Modeling Dependencies across Arbitrary Positions for High-Dimensional Long-Term Series Forecasting

Anonymous Author(s)
Affiliation
Address

Abstract

Modeling dependencies across arbitrary positions is crucial for accurate high-dimensional long-term time series forecasting, yet many existing methods face limited dependency modeling and computational inefficiency due to their reliance on localized paradigms and Transformer architectures. To address these, we propose replacing Self-Attention with autocorrelation, achieving two key innovations: 1) We propose calculating autocorrelation across both variable and time dimensions, which is a global paradigm, to model dependencies across arbitrary positions. 2) Our proposed Spectral Product Mechanism (SPM) reformulates autocorrelation as spectral product and reduces the complexity from $O(N^2)$ to $O(NlogN)$, while its Hadamard product-based correlation score matrix further reduces core computation to $O(N)$ compared to Self-Attention's $O(N^2)$ matrix multiplication. We further propose a Generalized Spectral Product Mechanism (GSPM), which extends traditional autocorrelation by mapping input into distinct feature representations, enabling modeling of complex dependencies through cross-feature correlations. Our method surpasses current state-of-the-art (SOTA) methods on numerous authoritative benchmarks, achieving the lowest average MSE and MAE across the high-dimensional Traffic and Electricity benchmarks. Anonymous code is available at: `https://anonymous.4open.science/r/SPM-61F7`

1 Introduction

Time series forecasting is widely applied in various fields such as finance [1], meteorology [2], and power [3] forecasting. High-dimensional multivariate long-term series prediction [4, 5], a crucial aspect of this field, aims to accurately predict future sequence trends over extended periods based on historical data with high-dimensional variables. This data form can be regarded as a 2D plane composed of the dimensions of time and variables. However, accurately modeling long-term dependencies in high-dimensional series is challenging because each sample point may have dependencies with any other sample point at arbitrary positions on the 2D plane. It requires the ability to model dependencies across arbitrary positions. These dependencies span both the variable (vertical) and time (horizontal) dimensions [6], and can be in arbitrary directions and distances. For instance, in traffic flow prediction, congestion at the first intersection (variable 1) is very likely to trigger congestion in the adjacent intersections (other variables), but this effect occurs with a time lag. Therefore, accurate prediction requires modeling the dependencies across time steps and variables, that is, having the ability to model the dependencies between any positions on the 2D plane.

However, most existing methods are local modeling paradigms and lack this ability. They tend to decompose the 2D plane into various local segments, then model these local relationships to capture long-term dependencies across variables. Specifically, they could be roughly divided into four categories as shown in Figure 1. Figure 1(a) shows the variable-independent paradigm [7, 8, 9, 10], which can only model dependencies in the time (horizontal) direction. Figure 1(b) shows the paradigm

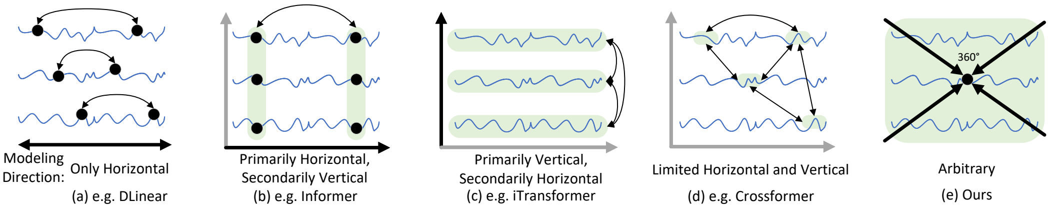


Figure 1: Most existing methods (a, b, c, d) focus on exploring rule-based paradigms. Instead, we abandon rules and directly model dependencies across arbitrary positions through the autocorrelation theory in the 2D plane (e). Then, we design a spectral product mechanism to reduce complexity.

that embeds all variable information at each time step into independent feature vectors [11, 12, 13]. It primarily models dependencies in the time (horizontal) direction with the variable (vertical) direction as secondary. In addition, Figure 1(c) shows the paradigm that independently embeds each variable series into feature vectors [4, 14]. It primarily models dependencies in the variable (vertical) direction with the time (horizontal) direction as secondary. To address these shortcomings, Yu, et al. [15] takes the paradigms in Figure 1(a) and (c) as two branches and then combines their results. However, the dependencies of different sampling points in any direction cannot be linearly decomposed into the superposition of horizontal and vertical directions as a whole. Therefore, this method does not accurately model the dependencies in any direction. Moreover, Figure 1(d) shows the paradigm that segments each variable sequence into patches and models dependencies between patches across variables [5]. It offers more oblique directional dependency modeling capabilities than the other paradigms. However, the fixed length of patch causes the loss of boundary information and semantic discontinuity in time series [16], resulting in inaccurate dependencies modeling. Building upon these paradigms, many SOTA methods extensively adopt the Transformer as the backbone model, resulting in high computational complexity.

To model dependencies across arbitrary positions while reducing the computational complexity of the Transformer, we propose a Spectral Product Mechanism (SPM). Unlike existing local modeling paradigms, SPM introduces a global modeling paradigm, as illustrated in Figure 1(e). Since the function of the Transformer is to calculate the correlation among local parts, we employ autocorrelation with the same function to replace it. We calculate autocorrelation across both the variable and time dimensions, which helps to model dependencies at arbitrary positions. In order to further model complex dependencies, we map the input to different feature representations and calculate their correlations. Since they essentially represent the same input, what is calculated is a generalized version of autocorrelation. It can separate different important features from the input signal and then capture more complex dependencies from the cross-feature correlations. We call this method GSPM.

However, the computational complexity of autocorrelation is still $O(N^2)$. Therefore, we reformulate it as self-convolution, then leverage convolution's simplifiability to transform self-convolution into the spectral product, which is $O(NlogN)$. This leads to three advantages. Firstly, it avoids the phase loss caused by the equivalence of the power spectrum with traditional method [12, 17]. Therefore, we do not adopt this traditional method. Secondly, since the spectral product is equivalent to the global convolution [18, 19], it can more conveniently incorporate the correlation scores into the input sequence. Thirdly, its global convolution avoids the local perception problem of CNN [20], achieving global modeling of Self-Attention. From the spectral product perspective, the Hadamard product-based correlation score matrix reduces the core computation to $O(N)$, while Self-Attention's matrix multiplication-based score matrix is $O(N^2)$. Despite Autoformer [12] and FEDformer [21] have respectively reduced complexity to $O(NlogN)$ and $O(N)$, their actual overhead of time, memory, and parameter number are respectively $\{56.12\%, 176.22\%\}$, $\{62.14\%, 12.74\%\}$, and $\{785.89\%, 2769.94\%\}$ higher than ours. Compared with the latest SOTA methods, our method achieves the lowest average MSE and MAE on the authoritative high-dimensional Traffic and Electricity benchmarks. Compared with Transformer- and Mamba- based methods, our method achieves excellent results in terms of resource overhead of time, memory, and parameter number, ranking 1st, 2nd, and 2nd, respectively.

Our contributions are: 1) We find that most existing methods cannot model dependencies across arbitrary positions. To address this, we propose calculating autocorrelation across both variable and time dimensions. It replaces the self-attention mechanism because it achieves the same function, which is to calculate the local correlation. 2) Then, we reformulate it as the Spectral Product Mechanism (SPM), reducing the complexity from $O(N^2)$ to $O(NlogN)$. 3) Finally, we extend the SPM to a generalized form (GSPM) to model more complex dependencies.

2 Related work

Existing approaches model the long-term dependencies across variables in several directions: 1) The methods such as PatchTST [7], DLinear [8], TimesNet [22], TimeMixer [9], and WPMixer [10] adopt a channel-independent strategy that runs the model in parallel across each variable sequence for independent prediction. This strategy determines the model coefficients by summing the ACFs of all channels, which can mitigate distribution drift and improve the model's prediction accuracy [23]. However, iTransformer [4] indicates that this approach might lead to suboptimal outcomes due to the neglect of inter-variable dependencies. 2) The methods like Informer [11], Autoformer [12], and Liu, et al. [24, 13, 21] process each time step's variable information independently through local segmentation and models them accordingly. This approach often results in attention maps lacking in meaningful information due to the insufficient semantic integration of variables, consequently reducing the accuracy of modeling variable relationships. Our method differs from FEDformer [21] in two aspects. The first is that FEDformer follows a local paradigm (Figure 1(b)), while ours follows a global modeling paradigm (Figure 1(e)). As a result, FEDformer applies 1D FFT in the time dimension, making it difficult to establish dependencies between arbitrary positions, while we can establish such dependencies through 2D FFT. The second is that we employ the spectrum to achieve the function of the Self-Attention mechanism with lower complexity, while FEDformer directly employs the vanilla Self-Attention mechanism to learn the spectrum.

3) Unlike like the above methods, Crossformer [5] segments each variable sequence into patches and models dependencies between patches across variables, allowing for more flexible global modeling. However, the fixed length of patch causes the loss of boundary information and semantic discontinuity of time series data [16]. Furthermore, patching increases the maximum path length at the point level [5]. Therefore, Crossformer [5] struggles to accurately model the dependencies across arbitrary positions. Similarly, Msgnet [25] only interacts with patches that have overlapping time ranges, learning positive and negative correlations. 4) To avoid the issues introduced by patching, iTransformer [4] and S-Mamba [14] embed entire variable sequences into feature vectors and model variable relationships directly. This operation has improved prediction accuracy and gained widespread recognition. Additionally, Leddam [15] adds a channel-independent learning branch on this basis to take into account the dependency modeling of inter- and intra-variables. Nevertheless, their design prevents the modeling across arbitrary positions, limiting further improvements.

3 Method

In this section, we first prove that the spectral product can achieve the function of Self-Attention (§ 3.2). Then, we present its specific design (§ 3.4) and how to extend it to the generalized form (§ 3.5).

3.1 Problem Definition

In multivariate time series forecasting, we represent the input multiple time series as $\boldsymbol{x} \in R^{M\times L}$, where M is the number of variate and L is the size of look-back window. For each single series of $i\text{-}th$ variate $\boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_L^{(i)}) \in R^{1\times L}$, where $i = 1, ..., M$, the goal is to forecast T future values $\boldsymbol{y}^{(i)} = (x_{L+1}^{(i)}, \ldots, x_{L+T}^{(i)}) \in R^{1\times T}$. We represent the multivariate prediction result as $\boldsymbol{y} \in R^{M\times T}$.

3.2 A Functional Replacement for Self-Attention

Essentially, the Self-Attention mechanism and autocorrelation achieve the same function because they both calculate the relevance among local parts through vector inner products. As shown in Figure 2, the first step is to prove that self-convolution in signal processing can achieve the calculation of autocorrelation. In signal processing, 2D discrete self-convolution's input are two same 2D discrete signals $f[x, y]$. It is defined as:

$$(f * f)[u, v] = \textstyle\sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f[x, y] \cdot f[u - x, v - y], \tag{1}$$

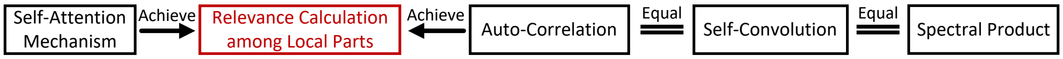


Figure 2: The red words refer to the function that needs to maintain consistency before and after replacement. Spectral product can replace the functions of the Self-Attention mechanism.

where x and y are the index, u and v are the delay, m and n are the index boundary. Flip one of the input signals and obtain:

$$(f * \tilde{f})[u,v] = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f[x,y] \cdot f[u+x, v+y], \tag{2}$$

where $\tilde{f}[x,y] = f[-x,-y]$. It is equivalent to the definition of 2D discrete autocorrelation:

$$R_f[u,v] = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f[x,y] \cdot f[x+u, y+v]. \tag{3}$$

Therefore, self-convolving a flipped input signal can achieve the calculation of autocorrelation. According to the convolution theorem [18], the Fourier transform of a convolution is given by:

$$\mathcal{F}\{f[x,y] * g[x,y]\} = \mathcal{F}\{f[x,y]\} \cdot \mathcal{F}\{g[x,y]\}, \tag{4}$$

it proves that the Fourier transform $\mathcal{F}$ of time convolution equals the spectral product (detailed proof is shown in Appendix). In this way, calculating the spectral product (Hadamard product) of the input signal and flipped input signal, then performing Inverse Fast Fourier Transform (IFFT) represents calculating their time-domain convolution, which further represents calculating autocorrelation:

$$\mathcal{F}^{-1}\{\mathcal{F}\{f[x,y]\} \cdot \mathcal{F}\{\tilde{f}[x,y]\}\} = R_f[u,v]. \tag{5}$$

Compared to the matrix multiplication in the Self-Attention mechanism, which also achieves the function of relevance calculation among local parts, Hadamard product in the spectral product reduces the computational complexity from $O(N^2)$ to $O(N)$. Additionally, FFT and IFFT have $O(NlogN)$ computational complexity.

3.3 Overview Framework

To effectively mitigate the impact of distribution drift between training and test data, we apply reversible instance normalization [26, 27] to the input, normalizing each sample point to have zero mean and unit standard deviation. Finally, these statistics are added to the prediction results.

Figure 3 shows the framework of the proposed method. Initially, since we need to calculate the autocorrelation across variables and time dimensions and reformulate it as a spectral product, we should apply 2D instead of 1D FFT. We apply 2D FFT to convert the input into a 2D spectrum, which is subsequently embedded as digital vectors to capture semantic information. These are expressed as:

$$\boldsymbol{x}_s = 2DFFT(\boldsymbol{x}), \quad \boldsymbol{x}_e = Embed(\boldsymbol{x}_s), \tag{6}$$

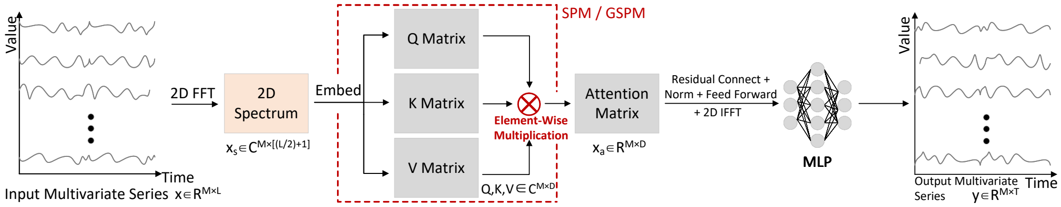
where $\boldsymbol{x}$, $2DFFT(\cdot)$, $\boldsymbol{x}_s$, $Embed(\cdot)$, and $\boldsymbol{x}_e$ represents the input multiple time series, 2D FFT, 2D spectrum, linear mapping embedding, and embedded 2D spectrum, respectively. Furthermore, $\boldsymbol{x} \in R^{M \times L}$, $\boldsymbol{x}_s \in C^{M \times [(L/2)+1]}$, and $\boldsymbol{x}_e \in C^{M \times D}$. Here, C, M, L, and D are the complex number field, number of variables, look-back window size, and embedding dimension, respectively.

Next, we input the embedded 2D spectrum to SPM to calculate autocorrelation. The output of SPM is an attention matrix. Like in Transformer, this matrix is further processed by the feed-forward network and others. We then perform 2D IFFT to obtain the output of the encoding stage.

$$\boldsymbol{x}_a = SPM(\boldsymbol{x}_e), \quad \boldsymbol{x}_o = 2DIFFT(Compose(LN(FFN(LN(\boldsymbol{Res})) + LN(\boldsymbol{Res})))), \tag{7}$$

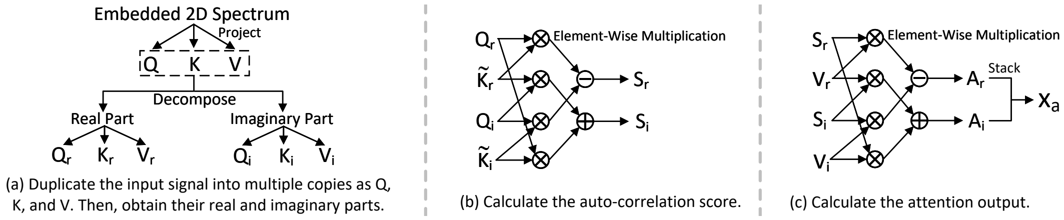
where $SPM(\cdot)$ and $\boldsymbol{x}_a$ represent the (Generalized) Spectral Product Mechanism and attention matrix (correlation score matrix), respectively. Let $\boldsymbol{Res} = \boldsymbol{x}_a + \boldsymbol{x}_e$ and it is a residue connection, while $\boldsymbol{x}_o$ represents the output of the encoding stage. Furthermore, $\boldsymbol{x}_a \in R^{M \times D}$, $\boldsymbol{Res} \in R^{M \times D}$, and $\boldsymbol{x}_o \in R^{M \times [(D-1)*2]}$. Let $LN(\cdot)$, $FFN(\cdot)$, $Compose(\cdot)$, and $2DIFFT(\cdot)$ represent the layer normalization, feed-forward network, composing the real and imaginary parts stacked in the batch size dimension into a complex number, and 2D IFFT, respectively.

Finally, we apply a multilayer perceptron (MLP) as a decoder. It maps the output of the encoding stage to the prediction result. Formally, this process is expressed as $\boldsymbol{y} = MLP(\boldsymbol{x}_o)$, where $MLP(\cdot)$ and $\boldsymbol{y}$ represent the multilayer perceptron and predicted multivariate series, respectively. Additionally, $\boldsymbol{y} \in R^{M \times T}$ and T represents the prediction length.



Value
Time
Input Multivariate Series x∈R^{M×L}
2D FFT
2D Spectrum
x_s∈C^{M×[(L/2)+1]}
Embed
SPM / GSPM
Q Matrix
K Matrix
V Matrix
Q,K,V∈C^{M×D}
Element-Wise Multiplication
Attention Matrix
x_a∈R^{M×D}
Residual Connect + Norm + Feed Forward + 2D IFFT
MLP
Value
Output Multivariate Series y∈R^{M×T}
Time

Figure 3: Framework of our method, where the SPM (GSPM) is highlighted in the dashed box. SPM employs spectral product equivalence to calculate autocorrelation across both variables and time dimensions, thereby modeling dependencies across arbitrary positions. Meanwhile, SPM achieves the function of local relevance calculation similar to the Self-Attention mechanism. However, the overall computational complexity of SPM is lower, which is $O(NlogN)$, and through the Hadamard product, the calculation of the most core correlation score matrix is further reduced to $O(N)$.



(a) Duplicate the input signal into multiple copies as Q, K, and V. Then, obtain their real and imaginary parts.

(b) Calculate the auto-correlation score.

(c) Calculate the attention output.

Figure 4: SPM's calculation steps follows the rules of complex multiplication and Hadamard product.

3.4 Spectral Product Mechanism

Figure 4 shows the details of SPM. The first step (Figure 4(a)) is to duplicate the input signal (embedded 2D spectrum) into multiple copies (Q, K, V) for calculating autocorrelation. Then we decompose to obtain the real and imaginary parts of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. These are expressed as follows.

$$\{\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\} = Project(\boldsymbol{x}_e), \quad \{\boldsymbol{Q}_r,\boldsymbol{Q}_i,\boldsymbol{K}_r,\boldsymbol{K}_i,\boldsymbol{V}_r,\boldsymbol{V}_i\} = Decompose(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}), \tag{8}$$

where $Project(\cdot)$, $Decompose(\cdot)$, $\boldsymbol{X}_r$, and $\boldsymbol{X}_i$ represent the identity mapping (duplication), decomposing operation, real part, and imaginary part of a complex number $\boldsymbol{X}$, respectively. Additionally, $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V},\boldsymbol{Q}_r,\boldsymbol{K}_r,\boldsymbol{V}_r,\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i \in R^{M\times D}$.

The second step (Figure 4(b)) involves calculating autocorrelation through multiplying the two copies of the input signal $\boldsymbol{Q} = \boldsymbol{Q_r} + \boldsymbol{Q_i}j$ and $\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{K}}_{\boldsymbol{r}} + \tilde{\boldsymbol{K}}_{\boldsymbol{i}}j$, where $\tilde{\boldsymbol{K}}$ represents the flipped $\boldsymbol{K}$. This is the spectral product and it requires the employment of Hadamard product instead of matrix multiplication. The calculation process can be expressed as follows.

$$\boldsymbol{Q}\tilde{\boldsymbol{K}} = (\boldsymbol{Q_r} + \boldsymbol{Q_i}j)(\tilde{\boldsymbol{K}}_{\boldsymbol{r}} + \tilde{\boldsymbol{K}}_{\boldsymbol{i}}j) = (\boldsymbol{Q_r}\tilde{\boldsymbol{K}}_{\boldsymbol{r}} - \boldsymbol{Q_i}\tilde{\boldsymbol{K}}_{\boldsymbol{i}}) + (\boldsymbol{Q_r}\tilde{\boldsymbol{K}}_{\boldsymbol{i}} + \boldsymbol{Q_i}\tilde{\boldsymbol{K}}_{\boldsymbol{r}})j, \tag{9}$$

where $(\boldsymbol{Q_r}\tilde{\boldsymbol{K}}_{\boldsymbol{r}} - \boldsymbol{Q_i}\tilde{\boldsymbol{K}}_{\boldsymbol{i}})$ and $(\boldsymbol{Q_r}\tilde{\boldsymbol{K}}_{\boldsymbol{i}} + \boldsymbol{Q_i}\tilde{\boldsymbol{K}}_{\boldsymbol{r}})$ respectively represent the real part $\boldsymbol{S_r}$ and the imaginary part $\boldsymbol{S_i}$ of the new complex number $\boldsymbol{S}$, which represents the correlation score matrix in time domain. It functions similarly to the Self-Attention score matrix. However, its element-wise multiplication (Hadamard product), compared with the matrix multiplication of Self-Attention, reduces the computational complexity from $O(N^2)$ to $O(N)$.

The third step (Figure 4(c)) is to apply the autocorrelation score to the input signal to adjust the connection between local signals according to the correlation. The process is achieved by multiplying the complex numbers $\boldsymbol{S} = \boldsymbol{S_r} + \boldsymbol{S_i}j$ and $\boldsymbol{V} = \boldsymbol{V_r} + \boldsymbol{V_i}j$. This is equivalent to performing global convolution on the input signal in the time domain using the correlation score matrix, so as to apply the score to the input. The calculation process can be expressed as follows.

$$\boldsymbol{SV} = (\boldsymbol{S_r} + \boldsymbol{S_i}j)(\boldsymbol{V_r} + \boldsymbol{V_i}j) = (\boldsymbol{S_r}\boldsymbol{V_r} - \boldsymbol{S_i}\boldsymbol{V_i}) + (\boldsymbol{S_r}\boldsymbol{V_i} + \boldsymbol{S_i}\boldsymbol{V_r})j, \tag{10}$$

where $(\boldsymbol{S_r}\boldsymbol{V_r} - \boldsymbol{S_i}\boldsymbol{V_i})$ and $(\boldsymbol{S_r}\boldsymbol{V_i} + \boldsymbol{S_i}\boldsymbol{V_r})$ respectively represent the real part $\boldsymbol{A_r}$ and the imaginary part $\boldsymbol{A_i}$ of the new complex number $\boldsymbol{A}$, which represents the attention output matrix. Then, we stack $\boldsymbol{A_r}$ and $\boldsymbol{A_i}$ in batch size dimension, obtaining the attention output $\boldsymbol{x}_a$. This is to facilitate subsequent feed-forward network and others to directly process the real part and imaginary part respectively.

3.5 Generalized Spectral Product Mechanism

To capture more abundant and discriminative dependencies, we extend the SPM to a generalized form (GSPM). Specifically, we first set the projection in Figure 4(a) to 3 different linear mappings (SPM adopts the identity mapping) to obtain different feature representations $\boldsymbol{Q}, \boldsymbol{K}$, and $\boldsymbol{V}$ of the input signal. This process is expressed as follows.

$$\boldsymbol{Q} = Linear_1(\boldsymbol{x}_e), \quad \boldsymbol{K} = Linear_2(\boldsymbol{x}_e), \quad \boldsymbol{V} = Linear_3(\boldsymbol{x}_e), \tag{11}$$

where $Linear_1(\cdot)$, $Linear_2(\cdot)$, and $Linear_3(\cdot)$ represent 3 different linear mappings, respectively. Additionally, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in R^{M \times D}$. Then, we remove the flipping operation of K in Figure 4(b) because the learnable linear mapping can already automatically achieve flipping. Other calculation processes of the spectral product are consistent with SPM. Their concise formal expression is $\boldsymbol{x}_a = Stack(\boldsymbol{QKV})$. The details of complex multiplication among $\boldsymbol{Q}, \boldsymbol{K}$, and $\boldsymbol{V}$, stack operations of $Stack(\cdot)$, and the attention output $\boldsymbol{x}_a$ have been introduced in the SPM section above.

In this way, $\boldsymbol{Q}, \boldsymbol{K}$, and $\boldsymbol{V}$ can be driven by data to represent different important features. GSPM extends the calculation object of autocorrelation in SPM from a single fixed feature to discriminative important features, then models more complex dependencies across feature representations.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We extensively conduct experiments on 14 authoritative public datasets, including Traffic, Electricity, Solar-Energy, PEMS (4 subsets), Weather, Exchange-Rate, ILI, and ETT (4 subsets) [12, 4, 14]. According to [28], these datasets are non-stationary with evolving trend and seasonal patterns, and can be used to evaluate the model's ability to handle non-stationary time series. Detailed descriptions and statistics information of these datasets can be found in the Appendix. Following [4], we merge and report average values of four subsets in ETT and PEMS.

Baselines. Due to space limitations, we can only give priority to presenting the latest SOTA methods published in 2025 and 2024. The corresponding relationships between their modeling paradigms and Figure 1 are as follows: Figure 1(a): WPMixer [10], TimeMixer [9], PatchTST [7]; Figure 1(b): Seq-Com [13]; Figure 1(c): S-Mamba [14], iTransformer [4]; Combining Figure 1(a) and (c): Leddam [15]. Earlier published methods and Crossformer [5] (Figure 1(d)) are reported in Appendix.

Implementation Details. Our method is implemented using PyTorch with Adam optimizer on a single RTX 3090 GPU. Following the above baselines, we adopt the widely used MSE loss function during training. The number of Transformer encoder layer, head, dropout rate, training epochs, and early stopping patience are configured with 1, 1, 0.2, 300, and 20, respectively. Due to space limitations, the full hyperparameter settings are reported in the Appendix. Following baselines [14, 13, 15, 4, 22], L is set at 36 for ILI and 96 for other datasets, while $T \in \{96, 192, 336, 720\}$.

4.2 Main Results

Compare with the latest SOTA methods. In this section, we focus on discussing the differences from other methods. Since GSPM and SPM essentially represent the same concept, that is, spectral product, they are collectively referred to as our method and discussed together. Their differences are discussed separately in the next section. As shown in Table 1, on the most authoritative Traffic dataset, our method achieves the lowest average MSE and MAE, surpassing not only the latest SOTA methods but also outperforming well-known methods like iTransformer [4] and PatchTST [7], with MSE 3.51% and 22.01% lower, respectively. On the highly authoritative Electricity dataset, our method also achieves the lowest average error. Specifically, our MSE is 3.55% to 24.54% lower than other methods, while MAE is 1.52% to 18.55% lower. On the other five datasets, except for ranking 2nd (0.272) and 4th (0.265) in two MAE metrics, our method ranks 1st in the other eight metrics.

Furthermore, it can be observed that on the two datasets with the highest dimensional variables (Traffic, Electricity), the methods (S-Mamba [14], Leddam [15], iTransformer [4]) of adopting the paradigm of modeling variable relationships (as shown in Figure 1(c)) are more effective than the other methods without adoption (as shown in Figure 1(a)). However, on other datasets with lower-dimensional variables, the situation is reversed. This indicates that modeling dependencies in only

Table 1: Quantitative results averaged from all prediction length $T \in \{96, 192, 336, 720\}$ of ours and latest SOTA methods. The full results and results of other methods can be found in the Appendix. Red and blue: 1st and 2nd. "-": unreported in its paper, unmeasurable as its code not open-sourced.

Models	GSPM		SPM		S-Mamba [14]		WPMixer [10]		Seq-Com [13]		Leddam [15]		iTransformer[4]		TimeMixer [9]		PatchTST [7]	
Venue	-		-		NC 25		AAAI 25		AAAI 25		ICML 24		ICLR 24		ICLR 24		ICLR 23	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	**0.413**	**0.276**	0.421	0.277	0.414	**0.276**	0.489	0.297	0.472	0.301	0.467	0.294	0.428	0.282	0.485	0.298	0.529	0.341
Electricity	0.167	0.263	**0.163**	**0.259**	0.170	0.265	0.177	0.267	0.192	0.282	0.169	0.263	0.178	0.270	0.182	0.273	0.216	0.318
Solar	**0.230**	0.266	0.233	0.265	0.240	0.273	0.237	**0.260**	-	-	**0.230**	0.264	0.233	0.262	0.246	0.291	0.270	0.307
PEMS	0.236	0.301	**0.210**	**0.283**	0.240	0.305	0.487	0.468	-	-	0.277	0.333	0.355	0.394	0.430	0.437	0.592	0.544
Weather	**0.239**	0.272	0.240	0.272	0.251	0.276	0.243	**0.269**	0.243	0.273	0.242	0.272	0.258	0.278	0.240	0.272	0.265	0.286
Exchange	**0.343**	**0.392**	0.414	0.425	0.367	0.408	0.391	0.418	0.356	0.400	0.421	0.427	0.360	0.403	0.418	0.425	0.352	0.397
ILI	**0.933**	**0.612**	1.114	0.654	2.817	1.126	2.093	0.900	-	-	1.993	0.888	1.856	0.873	1.806	0.845	1.633	0.801
ETT	0.346	**0.377**	**0.345**	**0.377**	0.380	0.398	0.356	0.379	0.363	0.389	0.367	0.394	0.383	0.399	0.367	0.389	0.401	0.409
1st Count:	**5**	**4**	3	3	0	1	0	2	0	0	1	0	0	0	0	0	0	0

one direction (either the variable or the time dimension) cannot adapt to a broader range of scenarios. Therefore, our global modeling paradigm (shown in Figure 1(e)) models the dependencies across variables and the time dimension. It treats the time and variable dimensions equally to model the dependencies at any position. In this way, our method can overcome this limitation and achieve the lowest average MSE and MAE in the greatest number of cases, significantly surpassing the latest SOTA methods.

Comparison between GSPM and SPM. On the one hand, GSPM performs better than SPM on five out of seven datasets in total. Its MSE decreases from 0.42% (Weather) to 17.15% (Exchange) and its MAE decreases from 0.36% (Traffic) to 7.76% (Exchange). This verifies the effectiveness of extending the traditional autocorrelation by mapping the input into distinct feature representations, which enables the modeling of complex dependencies through cross-feature correlations. This approach can improve the adaptability to generalized scenarios, capturing complex patterns.

On the other hand, SPM slightly reduces the MSE by 0.29% compared to GSPM on ETT, and on Electricity, it reduces the MSE and MAE by 2.40% and 1.52% respectively. Since there is no essential difference between them on ETT, we focus on interpreting why Electricity and Traffic, both having high-dimensional variables, show opposite performances. According to [28], in the spectrum of Traffic, the energy is dispersed among low, medium, and high frequencies, meaning that the time series changes more drastically and has richer patterns. While in the spectrum of Electricity, the energy is concentrated only on the lowest frequency and is close to zero at other frequencies, meaning that the time series changes less and has a more monotonous pattern. GSPM is designed to capture more complex patterns, but the monotonous pattern of Electricity lacks discriminative feature representations. In contrast, SPM is more suitable for this type of scenario.

4.3 Ablation Study

Component ablation. We explore it from three parts and report the results in Table 2. The first part is to investigate the role of modeling across both variable and time dimensions. We remove the FFT of variables, time, and all dimensions respectively, as shown in the left 1st to 3rd items. From an overall scenario perspective, all three of them are less effective than the 2D FFT of GSPM. Moreover, the error of removing the time dimension FFT is the largest, with its average MSE being 3.88% and 3.18% higher than that of variables and all dimensions respectively. This indicates that temporal dependence is the most fundamental and important feature, and it also shows that implementing FFT only on the variable dimension is less effective than simple time-domain modeling.

The second part is to explore the source of GSPM's non-linear modeling ability. We conduct the following studies respectively: removing the linear mapping of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ (left 4th), at this time GSPM degenerates into SPM, only two indicators decrease by 0.38% and 0.29%, but the average MSE increase by 9.97%, indicating that calculating autocorrelation from discriminative features is more effective; adding non-linear mapping to $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ (left 5th), only two indicators are on par with GSPM, but the average MSE increase by 11.02%, indicating that non-linear mapping is not needed when capturing the discriminative features of the input; adding a sigmoid non-linear activation function to the correlation score matrix (left 6th), the error is exactly the same as that of GSPM, indicating that non-linear mapping is not needed for the autocorrelation score either; removing the non-linear activation function from the feed-forward network (left 7th), only one indicator is on par with GSPM, but the average MSE increase by 4.99%; removing the non-linear activation function

Table 2: Component ablation study results. GSPM is the optimal one overall.

Model	w/o Vari FFT		w/o Time FFT		w/o FFT		w/o Line-QKV		w/ Nonl-QKV		w/ Sigm-Scor		w/o Nonl-FFN		w/o Nonl-MLP		w/ Mult		w/ Add		w/o FFT; w/ Add		GSPM	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	0.442	0.285	0.477	0.328	0.453	0.290	0.421	0.277	0.415	0.277	**0.413**	**0.276**	0.415	0.277	0.496	0.333	0.418	0.277	0.420	0.276	0.456	0.292	**0.413**	**0.276**
Electricity	0.178	0.269	0.173	0.269	0.178	0.269	0.163	0.259	0.168	0.264	0.167	0.263	0.167	0.263	0.194	0.284	0.171	0.267	**0.166**	**0.262**	0.179	0.270	0.167	0.263
Solar	0.239	0.273	0.243	0.274	0.242	0.273	0.233	**0.265**	0.232	0.266	**0.230**	0.266	0.232	0.266	0.271	0.288	0.244	0.277	0.232	**0.265**	0.247	0.278	**0.230**	0.266
Weather	0.241	0.273	0.241	0.273	0.240	0.273	0.240	0.272	**0.239**	0.272	**0.239**	0.272	0.241	0.273	0.248	0.279	0.241	0.274	0.241	0.273	0.240	**0.271**	**0.239**	0.272
Exchange	0.350	0.398	0.432	0.432	0.379	0.410	0.414	0.425	0.352	0.401	**0.343**	**0.392**	0.349	0.399	0.366	0.403	0.382	0.416	0.429	0.432	0.367	0.405	**0.343**	**0.392**
ILI	1.267	0.710	1.268	0.698	1.236	0.689	1.114	0.654	1.211	0.683	**0.933**	**0.612**	1.048	0.637	1.256	0.728	1.146	0.666	1.214	0.690	1.350	0.724	**0.933**	**0.612**
ETT	0.348	0.378	0.353	0.382	0.353	0.382	**0.345**	**0.377**	0.346	**0.377**	0.346	**0.377**	0.346	**0.377**	0.357	0.383	0.350	0.380	0.357	0.385	0.352	0.381	0.346	**0.377**
AVG	0.438	0.369	0.455	0.379	0.440	0.369	0.419	0.361	0.423	0.363	**0.381**	**0.351**	0.400	0.356	0.455	0.385	0.422	0.365	0.437	0.369	0.456	0.374	**0.381**	**0.351**
1st Count:	0	0	0	0	0	0	1	2	1	1	**6**	**5**	0	1	0	0	0	0	1	2	0	1	**6**	**5**

from the MLP (left 8th), the average MSE increase by 19.42%. The last two items indicate that the non-linear modeling ability of GSPM mainly comes from the MLP, followed by the FFN.

The third part is to study the effectiveness of the Hadamard product. We replace it with matrix multiplication (left 9th) and addition (left 10th) respectively, and their average MSE increase by 10.76% and 14.70% respectively, indicating that the Hadamard product is more suitable for the spectral product mechanism both in theory and in practice. In addition, addition is able to reduce the error by 0.001 in three indicators. This is because complex addition essentially projects the complex plane signal onto a new orthogonal basis and completely loses the original phase relationship. Therefore, when the scene is insensitive to phase, addition will be effective. Due to the linearity of FFT (proof in the Appendix), we conduct an additional control experiment on addition (left 11th, time-domain addition), and its average MSE increase by 4.35%. This is because when the addition is used in GSPM, the addition of real and imaginary parts is interleaved, which improves the feature interaction ability compared to time-domain addition.

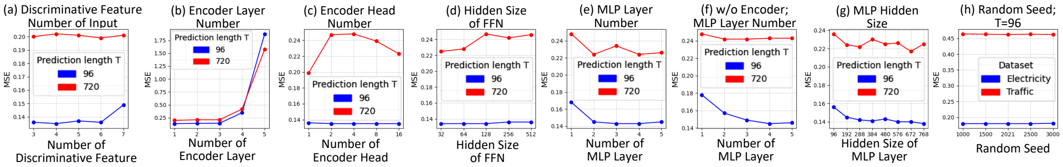


Figure 5: Hyperparameter sensitivity analysis on Electricity dataset. For the encoder and decoder, smaller and larger depths are respectively more optimal. The random seed has no effect.

Hyperparameter sensitivity analysis. We conduct this study in four parts. Firstly, we study the impact of the number of discriminative features (like $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$), as shown in Figure 5(a). Taking three as the basic version, the MSE is already low enough and only six can surpass it. This indicates that as long as there are three feature matrices, autocorrelation can be accurately calculated and values assigned. In addition, the impact of the number of features is not linear, which means that accuracy cannot be improved by simply increasing the number. The second part is to study the impact of the number of parameters in the encoder. Figures 5(b) and 5(c) respectively show that increasing the number of layers and heads of the encoder has no effect. Moreover, too many layers will lead to a significant increase in MSE, while the number of heads mainly affects the prediction effect over a longer period (720 step). Figure 5(d) shows that increasing the number of parameters in the feed-forward network will lead to a decrease in the prediction accuracy over a longer period. These three figures together indicate that increasing the number of parameters in the encoder brings no benefits because the autocorrelation scores will be distorted due to repeated calculations. In fact, we only use one layer and one head, confirming that we rely on an efficient theory, not parameter piling.

The third part is to study the impact of the number of parameters in the decoder. Figure 5(e) shows that increasing the number of layers of the MLP can generally reduce the MSE, and it reaches the minimum at four layers. To verify whether the same result can be obtained using only the MLP, we further conduct experiments by removing the encoder, as shown in Figure 5(f). At this time, for the shorter-term (96 step) prediction, the MSE becomes higher initially, and then the lowest MSE appears later at a deeper layer, and it increases by 1.40%. This not only leads to a larger number of parameters and higher costs but also a larger error. For the longer-term (720 step) prediction, its lowest MSE increases by 8.04%. Therefore, our autocorrelation calculation method plays a very important role, and the same effect cannot be achieved using only the MLP. In addition, we also studied the impact of the hidden size of the MLP, as shown in Figure 5(g). A larger size generally brings higher accuracy, and it reaches the overall optimal value when the size is 672. This is because the encoder models the point-to-point autocorrelation dependencies, which are large in scale and complexity. Therefore, the decoder needs to provide a stronger non-linear decoding ability. As the number of layers and hidden

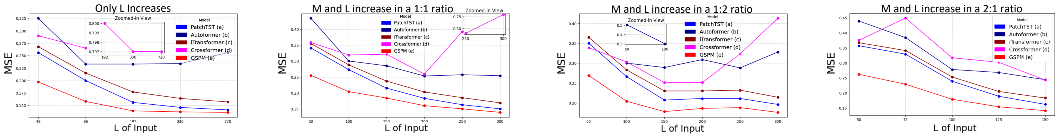


Figure 6: Influence of look-back window size on Electricity dataset with T=96 and unified hyper-parameters. GSPM achieves consistently decreased and lower MSE than baselines under all settings.

size of the decoder increase, the non-linear decoding ability it provides becomes stronger. However, when the needs of the encoder are met (four layers and 672 size), there is no gain. The fourth part focuses on studying the impact of different random seeds, as shown in Figure 5(h). Changing the random seed over a large range does not change the MSE, so our method is not affected by it.

Influence of look-back window size. We specifically study the impact of the common input time length (L) and the input size at any position. The latter refers to the situation where the variable (M) and time (L) dimensions change simultaneously. We increase M and L at multiple ratios and compare with four most representative local paradigm methods shown in Figure 1(a), (b), (c), and (d). The first sub-figure of Figure 6 shows that GSPM is optimal in terms of the conventional utilization of time length input. Its MSE not only gradually decreases but is also the lowest throughout the process. The other three sub-figures illustrate that GSPM performs the same when the input size at any position with multiple ratios increases. These jointly verify that GSPM fully leverages more input by modeling dependencies across arbitrary positions, resulting in more accurate high-dimension long-term predictions.

Resource overhead analysis. Since our goal is to reduce the overhead of Transformer, our baselines are the Transformer-based methods and the latest Mamba-based method (Table 3). GSPM has the second-lowest computational complexity, outperforming most square-level methods. It trails only FEDformer and S-Mamba's linear-level complexity. Crucially, under respective hyper-parameters, GSPM's average MSE is much lower than FEDformer's, giving it higher practical value. GSPM also beats S-Mamba in MSE, validating its theoretical effectiveness. The autocorrelation score calculation in GSPM has $O(N)$ complexity. Although FFT adds $O(LlogL)$, its impact on time, memory, and parameters is minimal (4.38%, 1.61%, 0% respectively), making GSPM almost $O(N)$ overall.

In time overhead, GSPM ranks 1st across all prediction lengths. Notably, FEDformer, despite its lowest complexity, has an average time overhead 1.77 times and 1.65 times larger than Autoformer and PatchTST respectively, proving lower complexity does not always translate to better practicality. For memory and parameter overhead, GSPM ranks 2nd in the average prediction length and 1st in the longest prediction. This shows that the advantage of GSPM lies in ultra-long-term prediction, which is exactly the problem scenario we are targeting. By comprehensively comparing the four indicators, it can be verified that GSPM has achieved our goal. While achieving the lowest MSE by modeling dependencies across arbitrary positions, GSPM effectively reduces the overhead of Transformer.

Table 3: Fair comparison of resource overhead on the largest-scale dataset Traffic employing unified hyper-parameters. Red and blue: 1st and 2nd. For dynamic "M" or "L", use "N" for representation.

Model	Complexity	Time (s/epoch)					Memory (GB)					Parameter (M)					AVG
		96	192	336	720	AVG	96	192	336	720	AVG	96	192	336	720	AVG	MSE
Autoformer [12]	$O(LlogL)$	17.564	25.063	29.063	42.284	28.494	1.406	2.282	3.134	5.440	3.066	1.369	1.468	**1.468**	1.468	1.444	0.628
FEDformer [21]	$O(N)$	41.203	45.659	49.687	65.099	50.412	1.286	1.572	2.160	3.510	2.132	4.515	4.515	4.515	4.515	4.515	0.61
Crossformer [5]	$O((L/p)^2)$	63.143	61.121	99.538	109.653	83.364	15.546	26.348	42.114	85.598	42.402	2.332	2.999	3.999	6.665	3.999	0.55
PatchTST [7]	$O((L/p)^2)$	24.632	26.813	31.090	39.568	30.526	6.522	6.254	6.962	7.542	6.820	0.249	0.397	0.618	1.209	0.618	0.529
iTransformer [4]	$O(M^2)$	11.455	14.808	19.731	31.556	19.388	1.776	2.088	2.342	2.880	2.272	**0.125**	**0.137**	0.156	0.205	**0.156**	0.428
Leddam [15]	$O(M^2 + (L/p)^2)$	118.715	166.338	170.349	179.943	158.836	2.130	1.686	1.776	2.528	2.030	0.305	3.293	3.367	3.565	2.633	0.467
S-Mamba [14]	$O(N)$	11.661	15.988	21.254	32.552	20.364	**1.278**	**1.432**	**1.706**	2.806	**1.806**	0.189	0.202	0.220	0.270	0.220	0.414
GSPM	$O(NlogN)$	**10.154**	**14.016**	**18.639**	**30.195**	**18.251**	1.582	1.594	1.998	**2.390**	1.891	0.140	0.149	0.163	**0.200**	0.163	**0.413**

5 Conclusion

To achieve more accurate high-dimensional long-term time series forecasting, we propose a new global modeling paradigm. It models dependencies at arbitrary position across variable and time dimensions. To reduce Transformer's cost, we replace the Self-Attention with autocorrelation and reformulate it into Spectral Product Mechanism (SPM), reducing complexity from $O(N^2)$ to $O(NlogN)$. For more complex patterns, we introduce GSPM, which maps and models cross-feature correlations. Our method surpasses current SOTA methods on numerous authoritative benchmarks.

References

[1] T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, “Master: Market-guided stock transformer for stock price forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 162–170.

[2] K. Chen, T. Han, J. Gong, L. Bai, F. Ling, J.-J. Luo, X. Chen, L. Ma, T. Zhang, R. Su *et al.*, “Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead,” *arXiv preprint arXiv:2304.02948*, 2023.

[3] A. Hussein and M. Awad, “Time series forecasting of electricity consumption using hybrid model of recurrent neural networks and genetic algorithms,” *Measurement: Energy*, vol. 2, p. 100004, 2024.

[4] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, “itransformer: Inverted transformers are effective for time series forecasting,” *arXiv preprint arXiv:2310.06625*, 2023.

[5] Y. Zhang and J. Yan, “Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting,” in *The eleventh international conference on learning representations*, 2023.

[6] A. Behrouz, M. Santacatterina, and R. Zabih, “Chimera: Effectively modeling multivariate time series with 2-dimensional state space models,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 119 886–119 918, 2024.

[7] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A time series is worth 64 words: Long-term forecasting with transformers,” *arXiv preprint arXiv:2211.14730*, 2022.

[8] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are transformers effective for time series forecasting?” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[9] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, “Timemixer: Decomposable multiscale mixing for time series forecasting,” *arXiv preprint arXiv:2405.14616*, 2024.

[10] M. M. N. Murad, M. Aktukmak, and Y. Yilmaz, “Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 19 581–19 588.

[11] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[12] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting,” *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[13] X. Chen, P. Qiu, W. Zhu, H. Li, H. Wang, A. Sotiras, Y. Wang, and A. Razi, “Sequence complementor: Complementing transformers for time series forecasting with learnable sequences,” vol. 39, 2025.

[14] Z. Wang, F. Kong, S. Feng, M. Wang, X. Yang, H. Zhao, D. Wang, and Y. Zhang, “Is mamba effective for time series forecasting?” *Neurocomputing*, p. 129178, 2024.

[15] G. Yu, J. Zou, X. Hu, A. I. Aviles-Rivero, J. Qin, and S. Wang, “Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling,” in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=87CYNyCGOo

[16] Q. Huang, L. Shen, R. Zhang, J. Cheng, S. Ding, Z. Zhou, and Y. Wang, “Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 11, 2024, pp. 12 608–12 616.

[17] N. Wiener, “Generalized harmonic analysis,” *Acta mathematica*, vol. 55, no. 1, pp. 117–258, 1930.

[18] B. Hunt, “A matrix theory proof of the discrete convolution theorem,” *IEEE Transactions on Audio and Electroacoustics*, vol. 19, no. 4, pp. 285–288, 1971.

[19] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, "Frequency-domain mlps are more effective learners in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[20] S. Gao, Z.-Y. Li, Q. Han, M.-M. Cheng, and L. Wang, "Rf-next: Efficient receptive field search for convolutional neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 2984–3002, 2022.

[21] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

[22] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[23] L. Han, H.-J. Ye, and D.-C. 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.

[24] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International conference on learning representations*, 2021.

[25] W. Cai, Y. Liang, X. Liu, J. Feng, and Y. Wu, "Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 10, 2024, pp. 11 141–11 149.

[26] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021.

[27] W. Fan, P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu, "Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 6, 2023, pp. 7522–7529.

[28] W. Ye, S. Deng, Q. Zou, and N. Gui, "Frequency adaptive normalization for non-stationary time series forecasting," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: [Yes]

   Justification: We claim the contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in the Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the full set of assumptions and a complete (and correct) proof for each theoretical result.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We open the source codes and provide detailed implementations and hyper-parameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We open the source codes and provide detailed implementations and hyperparameters.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details, open-sourced codes, detailed implementations, and complete hyper-parameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: In the task of this paper, baseline papers all report MSE and MAE, instead of error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the information of computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Introduction section, we discuss our societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: We employ the public datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite and mention the creators of these assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.