Deep Frequency Derivative Learning for Non-stationary Time Series Forecasting

Wei Fan¹, Kun Yi²*, Hangting Ye³, Zhiyuan Ning⁴, Qi Zhang⁵, Ning An⁶
¹Medical Sciences Division, University of Oxford
²School of Computer Science and Technology, Beijing Institute of Technology
³School of Artificial Intelligence, Jilin University
⁴Computer Science and Technology, Tongji University
⁵Department of Computer Science and Technology, Tongji University
⁶School of Computer Science and Information Engineering, Hefei University of Technology

wei.fan@wrh.ox.ac.uk, yikun@bit.edu.cn, yeht2118@mails.jlu.edu.cn,
ningzhiyuan@cnic.cn, zhangqi_cs@tongji.edu.cn, ning.g.an@acm.org

Abstract

While most time series are non-stationary, it is inevitable for models to face the distribution shift issue in time series forecasting. Existing solutions manipulate statistical measures (usually mean and std.) to adjust time series distribution. However, these operations can be theoretically seen as the transformation towards zero frequency component of the spectrum which cannot reveal full distribution information and would further lead to information utilization bottleneck in normalization, thus hindering forecasting performance. To address this problem, we propose to utilize the whole frequency spectrum to transform time series to make full use of data distribution from the frequency perspective. We present a deep frequency derivative learning framework, DERITS, for non-stationary time series forecasting. Specifically, DERITS is built upon a novel reversible transformation, namely Frequency Derivative Transformation (FDT) that makes signals derived in the frequency domain to acquire more stationary frequency representations. Then, we propose the Order-adaptive Fourier Convolution Network to conduct adaptive frequency filtering and learning. Furthermore, we organize DERITS as a parallel-stacked architecture for the multi-order derivation and fusion for forecasting. Finally, we conduct extensive experiments on several datasets which show the consistent superiority in both time series forecasting and shift alleviation.

1 Introduction

Time series forecasting has been playing an important role in a variety of real-world industries, such as traffic analysis [Ben-Akiva et al., 1998], weather prediction [Lorenz, 1956], financial estimation [King, 1966; Ariyo et al., 2014], energy planning [Deb et al., 2017], etc. Following by classic statistical methods (e.g., ARIMA [Whittle, 1963]), many

*Corresponding Author.

Figure 1: Given one time series and its frequency spectrum, the main comparison between existing works (a) and our method (b).

... deep machine learning-based time series forecasting methods [Salinas et al., 2020; Oreshkin et al., 2019] have recently achieved superior performance in different scenarios. Despite the remarkable success, the non-stationarity widely existing in time series data has still been a critical but under-addressed challenge for accurate forecasting [Priestley and Rao, 1969; Huang et al., 1998; Brockwell and Davis, 2009].

Since time series data are usually collected at a high frequency over a long duration, such non-stationary sequences with millions of timesteps inevitably let forecasting models face the distribution shifts over time. This would lead to performance degradation at test time [Kim et al., 2022] due to the covariate shift or the conditional shift [Woo et al., 2022a]. For this issue, pioneer works [Ogasawara et al., 2010] propose to normalize time series data with global statistics; one recent work [Kim et al., 2022] proposes to use instance statistics to normalize time series against distribution shifts. Then, some work brings statistical information into self-attention computation [Liu et al., 2022b]; another work [Fan et al., 2023] transform time series with learnable statistics and consider shifts input and output sequences, and [Liu et al., 2023] utilize predicted sliced statistics for adaptive normalization.

Most of these existing works focus on transforming each timestep of time series with certain statistics (usually mean and std.) After our careful theoretical analysis¹, we have found that these operations can actually be regarded as the

¹We leave the details of our theoretical analysis in Appendix B.1
normalization towards the zero frequency component of the spectrum in the frequency domain, as shown in Figure 1(a). However, they cannot fully utilize distribution information of time series signals; moreover, this would lead to information utilization bottleneck in normalization and thus hinders the performance of time series forecasting. To address this problem, we propose to utilize the entire frequency spectrum for the transformation of time series to make full use of distribution information of time series from the frequency perspective and in the meanwhile transform time series into more stationary space thus making more accurate forecasting. Figure 1(b) has shown our method with the whole frequency spectrum.

Motivated by this view, we then present a deep frequency derivative learning framework, DeRiTS, for non-stationary time series forecasting. The core idea of DeRiTS lies in two folds: (i) employing the whole frequency spectrum to take the derivative of time series signals, and (ii) learning frequency dependencies on more stationary transformed representations. Specifically, we first propose a novel transformation for time series signals in DeRiTS, namely Frequency Derivative Transformation (FDT), which mainly includes two stages. In the first stage, the raw signals in the time domain are transformed into the frequency domain with Fourier transform [Nussbaumer and Nussbaumer, 1982] for further learning. In the second stage, the transformed frequency components are derived with respect to timestamps to get more stationary frequency representations. Inspired by the derivative in mathematics [Hirsa and Neftci, 2013], FDT let models aim for modeling gradients of signals rather than raw input signals, which could mitigate their burden of forecasting with distribution shifts by resolving non-stationary factors (e.g., the shift of trends) in the original time series through one- or high-order derivation.

After acquiring more stationary representations, we further propose a novel architecture, Order-adaptive Fourier Convolution Network (OFCN) in DeRiTS for the frequency filtering and dependency learning to accomplish the forecasting. Concretely, OFCN is composed of (i) Order-adaptive frequency filter that adaptively extracts meaningful patterns by excluding high-frequency noises for derived signals of different orders, and (ii) Fourier convolutions that conduct dependency mappings and learning for complex values in the frequency domain. Since OFCN is operating in the projection space by FDT, we thus utilize the inverse Frequency Derivative Transformation to recover the predicted frequency components back to the original time domain. Inspired by previous work [Kim et al., 2022], we let all stages of FDT fully reversible and symmetric and make OFCN predict in the more stationary frequency space, which reveals our superiority in enhancing forecasting against distribution shifts. Furthermore, in order for the multi-order derivative learning, we have organized DeRiTS as a parallel-stacked architecture to fuse representations of different orders. Specifically, DeRiTS is composed of several parallel branches, each of which represents an order of derivation and prediction corresponding for its FDT and OFCN. Note that the Fourier convolution adapted in each branch is not parameter-sharing and after the distinct processing of different branches, the outputs are fused to achieve the final time series forecasting. In summary, our main contribution can be listed as follows:

- Motivated by our theoretical analysis towards existing time series normalization techniques from the frequency spectrum perspective, we propose to utilize the entire frequency spectrum for the transformation of time series.
- We present a deep frequency derivative learning framework, namely DeRiTS, built upon our proposed Frequency Derivative Transformation (with its inverse) for non-stationary time series forecasting.
- We introduce the novel Order-adaptive Fourier Convolution Network, for the frequency dependency learning and organize DeRiTS as a parallel-stacked architecture to fuse multi-order representations for forecasting.
- We have conducted extensive experiments on seven real-world datasets, which have demonstrated the consistent superiority compared with state-of-the-art methods in both time series forecasting and shift alleviation.

2 Related Work

2.1 Time Series Forecasting with Non-stationarity

Time series forecasting is a longstanding research topic. Traditionally, researchers have proposed statistical approaches, including exponentially weighted moving averages [Holt, 1957] and ARMA [Whittle, 1951]. Recently, with the advanced development of deep learning, many deep time series forecasting methods have been developed, including RNN-based methods (e.g., deepAR [Salinas et al., 2020], LSTM [Lai et al., 2018]), CNN-based methods (e.g., SCINet [Liu et al., 2021], TCN [Bai et al., 2018]), MLP-based Methods (e.g., DLinear [Zeng et al., 2023], N-BEATS [Oreshkin et al., 2020]) and Transformer-based methods (e.g., Autoformer [Wu et al., 2021], PatchTST [Nie et al., 2023]). While time series are non-stationary, existing works try to normalize time series with global statistics [Ogasawara et al., 2010], instance statistics [Kim et al., 2022], learnable statistics [Fan et al., 2023] and sliced statistics [Liu et al., 2023] in order to relieve the influence of distribution shift on forecasting.

2.2 Frequency Analysis in Time Series Modeling

The frequency analysis has been widely used to extract knowledge of the frequency domain in time series modeling and forecasting [Yi et al., 2023a]. Specifically, SFM [Zhang et al., 2017] adopts Discrete Fourier Transform to decompose the hidden state of time series by LSTM into frequency components; StemGNN [Cao et al., 2020] adopts Graph Fourier Transform to perform graph convolutions and FourierGNN [Yi et al., 2024] uses Discrete Fourier Transform to computes series-wise correlations. Autoformer [Wu et al., 2021] replaces self-attention in Transformer [Vaswani et al., 2017] and proposes the auto-correlation mechanism implemented by Fast Fourier Transform. FEDformer [Zhou et al., 2022] introduces Discrete Fourier Transform-based frequency enhanced attention. DEPTS [Fan et al., 2022] utilizes Discrete Cosine Transform to extract periodic patterns. In addition, [Woo et al., 2022b] transforms hidden features of time
series into the frequency domain with Discrete Fourier Transform to enhance the time series representation learning. [Yi et al., 2023b] combines Fast Fourier Transform with multi-layer perceptrons for time series forecasting.

3 Problem Formulation

Time Series Forecasting Let \( x = [x_1; x_2; \ldots; x_T] \in \mathbb{R}^{T \times D} \) be regularly sampled multi-variate time series with \( T \) timestamps and \( D \) variates, where \( x_t \in \mathbb{R}^D \) denotes the multi-variate values at timestamp \( t \). In the task of time series forecasting, we use \( X_t \in \mathbb{R}^{L \times D} \) to denote the lookback window, a length-\( L \) segment of \( x \) ending at timestamp \( t \) (exclusive), namely \( X_t = x_{t-L:t} = [x_{t-L}; x_{t-L+1}; \ldots; x_{t-1}] \). Similarly, we represent the horizon window as a length-\( H \) segment of \( x \) starting from timestamp \( t \) (inclusive) as \( Y_t \), so we have \( Y_t = x_{t:t+H} = [x_t; x_{t+1}; \ldots; x_{t+H-1}] \). The classic time series forecasting formulation is to project lookback values \( X_t \) into horizon values \( Y_t \). Specifically, a typical forecasting model \( F_{\theta} : \mathbb{R}^{L \times D} \rightarrow \mathbb{R}^{H \times D} \) produces forecasts by \( \hat{Y}_t = f_{\theta}(X_t) \) where \( \hat{Y}_t \) stands for the forecasting results and \( \theta \) encapsulates the model parameters.

Non-stationarity and Distribution Shifts In this paper, we aim to study the problem of non-stationarity in deep time series forecasting. As aforementioned in Section 1, long time series with millions of timestamps let forecasting models face distribution shifts over time due to the non-stationarity. The distribution shifts in time series forecasting are usually the covariate shift [Wiles et al., 2021; Woo et al., 2022a]. Specifically, given a stochastic process, let \( p(x_t, x_{t-1}, \ldots, x_{t-L+1}) \) be the unconditional joint distribution of a length \( L \) segment where \( x_t \) is the value of univariate time series at timestamp \( t \). The stochastic process experiences covariate shift if any two segments are drawn from different distributions, i.e. \( p(x_t, x_{t-1}, x_{t-L+1}, \ldots, x_{t-1}) \neq p(x_t', x_{t-1}, x_{t-L+1}, \ldots, x_{t-1}) \), \( \forall t \neq t' \). Subsequently, let \( p(x_t \mid x_{t-1}, \ldots, x_{t-L}) \) represents the conditional distribution of \( x_t \), such a stochastic process experiences conditional shift if two segments have different conditional distributions, i.e. \( p(x_t' \mid x_{t-1}, \ldots, x_{t-L+1}, x_{t-L}) \neq p(x_t' \mid x_{t-1}, \ldots, x_{t-L+1}, x_{t-L}) \), \( \forall t \neq t' \).

4 Methodology

In this section, we elaborate on our proposed deep frequency derivative learning framework, DeRiTS, designed for non-stationary time series forecasting. First, we introduce our novel reversible transformation, Frequency Derivative Transformation (FDT) in Section 4.1. Then, to fuse multi-order information, we present the parallel-stacked frequency derivative learning architecture in Section 4.2. Finally, we introduce our Order-adaptive Fourier Convolution Network (OCFN) for frequency learning in Section 4.3.

4.1 Frequency Derivative Transformation

As aforementioned in Section 1, to fully utilize the whole frequency spectrum for the transformation of time series with sufficient distribution information, we propose the novel Frequency Derivative Transformation (FDT) to achieve more stationary frequency representations of time series signals. For this aim, FDT mainly includes two distinct stages respectively for domain transformation and frequency derivation.

Domain Transformation

In the first stage, to be specific, we make use of fast Fourier transform [Nussbaumer and Nussbaumer, 1982] to enable the decomposition of time series signals from the time domain into their inherent frequency components. Formally, given the time domain input signals \( X(t) \), we convert it into the frequency domain by:

\[
X(f) = \mathcal{F}(X(t)) = \int_{-\infty}^{\infty} X(t) e^{-j2\pi ft} dt = \int_{-\infty}^{\infty} X(t) \cos(2\pi ft) dt + j \int_{-\infty}^{\infty} X(t) \sin(2\pi ft) dt,
\]

where \( \mathcal{F} \) is the fast Fourier transform, \( f \) is the frequency variable. \( t \) is the integral variable, and \( j \) is the imaginary unit, defined as the square root of -1. \( \int_{-\infty}^{\infty} X(t) \cos(2\pi ft) dt \) is the real part of \( X \) and is abbreviated as \( \Re(\mathcal{F}(X)) \); \( \int_{-\infty}^{\infty} X(t) \sin(2\pi ft) dt \) is the imaginary part and is abbreviated as \( \Im(\mathcal{F}(X)) \). After that we can rewrite \( X \) as \( X = \mathcal{F}(X) + j \mathcal{I}(X) \).

Frequency Derivation

In the second stage, with the transformed frequency components, we propose to utilize the whole frequency spectrum for the signal derivation, in order to represent time series in a more stationary space. The basic idea is to perform our proposed Fourier Derivative Operator in the frequency domain, which is defined as follows:

**Definition 1 (Fourier Derivative Operator).** Given the time domain input signals \( X(t) \) and its corresponding frequency components \( \mathcal{F}(X) \), we then define \( \mathcal{R}(\mathcal{F}(X)) := (j2\pi f)^k \mathcal{F}(X) \) as the Fourier Derivative Operator (FDO), where \( f \) is the frequency variable and \( j \) is the imaginary unit.

In the derivation, different order usually represents different signal representations. We propose to incorporate multi-order information in DeRiTS to further enhance the forecasting. For this aim, we extend above definition and further define the \( k \)-order Fourier Derivative Operator \( \mathcal{R}_k \) as:

\[
\mathcal{R}_k(\mathcal{F}(X)) = (j2\pi f)^k \mathcal{F}(X).
\]

With such two stages, we can finally write the \( k \)-order Frequency Derivation Transformation FDT\(_k\) as:

\[
FDT_k(X(t)) = (j2\pi f)^k \mathcal{F}(X(t))
\]

where \( X(t) \) is the time domain input signal; \( \mathcal{F} \) stands for fast Fourier transform and \( f \) is the frequency variable.

**Proposition 1.** Given \( X(t) \) in the time domain and \( \mathcal{F}(X) \) in the frequency domain correspondingly, the \( k \)-order Fourier Derivative Operator on \( \mathcal{F}(X) \) is equivalent to \( k \)-order derivation on \( X(t) \) with respect to \( t \) in the time domain, written by:

\[
(j2\pi f)^k X(t) = \frac{d^k X(t)}{dt^k},
\]

where \( \mathcal{F} \) is Fourier transform, \( \frac{d^k}{dt^k} \) is \( k \)-order derivative with respect to \( t \), and \( j \) is the imaginary unit.
We leave the detailed proof in Appendix B.2. With such an equivalence, we can find out FDT can actually achieve more stationary representations in the lower order by derivation. For example, the shifts caused by a single trend signal in time series can nearly be zero. We include the specific analysis in Appendix D. Then, with less distribution shifts and non-stationarity by FDT, the deep networks can have large potential to perform more accurate forecasting.

4.2 Frequency Derivative Learning Architecture

The main architecture of DeRITS is depicted in Figure 2, which is built upon the Frequency Derivative Transformation and its inverse for the frequency derivative learning.

**FDT/iFDT** As mentioned in Section 4.1, DeRITS needs to conduct predictions in a more stationary frequency space achieved by frequency derivative transformation. We naturally need to recover the predictions back to the time domain for final forecasting and evaluation. To make FDT fully reversible, we let both stages of FDT reversible, including Fourier transform and Fourier Derivation. Specifically, following Equation (3), we can symmetrically write the inverse frequency derivative transformation (iFDT) of k order as:

\[
\text{iFDT}_k(f) = \mathcal{F}^{-1}\left(\frac{1}{(j2\pi)^k}f\right),
\]

where \(\mathcal{F}^{-1}\) is the inverse process of Fourier Derivative Operator of k order; \(\mathcal{F}^{-1}\) is the inverse Fourier transform; \(\mathcal{X}(f)\) is the frequency components that need to be recovered to the time domain. Actually, the inverse process \(\mathcal{F}^{-1}\) is equivalent to an integration operator in the time domain. More details can be found in Appendix D.

**The Parallel-Stacked Architecture**

To conduct the multi-order frequency derivation transformation and learning, we have organized our DeRITS framework as a parallel-stacked architecture, where each branch represents an order of frequency derivation learning, as shown in Figure 2. Let DeRITS have \(K\) branches in total. For each branch, we first take lookback values \(X_t\) to frequency derivative transformation by:

\[
X^k_t = \text{FDT}_k(X_t), \hspace{1cm} k = 1, 2, \ldots, K
\]

where FDT\(_k\) is the k-order FDT and \(X^k_t\) is the frequency derivative representation for \(X^k_t\) at timestamp \(t\). Then, the learned representations for each branch are taken to the Fourier Convolution Network (FCN) for frequency dependency learning. Since our FCN is order-adaptive in each parallel branch, we also take \(k\) as input with the computation by:

\[
\mathcal{H}^k_t = \text{Order-adaptiveFourierConvolution}(k, x^k_t)
\]

where \(\mathcal{H}^k_t\) are the predicted frequency components for \(x^k_t\). Note that FourierConvolution is not parameter-sharing for different branches. After that, we recover the predictions to the time domain by:

\[
\hat{Y}_t = \text{MultilayerPerceptron}(H^1_t, H^2_t, \ldots, H^K_t)
\]

where \(\hat{Y}_t\) are forecasting results by DeRITS for evaluation.

4.3 Order-adaptive Fourier Convolution Network

Apart from the frequency derivative transformation, another aim of DeRITS is to accomplish the dependency learning with the derived signals in the frequency domain. We thus introduce a novel network architecture, namely Order-adaptive Fourier Convolution Network (OFCN) to enable the frequency learning. Specifically, OFCN is composed of two important components, i.e., order-adaptive frequency filter and Fourier convolutions, which are illustrated as follows:

**Order-adaptive Frequency Filter** We aim to fuse multi-order derived signal information for forecasting, while it is notable that different order corresponds to different frequency comments. Since time series include not only meaningful patterns but also high-frequency noises, we develop an order-adaptive frequency filter to enhance the learning process.

Supposing there are \(S\) frequencies in \(x^k_t\), we sort \(x^k_t\) on the frequencies in a descending order of amplitude for each frequency to get \(\tilde{x}^k_t\) for \(k\) order. Then, we design an adaptive mask \(m_k\) to concentrate on only \(\frac{S}{2^{(K-k)}}\) frequency components of \(\tilde{x}^k_t\) for further learning. We write the adaptive frequency filtering process by:

\[
\mathcal{H}^k_t = m_k \odot v_k \tilde{x}^k_t = \frac{s}{S/2^{(K-k)}} \odot v_k \tilde{x}^k_t
\]
where \( \mathbf{m}_k \) is the mask vector of length \( S \) for filtering; \( \mathbf{v}_k \) is a randomly-initialized vector of order \( k \) which is learnable; \( \mathcal{H}^k_t \) are the filtered frequency representations. In particular, Equation (11) is inspired by that the low-order derived representations include more noises than more stationary high-order representations and thus should be filtered. To filter frequencies, we design an exponential-masking mechanism for \( \mathbf{m}_k \) to select \( \frac{S}{2(2^k-1)} \) frequencies while filtering others.

**Fourier Convolutions** Given the filtered signal representations in the frequency domain, the subsequent step involves acquiring the dependencies for time series forecasting. Considering that the representations are complex values, it is intuitive to devise a network in which all operations are conducted in the frequency domain. According to the convolution theorem [Katznelson, 1970], the Fourier transform of a convolution of two signals equals the point-wise product of their Fourier transforms in the frequency domain. Accordingly, we employ Fourier convolution layers that involve performing a product in the frequency domain, to capture these dependencies. Specifically, given \( \mathcal{H}^k_t \) achieved by order-adaptive filtering, we compute it as follows:

\[
\mathcal{H}^k_t = \text{FourierConvolution}(\mathcal{H}^k_t) = \mathcal{H}^k_t \mathcal{W}_k 
\]  

(11)

where \( \mathcal{W}_k \) is the weighted matrix to conduct the convolutions in the frequency domain; \( \mathcal{H}^k_t \) is the output by our order-adaptive Fourier convolution network when the order is \( k \). Equation (11) is intuitive which aims to directly learn the dependencies on the filtered components for forecasting.

## 5 Experiments

In this section, in order to evaluate the performance of our model, we conduct extensive experiments on six real-world time series benchmarks to compare with the state-of-the-art time series forecasting methods. Furthermore, we conduct a detailed analysis of our model for better evaluation.

### 5.1 Experimental Setup

**Datasets** We follow previous work [Wu et al., 2021; Zhou et al., 2022; Nie et al., 2023; Yi et al., 2023b] to evaluate our DeRTS on different representative datasets from various application scenarios, including Electricity [Asuncion and Newman, 2007], Trafﬁc [Wu et al., 2021], ETT [Zhou et al., 2021], Exchange [Lai et al., 2018], ILI [Wu et al., 2021], and Weather [Wu et al., 2021]. We preprocess all datasets following the recent frequency learning work [Yi et al., 2023b] to normalize the datasets and split the datasets into training, validation, and test sets by the ratio of 7:2:1. We leave more dataset details in Appendix A.1.

**Baselines** We conduct a comprehensive comparison of the forecasting performance between our model DeRTS and several representative and state-of-the-art (SOTA) models on the six datasets, including Transformer-based models: Informer [Zhou et al., 2021], Autoformer [Wu et al., 2021], FEDformer [Zhou et al., 2022], PatchTST [Nie et al., 2023]; MLP-based model: LSTF-Linear [Zeng et al., 2023]; Frequency domain-based model: FreTS [Yi et al., 2023b]. Besides, we also consider the existing normalization methods towards distribution shifts in time series forecasting, including RevIN [Kim et al., 2022], NSTransformer [Liu et al., 2022b].
and Dish-TS [Fan et al., 2023]. All the baselines we reproduced are implemented based on their official code and we leave more baseline details in Appendix A.2.

**Implementation Details** We conduct our experiments on a single NVIDIA RTX 3090 24GB GPU with PyTorch 1.8 [Paszke et al., 2019]. We take MSE (Mean Squared Error) as the loss function and report the results of MAE (Mean Absolute Errors) and RMSE (Root Mean Squared Errors) as the evaluation metrics. A lower MAE/RMSE indicates better performance of time series forecasting. More detailed information about the implementation is included Appendix A.3.

### 5.2 Overall Performance

To verify the effectiveness of DeRiTS, we conduct the performance comparison of multivariate time series forecasting in several benchmark datasets. Table 1 presents the overall forecasting performance in the metrics of MAE and RMSE under different prediction lengths. In brief, the experimental results demonstrate that DeRiTS achieves the best performances in most cases as shown in Table 1. Quantitatively, compared with the best results of transformer-based models, DeRiTS has an average decrease of more than 20% in MAE and RMSE. Compared with more recent frequency learning model, FreTS [Yi et al., 2023b] and the state-of-the-art transformer model, PathcTST [Nie et al., 2023], DeRiTS can still outperform them in general. This has shown the great potential of DeRiTS in the time series forecasting task.

### 5.3 Comparison with Normalization Techniques

In this section, we further compare our performance with the recent normalization technique, RevIN [Kim et al., 2022] and Dish-TS [Fan et al., 2023] that handle distribution shifts in time series forecasting. Table 2 has shown the performance comparison in time series forecasting taking the LTSF-Linear [Zeng et al., 2023]. Since FDT transforms signals to the frequency domain, we implement a simple single-layer Linear model in the frequency domain. From the results, we can observe that the existing RevIN and Dish-TS can only improve the backbone in some shifted datasets. In some situations, it might lead to worse performances. Nevertheless, our FDT can usually achieve the best performance. A potential explanation is that FDT transforms data with full frequency spectrum and thus achieves stable improvement, while other normalization techniques cannot reveal full data distribution and thus cannot make use of them for transformation.

### 5.4 Model Analysis

**Impact of Frequency Derivative Transformation** It is notable that our proposed FDT plays an important role in DeRiTS, and we aim to analysis the impact of FDT on the model performance. Thus, we consider a special case of FDT, which is when we set $k = 0$, the derivation is removed and FDT is degraded to naive Fourier transform. In addition to this setting, we also vary different orders ($k$) of derivation to test the effectiveness. Table 3 has shown the results on three datasets. We can easily observe that the performance of DeRiTS can beat the variant version without the derivation, which has demonstrate the effectiveness of FDT. Moreover, with the increase of order, the original time series would be derived too much. This might cause the information loss which leads to performance degradation accordingly.

**Impact of Multi-order Stacked Architecture** As aforementioned in Section 4.2, we organize DeRiTS as a parallel-stacked architecture for multi-order fusion. Thus we aim to study the impact of such a stacked architecture. In contrast to multi-order stacked DeRiTS, we have considered another situation of individual-order DeRiTS that removes the parallel-stacked architecture. Figure 3 has shown the performance comparison with individual-order DeRiTS in the Ex-

---

### Table 2: Performance comparisons on MAE and RMSE with state-of-the-art normalization techniques in time series forecasting taking LTSF-Linear as the backbone.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Performance Comparison</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LTSF-Linear</strong></td>
<td>+RevIN</td>
<td>0.030</td>
<td>0.035</td>
</tr>
<tr>
<td></td>
<td>+Dish-TS</td>
<td>0.035</td>
<td>0.040</td>
</tr>
<tr>
<td></td>
<td>+FDT</td>
<td>0.030</td>
<td>0.035</td>
</tr>
</tbody>
</table>

### Table 3: The impact of frequency derivative transformation with order $k$. For Exchange and Weather datasets, the prediction length and the lookback window size are 96. For ILI dataset, the prediction length and the lookback window size are 36 due to length limitation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Metrics</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange</td>
<td>$k = 0$</td>
<td>0.041</td>
<td>0.058</td>
</tr>
<tr>
<td></td>
<td>$k = 1$</td>
<td>0.037</td>
<td>0.053</td>
</tr>
<tr>
<td></td>
<td>$k = 2$</td>
<td>0.035</td>
<td>0.050</td>
</tr>
<tr>
<td>Weather</td>
<td>$k = 3$</td>
<td>0.036</td>
<td>0.052</td>
</tr>
<tr>
<td>ILI</td>
<td>$k = 0$</td>
<td>0.15</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>$k = 1$</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>$k = 2$</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>$k = 3$</td>
<td>0.20</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Figure 3: The forecasting performance (MAE) comparison between original DeRiTS (multi-order) and its individual-order variant. The lower values indicate the better forecasting performance.
change, ILI, and Weather datasets. It can be easily observed that without the parallel-stacked architecture for multi-order fusion, the individual-order DErITs achieves much worse performance than the original one even under different orders, which signifies the necessity of our multi-order design in the frequency derivative learning architecture.

**Lookback Analysis** We aim to examine the impact of the lookback window on forecasting performance of DErITs. Figure 4 has demonstrated the experimental results on the Exchange, Weather, and ILI datasets. Specifically, we maintain the prediction length as 96 and vary the lookback length from 48 to 288 on Exchange and Weather datasets. For the ILI dataset, we keep the prediction length at 36 and alter the lookback window size from 24 to 72. From the results, we can observe that in most cases, larger lookback length would bring up less prediction errors; this is because larger input includes more temporal information. Also, larger input length would also bring more noises hindering forecasting, while our DErITs can achieve comparatively stable performance.

![Figure 4: Impact of lookback length on forecasting. Metrics MAE and RMSE are reported with the length of lookback window prolonged and the prediction length fixed.](image)

**Table 4: Efficiency analysis. We report the training time of DErITs and Non-Stationary (NS) transformer-based methods.**

<table>
<thead>
<tr>
<th>Length</th>
<th>96</th>
<th>192</th>
<th>336</th>
<th>480</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS-FEDformer</td>
<td>137.7</td>
<td>160.4</td>
<td>192.8</td>
<td>227.2</td>
</tr>
<tr>
<td>NS-Autoformer</td>
<td>44.41</td>
<td>59.23</td>
<td>78.29</td>
<td>101.5</td>
</tr>
<tr>
<td>NS-Transformer</td>
<td>30.24</td>
<td>41.38</td>
<td>50.21</td>
<td>61.88</td>
</tr>
<tr>
<td>DErITs</td>
<td>12.57</td>
<td>13.87</td>
<td>14.93</td>
<td>15.93</td>
</tr>
</tbody>
</table>

**Efficiency Analysis** To conduct the efficiency analysis for our framework, we report the training time of DErITs across various prediction lengths, and we also include the training time of the Non-Stationary transformer [Liu et al., 2022b] for comparison, coupled with its corresponding backbones such as FEDformer, Autoformer and Transformer. The experiments are conducted under the prediction length with the same input length of 96 on the Exchange dataset. As shown in Table 4, the results prominently highlight that our model exhibits superior efficiency metrics. Our DErITs significantly reduces the number of parameters thus enhancing the computation speed. In particular, our model showcases an average speed that is several times faster than the baselines. These findings underscore the efficiency gains achieved by our model, positioning it as a compelling choice for non-stationary time series forecasting.

![Figure 5: Visualization comparison of original signals and derived signals with Fourier derivative transformation.](image)

**5.5 Visualization Analysis**

**FDT and Non-stationarity** To study the Fourier derivative transformation in DErITs, we visualize the original signals and derived signals for comparison. Since the direct outputs of FDT are complex values that are difficult to visualize completely, we thus transform the derived frequency components back to the time domain via inverse FDT, which allows us to show the corresponding time values for visualization. Specifically, we choose two non-stationary time series from the Weather dataset and Exchange dataset, respectively. As shown in Figure 5, we can observe the original signals have included obvious non-stationary oscillations and trends. In contrast, the derived signals exhibit a larger degree of stationarity compared with raw data. This further reveals that learning in the derivative representation of signals is more stationary and thus can achieve better performance.

**Case Study of Forecasting** To further analysis the model performance, we carry out the case study for non-stationary time series forecasting. Figure 6 demonstrates the visualization of forecasting results of DErITs and NSTransformer [Liu et al., 2022b] with the prediction length as 96 and lookback length as 96. Upon careful observation of it, it becomes evident that DErITs can be capable of aligning with the ground truth when the time series distribution is largely shifted, while the baseline method deviates from the true values. This demonstrates our model’s adaptability to shifts. We include more visualizations in Appendix E.

**6 Conclusion Remarks**

In this paper, we propose to address non-stationary time series forecasting from the frequency perspective. Specifically, we utilize the whole frequency spectrum for the transformation of time series in order to make full use of time series distribution. Motivated by this point, we propose a deep frequency derivative learning framework DErITs for non-stationary forecasting, which is mainly composed of the Frequency Derivative Transformation and the Order-adaptive Fourier Convolution Network with a parallel-stacked architecture. Extensive experiments on real-world stacked datasets have demonstrated its superiority. Moreover, distribution shifts and non-stationarity are actually a pervasive and crucial topic for time series forecasting. Thus we hope that the new perspective of frequency derivation together with the DErITs framework can facilitate more future related research.
Acknowledgments
This work was partially supported by the National Natural Science Foundation of China (No. 62072153) the Anhui Provincial Key Technologies R&D Program (No. 2022h11020015), the Shanghai Baiyulan Talent Plan Pujiang Project (23PJ1413800), and the National Natural Science Foundation of China (No. 623B2043).

References


