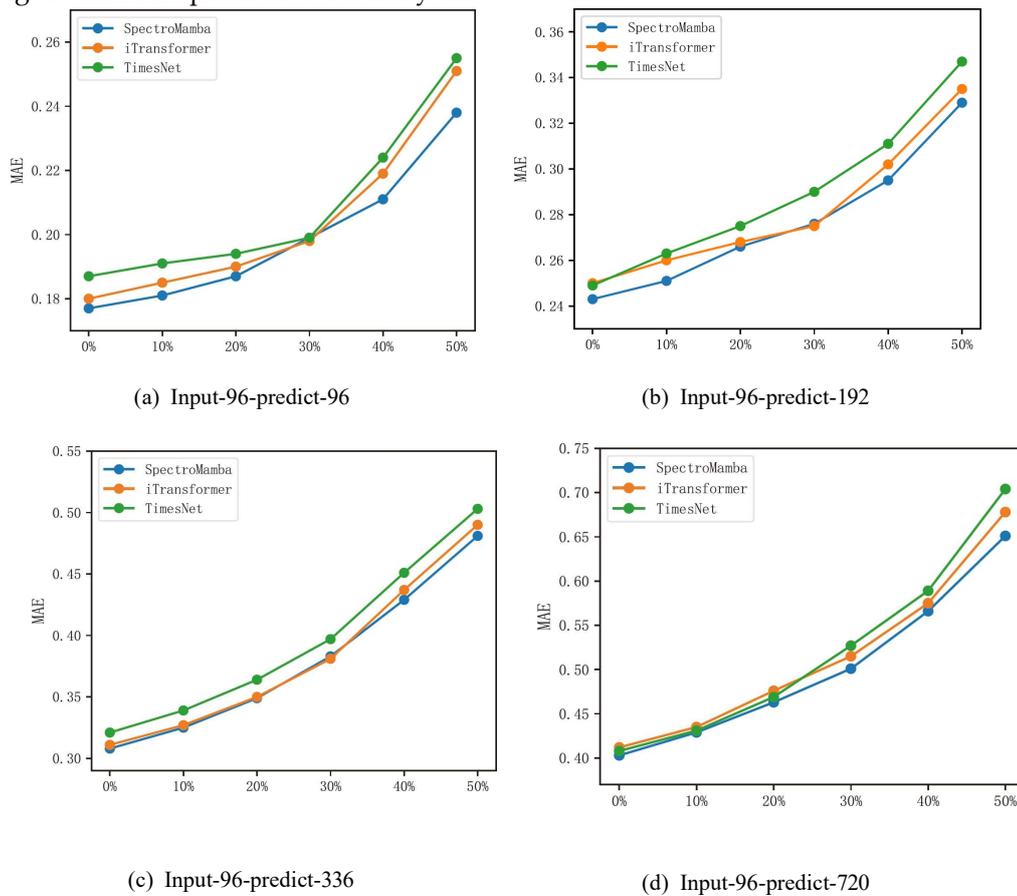


**Figure 5.** Visualization of input-96-predict-720 results on the ETTm1 dataset.

As shown in Table 3, the spectral attention module consistently demonstrated significant advantages across all prediction horizons, particularly in handling datasets with complex spectral features. For instance, in the 720-step prediction task on the ETTm1 dataset, the model equipped with the spectral attention module achieved MSE and MAE values of 0.485 and 0.455, respectively, compared to 0.450 and 0.468 for the traditional self-attention mechanism. This represents improvements of 7.2% and 2.8% in MSE and MAE, respectively.

These experimental results highlight the superior ability of the spectral attention module to capture spectral features and complex temporal dependencies in the data, compared to the traditional self-attention mechanism. Particularly for long-term prediction tasks, the spectral

attention module excels at extracting and leveraging multi-scale features, significantly enhancing the model's predictive accuracy and robustness.



**Figure 6.** Comparison of the average MAE performance across three runs on different datasets under varying noise levels.

To systematically evaluate the performance stability of the proposed model when confronted with noise in the input data, we conducted experiments on the ETTm2 dataset. In these experiments, Gaussian noise was introduced to a selected subset of the input data to simulate real-world uncertainties and disturbances. Specifically, for any given time series input  $X_{(t-T+1:t)}$ , Gaussian perturbations were applied, as mathematically expressed below:

$$\begin{aligned}\hat{X}_{(t-T+1:t)} &= X_{(t-T+1:t)} + M \otimes \hat{\tau} \\ \hat{\tau} &= (V_{max}^i - V_{min}^i) \times \tau\end{aligned}\quad (25)$$

where  $M$  represents a randomly generated binary mask used to adjust the noise injection ratio, this ratio was set at 10%, 20%, 30%, 40%, and 50% to comprehensively evaluate the model's performance under different noise levels.  $V_{max}^i$  and  $V_{min}^i$  refer to the maximum and minimum values of the  $i$ -th feature dimension, respectively, while  $\tau \sim Gauss(0, 1)$  is a random sample drawn from a standard normal distribution, serving as the noise source.

Figure 6 presents a comparison of the MAE for the SpectroMamba, iTransformer, and TimesNet models on the ETTm2 dataset under different noise levels. As the noise intensity increased, SpectroMamba exhibited a slower MAE growth trend compared to iTransformer and TimesNet. This result not only confirms the superior robustness of SpectroMamba in handling noisy data but also provides strong theoretical support for its ability to manage abnormal data fluctuations in practical applications.

## 7. Conclusion

In this paper, we presented SpectroMamba, a novel framework for multivariate time series forecasting (MTSF) that effectively models both temporal dependencies and cross-variable relationships. Our proposed Spectral Attention Module (SAM) integrates both time-domain and frequency-domain information, capturing local and global dependencies in time series data. The module incorporates the Extended Discrete Fourier Transform (EDFT) to address frequency misalignment issues and employs a Complex-Valued Spectral Attention Mechanism (CV-SAM) to identify and utilize intricate relationships among frequency combinations. Complementing SAM, the Bidirectional Variable Mamba (BV-Mamba) efficiently captures cross-variable relationships, overcoming the trade-offs between variable-mixing and variable-independent methods. Through comprehensive experiments on multiple real-world datasets, SpectroMamba consistently outperformed state-of-the-art models in both short-term and long-term forecasting tasks. These results demonstrate its ability to extract meaningful dependencies from complex multivariate data, significantly enhancing predictive accuracy.

## References

1. Zhuang, W., Fan, J., Fang, J., Fang, W., Xia, M.: Rethinking general time series analysis from a frequency domain perspective. *Knowledge-Based Systems* **301**, 112281 (2024)
2. Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., Niu, Z.: Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems* **36** (2024)
3. Luo, Y., Lyu, Z., Huang, X.: Tfdnet: Time-frequency enhanced decomposed network for long-term time series forecasting. *arXiv preprint arXiv:2308.13386* (2023)
4. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, (2015)
5. Hochreiter, S.: *Long short-term memory*. Neural Computation MIT-Press (1997)
6. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018)
7. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: 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)
8. Xu, Z., Zeng, A., Xu, Q.: Fits: Modeling time series with 10k parameters. *arXiv preprint arXiv:2307.03756* (2023)
9. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186* (2022)
10. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: *The Eleventh International Conference on Learning Representations* (2023)
11. Qi, S., Wen, L., Li, Y., Yang, Y., Li, Z., Rao, Z., Pan, L., Xu, Z.: Enhancing multivariate time series forecasting with mutual information-driven cross-variable and temporal modeling. *arXiv preprint arXiv:2403.00869* (2024)
12. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022)
13. Zhou, X., Wang, W., Buntine, W., Qu, S., Sriramulu, A., Tan, W., Bergmeir, C.: Scalable transformer for high dimensional multivariate time series forecasting. In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3515–3526 (2024)
14. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 11121–11128 (2023)

15. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International Conference on Machine Learning, pp. 27268–27286 (2022). PMLR
16. Yu, G., Zou, J., Hu, X., Aviles-Rivero, A.I., Qin, J., Wang, S.: Revitalizing multi-variate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. arXiv preprint arXiv:2402.12694 (2024)
17. Yu, C., Wang, F., Shao, Z., Sun, T., Wu, L., Xu, Y.: Dsformer: A double sampling transformer for multivariate time series long-term prediction. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 3062–3072 (2023)
18. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2023)
19. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems **34**, 22419–22430 (2021)
20. Zhou, T., Ma, Z., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R., *et al.*: Film: Frequency improved legendre memory model for long-term time series forecasting. Advances in neural information processing systems **35**, 12677–12690 (2022)
21. Sun, F.-K., Boning, D.S.: Fredo: frequency domain-based long-term time series forecasting. arXiv preprint arXiv:2205.12301 (2022)
22. Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., *et al.*: Spectral temporal graph neural network for multivariate time-series forecasting. Advances in neural information processing systems **33**, 17766–17778 (2020)
23. Chiu, C.-C., Raffel, C.: Monotonic chunkwise attention. arXiv preprint arXiv:1712.05382 (2017)
24. Wang, J., Zhu, W., Wang, P., Yu, X., Liu, L., Omar, M., Hamid, R.: Selective structured state-spaces for long-form video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6387–6397 (2023)
25. Ali, A., Zimerman, I., Wolf, L.: The hidden attention of mamba models. arXiv preprint arXiv:2403.01590 (2024)
26. Yang, S., Wang, Y., Chen, H.: Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 296–306 (2024). Springer
27. Qiao, Y., Yu, Z., Guo, L., Chen, S., Zhao, Z., Sun, M., Wu, Q., Liu, J.: Vlmamba: Exploring state space models for multimodal learning. arXiv preprint arXiv:2403.13600 (2024)
28. Lu, Y., Wang, S., Wang, Z., Xia, P., Zhou, T., *et al.*: Lfmamba: Light field image super-resolution with state space model. arXiv preprint arXiv:2406.12463 (2024)
29. Burke, K., Wagner, L.O.: Dft in a nutshell. International Journal of Quantum Chemistry **113**(2), 96–101 (2013)
30. Oppenheim, A.V., Verghese, G.C.: Signals, Systems & Inference. Pearson London, (2017)
31. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
32. Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. Advances in neural information processing systems **34**, 572–585 (2021)
33. Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C.: Hippo: Recurrent memory with optimal polynomial projections. Advances in neural information processing systems **33**, 1474–1487 (2020)
34. Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., Choo, J.: Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations (2021)
35. Lai, G., Chang, W.-C., Yang, Y., Liu, H.: Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95–104 (2018)

36. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science* **7**, 623 (2021)
37. Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems* **35**, 9881–9893 (2022)
38. Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., Yu, R.: Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424* (2023)
39. Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., Xu, Q.: Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems* **35**, 5816–5828 (2022)
40. Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting* **36**(3), 1181–1191 (2020)
41. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014)

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.