

FreIE: Low-Frequency Spectral Bias in Neural Networks for Time-Series Tasks

Jialong Sun¹, Xinpeng Ling², Jiaxuan Zou³, Jiawen Kang⁴, and Kejia Zhang^{5,*}

¹School of Mathematical Science, Heilongjiang University, Harbin, China
20212644@s.hlju.edu.cn

²Software Engineering Institute, East China Normal University, Shanghai, China
xpling@stu.ecnu.edu.cn

³Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China
jiaxuanzou@stu.xjtu.edu.cn

⁴School of Automation, Guangdong University of Technology, Guangzhou, China
kavinkang@gdut.edu.cn

⁵School of Computer Science and Big Data (School of Cybersecurity), Heilongjiang University, Harbin, China

Abstract

The inherent autocorrelation of time series data presents an ongoing challenge to multivariate time series prediction. Recently, a widely adopted approach has been the incorporation of frequency domain information to assist in long-term prediction tasks. Many researchers have independently observed the spectral bias phenomenon in neural networks, where models tend to fit low-frequency signals before high-frequency ones. However, these observations have often been attributed to the specific architectures designed by the researchers, rather than recognizing the phenomenon as a universal characteristic across models. To unify the understanding of the spectral bias phenomenon in long-term time series prediction, we conducted extensive empirical experiments to measure spectral bias in existing mainstream models. Our findings reveal that virtually all models exhibit this phenomenon. To mitigate the impact of spectral bias, we propose the FreLE (Frequency Loss Enhancement) algorithm, which enhances model generalization through both explicit and implicit frequency regularization. This is a plug-and-play model loss function unit. A large number of experiments have proven the superior performance of FreLE. Code is available at <https://github.com/Chenxing-Xuan/FreLE>.

Keywords: Time series forecasting, Fourier transform, Implicit Regularization.

1 Introduction

Time series data consists of numerical values associated with time. Long-term time series prediction is crucial across various domains, including weather forecasting and intelligent manufacturing [1,2]. However, due to the inherent complexity of time series data, existing deep learning approaches that directly predict time-domain data often yield suboptimal performance. In recent years, a promising approach has emerged that leverages frequency-domain information to improve prediction accuracy.

Modeling long-term time series prediction using quasi-periodic dynamical systems reveals that both linear and nonlinear time-domain prediction optimization objectives are highly non-convex. However, by mapping the optimization objective to the frequency domain, the global

*Corresponding author: zhangkejia@hlju.edu.cn

optimal solution of the error surface can be efficiently computed using the Koopman-FFT method [3]. This theoretical foundation has significantly inspired researchers to incorporate frequency-domain information into long-term time series prediction. Building on Koopman’s work, a method has been proposed that transforms frequency-domain information into 2D, converting frequency sequence data into frequency image data. This method employs 2D kernel modeling to capture implicit frequency relationships between different sequences, thereby enhancing time-domain learning performance [4]. Additionally, given the complex information resulting from frequency-domain transformations, complex-valued neural networks can be employed to achieve efficient long-term time series prediction with a reduced number of parameters [5]. Recent studies have also provided both theoretical proof and empirical analysis demonstrating that using frequency-domain loss functions can decouple the complexity of time series [6], further improving model performance in long-term time series prediction.

However, as the saying goes, "there is no free lunch." While frequency-domain information offers researchers a potentially limitless framework for machine learning, it also presents inevitable challenges, particularly concerning the "selection of spectral information." After decomposing a signal into its spectral components, determining how to effectively utilize both low-frequency and high-frequency information within a machine learning framework has become a central area of investigation. Low-frequency signals represent stable events with higher intensity over time but fail to capture the variability of these events. In contrast, high-frequency signals reflect more volatile and trend-based events over time but are highly susceptible to noise interference. The question remains: how should these signals be leveraged in models? Based on the Johnson-Lindenstrauss Lemma, one approach employs a random dimensionality reduction method that selectively chooses specific frequency signal features for auxiliary prediction, effectively mitigating noise interference in high-frequency features [7]. Another method, grounded in the Parseval Theorem, proposes a multilayer perceptron (MLP) model architecture that applies equal signal strength in both the time and frequency domains to jointly learn time-domain signal features [8]. While these approaches have significantly improved long-term time series prediction performance, they have yet to fully address the original question. How should we truly understand the role of spectral information in time series prediction?

Interestingly, when researchers investigate the role of spectral information in time series prediction, they often reach the same conclusion: in implicit neural representations (INR) networks, a tendency toward simple solutions is observed during the reconstruction process, with most solutions being linear combinations of low-frequency signals [9]. In studies of Transformer attention mechanisms, researchers have found that, during prediction, the Transformer architecture first learns low-frequency signal features before progressing to high-frequency signal features [10]. This learning sequence is believed to be influenced by the attention mechanism’s inherent bias toward low-frequency signals. While these findings provide in-depth insights into the mechanisms of frequency learning, the researchers unanimously agree that this frequency preference phenomenon is an intrinsic characteristic of specific models. In the following, we will refer to this as the "spectral bias phenomenon" and conduct a comprehensive investigation.

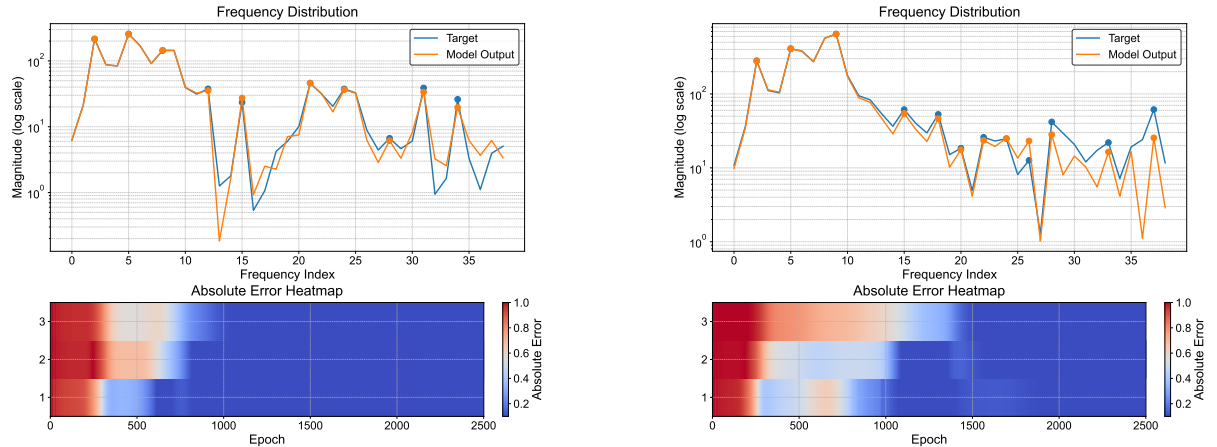
Some researchers have examined the "spectral bias phenomenon" from the perspective of numerical solutions to partial differential equations in neural networks. When solving the Poisson-Boltzmann equation, decomposing the loss function into low-frequency and high-frequency components can significantly enhance numerical stability. Colleagues have conducted extensive experiments to verify the existence of the "spectral bias phenomenon" in two-layer deep neural networks (2-DNNs) and provided theoretical proof of its presence in two-layer infinitely wide DNNs [11]. Furthermore, a variational dynamics theory based on linear assumptions confirmed the "spectral bias phenomenon" in existing neural networks. The theory proposed that this phenomenon primarily depends on the nonlinear transformation of the activation function and recommended using the Ricker activation function to mitigate it [12,13]. However, the question remains: *Can this approach be extended to time series prediction tasks? Is there a*

simpler method for understanding and addressing the "spectral bias phenomenon"?

This remains an unresolved issue in the field of time series prediction.

This work aims to investigate the existence of the spectral bias phenomenon in neural networks with various architectures and to improve time series prediction performance by addressing this phenomenon. We introduce FreLE (Frequency Loss Enhancement), an adaptive frequency enhancement algorithm designed to mitigate the spectral bias phenomenon observed in neural networks during time series prediction tasks. To validate the effectiveness of our method, we conducted extensive preliminary experiments and compared it with existing machine learning methods, highlighting the similarities and differences between various approaches. The main contributions of this paper are summarized as follows:

- **Theoretical Research:** Building on the existing 2-DNN spectral bias dynamics theory, we conduct extensive empirical research on existing temporal neural networks. Our findings confirm that various neural network architectures exhibit the spectral bias phenomenon.
- **Algorithm Design:** The FreLE algorithm we propose consists of two key components: frequency explicit regularization and frequency implicit regularization. These components are designed to perform two tasks—denoising and balancing signals of different frequencies. The roles and irreplaceability of these components are further analyzed through ablation experiments.
- **Experimental Effect:** We conducted extensive experiments to validate the effectiveness of FreLE, which achieved first place 38 times and second place 18 times across seven real-world datasets, demonstrating its theoretical superiority.



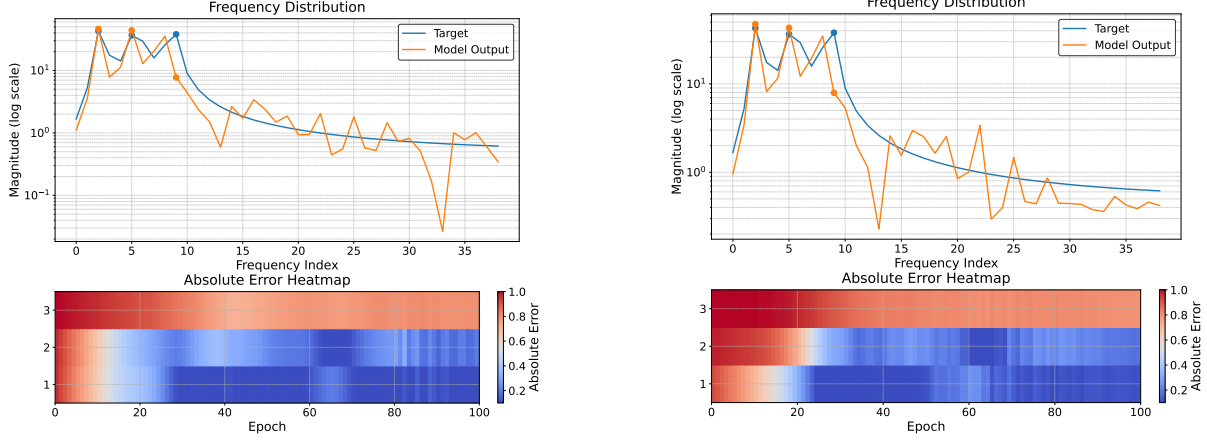
(a) The spectral loss graph for synthetic data $y = \sin x + \sin 2x + \sin 3x$

(b) The spectral loss graph for synthetic data $datay = \sin x + 2 \sin 2x + 3 \sin 3x$

Figure 1: The spectral loss graph for a 2-DNN across different synthetic datasets. The line graph represents the frequency comparison between the original data and the output data in the final iteration, and the heatmap illustrates the decrease in the RMSE loss metric as the iterations progress, showing how the three primary frequencies change with the number of iterations

2 Preliminary Analysis

We investigate the spectral bias phenomenon in neural networks through three different experiments: 1) examining the spectral bias phenomenon in simple time series models using synthetic



(a) The spectral loss graph based on σ_{relu}

(b) The spectral loss graph based on σ_{tanh}

Figure 2: The spectral loss graph for LSTM across different σ .

datasets; 2) exploring the spectral bias phenomenon in various classical models on real-world datasets; 3) evaluating the effectiveness of existing dynamic theories in mitigating the spectral bias phenomenon. The symbols and formulas associated with the spectral dynamics theory for neural networks are introduced in Sec. 2.1. The experimental analysis of synthetic datasets is presented in Sec. 2.2. The analysis of real-world datasets is discussed in Sec. 2.3. The experimental improvements based on dynamic theory are outlined in Sec. 2.4.

2.1 Spectral Dynamics in Neural Networks

This section will explore three key aspects of the study of spectral dynamics in neural networks: spectral visualization, spectral dynamics hypotheses, and the formulation of spectral dynamics equations under different activation functions.

2.1.1 Spectral Visualization

In machine learning theory, a phenomenon related to spectral bias that has garnered the attention of mathematicians: when using synthetic data formed by the sum of multiple sine signals (e.g., $y = \sin x + 2 \sin 2x + 3 \sin 3x$) for deep learning training, it is often observed that as the number of iterations increases, the low-frequency signals converge rapidly, while the high-frequency signals converge more slowly [11]. Before extending this issue to the time series domain, we replicate the spectral bias phenomenon in 2-DNNs to facilitate further in-depth discussion. The visual representation is shown in Fig. 1.

2.1.2 Spectral Dynamics Formula

For the loss function of a two-layer wide neural network, its Fourier expansion can be expressed, and the relationship between the derivative of the expanded loss function $\mathcal{F}$ with respect to time t can be described as follows [12–17]:

$$\begin{aligned}
 \partial_t \mathcal{F}[u](\xi, t) &= -\mathcal{L}[\mathcal{F}[u_\rho]], \\
 \mathcal{L}[\mathcal{F}[u_\rho]] &\approx \frac{\Gamma^*(d/2)}{\|\xi\|^{d-1}} E[a^{-1}(0) \mathcal{F}[\mathbf{K}] H(\xi)] \mathcal{F}[u_\rho](\xi), \\
 \Gamma^*(d/2) &= \frac{\Gamma(d/2)}{2\sqrt{2}\pi(d+1)/2\sigma}, \\
 H(\xi) &= -\frac{\|\xi\|}{a(0)},
 \end{aligned} \tag{1}$$

where, $F[u]$ and $F[u_P]$ represent the loss functions under different sampling densities. $F[u_P]$ describes the samples obtained from the current iteration of training, where the loss function is influenced solely by the batch size of data in a single iteration. ξ represents the frequency of the current signal, $a(0)$ is the randomly initialized weight of the neural network’s linear layer, and $b(0)$ is the randomly initialized weight of the neural network’s activation function. σ denotes the activation function, and $\mathbf{K}(x) \triangleq (\sigma(x), b\sigma'(x))'$. Therefore, the form of the Linear Frequency Principle (LFP) derived above is closely related to the choice of the activation function σ , and the general form of the Fourier expansion of the loss function can be obtained as follows:

$$\partial_t \mathcal{F}[u](\xi, t) = (\gamma_\sigma(\xi))^2 [\mathcal{F}[u_P]] \quad (2)$$

where, $\gamma_\sigma(\xi)$ is the frequency decay function obtained for different activation functions. Theorem 2 emphasizes the expression that **after performing a Fourier transform on the loss function, different frequencies follow different decay schemes in the gradient expression**. This decay scheme is often related to the choice of activation function. More specifically, the expressions for γ_σ in the ReLU and Tanh activation functions are shown in Theorem 1.

Theorem 1 (*Decay functions of different activation functions*). *The γ_{relu} of the ReLU activation function can be expressed as:*

$$(\gamma_{\text{relu}}^2(\xi)) = \mathbb{E} \left[\frac{a(0)^3}{16\pi^4 \|\xi\|^{d+3}} + \frac{b(0)^2 a(0)}{4\pi^2 \|\xi\|^{d+1}} \right]. \quad (3)$$

The γ_{tanh} of the tanh activation function can be expressed as:

$$\begin{aligned} (\gamma_{\text{tanh}}^2(\xi)) = & \frac{1}{\|\xi\|^{d-1}} \mathbb{E}_{a,r} \left[\frac{\pi^2}{r} \text{csch}^2 \left(\frac{\pi \|\xi\|}{r} \right) \right. \\ & \left. + \frac{4\pi^4 a^2 \|\xi\|^2}{r^3} \text{csch}^2 \left(\frac{\pi \|\xi\|}{r} \right) \right]. \end{aligned} \quad (4)$$

Theorem 1 shows that the spectral bias decays according to the power of the frequency of the spectral signal. As the frequency increases, the gradient of the loss function rapidly decays to zero. This represents a classical dynamical theoretical analysis in machine learning [12, 13, 18]. However, this theoretical result also raises a new issue: ***Q: Some classical time series models, such as RLinear, DLinear, and FITS [5, 19, 20], do not incorporate activation functions during the prediction process. Therefore, is the frequency preference principle widely observed in time series models? Can it be improved by introducing or modifying the activation function?***

A: In the subsequent experiments of Secs. 2.2-2.5, more extensive empirical tests will be conducted to demonstrate that the spectral bias phenomenon in time series tasks cannot be solely attributed to the effects of activation functions. The spectral bias phenomenon is widely observed in both linear and nonlinear models. Moreover, alleviating spectral bias in 2-DNNs by modifying the activation function did not yield favorable results in the time series domain. Instead, the activation function’s hyperparameters significantly impacted the convergence speed and final performance.

2.2 LSTM Experiment on Synthetic Datasets

Based on the experiments in Sec. 2.1.3, the 2-DNN was replaced with an LSTM neural network for training, with the activation functions being ReLU and Tanh. In a large number of experiments, significant spectral phenomena were still observed. Some experimental results are shown in Fig. 2.

2.3 Experiments on Real-World Datasets

We have compiled a list of classic time series models from recent years and categorized them based on their structure into two main categories: MLP models and Transformer models. These categories are divided into two subcategories: whether frequency domain data was incorporated during the training process. The models discussed are summarized as follows: 1) MLP models without frequency domain data: DLinear [20], Tide [21]; 2) MLP models with frequency domain data: TimesNet [4], FreTS [8], FreDF [6], FITS [5]; 3) Transformer models without frequency domain data: Autoformer [22], CrossFormer [23]; 4) Transformer models with frequency domain data: FEDformer [7].

In addition, we define the top 10% of the Fourier transform frequency results as low-frequency signals, the 10%-50% range as mid-frequency signals, and the 50%-100% range as high-frequency signals. We also introduce the concept of global signals, referring to the entire signal obtained after the Fourier transform. The results of the spectral bias phenomenon calculated on the ETTH2 dataset are presented in Table 1. It can be observed that most models with strong predictive performance are based on frequency-domain information. Meanwhile, the strict spectral bias phenomenon appears in all models except TimesNet, which exhibits convergence characteristics in the high-frequency range. This is due to the 2D-FFT performed by TimesNet, which disrupts frequency information and causes overfitting of high-frequency signals. The low-frequency and mid-frequency signals, both of equal importance, are not well fitted, resulting in the model having the best frequency fitting but weaker performance than other models.

Table 1: For the long-term forecasting task on the ETTH2 dataset, LIL is configured with a past sequence length of 36, while other settings are set to 96. Models marked with an asterisk * use frequency information to assist in prediction. LF, MF, HF, and GF represent low-frequency, mid-frequency, high-frequency, and global frequency, respectively. The evaluation metric used is RMSE. The **best** results are highlighted.

Domain	Frequency domain indicators				Time domain indicators	
Metrics	LF	MF	HF	GF	MAE	MSE
Delinear	0.5635	0.6261	1.7671	1.0598	0.3963	0.3425
Tide	0.3951	0.8379	0.8641	0.6643	0.3384	0.2894
TimesNet*	0.3994	0.6707	0.0481	0.2453	0.3640	0.3198
FreTS*	0.7969	0.2338	1.5131	0.9283	0.4043	0.3511
FreDF*	0.2963	0.9051	1.4407	1.0087	0.3438	0.2940
FITS*	0.1476	0.7151	1.0750	0.8291	0.3367	0.2718
Crossformer	0.5496	0.9393	2.1264	1.3737	0.5925	0.6985
Autoformer	0.2551	0.6650	1.6012	0.8788	0.4230	0.3972
Transformer	1.8012	2.3223	2.9539	1.9031	1.1441	2.0782
Fedformer*	0.4671	1.3907	1.4540	0.8367	0.3912	0.3470

2.4 Neural Network Optimization Based on Dynamic Theory

In the research by Xv et al. [12], it is stated that the impact of spectral bias can be mitigated by disrupting the monotonicity of activation functions. The classical wavelet transform function, Ricker [24], demonstrates excellent expressive performance within a 2-DNN, with its mathematical expression given by:

$$\sigma_{ricker} = \frac{\pi^{1/4}}{15a} \left(1 - \left(\frac{x}{a} \right)^2 \right) \exp \left(- \left(\frac{x}{\sqrt{2}a} \right)^2 \right) \quad (5)$$

where a is an adjustable hyperparameter.

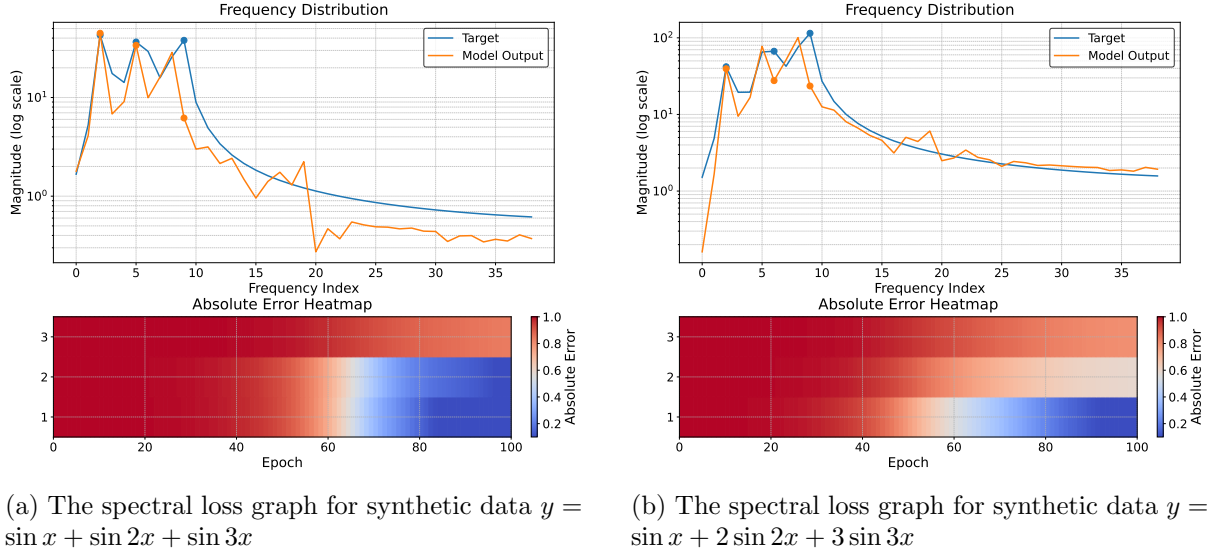


Figure 3: Under different synthetic datasets, frequency amplitude loss diagram of σ_{ricker} with $a = 1$.

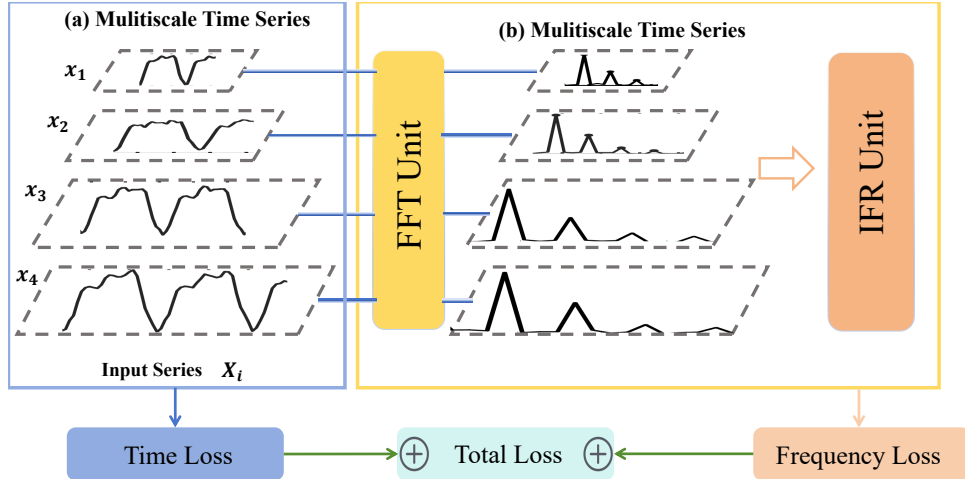


Figure 4: The framework diagram of FreLE, where IFR Unit refers to the Implicit Frequency Regularization module.

In the empirical experiments of this section, we will investigate whether replacing the activation function in LSTM with Ricker leads to improved performance. Some experimental results are shown in Figure 3. It can be observed that while Ricker mitigates spectral bias in some experiments, its effect is negligible when dealing with more complex spectral phenomena, even with minor changes to the coefficients of different signal components. This suggests the need to explore an entirely new approach to address the issue of spectral bias in time series forecasting tasks.

3 Frequency Enhancement: Methods

In this section, we will elaborate on how the FreLE algorithm balances frequency information and removes noise by separately discussing its two key components: explicit frequency regularization and implicit frequency regularization. The framework diagram of FreLE is shown in Figure 4.

3.1 Explicit Frequency Regularization

For a given time series X and its predicted value $\hat{X}$, the time series forecasting task can be described as an optimization problem:

$$\begin{aligned} \min_{\theta} \quad & \mathcal{L}_{\theta}^t \\ \mathcal{L}_{\theta}^t = & \frac{1}{n} \sum_{i=1}^n \|X_i - \hat{X}_i\| \end{aligned} \quad (6)$$

To redefine this problem explicitly with a frequency loss function (7), we can incorporate it as:

$$\begin{aligned} \min_{\theta} \quad & \delta \mathcal{L}_{\theta}^f + (1 - \delta) \mathcal{L}_{\theta}^t \\ \mathcal{L}^f = & \frac{1}{n} \sum_{i=1}^N \|\mathcal{F}(X_i) - \mathcal{F}_{\theta}(\hat{X}_i)\| \end{aligned} \quad (7)$$

where, δ serves as a parameter for balancing between two types of losses. An interesting research question is whether, by using explicit regularization alone, significant optimization effects can already be achieved when $\delta = 1$.

3.2 Implicit Frequency Regularization

The purpose of explicit frequency regularization is to incorporate frequency as a penalty term, preventing the neural network from converging too quickly after fitting the low-frequency signals. This is particularly relevant because complex neural networks (e.g., iTransformer [25], DLinear [20], TimeXer [26]) typically converge on the ETTh1 dataset in an average of just eight epochs. By introducing frequency as a penalty term, the model can continue learning even after reaching its original extremum. However, simply adding explicit regularization does not effectively extend the number of training epochs. In their research, Wu et al. [27] thoroughly explored the enhancement of model generalization through prolonged training durations. Unlike modifying the loss function with explicit regularization, implicit regularization offers a more practical approach to improving the model's generalization capability. Therefore, this section will discuss how implicit regularization can slow down the model's learning process.

Before explaining how to achieve implicit frequency regularization, first present Theorem 2.

Theorem 2 (*Multi-dimensional Fourier separation theorem*). *A two-dimensional Fourier transform can be decomposed into two one-dimensional Fourier transforms, serving as an example of the Fourier transform applied to two-dimensional vectors. It follows:*

$$\begin{aligned} & \mathcal{F}(k_x, k_y) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \\ &= \int_{-\infty}^{\infty} \left[\left(\int_{-\infty}^{\infty} f(x, y) e^{-i2\pi k_x x} dx \right) e^{-i2\pi k_y y} \right] dy \end{aligned} \quad (8)$$

In subsequent processing, Fourier transforms of multidimensional signals will be treated as single-channel Fourier transforms to reduce computational complexity. However, directly incorporating Fourier-transformed results into neural networks introduces significant noise interference, making denoising a crucial step. Traditional windowing methods, while commonly used, fail to achieve effective denoising. Applying windows to frequency information leads to rapid attenuation of frequencies other than the primary frequency, significantly degrades neural network performance [28, 29]. Therefore, this section will explore a novel denoising method and integrate it into the process of implicit regularization.

The latest research proposes an adaptive frequency processing approach that normalizes the amplitude adaptively across different frequency bands [30]. This is an effective solution for handling noise in frequency information. As is well known, in the frequency information obtained from the Fourier transform, local maximum values within a small range represent more significant extremal components. Therefore, before calculating the loss function, we progressively detect whether a frequency is a local maximum within a frequency width d , starting from the low frequencies. If ξ_i is a local maximum of the signal amplitude, we perform a signal assignment correction on it:

$$\xi_i^* = \frac{i}{\eta} \xi_i \quad (9)$$

where i represents the number of frequency components and η is a dimensional balance constant, the core concept lies in adjusting the parameters of the frequency components before computing the loss function. This ensures that the frequency components do not appear as independent computational graphs during gradient computation, leading to smoother gradients. The pseudo-code for FreLE, consisting of two modules, is shown in Algorithm 1.

Algorithm 1 FreLE Algorithm

Require: Time series data X_i , loss balance constant δ , frequency width d , dimensional balance constant η .

Ensure: $\delta\mathcal{L}^f + (1 - \delta)\mathcal{L}^t$

- 1: Frequency: $f[i] \leftarrow \mathcal{F}(X_i)$
 - 2: Amplitude: $A[i] \leftarrow \|f[i]\|$
 - 3: **for** each frequency $f[i]$ in f **do**
 - 4: **if** $A[i] = \max\{A[i - \lfloor \frac{d}{2} \rfloor], \dots, A[i + \lceil \frac{d}{2} \rceil]\}$ **then**
 - 5: $f[i] \leftarrow \frac{i}{\eta} f[i]$
 - 6: **end if**
 - 7: **end for**
 - 8: $\mathcal{L}^f = \text{MAE}(f[i] - \hat{f}[i])$
 - 9: $\mathcal{L}^t = \text{MAE}(X_i - \hat{X}_i)$ **return** $\delta\mathcal{L}^f + (1 - \delta)\mathcal{L}^t$
-

4 Experiment

To verify the effectiveness of FreLE, we will examine it from the following four perspectives:

1. **Performance:** *Does FreLE work?* In Sec. 4.2, we used classical public datasets to compare the performance metrics of FreLE with classical baselines from 2020 to 2024, demonstrating FreLE’s superior performance.
2. **Echanism:** *Why does it work?* In Sec. 4.3, ablation experiments are conducted on the two existing modules separately, and the frequency signal processing method proposed in the 2024 paper is integrated into our module for performance comparison. This demonstrates that the proposed explicit regularization, combined with implicit regularization, is irreplaceable.
3. **Sensitivity:** *Does it require repeated adjustment of hyperparameters?* In Sec. 4.4, we discuss the sensitivity analysis of the hyperparameter δ and validate that FreLE is not sensitive to hyperparameters.
4. **Efficiency:** *Is it effective when reducing the number of parameters?* In Sec. 4.5, performance variation curves with different parameter quantities are presented, demonstrating that FreLE’s method can be effectively utilized in stringent computational environments by reducing the number of parameters while maintaining strong performance.

4.1 Set Up

4.1.1 Baselines.

In our experiments, the comparison baselines we adopted are primarily drawn from studies published between 2020 and 2024. These models can be categorized into three main groups based on their architectures: 1) Methods based on MLP: DLinear [20], RLinear [19], TiDE [21], FreTS [8]; 2) Methods based on the Transformer architecture: Autoformer [22], FEDformer [7], Fredformer [10], iTransformer [25], Stationary [31], TimesX [32]; 3) Other well-known models: TimesNet [4].

4.1.2 Datasets.

The datasets used for long-term forecasting include: ETT (h1, h2, m1, m2), Weather, Traffic, and Electricity [33,34]. The information these datasets provide is summarized in Table 2.

4.1.3 Implementation.

Regarding the reproduction of the baseline, it is based on the script of TimesNet [4] and FreDF [6]. Our experiments are conducted on GPU RTX 4090 and CPU with 14 cores, AMD EPYC 7453. The FreLE loss module is inserted into the DLinear model.

4.2 Result

Table 3 presents the prediction performance of different models across four selected datasets, with an input sequence length of 96 and prediction lengths of 96, 192, 336, and 720. In the complete set of seven datasets, FreLE achieved 21 first-place rankings and 17 second-place rankings.

4.3 Ablation Studies

In this section, we will verify the irreplaceability of implicit regularization. The modules for explicit regularization have been discussed in several classical papers [6,35]. However, explicit regularization methods have certain drawbacks, such as introducing Fourier noise and significant variations in the amplitudes of frequency components. Many recent studies have highlighted that adaptive normalization methods can alleviate the disadvantages of explicit regularization [30,36]. Compared to traditional normalization methods, can the implicit regularization proposed by FreLE better address the shortcomings of explicit regularization? As shown in the ablation and module comparison experiments in Table 4, the performance of the FreLE module is optimal across the four datasets—ETTh1, ETTh2, ECL, and Weather. It outperforms traditional normalization methods in extracting frequency features from time series.

Table 2: Benchmark dataset summary

Datasets	Weather	Electricity	ETTh1	ETTh2	ETTh1	ETTh2	Traffic
#Frequency	10min	Hourly	Hourly	Hourly	15min	15min	Hourly
#Channel	21	321	7	7	7	7	862
#D	21	321	7	7	7	7	862
#Timesteps	52969	26304	17420	17420	69680	69680	17544

Table 3: Multivariate forecasting results with prediction lengths $S \in \{96, 192, 336, 720\}$ for all datasets and fixed look-back length $T = 96$. Experimental results for some datasets, with the **best** and **second best** results are highlighted.

Models	FreIE (Ours)		iTransformer 2023		RLinear 2023		Fredformer 2024		Crossformer 2023		TiDE 2023		TimesNet 2022		DLinear 2023		FreTS 2022		FEDformer 2022		Stationary 2022		Autoformer 2021	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.319 0.356	0.334	0.368	0.355	0.376	<u>0.326</u> <u>0.361</u>	0.404	0.426	0.364	0.387	0.338	0.375	0.345	0.372	0.339	0.374	0.379	0.419	0.386	0.398	0.505	0.475	
	192	<u>0.371</u> 0.379	0.377	0.391	0.391	0.392	0.363 <u>0.380</u>	0.450	0.451	0.398	0.404	0.374	0.387	0.380	0.389	0.382	0.397	0.426	0.441	0.459	0.444	0.553	0.496	
	336	0.393 0.401	0.426	0.420	0.424	0.415	<u>0.395</u> <u>0.403</u>	0.532	0.515	0.428	0.425	0.410	0.411	0.413	0.413	0.421	0.426	0.445	0.459	0.495	0.464	0.621	0.537	
	720	<u>0.463</u> <u>0.443</u>	0.491	0.459	0.487	0.450	0.453 0.438	0.666	0.589	0.487	0.461	0.478	0.450	0.474	0.453	0.485	0.462	0.543	0.490	0.585	0.516	0.671	0.561	
	Avg	0.386 0.394	0.407	0.410	0.414	0.407	<u>0.387</u> <u>0.400</u>	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407	0.407	0.415	0.448	0.452	0.481	0.456	0.588	0.517	
ETTm2	96	0.174 0.256	0.180	0.264	0.182	0.265	<u>0.177</u> <u>0.259</u>	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.292	0.190	0.282	0.203	0.287	0.192	0.274	0.255	0.339	
	192	0.239 0.295	0.250	0.309	0.246	0.304	<u>0.243</u> <u>0.301</u>	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.362	0.260	0.329	0.269	0.328	0.280	0.339	0.281	0.340	
	336	0.300 0.334	0.311	0.348	0.307	0.342	<u>0.302</u> <u>0.340</u>	0.597	0.542	0.377	0.422	0.321	0.351	0.369	0.427	0.373	0.405	0.325	0.366	0.334	0.361	0.339	0.372	
	720	0.399 0.392	0.412	0.407	0.407	0.398	<u>0.397</u> <u>0.396</u>	1.730	1.042	0.558	0.524	0.408	0.403	0.554	0.522	0.517	0.499	0.421	0.415	0.417	0.413	0.433	0.432	
	Avg	0.278 0.319	0.288	0.332	0.286	0.327	<u>0.279</u> <u>0.324</u>	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401	0.335	0.379	0.305	0.349	0.306	0.347	0.327	0.371	
ETTTh1	96	0.371 0.392	0.386	0.405	0.386	<u>0.395</u>	<u>0.373</u> 0.392	0.423	0.448	0.479	0.464	0.384	0.412	0.386	0.400	0.399	0.433	0.376	0.419	0.513	0.491	0.449	0.459	
	192	0.425 <u>0.423</u>	0.441	0.436	0.437	0.424	<u>0.433</u> 0.420	0.471	0.474	0.525	0.492	0.436	0.429	0.437	0.432	0.453	0.443	0.437	0.448	0.534	0.504	0.500	0.482	
	336	0.467 <u>0.445</u>	0.487	0.458	0.479	0.446	<u>0.470</u> 0.437	0.570	0.546	0.565	0.515	0.491	0.469	0.481	0.459	0.503	0.475	0.479	0.465	0.588	0.535	0.521	0.496	
	720	<u>0.476</u> <u>0.473</u>	0.503	0.491	0.456 0.470		<u>0.467</u> 0.488	0.653	0.621	0.594	0.558	0.521	0.500	0.519	0.516	0.596	0.565	0.506	0.507	0.643	0.616	0.514	0.512	
	Avg	0.434 <u>0.433</u>	0.454	0.447	<u>0.435</u> <u>0.433</u>		<u>0.435</u> 0.426	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.488	0.474	0.450	0.460	0.570	0.537	0.496	0.487	
ETTTh2	96	<u>0.284</u> <u>0.336</u>	0.297	0.349	0.288 0.338		0.293 0.342	0.745	0.584	0.400	0.440	0.340	0.374	0.333	0.387	0.350	0.403	0.358	0.397	0.476	0.458	0.346	0.388	
	192	0.370 0.388	0.380	0.400	0.374	0.390	<u>0.371</u> <u>0.389</u>	0.877	0.656	0.528	0.509	0.402	0.414	0.477	0.476	0.472	0.475	0.429	0.439	0.512	0.493	0.456	0.452	
	336	<u>0.413</u> 0.403	0.428	0.432	0.415	0.426	0.382 <u>0.409</u>	1.043	0.731	0.643	0.571	0.452	0.452	0.594	0.541	0.564	0.528	0.496	0.487	0.552	0.551	0.482	0.486	
	720	0.412 0.429	0.427	0.445	<u>0.415</u> <u>0.434</u>		0.431 0.446	1.104	0.763	0.874	0.679	0.462	0.468	0.831	0.657	0.815	0.654	0.463	0.474	0.562	0.560	0.515	0.511	
	Avg	<u>0.370</u> 0.389	0.383	0.407	0.374	0.398	0.365 <u>0.393</u>	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515	0.550	0.515	0.437	0.449	0.526	0.516	0.450	0.459	
ECL	96	0.144 0.249	0.148	0.240	0.201	0.281	<u>0.147</u> <u>0.241</u>	0.219	0.314	0.237	0.329	0.168	0.272	0.197	0.282	0.189	0.277	0.193	0.308	0.169	0.273	0.201	0.317	
	192	0.162 <u>0.255</u>	0.162 0.253	0.201	0.283		<u>0.165</u> 0.258	0.231	0.322	0.236	0.330	0.184	0.289	0.196	0.285	0.193	0.282	0.201	0.315	0.182	0.286	0.222	0.334	
	336	0.179 0.285	<u>0.178</u> 0.269	0.177 <u>0.273</u>	0.181	0.305	<u>0.181</u> 0.305	0.246	0.337	0.249	0.344	0.198	0.300	0.209	0.301	0.207	0.296	0.214	0.329	0.200	0.304	0.231	0.338	
	720	0.213 0.296	0.225	0.317	0.257	0.331	0.213 <u>0.304</u>	0.280	0.363	0.284	0.373	<u>0.220</u> 0.320	0.245	0.333	0.245	0.332	0.246	0.355	0.222	0.321	0.254	0.361		
	Avg	0.175 0.271	0.178	<u>0.270</u>	0.219	0.298	<u>0.177</u> 0.269	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300	0.209	0.297	0.214	0.327	0.193	0.296	0.227	0.338	
Traffic	96	0.412 0.294	0.395 0.268	0.649	0.389		<u>0.406</u> <u>0.277</u>	0.522	0.290	0.805	0.493	0.593	0.321	0.650	0.396	0.528	0.341	0.587	0.366	0.612	0.338	0.613	0.388	
	192	0.437 0.286	0.417 0.276	0.601	0.366		<u>0.426</u> 0.290	0.530	0.293	0.756	0.474	0.617	0.336	0.598	0.370	0.193	0.282	0.604	0.373	0.613	0.340	0.616	0.382	
	336	0.438 <u>0.282</u>	<u>0.433</u> 0.283	0.609	0.369		0.432 0.281	0.558	0.305	0.762	0.477	0.629	0.336	0.605	0.373	0.551	0.345	0.621	0.383	0.618	0.328	0.622	0.337	
	720	0.459 0.299	0.467	0.302	0.647	0.387	<u>0.463</u> <u>0.300</u>	0.589	0.328	0.719	0.449	0.640	0.350	0.645	0.394	0.598	0.367	0.626	0.382	0.653	0.355	0.660	0.408	
	Avg	0.436 0.290	0.428 0.282	0.626	0.378		<u>0.431</u> <u>0.287</u>	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383	0.552	0.348	0.610	0.376	0.624	0.340	0.628	0.379	
Weather	96	0.171 <u>0.212</u>	0.174	0.214	0.192	0.232	<u>0.163</u> 0.207	0.158 0.230	0.202	0.261		0.172	0.220	0.196	0.255	0.184	0.239	0.217	0.296	0.173	0.223	0.266	0.336	
	192	0.219 0.245	0.221	0.254	0.240	0.271	<u>0.211</u> <u>0.251</u>	0.206 0.277	0.242	0.298	0.219	0.223	0.275	0.296	0.261	0.340	0.276	0.336	0.245	0.285	0.307	0.367		
	336	0.258 0.304	0.278	<u>0.296</u>	0.292	0.307	<u>0.267</u> 0.292	0.272	0.335	0.287	0.335	0.280	0.306	0.283	0.335	0.272	0.316	0.339	0.380	0.321	0.338	0.359	0.395	
	720	0.341 <u>0.348</u>	0.358	0.349	0.364	0.353	<u>0.343</u> 0.341	0.398	0.418	0.351	0.386	0.365	0.359	0.345	0.381	0.340	0.363	0.403	0.428	0.414	0.410	0.419	0.428	
	Avg	<u>0.247</u> <u>0.277</u>	0.258	0.279	0.272	0.291	0.246 0.272	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.255	0.299	0.309	0.360	0.288	0.314	0.338	0.382	
1 st Count	21	17	4	6	3	2	<u>6</u> <u>10</u>	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	

Table 4: Averaged results for each setting in the ablation study. EFR stands for Explicit Frequency Regularization, IFR stands for Implicit Frequency Regularization, and AN stands for Adaptive Normalization.

Setting	EFR-IFR		EFR		EFR-AN	
	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.386	0.394	0.411	0.432	0.407	0.435
ETTm2	0.278	0.319	0.293	0.325	0.280	0.351
ECL	0.175	0.271	0.197	0.311	0.251	0.294
Weather	0.247	0.277	0.254	0.291	0.255	0.283

4.4 Hyperparameter Sensitivity

In this section, we conduct a sensitivity analysis on the frequency loss balance hyperparameter. For the ETTm1 and ECL datasets, we select points at 0.1 intervals for $\delta \in [0, 1]$ and perform experiments. The relationship between the hyperparameter and model performance is illustrated in Figure A. It can be observed that when $\delta = 0$, the model performs worst, as the frequency regularization method is not applied. Additionally, directly setting $\delta = 1$ without hyperparameter tuning also yields good experimental performance. This observation is consistent with the frequency decoupling phenomenon discussed in FreDF [6]. Notably, at $\delta = 0.3$, the experimental performance is generally optimal, with the loss values for both frequency domain and time domain losses being nearly identical. This indicates that the best experimental results are achieved when the importance of both tasks is balanced equally.

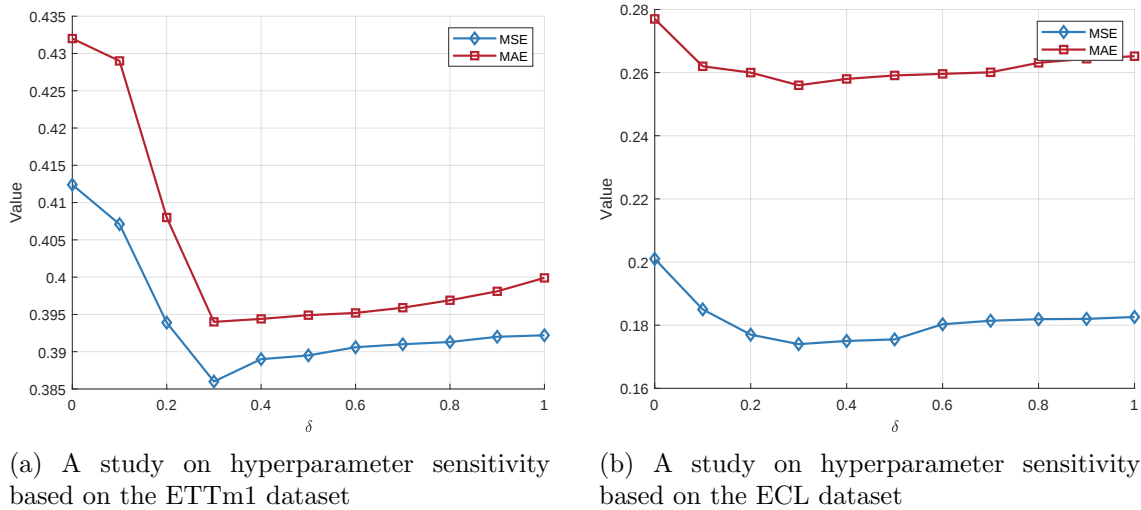
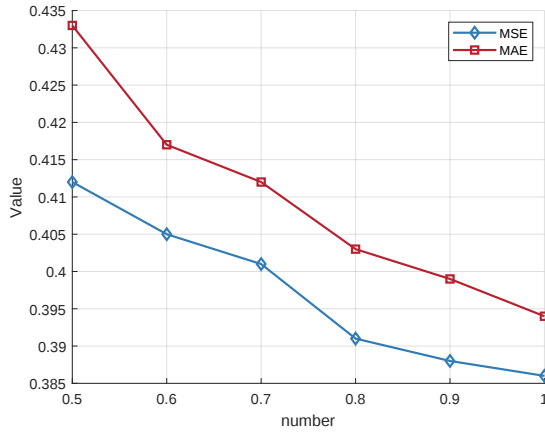


Figure 5: Hyperparameter Sensitivity.

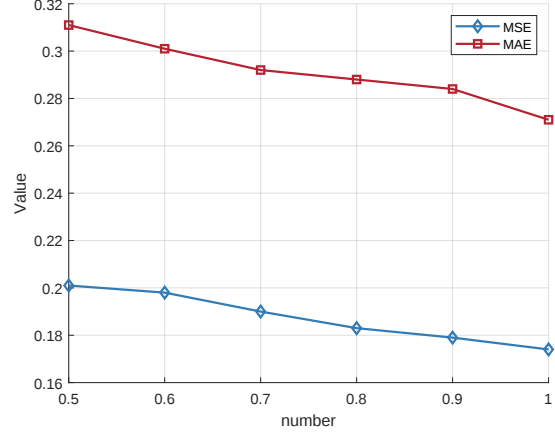
4.5 Parameter-Performance Curve Analysis

In this section, we will reduce the parameter usage of FreLE to analyze its primary impact on the model. The model parameters of FreLE mainly arise from those involved in calculating frequency signals for each layer in explicit regularization. In this section, we will introduce the method of amplitude filtering to reduce the parameter quantity adaptively [37, 38]: set a threshold ϵ_ξ for the signal amplitude, and let $\xi_i = 0$ if and only if $|\xi| < \epsilon_\xi$. The relationship between parameters and performance is illustrated in Figure 6, where the number of neural network parameters is defined as $num \in [0.5, 1]$.

When the parameter retention rate is $num = 0.8$, the model has already stabilized, as observed from the experimental results on ETTm1 and ECL. By reducing 20% of the model parameters while maintaining performance, the model’s performance only decreases by 2%.



(a) The relationship between the percentage of parameter 'num' retained and model performance on the ETTm1 dataset.



(b) The relationship between the percentage of parameter 'num' retained and model performance on the ECL dataset.

Figure 6: Parameter-Performance Curve.

5 Conclusion

This paper adopts the dynamic approach of spectral bias as its starting point and thoroughly investigates the phenomenon of spectral bias in 2-DNNs and time series models. After validating through extensive empirical experiments that nearly all time series models exhibit the spectral bias phenomenon, we propose the FreLE algorithm, which consists of two modules: explicit regularization and implicit regularization. Our extensive experiments on seven datasets demonstrate the high efficiency of the FreLE algorithm. Furthermore, the ablation experiments, sensitivity analysis, and discussions on computational efficiency indirectly confirm the irreplaceable role of the implicit regularization module in FreLE. In the future, we plan to explore the development of new optimization algorithms by leveraging the implicit regularization method we have adopted, with the goal of better utilizing the information priors provided during the Fourier transformation process to address a broader range of problems.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62271234, the Open Foundation of State Key Laboratory of Public Big Data (Guizhou University) under Grant No. PBD2022-16, the Fundamental Research Funds for Heilongjiang Universities under Grant 2022-KYYWF-1042, Double First-Class Project for Collaborative Innovation Achievements in Disciplines Construction in Heilongjiang Province under Grant No. LJGXCG2022-054 and LJGXCG2023-028.

References

- [1] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3d neural networks," *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.
- [2] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

- [3] H. Lange, S. L. Brunton, and J. N. Kutz, “From fourier to koopman: Spectral methods for long-term time series prediction,” *Journal of Machine Learning Research*, vol. 22, no. 41, pp. 1–38, 2021.
- [4] 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.
- [5] Z. Xu, A. Zeng, and Q. Xu, “Fits: Modeling time series with 10k parameters,” *arXiv preprint arXiv:2307.03756*, 2023.
- [6] H. Wang, L. Pan, Z. Chen, D. Yang, S. Zhang, Y. Yang, X. Liu, H. Li, and D. Tao, “Fredf: Learning to forecast in frequency domain,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.02399>
- [7] 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.
- [8] 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.
- [9] M. Li, K. Liu, H. Chen, J. Bu, H. Wang, and H. Wang, “Tsinr: Capturing temporal continuity via implicit neural representations for time series anomaly detection,” *arXiv preprint arXiv:2411.11641*, 2024.
- [10] X. Piao, Z. Chen, T. Murayama, Y. Matsubara, and Y. Sakurai, “Fredformer: Frequency debiased transformer for time series forecasting,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2400–2410.
- [11] J. Geiping, M. Goldblum, P. E. Pope, M. Moeller, and T. Goldstein, “Stochastic training is not necessary for generalization,” *arXiv preprint arXiv:2109.14119*, 2021.
- [12] Y. Zhang, Z.-Q. J. Xu, T. Luo, and Z. Ma, “Explicitizing an implicit bias of the frequency principle in two-layer neural networks,” *arXiv preprint arXiv:1905.10264*, 2019.
- [13] T. Luo, Z. Ma, Z.-Q. J. Xu, and Y. Zhang, “Theory of the frequency principle for general deep neural networks,” *arXiv preprint arXiv:1906.09235*, 2019.
- [14] A. Jacot, F. Gabriel, and C. Hongler, “Neural tangent kernel: Convergence and generalization in neural networks,” *Advances in neural information processing systems*, vol. 31, 2018.
- [15] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, “Deep neural networks as gaussian processes,” *arXiv preprint arXiv:1711.00165*, 2017.
- [16] B. Hanin and M. Nica, “Finite depth and width corrections to the neural tangent kernel,” *arXiv preprint arXiv:1909.05989*, 2019.
- [17] J. Sohl-Dickstein, R. Novak, S. S. Schoenholz, and J. Lee, “On the infinite width limit of neural networks with a standard parameterization,” *arXiv preprint arXiv:2001.07301*, 2020.
- [18] T. Luo, Z.-Q. J. Xu, Z. Ma, and Y. Zhang, “Phase diagram for two-layer relu neural networks at infinite-width limit,” *Journal of Machine Learning Research*, vol. 22, no. 71, pp. 1–47, 2021.

- [19] Z. Li, S. Qi, Y. Li, and Z. Xu, “Revisiting long-term time series forecasting: An investigation on linear mapping,” *arXiv preprint arXiv:2305.10721*, 2023.
- [20] 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.
- [21] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, “Long-term forecasting with tide: Time-series dense encoder,” *arXiv preprint arXiv:2304.08424*, 2023.
- [22] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with autocorrelation for long-term series forecasting,” *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.
- [23] 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.
- [24] Y. Wang, “Frequencies of the ricker wavelet,” *Geophysics*, vol. 80, no. 2, pp. A31–A37, 2015.
- [25] 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.
- [26] Y. Wang, H. Wu, J. Dong, G. Qin, H. Zhang, Y. Liu, Y. Qiu, J. Wang, and M. Long, “Timexer: Empowering transformers for time series forecasting with exogenous variables,” *arXiv preprint arXiv:2402.19072*, 2024.
- [27] L. Wu and W. J. Su, “The implicit regularization of dynamical stability in stochastic gradient descent,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 37 656–37 684.
- [28] F. J. Harris, “On the use of windows for harmonic analysis with the discrete fourier transform,” *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 2005.
- [29] C. Mateo and J. A. Talavera, “Short-time fourier transform with the window size fixed in the frequency domain,” *Digital Signal Processing*, vol. 77, pp. 13–21, 2018.
- [30] W. Ye, S. Deng, Q. Zou, and N. Gui, “Frequency adaptive normalization for non-stationary time series forecasting,” *arXiv preprint arXiv:2409.20371*, 2024.
- [31] Y. Liu, H. Wu, J. Wang, and M. Long, “Non-stationary transformers: Exploring the stationarity in time series forecasting,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.
- [32] Y. Wang, H. Wu, J. Dong, Y. Liu, Y. Qiu, H. Zhang, J. Wang, and M. Long, “Timexer: Empowering transformers for time series forecasting with exogenous variables,” *Advances in Neural Information Processing Systems*, 2024.
- [33] 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, 2021, pp. 11 106–11 115.
- [34] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, “Deep time series models: A comprehensive survey and benchmark,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.13278>

- [35] H. Lange, S. L. Brunton, and J. N. Kutz, “From fourier to koopman: Spectral methods for long-term time series prediction,” *Journal of Machine Learning Research*, vol. 22, no. 41, pp. 1–38, 2021.
- [36] K. Yi, J. Fei, Q. Zhang, H. He, S. Hao, D. Lian, and W. Fan, “Filtarnet: Harnessing frequency filters for time series forecasting,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.01623>
- [37] J. Karki, “Active low-pass filter design,” *Texas Instruments application report*, 2000.
- [38] J. van Driel, C. N. Olivers, and J. J. Fahrenfort, “High-pass filtering artifacts in multivariate classification of neural time series data,” *Journal of Neuroscience Methods*, vol. 352, p. 109080, 2021.
- [39] K. Ahn, J. Zhang, and S. Sra, “Understanding the unstable convergence of gradient descent,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 247–257.
- [40] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, “Fantastic generalization measures and where to find them,” *arXiv preprint arXiv:1912.02178*, 2019.
- [41] Z. Liu, W. Cai, and Z.-Q. J. Xu, “Multi-scale deep neural network (mscalednn) for solving poisson-boltzmann equation in complex domains,” *arXiv preprint arXiv:2007.11207*, 2020.
- [42] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 5301–5310.
- [43] P. Verma, “Neural architectures learning fourier transforms, signal processing and much more....” 2023. [Online]. Available: <https://arxiv.org/abs/2308.10388>
- [44] L. B. Godfrey and M. S. Gashler, “Neural decomposition of time-series data for effective generalization,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 2973–2985, 2017.
- [45] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
- [46] Y. Chen, S. Liu, J. Yang, H. Jing, W. Zhao, and G. Yang, “A joint time-frequency domain transformer for multivariate time series forecasting,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.14649>

Recent Work on Spectral Bias Phenomenon

Progress in the interpretability of deep learning has been challenging. Compared to traditional modeling theories, the vast number of parameters in deep learning should theoretically suggest a negative outcome: overfitting. However, despite the increasing number of parameters in deep learning network architectures, overfitting, as predicted by traditional modeling theory, does not seem to occur. Thus, developing a robust theoretical understanding of this non-overfitting phenomenon has become increasingly important.

Some researchers aim to establish a theoretical framework for neural networks by beginning with idealized assumptions about DNNs models and applying classical optimization theories through rigorous mathematical proofs. For example, when the width of a neural network approaches infinity, the training dynamics under gradient descent optimization can be approximated by a linearized model governed by the Neural Tangent Kernel (NTK) [14, 15]. Neural

networks excel at learning both simple and complex interaction effects within data but struggle with interactions of moderate complexity, a phenomenon known as the "representation bottleneck".

While these studies provide a solid theoretical foundation for neural networks, they all rely on complex mathematical assumptions. A significant challenge is that during the training process, the gradient's sharpness often exceeds theoretical thresholds [39], undermining the reliability of some classical assumptions and rendering them insufficient to explain the behavior of general neural networks. Furthermore, a case study suggests that norm-based complexity measures perform poorly in stochastic optimization, sometimes even adversely affecting the generalization of neural networks [40]. This reality encourages the exploration of phenomenological approaches to better understand neural network theory.

The frequency principle is a recently discovered phenomenological approach to explaining neural network phenomena. Xu et al. [41] observed that over-parameterized DNNs tend to use low-frequency functions to fit training data. These networks initially capture the low-frequency components of the training data and while maintaining the high-frequency components at a smaller magnitude. To extract high-frequency components from the training data, techniques such as the discrete Fourier transform or the design of relaxed objective functions can be employed, which convert high-frequency signals to low-frequency signals. The frequency principle has also been applied to guide the solution of partial differential equations (PDEs). Various researchers have repeatedly demonstrated the reliability of this principle. For instance, Rahaman et al. [42] proposed the concept of spectral bias in the learning process of neural networks. Additionally, Prateek Verma [43], in a technical report at Stanford University, introduced the implicit Fourier transform operations within neural network architectures. These studies have thoroughly investigated methods for studying time series in the frequency domain. Researchers in time series analysis recognized the potential of frequency domain features early on. Work related to frequency domain features has been continuously proposed: as early as Godfrey et al.'s study [44], Fourier decomposition was used to enhance model generalization. However, early neural network architectures were unsuitable for multimodal learning in both the time and frequency domains, and significant progress was not made until the introduction of the Transformer deep learning model architecture.

The self-attention and multi-head attention mechanisms in Transformers significantly accelerated the development of multimodal learning. Since then, many time series forecasting models based on the Transformer architecture that integrate time and frequency domains have been proposed, such as TFT [45], FEDformer [7], and JTFT [46]. The excellent performance of the Transformer architecture in learning frequency domain features has given scholars confidence to apply this method in the MLP domain. Some notable MLP methods, such as Timenet [4], FreDF [6], and FITS [5], have been continuously explored by researchers.