ATFNet: Adaptive Time-Frequency Ensembled Network for Long-term Time Series Forecasting

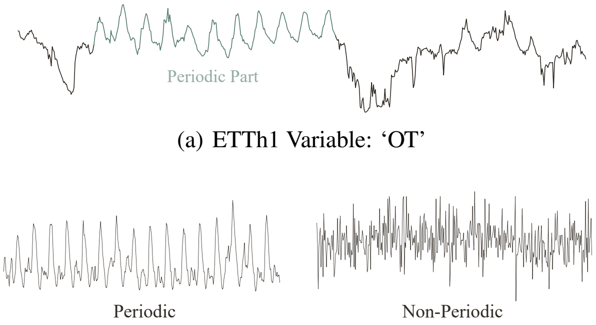
Hengyu Ye [1] Jiadong Chen [1] Shijin Gong [2] Fuxin Jiang [3] Tieying Zhang [3] Jianjun Chen [3] Xiaofeng Gao [1]

Abstract

The intricate nature of time series data analysis benefits greatly from the distinct advantages offered by time and frequency domain representations. While the time domain is superior in representing local dependencies, particularly in non-periodic series, the frequency domain excels in capturing global dependencies, making it ideal for series with evident periodic patterns. To capitalize on both of these strengths, we propose ATFNet, an innovative framework that combines a time domain module and a frequency domain module to concurrently capture local and global dependencies in time series data. Specifically, we introduce Dominant Harmonic Series Energy Weighting, a novel mechanism for dynamically adjusting the weights between the two modules based on the periodicity of the input time series. In the frequency domain module, we enhance the traditional Discrete Fourier Transform (DFT) with our Extended DFT, designed to address the challenge of discrete frequency misalignment. Additionally, our Complex-valued Spectrum Attention mechanism offers a novel approach to discern the intricate relationships between different frequency combinations. Extensive experiments across multiple real-world datasets demonstrate that our ATFNet framework outperforms current state-of-the-art methods in long-term time series forecasting. Codes are publicly available at https://github.com/YHYHYHYHYHY/ATFNet.

1. Introduction

Real-world time series display differentiated periodic patterns. As illustrated in Figure 1, different segments of the same time series, as well as distinct time series within the same dataset, exhibit periodic characteristics with noticeable variations. Periodic time series typically exhibit more global dependencies, where the values of the series at one point in time are influenced by the values at other points in time that are separated by a specific interval. Non-periodic time series, on the other hand, exhibit more local dependencies. In this case, the values at one point in time may be influenced by nearby values in time, but there may not be a consistent and predictable relationship over longer periods.



(a) ETTh1 Variable: ‘OT’

(b) Electricity Variable: ‘3’

(c) Electricity Variable: ‘125’

Figure 1. Real-world time series with distinct periodic pattern.

Time domain and frequency domain are two fundamental representations used to analyze time series data. In time domain, the analysis focuses on the variations in amplitude over time, allowing for the identification of local dependencies and transient behavior within the signal. Conversely, frequency domain analysis aims to represent time series in terms of their frequency components, providing insights into global dependencies and spectral characteristics of the data. Combining the advantages of both domains is a promising approach to address the challenge of dealing with the mixing of distinct periodic properties in real-world time series, but ***how to effectively combine the advantages of time domain and frequency domain*** *(Q_1)* remains a challenge.

Early research predominantly focused on time-domain analysis. Over time, numerous intricate methodologies and sophisticated models have been developed to capture and exploit information in time domain, frequently relying on approaches like different neural network architectures (Zhou

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China [2] School of Management, University of Science and Technology of China, Hefei, Anhui, China [3] ByteDance Inc.. Correspondence to: Xiaofeng Gao <gaoxf@cs.sjtu.edu.cn>.

et al., 2021; Zhang & Yan, 2022). Compared to the advancements in time domain, there are still many unexplored areas in frequency domain. In recent years, some works have employed frequency domain to better handle global dependencies of time series (Zhou et al., 2022b; Yi et al., 2023; Zhou et al., 2022a). Most these works incorporate the frequency domain as an intermediary module instead of directly forecasting time series in frequency domain. Directly forecasting in frequency domain enables leveraging more spectral information to enhance the accuracy of time series predictions. FITS (Xu et al., 2023) directly forecasts spectrum of complete series with spectrum of input series as input in the frequency domain, and achieves promising results through lightweight linear layers, which has demonstrated the effectiveness and potential of this approach.

However, there are still some challenges associated with direct spectrum prediction in the frequency domain. One of these challenges is the ***potential misalignment of frequencies between the input spectrum and the complete spectrum (*** Q_2 ***)***, resulting from the use of the Discrete Fourier Transform (DFT). This misalignment makes it challenging to accurately represent the information of specific frequencies in the complete spectrum within the input spectrum, leading to prediction inaccuracies. Another challenge lies in ***how to effectively extract information about frequency combinations (*** Q_3 ***)***. Extracting spectral features is a complex task, as the harmonic series appeared in groups within the spectrum contain a significant amount of information. Determining how to effectively capture and utilize this information about frequency combinations remains an open question.

To address the challenges mentioned above, we propose ATFNet. It incorporates a time domain module and a frequency module to simultaneously handle local and global dependencies. Additionally, we introduce a novel weighting mechanism that dynamically allocates weights to the two modules. Our solutions for above challenges together with the contribution of this paper are listed as follows:

A_1: We propose the ***Dominant Harmonic Series Energy Weighting*** , which is able to generate appropriate weights for the time domain and frequency domain modules based on the level of periodicity exhibited by the input series. This allows us to leverage the advantages of both domains effectively when dealing with time series with distinct periodic patterns. We theoretically and empirically prove the effectiveness of out proposed method, which also provides with an effective way to measure the periodicity of a time series.

A_2: Drawing inspiration from signal processing techniques employed to enhance measurement accuracy (Gough, 1994; Santamaria et al., 2000; Luo et al., 2016), we introduce ***Extended DFT*** to align the discrete frequencies of the input spectrum with the complete spectrum, improving the accuracy of representing specific frequencies.

A_3: We introduce the attention mechanism into frequency domain and propose ***Complex-valued Spectrum Attention (CSA)***. This approach allows us to capture information from various frequency combinations, which provides with an **effective way of applying attention in addressing frequency domain representations**.

Extensive expirements on 8 real-world datasets demonstrates that our proposed ATFNet achieves more promising results compared with other state-of-the-art methods.

2. Related Work

Time domain forecasting methods. Early time series forecasting predominantly employed traditional statistical methods, such as ETS (Holt, 2004) and ARIMA (Box & Jenkins, 1968), which are are foundational yet often limited in capturing complex temporal dynamics. In recent years, the advent of deep learning has significantly advanced this field, particularly for long-term forecasting. RNN-based methods (Hochreiter & Schmidhuber, 1997; Chung et al., 2014; Lai et al., 2018) and CNN-based methods (Bai et al., 2018; Liu et al., 2022) have demonstrated enhanced capabilities in extracting intricate patterns from time series data. Furthermore, Transformer-based methods (Zhou et al., 2021; Kitaev et al., 2020; Zhang & Yan, 2022; Nie et al., 2022) can discover the temporal dependence between time points through the attention mechanism. Additionally, MLP-based methods (Oreshkin et al., 2019; Challu et al., 2022; Zeng et al., 2023) provide a straightforward yet effective approach to learning non-linear temporal dependencies of time series.

Frequency domain forecasting methods. The frequency domain offers a unique perspective on time series data by revealing the underlying periodicity and frequency components. Within this domain, a variety of forecasting methods have emerged recently, specifically designed to harness these characteristics and effectively capture the global dependencies inherent in time series data. Specifically, Autoformer (Wu et al., 2021) replaces the self-attention mechanism with an Auto-Correlation mechanism based on main periods detected using Fast Fourier Transform (FFT). TimesNet (Wu et al., 2022) transforms 1D time series into 2D space according to the main periods detected from the topk frequency spectrum amplitude by FFT. FilM (Zhou et al., 2022a) employs the Legendre Polynomials projection technique to approximate historical information, utilizes FFT for frequency feature extraction by Frequency Enhanced Layer. FreDo (Sun & Boning, 2022) proposes a simple yet strong model called AverageTile, transforming the tiled series to frequency domain with DFT to more accurately extract features from frequency domain.

Complex number processing methods. Extending time series analysis into the frequency domain necessitates tran-

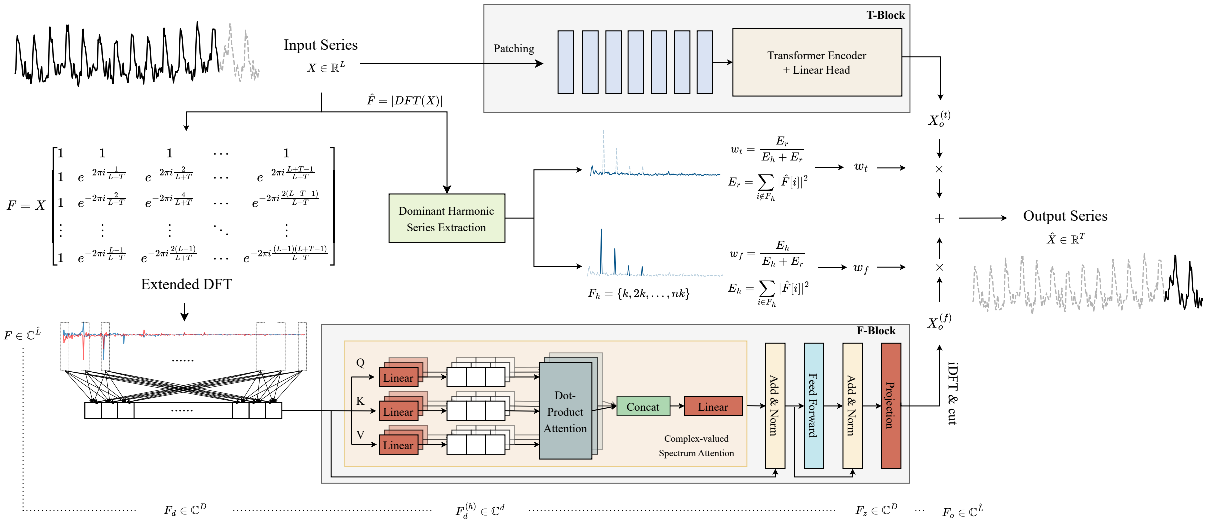

Input Series
$X \in \mathbb{R}^L$
T-Block
Patching
Transformer Encoder + Linear Head
$\hat{F} = |DFT(X)|$
$X_o^{(t)}$
$F = X \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-2\pi i \frac{1}{L+T}} & e^{-2\pi i \frac{2}{L+T}} & \cdots & e^{-2\pi i \frac{L+T-1}{L+T}} \\ 1 & e^{-2\pi i \frac{2}{L+T}} & e^{-2\pi i \frac{4}{L+T}} & \cdots & e^{-2\pi i \frac{2(L+T-1)}{L+T}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-2\pi i \frac{L-1}{L+T}} & e^{-2\pi i \frac{2(L-1)}{L+T}} & \cdots & e^{-2\pi i \frac{(L-1)(L+T-1)}{L+T}} \end{bmatrix}$
Dominant Harmonic Series Extraction
$w_t = \frac{E_r}{E_h + E_r}$
$E_r = \sum_{i \notin F_h} |\hat{F}[i]|^2$
w_t
×
+
Output Series
$\hat{X} \in \mathbb{R}^T$
$w_f = \frac{E_h}{E_h + E_r}$
$E_h = \sum_{i \in F_h} |\hat{F}[i]|^2$
w_f
$F_h = \{k, 2k, \ldots, nk\}$
Extended DFT
$F \in \mathbb{C}^{\hat{L}}$
......
......
$X_o^{(f)}$
F-Block
Q
K
V
Linear
Linear
Linear
Dot-Product Attention
Concat
Linear
Complex-valued Spectrum Attention
Add & Norm
Feed Forward
Add & Norm
Projection
iDFT & cut
$F_d \in \mathbb{C}^D$
$F_d^{(h)} \in \mathbb{C}^d$
$F_z \in \mathbb{C}^D$
$F_o \in \mathbb{C}^{\hat{L}}$

Figure 2. Model architecture of ATFNet. ATFNet is mainly composed of three sub-parts: 1) T-Block to capture local dependency from time domain; 2) F-Block to capture global dependency from frequency domain. The Extended DFT is used to generate frequency-aligned spectrum of input series. 3) The Dominant Harmonic Series Energy Weighting to allocate appropriate weights for F-Block and T-Block according to the periodic property of input series.

sitioning from the real number field to the complex number field. The method of processing complex number is critical, as simplistic treatment as real numbers could result in missed analytical insights. StemGNN (Cao et al., 2020) combines Graph Fourier Transform (GFT) for inter-series correlations and Discrete Fourier Transform (DFT) for temporal dependencies. However, its approach of transforming complex number into real numbers, and treating the real and imaginary parts separately, might not fully exploit the nuances of complex data interactions. In contrast, several methods have utilized complex-valued neural networks for more effective frequency domain transformations. FEDformer (Zhou et al., 2022b) introduces a Frequency Enhanced Block, available in two versions based on DFT and Discrete Wavelet Transform (DWT), which transforms the original time series into the frequency domain using learnable complex-valued weights to better capture frequency domain relationships. FITS (Xu et al., 2023) employs FFT to convert time series to frequency domain and uses a complex-valued linear layer structure, supplemented with a low-pass filter to attenuate high-frequency noise. FreTS (Yi et al., 2023) implements a specially designed real-valued MLP structure that mimics operations on complex number, managing both the real and imaginary parts of frequency coefficients simultaneously.

3. Preliminary

3.1. Problem Formulation

Long-term time series forecasting can be regarded as a sequence-to-sequence problem. Given a multi-variate time series $X = [x_1, x_2, \ldots, x_L] \in \mathbb{R}^{N \times L}$, where L is the look back window size and N is the number of variates, the target is to predict the future T steps $\hat{X} = [x_{L+1}, x_{L+2}, \ldots, x_{L+T}] \in \mathbb{R}^{N \times T}$.

3.2. Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a mathematical transformation that converts a time series into its frequency domain representation, revealing the frequency components present in the data. Specifically, for an input series with length L , the DFT spectrum is calculated as follows:

$$F[k] = \sum_{n=0}^{L-1} X[n] e^{-2\pi i \frac{kn}{L}}, k = 0, 1, \ldots, L-1. \quad (1)$$

Here, $F[k]$ represents the frequency-domain representation (the spectrum) at frequency index k, X represents the time-domain input sequence, L is the number of data points in the sequence, and i is the imaginary unit ($i^2 = -1$).

The Inverse Discrete Fourier Transform (iDFT) is the reverse operation of the DFT. It takes a frequency-domain signal, obtained through the DFT, and reconstructs the original time-domain signal:

$$X[n] = \frac{1}{L}\sum_{k=0}^{L-1} F[k]e^{2\pi i \frac{kn}{L}}, n = 0, 1, \ldots, L-1. \quad (2)$$

3.3. Energy in Time & Frequency Domain

The term '**Energy**' is commonly used in signal processing to refer to a concept that captures the overall strength or magnitude of a signal. In the context of a univariate time series represented by $x \in \mathbb{R}^L$, the energy of x is defined as the sum of the squared values of its individual data points, expressed as $E_t = \sum_{n=0}^{L-1} x[n]^2$ (Oppenheim & Verghese, 2015). Denoting the Discrete Fourier Transform (DFT) spectrum of x by $F \in \mathbb{C}^L$, we can define the energy of F in a similar manner as $E_f = \sum_{n=0}^{L-1} |F[n]|^2$.

The energy distribution within the spectrum of a time series provides valuable insights into the presence and strength of periodic patterns. When a significant amount of energy is concentrated at dominant frequencies, it indicates the presence of strong periodic behavior in the data. On the other hand, a more uniform energy distribution across the spectrum suggests a lack of pronounced periodicity.

4. Proposed Method

The structure of our proposed ATFNet is depicted in Figure 2. We employ a channel-independence scheme, which has been demonstrated to be effective by Nie et al. (2022), to prevent the mixture of spectra from different channels. Mixing such spectra could have a detrimental impact since channels may possess distinct global patterns.

For an input univariate series $X \in \mathbb{R}^L$, the T-Block directly processes it in the time domain, resulting in an output of $X_o^{(t)} \in \mathbb{R}^T$. We utilize the Extended Discrete Fourier Transform (DFT) to convert X into the frequency domain, generating an extended spectrum denoted as $F \in \mathbb{C}^{L+T}$. After transforming the spectrum back to the time domain using the inverse DFT (iDFT), the F-Block produces the output $X_o^{(f)} \in \mathbb{R}^T$. Finally, the output of the T-Block, $X_o^{(t)}$, and the output of the F-Block, $X_o^{(f)}$, are combined with weights to generate the final output $\hat{X} \in \mathbb{R}^T$. These weights are derived from the Dominant Harmonic Series Energy Weighting. The entire structure is formalized as:

$$\begin{aligned}
F &= \textbf{Extended DFT}(X), \\
X_o^f &= \textbf{F-Block}(F), \\
X_o^t &= \textbf{T-Block}(X), \\
w_t, w_f &= \textbf{Weighting}(X), \\
\hat{X} &= w_t X_o^{(t)} + w_f X_o^{(f)}.
\end{aligned} \quad (3)$$

In the following subsections, we will provide detailed explanations of the different sub-parts comprising ATFNet.

4.1. Extended DFT

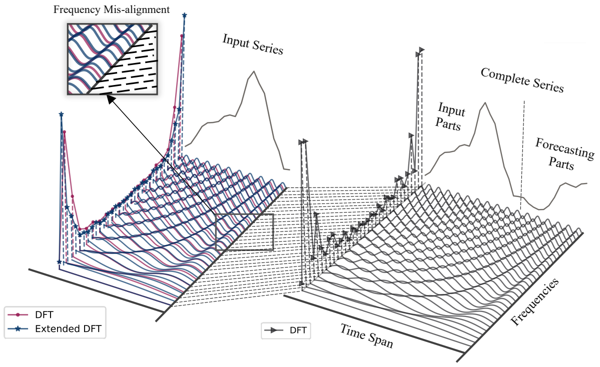


Figure 3. By employing Extended DFT, it becomes possible to obtain the spectrum of the input series with a discrete frequency group that aligns with the DFT spectrum of the complete series.

As illustrated in Section 1, using the traditional DFT may introduce a misalignment of frequencies between the spectra of input series and the entire series (as depicted in Figure 3). Consequently, forecasting models built on the input series may lack access to complete and accurate frequency information, leading to less accurate predictions for the complete series.

To address this issue, we propose Extended DFT, which overcomes the limitation imposed by the input length. This allows us to obtain an input spectrum that aligns with the DFT frequency group of the complete series. Specifically, based on Eq. 1, we replace the original complex exponential basis with the DFT basis of complete series:

$$F[k] = \sum_{n=0}^{L-1} x[n]e^{-2\pi i \frac{kn}{L+T}}, k = 0, 1, \ldots, L+T-1. \quad (4)$$

By using Eq. 4, we obtain a spectrum of length $L + T$ that aligns with the DFT spectrum of the complete series. For real-valued time series, the conjugate symmetry of the output spectrum is an important property of the DFT. Exploiting this property, we can reduce the computational cost by considering only the first half of the output spectrum, as the second half provides redundant information. Theorem 1 demonstrates that Extended DFT maintains this property.

Theorem 1. *Let $F \in \mathbb{C}^{L+T}$ represent the output spectrum of the input series $X \in \mathbb{R}^L$ obtained through the Extended DFT, where L denotes the input length and T denotes the prediction length. We have the following statement holds*

$$\begin{aligned}
\mathrm{Re}(F[k]) &= \mathrm{Re}(F[L+T-k]), \\
\mathrm{Im}(F[k]) &= -\mathrm{Im}(F[L+T-k]),
\end{aligned} \quad (5)$$

which implies that F exhibits conjugate symmetry. Here, $k = [1, 2, \dots, L+T-1]$, $\mathrm{Re}(\cdot)$ represents the real part, and $\mathrm{Im}(\cdot)$ represents the imaginary part.

See Appendix A.1 for a detailed proof.

4.2. F-Block

The architecture of F-Block is based on the Transformer Encoder, with all parameters to be complex-valued. Notably, all calculations within the F-Block are performed in the complex number field. The F-Block takes a univariate spectrum $F \in \mathbb{C}^{\hat{L}}$ as input, which is generated by the Extended DFT. Here, $\hat{L} = \lfloor (L+T)/2 \rfloor + 1$ represents the length of the spectrum.

Firstly, we utilize RevIN (Kim et al., 2021) to process the input spectrum F. While RevIN was originally developed to address distribution shift in the time domain, we have found it to be effective in handling frequency domain spectra as well. This approach enables us to transform the spectra of series with distinct global features into a similar distribution. To accomplish this, we normalize F by subtracting its mean and dividing it by its standard deviation. The standard deviation is calculated using $|F| \in \mathbb{R}^{\hat{L}}$. Afterwards, we add them back at the end of the structure. Next, we embed $F \in \mathbb{C}^{\hat{L}}$ into $F_d \in \mathbb{C}^D$ using a trainable linear projection $W_{emb} \in \mathbb{C}^{D \times \hat{L}}$. Since there are few chronological dependencies within the frequency domain spectrum, we disable the use of positional embeddings.

Complex-valued Spectrum Attention. We applied a modified multi-head attention mechanism rather than the original one in Vaswani et al. (2017). For each head $h = 1, 2, \dots, H$, it projects the embedded spectrum $F_d \in \mathbb{C}^D$ to dimension d across the spectrum dimension with trainable projections. Specifically, $Q_h = F_d^T W_h^Q$, $K_h = F_d^T W_h^K$, $V_h = F_d^T W_h^V$, where $W_h^Q, W_h^K, W_h^V \in \mathbb{C}^{D \times d}$ are learnable parameters. On each head we perform complex-valued Dot-Product Attention:

$$\textbf{Attention}(Q_h, K_h, V_h) = \textbf{Softmax}(|Q_h K_h^T|)V_h. \tag{6}$$

Then the final output of Complex-valued Spectrum Attention is calculated as follows:

$$\begin{aligned} \mathrm{head}_h &= \textbf{Attention}(Q_h, K_h, V_h), \\ \mathrm{CSA}(F_d) &= \textbf{Concat}(\mathrm{head}_1, \dots, \mathrm{head}_H)^T W_O, \end{aligned} \tag{7}$$

where $W_O \in \mathbb{C}^{hd \times D}$ are learnable parameters.

We follow the design of LayerNorm and FeedForward layers with residual connections in Transformer, with expanding to complex number field. After M layers of Encoder, the output F_z is linearly projected to $F_o \in \mathbb{C}^{\hat{L}}$. We convert F_o to time domain using iDFT, taking the last T points (the forecasting part) as the final output of F-Block $X_o^{(f)} \in \mathbb{R}^T$. The F-Block applies a fully Complex-Valued Neural Network (CVNN) architecture, we further discuss the effectiveness of CVNN in Appendix B.

4.3. T-Block

The T-Block is responsible for capturing local dependency in time series, which can be easier to be handled in time domain. Patching is an intuitive and effective way to capture local dependency in time series. Following (Nie et al., 2022), We employ patching together with the Transformer Encoder architecture as T-Block. It first devides the input series $X \in \mathbb{R}^L$ into patches $X_p \in \mathbb{R}^{P \times N}$, where P is the length of patches and N is the number of patches. The time series patches X_p are then embed and fed into the Transformer Encoder. With a linear projection, it generates the final output $X_o^{(t)} \in \mathbb{R}^T$. We also employ RevIN in T-Block to deal with distribution shift problem.

4.4. Dominant Harmonic Series Energy Weighting

As depicted in Figure 4, periodic time series consistently exhibits the existence of at least one harmonic group in their frequency domain spectrum, wherein the dominant harmonic group exhibits the highest concentration of spectrum energy. Conversely, this characteristic is seldom witnessed in the spectrum of non-periodic time series, where the distribution of energy is more uniform. The following Theorem 2 shows that the extent of concentration of energy within the dominant harmonic series in the frequency spectrum can reflect the periodicity of a time series.

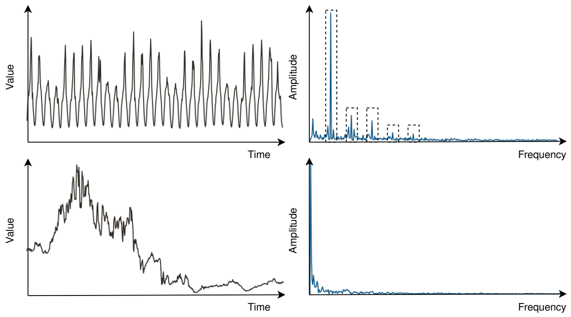


Figure 4. **Top:** Periodic time series, sampled from Traffic dataset. It can be observed that there exists an obvious harmonic group. **Bottom:** Non-periodic time series, sampled from Weather dataset. There are no obvious harmonic group in non-periodic series.

Theorem 2. *Let $f(x)$ denote a continuous time series defined on the interval $[0, L]$. We denote the discrete form of length-L realizations of $f(x)$, sampled at an interval of 1, as $X = [f(0), f(1), \dots, f(L-1)]$. $f(x)$ can be decomposed into two parts: $f(x) = f_p(x) + f_r(x)$, where $f_p(x)$ is a strictly periodic function with a period of τ (assuming $L = k\tau$ and $L, k, \tau \in \mathbb{N}^*$), i.e. $f_p(x-\tau) = f_p(x), \forall x \in$*

$[\tau, L]$. Let $E_p = \sum_{x=0}^{L-1} f_p^2(x)$ denotes the energy of periodic part and $E_r = \sum_{x=0}^{L-1} f_r^2(x)$ denotes the energy of the remaining part. Let $\lambda = E_p/E_r$, where λ represents the magnitude of the periodic signal compared to the remaining signal, and varies with different dividing methods. Considering the frequency resolution limitations imposed by the sampling interval, we assume that the energy in the frequency component above the Nyquist frequency (π) is small enough to be neglected. For simplicity, we additionally assume $\sum_{x=0}^{L-1} f(x) = 0$. Under these assumptions, we have

$$\frac{E_h}{E_f} = \frac{\sum_{n=0}^{T-1} |F[nk]|^2}{\sum_{n=0}^{L-1} |F[n]|^2} \geq \frac{\lambda - 2\sqrt{\lambda}}{\lambda - 2\sqrt{\lambda} + 1}, \tag{8}$$

where E_h represents the total energy of the dominant harmonic group with a basis frequency $k = L/\tau$, E_f represents the total energy of the whole spectrum, and $F(\cdot)$ represents the DFT components at a specific frequency.

See Appendix A.2 for a detailed proof. The ratio E_h/E_f in Eq. 8 serves as a metric for quantifying the concentration of energy within the dominant harmonic series in the spectrum of a time series. Theorem 2 establishes a lower bound for this ratio, which increases monotonically with λ for $\lambda > 1$. Intuitively, when a time series exhibits a more prominent periodic pattern, it can be decomposed into components that correspond to a larger value of λ. Consequently, our theorem implies that such a time series possesses a higher concentration of energy within dominant harmonic series.

Based on this property, we aim to employ the energy proportion of the dominant harmonic series as a metric for quantifying the extent to which a time series exhibits periodicity. To identify the dominant harmonic series, the crucial task is to determine the fundamental frequency. In addressing this challenge, we can draw insights from audio processing methodologies. Several methods can be employed for this purpose: 1) Naive method which identifies the frequency with the highest amplitude value as the fundamental frequency. 2) Rule-based pitch detection algorithms such as YIN(De Cheveigné & Kawahara, 2002) and SWIPE(Camacho & Harris, 2008). 3) Data-driven pitch detection algorithms such as CREPE(Kim et al., 2018) and SPICE(Gfeller et al., 2020).

Specifically, we begin by identifying the fundamental frequency through one of the aforementioned methods. Next, We consider this component along with its n harmonics, namely $F[2k], \ldots, F[nk]$, and calculate the total energy E_h. Subsequently, we determine the weights for the F-Block by computing the ratio E_h/E_f, where E_f represents the total energy of the spectrum. The detailed procedure is presented in Algorithm 1.

Algorithm 1 Dominant Harmonic Series Energy Weighting

```
Input: Time series $x$, length $L$, number of harmonics $N$
Output: The weight $w_f$, $w_t$ of the F-Block and T-Block
$F = \text{DFT}(x)$
$k = \text{Pitch Detection}(F)$
$E_f = \sum_{i=0}^{\lfloor L/2 \rfloor + 1} |F[i]|^2$
$E_h = \sum_{n=1}^{N} |F[nk]|^2$
$w_f = E_h/E_f$
$w_t = 1 - w_f$
```

Through our extensive attempts, we have observed that the naive method demonstrates commendable accuracy across a majority of real-world time series datasets, all while maintaining a low computational cost. In contrast, alternative approaches are hindered by the issue of computational complexity. Additionally, data-driven methods necessitate labeled pitch data, which is often challenging to obtain, posing a significant obstacle to their practical utilization. In the experiments part, we adopt the naive method as default for detecting fundamental frequency in ATFNet . Further exploration of employing various pitch detection algorithms is discussed in Appendix C.

5. Experiments

We evaluate the performance of ATFNet architecture through extensive experiments on 8 real-world datasets, including energy, weather, and traffic. We choose some state-of-the-art long-term time series forecasting models as baselines, including Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022b), iTransformer (Liu et al., 2023), Crossformer (Zhang & Yan, 2022), PatchTST (Nie et al., 2022), TimesNet (Wu et al., 2022), SCINet (Liu et al., 2022), DLinear (Zeng et al., 2023), FiLM (Zhou et al., 2022a), FreTS (Yi et al., 2023), FITS (Xu et al., 2023), and StemGNN (Cao et al., 2020). The codes are uploaded to https://github.com/YHYHYHYHYHY/ATFNet. to achieve reproducibility.

For all the models, we test the performance under 4 prediction length $T \in \{96, 192, 336, 720\}$. As for the look-back window size L, we run three distinct sizes $L \in \{96, 192, 336\}$ and always choose the best results to deal with the problem that not all models are suitable for the same look-back window size.

5.1. Long-term Time Series Forecasting

Comprehensive forecasting results for multi-variate and univariate forecasting task are listed in Table 1 and Table 2 with the best in red and the second underlined. The lower MSE/MAE indicates the more accurate prediction result. We adopt the naive method to identify the fundamental frequency for ATFNet.

For the multi-variate long-term forecasting task, ATFNet

Table 1. Multi-variate long-term time series forecasting results on 8 datasets. The best results are in bold and the second best results are underlined. We only display the average results of all prediction lengths $T \in \{96, 192, 336, 720\}$ here, see Table 9 for full results.

Models	ATFNet		Autoformer		FEDformer		iTransformer		Crossformer		PatchTST		TimesNet		SCINet		DLinear		FiLM		FreTS		FITS		StemGNN	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.500**	**0.508**	0.672	0.608	0.649	0.601	0.538	0.528	0.529	0.533	0.528	0.519	0.616	0.577	0.837	0.674	0.533	0.532	0.596	0.549	0.550	0.533	1.641	0.905	0.703	0.639
ETTh2	**0.229**	**0.335**	0.265	0.361	0.262	0.359	0.249	0.344	0.248	0.355	0.241	0.336	0.277	0.364	0.321	0.403	0.238	0.344	0.267	0.352	0.236	0.341	0.413	0.452	0.393	0.456
ETTm1	**0.388**	**0.425**	0.809	0.633	0.539	0.540	0.423	0.442	0.405	0.440	0.396	0.429	0.457	0.473	0.475	0.473	0.413	0.427	0.457	0.449	0.417	0.437	0.617	0.549	0.644	0.599
ETTm2	0.162	**0.270**	0.185	0.296	0.175	0.283	0.171	0.279	0.171	0.288	0.165	0.273	0.176	0.283	0.180	0.290	**0.161**	0.272	0.173	0.278	0.167	0.279	0.197	0.305	0.239	0.357
Traffic	**0.403**	**0.279**	0.630	0.387	0.608	0.374	0.420	0.296	0.948	0.563	0.423	0.295	0.816	0.497	0.533	0.387	0.456	0.320	0.491	0.345	0.466	0.318	0.870	0.566	0.847	0.478
Electricity	**0.161**	**0.257**	0.215	0.327	0.222	0.335	0.163	0.259	0.188	0.285	0.170	0.270	0.288	0.375	0.184	0.289	0.173	0.275	0.185	0.284	0.173	0.273	0.313	0.409	0.230	0.334
Weather	**0.228**	0.268	0.328	0.371	0.321	0.370	0.238	0.272	0.230	0.288	0.229	**0.265**	0.265	0.296	0.267	0.299	0.247	0.300	0.270	0.301	0.248	0.298	0.318	0.340	0.320	0.390
Solar	0.198	0.261	0.747	0.649	0.285	0.392	0.219	0.272	**0.193**	**0.249**	0.205	0.272	0.325	0.362	0.264	0.327	0.258	0.321	0.296	0.328	0.231	0.290	0.392	0.450	0.235	0.301

achieves the most promising results on both MSE and MAE compared to other baseline models. Through the full results in Table 9, it can be further observed that when the forecasting horizon is large (720), ATFNet achieves the best results. Additionally, even in the remaining dataset (ETTm2), ATFNet achieves the second-best result. These results highlight the effectiveness of ATFNet in handling long-term time series forecasting and capturing the complex relationships and dependencies that arise over extended time periods.

As for the addressing of datasets exhibiting distinct periodic patterns, we select 3 representive datasets for clarification. **1) Traffic dataset**: This dataset exhibits periodic characteristics. As shown in Table 9, ATFNet consistently achieves the best results in terms of MSE and MAE across all settings, including four distinct forecasting horizons. Notably, ATFNet exhibits a relative improvement of 4.0% in average MSE compared to the second best (also the current SOTA) iTransformer. The results in Table 3 also indicate that the F-Block, which leverages our proposed CSA to capture global dependencies in periodic time series, contributes significantly to the performance improvements. **2) Weather dataset**: This dataset represents non-periodic time series. We acknowledge that ATFNet does not achieve significantly promising results in the Weather dataset, as our proposed model does not include specific modifications tailored for handling non-periodic time series with few global dependencies. However, the design of including Transformer with patching (the idea borrowed from PatchTST) contributes a lot to remain good performance by effectively capturing important local patterns and relationships within the time series. **3) ETTh1 dataset**: This dataset represents a mixture of both periodic and non-periodic time series. In this case, ATFNet achieves relatively more advanced results, demonstrating a relative improvement of 5.3% in average MSE compared to the second-best model, PatchTST. Table 3 also reveals that both the T-Block and F-Block individually do not achieve such results, indicating that our proposed ensemble method, the Dominant Harmonic Group Energy Weighting, effectively utilizes the advantages of both modules and contributes to a significant improvement when addressing datasets with a mixture of periodic and non-periodic time series.

We also experiment with univariate datasets where the results are provided in Table 2, which also demonstrates state-of-the-art performance.

5.2. Dominant Harmonic Series Energy Weighting

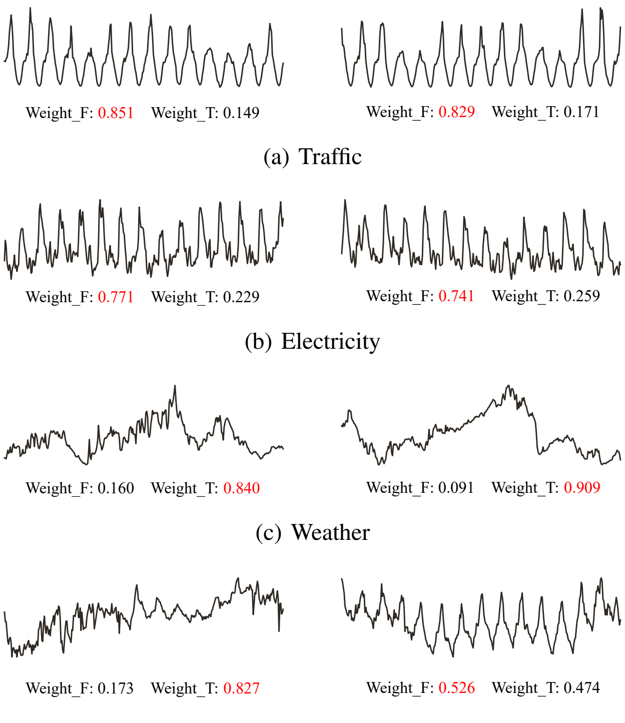
The purpose of Dominant Harmonic Series Energy Weighting is to adjust weights allocating to F-Block and T-Block according to the periodic property of a certain time series. As depicted in Figure 5, we take 8 time series from 4 different datasets to test whether our proposed Dominant Harmonic Series Energy Weighting can generate appropriate weights for distinct time series. We can observe that for time series with good periodic pattern, such as the time series in Traffic and Electricity dataset, are allocated more weights for F-Block. While for other time series without obvious periodic pattern, such as the time series in Weather dataset and the first time series in ETTh1 dataset, T-Block is allocated more weights.

5.3. Ablation Study

In this part, we conduct several experiments to analyze the contribution of different sub-modules of ATFNet, and the results are listed in Table 3. We designed 3 distinct versions of ATFNet. 1) ATFNet v1 uses traditional DFT instead of Extended DFT . 2) ATFNet v2 use the simple average instead of Dominant Harmonic Series Energy Weighting. 3) the complex-valued network of ATFNet is implemented with 2 independent real-valued channels in ATFNet v3. We also include single T-Block and F-Block to test how ATFNet

Table 2. Uni-variate long-term time series forecasting results on ETT datasets. The ETT datasets have a target feature 'Oil Temperature', and we take it as the uni-variate time series for forecasting. The best results are in bold and the second best results are underlined. We only display the average results of all prediction lengths $T \in \{96, 192, 336, 720\}$ here, see Table 10 in Appendix for full results.

Models	ATFNet		Autoformer		FEDformer		PatchTST		TimesNet		SCINet		DLinear		FiLM		FreTS		FITS		TFDNet	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.118	0.270	0.124	0.279	0.129	0.284	0.134	0.285	0.126	0.277	0.158	0.313	0.115	0.269	0.124	0.279	**0.114**	**0.265**	0.212	0.361	0.138	0.293
ETTh2	**0.293**	**0.428**	0.296	0.436	0.315	0.445	0.332	0.452	0.317	0.447	0.429	0.519	0.302	0.440	0.305	0.438	0.346	0.460	0.690	0.650	0.386	0.491
ETTm1	**0.088**	**0.217**	0.112	0.258	0.105	0.244	0.089	0.221	0.090	0.222	0.094	0.230	0.088	0.221	0.098	0.234	0.091	0.226	0.120	0.264	0.098	0.232
ETTm2	**0.203**	**0.331**	0.298	0.417	0.243	0.369	0.214	0.338	0.229	0.354	0.236	0.363	0.221	0.350	0.259	0.377	0.225	0.348	0.357	0.458	0.252	0.368



(a) Traffic

(b) Electricity

(c) Weather

(d) ETTh1

Figure 5. Examples of weights allocation for distinct time series from 4 different datasets.

utilize the advantages of both to enhancing forecasting performance.

Extended DFT. From the results we can observe that when employing Extended DFT, ATFNet can perform better than ATFNet v1 with conventional DFT. This indicates that with the help of Extended DFT, the model can better capture the spectral similarity between input and complete sequence.

Dominant Harmonic Series Energy Weighting. We can observe that ATFNet achieves better performance than both T-Block and F-Block, which indicates that our proposed ATFNet succeeds in combining 2 distinct models efficiently. Besides, ATFNet outperforms ATFNet v2, which utilizes a simple average stategy instead of Dominant Harmonic Series Energy Weighting. This proves that our proposed weighting method can allocate appropriate weights for time series with distinct periodic property and make full use of the advantages of both T-Block and F-Block.

Table 3. Ablation study results. The best results are in bold and the second best results are underlined. We only display the average results of all prediction lengths $T \in \{96, 192, 336, 720\}$ here, see Table 11 in Appendix for full results.

Models	ATFNet		F-Block		T-Block		ATFNet v1		ATFNet v2		ATFNet v3	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.500**	**0.508**	0.526	0.521	0.528	0.519	0.506	0.509	0.511	0.511	0.510	0.512
ETTm1	**0.388**	**0.425**	0.408	0.433	0.396	0.429	0.393	0.428	0.393	0.427	0.392	0.427
Traffic	**0.403**	**0.279**	0.406	0.283	0.423	0.295	0.405	0.279	0.404	0.282	0.410	0.282
Weather	**0.228**	0.268	0.237	0.283	0.229	**0.265**	0.233	0.274	0.235	0.284	0.229	0.274

Complex-valued Network. Comparing the performance of ATFNet and ATFNet v3, We find that using a real-valued network to approximate a complex-valued network leads to performance losses. Previous work (Cao et al., 2020) overlooked this aspect, resulting in sub-optimal results.

5.4. Varying Look-back Window Size.

Previous works (Zeng et al., 2023) have noticed that longer input length cannot always contribute to better forecasting performance, especially for most of Transformer-based methods, since time points in too early stage may be less related to current and future temporal patterns, which can distract the attention and lead the models to more likely to overfit the training data. Although our proposed F-Block is also based on attention mechanism, it can avoid such problem since the dependencies in frequency domain get less influence by longer historical input than time domain.

From the results shown in Figure 6, it can be observed that the performance of our proposed ATFNet consistently improves as the look-back window size increases. This

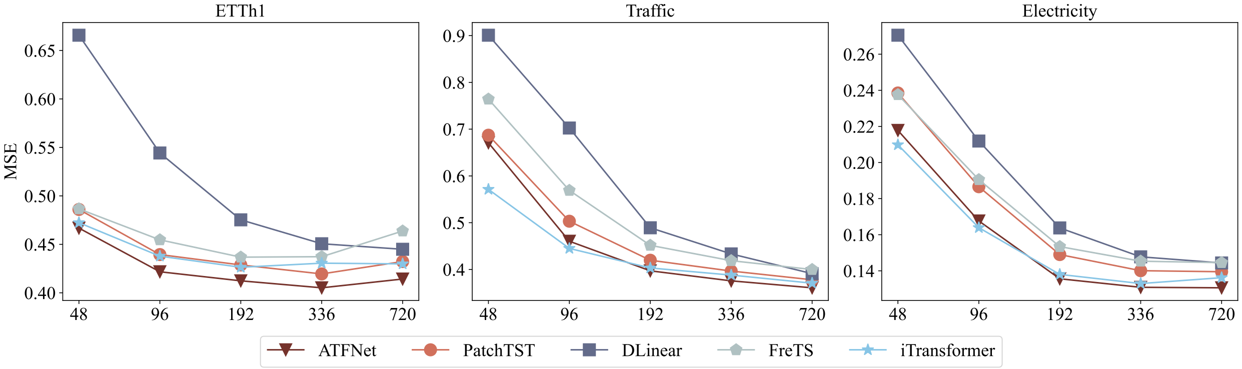


Figure 6. Multi-variate long-term time series forecasting results (MSE) with 5 different look-back window size $L \in \{48, 96, 192, 336, 720\}$. We evaluate the performance of ATFNet with 4 baseline models on ETTh1, Traffic and Electricity datasets. The prediction length is $T = 96$. We also test on other prediction length $T \in \{192, 336, 720\}$. See Figure 10 in Appendix for full results.

finding suggests that our methods are capable of capturing more information from longer input sequences, which helps in avoiding overfitting and leads to enhanced performance.

6. Conclusion

In this study, we introduce ATFNet, a novel approach that innovatively integrates a frequency domain module with a time domain module for time series forecasting. This dual-module ensemble is designed to effectively capture both global and local dependencies inherent in time series data. A standout feature of ATFNet is the Dominant Harmonic Series Energy Weighting mechanism, which intelligently adapts the contribution of each module based on the specific periodicity patterns of the time series. It also provides with a valuable way to measure the extent to which a time series exhibit periodicity, which is verified to be effective both theoretically and empirically. For the frequency module, we propose Extended DFT and Complex-valued Spectrum Attention, which play a crucial role in extracting essential frequency domain insights and enhancing the application of Transformer in the frequency domain. Extensive experiments conducted on various real-world datasets demonstrate that ATFNet outperforms existing state-of-the-art time series forecasting methods. This advancement not only illustrates the efficacy of our model but also lays the groundwork for future explorations into the combined use of time and frequency domain analyses for enhanced forecasting accuracy.

References

Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Bassey, J., Qian, L., and Li, X. A survey of complex-valued neural networks. *arXiv preprint arXiv:2101.12249*, 2021.

Box, G. E. and Jenkins, G. M. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

Camacho, A. and Harris, J. G. A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America*, 124(3):1638–1652, 2008.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., and Dubrawski, A. N-hits: Neural hierarchical interpolation for time series forecasting, 2022.

Chiheb, T., Bilaniuk, O., Serdyuk, D., et al. Deep complex networks. In *International Conference on Learning Representations*, 2017.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

De Cheveigné, A. and Kawahara, H. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

Gfeller, B., Frank, C., Roblek, D., Sharifi, M., Tagliasacchi, M., and Velimirović, M. Spice: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1118–1128, 2020.

Gough, P. T. A fast spectral estimation algorithm based on the fft. *IEEE transactions on signal processing*, 42(6): 1317–1322, 1994.

Hirose, A. and Yoshida, S. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and learning systems*, 23(4):541–551, 2012.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Holt, C. C. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

Kim, J. W., Salamon, J., Li, P., and Bello, J. P. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161–165. IEEE, 2018.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., and Xu, Q. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

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

Luo, J., Xie, Z., and Xie, M. Interpolated dft algorithms with zero padding for classic windows. *Mechanical Systems and Signal Processing*, 70:1011–1025, 2016.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

Oppenheim and Verghese. *Signals, Systems, and Inference*. Pearson, 2015. ISBN 9780133944211.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Parseval and Mémoire, M.-A. Sur les séries et sur l'intégration complète d'une équation aux différences partielles linéaire du second ordre, à coefficients constants. *Sciences, mathématiques et physiques*, pp. 638–648, 1806.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Santamaria, I., Pantaleon, C., and Ibanez, J. A comparative study of high-accuracy frequency estimation methods. *Mechanical Systems and Signal Processing*, 14(5):819–834, 2000.

Sun, F.-K. and Boning, D. S. Fredo: Frequency domain-based long-term time series forecasting. *arXiv e-prints*, pp. arXiv–2205, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2022.

Xu, Z., Zeng, A., and Xu, Q. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

Xu, Z., Zeng, A., and Xu, Q. Fits: Modeling time series with $10k$ parameters, 2024.

Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., and Niu, Z. Frequency-domain mlps are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Zhou, T., Ma, Z., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R., et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems*, 35:12677–12690, 2022a.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022b.

A. Proofs

A.1. Theorem 1

Theorem 1. *Let $F \in \mathbb{C}^{L+T}$ represent the output spectrum of the input series $X \in \mathbb{R}^L$ obtained through the Extended DFT, where L denotes the input length and T denotes the prediction length. We have the following statement holds*

$$\begin{aligned} \mathrm{Re}(F[k]) &= \mathrm{Re}(F[L+T-k]), \\ \mathrm{Im}(F[k]) &= -\,\mathrm{Im}(F[L+T-k]), \end{aligned} \tag{9}$$

which implies that F exhibits conjugate symmetry. Here, $k = [1, 2, \ldots, L+T-1]$, $\mathrm{Re}(\cdot)$ represents the real part, and $\mathrm{Im}(\cdot)$ represents the imaginary part.

Proof. Following Eq. 4, we have

$$F[k] = \sum_{n=0}^{L-1} x[n] e^{-2\pi i \frac{kn}{L+T}}, \tag{10}$$

$$\begin{aligned} F[L+T-k] &= \sum_{n=0}^{L-1} x[n] e^{-2\pi i \frac{(L+T-k)n}{L+T}} \\ &= \sum_{n=0}^{L-1} x[n] e^{-2\pi i n} \cdot e^{2\pi i \frac{kn}{L+T}} \\ &= \sum_{n=0}^{L-1} x[n] e^{2\pi i \frac{kn}{L+T}}. \end{aligned} \tag{11}$$

Applying Euler's Formula, we obtain

$$\mathrm{Re}(F[k]) = \mathrm{Re}(F[L+T-k]) = \sum_{n=0}^{L-1} x[n] \cos(2\pi \frac{kn}{L+T}), \tag{12}$$

$$\mathrm{Im}(F[k]) = -\,\mathrm{Im}(F[L+T-k]) = -\sum_{n=0}^{L-1} x[n] \sin(2\pi \frac{kn}{L+T}). \tag{13}$$

Thus, we complete the proof of Theorem 1. □

A.2. Theorem 2

Theorem 2. *Let $f(x)$ denote a continuous time series defined on the interval $[0, L]$. We denote the discrete form of length-L realizations of $f(x)$, sampled at an interval of 1, as $X = [f(0), f(1), \ldots, f(L-1)]$. $f(x)$ can be decomposed into two parts: $f(x) = f_p(x) + f_r(x)$, where $f_p(x)$ is a strictly periodic function with a period of τ (assuming $L = k\tau$ and $L, k, \tau \in \mathbb{N}^*$), i.e. $f_p(x-\tau) = f_p(x), \forall x \in [\tau, L]$. Let $E_p = \sum_{x=0}^{L-1} f_p^2(x)$ denotes the energy of periodic part and $E_r = \sum_{x=0}^{L-1} f_r^2(x)$ denotes the energy of the remaining part. Let $\lambda = E_p/E_r$, where λ represents the magnitude of the periodic signal compared to the remaining signal, and varies with different dividing methods. Considering the frequency resolution limitations imposed by the sampling interval, we assume that the energy in the frequency component above the Nyquist frequency (π) is small enough to be neglected. For simplicity, we additionally assume $\sum_{x=0}^{L-1} f(x) = 0$. Under these assumptions, we have*

$$\frac{E_h}{E_f} = \frac{\sum_{n=0}^{T-1} |F[nk]|^2}{\sum_{n=0}^{L-1} |F[n]|^2} \geq \frac{\lambda - 2\sqrt{\lambda}}{\lambda - 2\sqrt{\lambda} + 1}, \tag{14}$$

where E_h represents the total energy of the dominant harmonic group with a basis frequency $k = L/\tau$, E_f represents the total energy of the whole spectrum, and $F(\cdot)$ represents the DFT components at a specific frequency.

Proof. Since $f_p(x)$ is strictly periodic, it can be decomposed into its Fourier series. Based on the assumption that the energy in the frequency component above the Nyquist frequency (π) is small enough to be neglected, we decompose $f_p(x)$ into the first $q = \lceil \tau/2 \rceil - 1$ Fourier series as follows:

$$f_p(x) = \sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x}. \tag{15}$$

where the Fourier coefficients c_m are determined through the integral:

$$c_m = \frac{1}{\tau} \int_0^\tau f(x) e^{2\pi i \frac{m}{\tau} x} dx. \tag{16}$$

The energy of $f_p(x)$ is then computed using

$$\begin{aligned} E_p &= \sum_{x=0}^{L-1} \Big[\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} \Big]^2 \\ &= k \sum_{x=0}^{\tau-1} \Big[\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} \Big]^2 \\ &= k \sum_{x=0}^{\tau-1} \sum_{m=-q}^{q} \sum_{n=-q}^{q} c_m c_n e^{2\pi i \frac{m+n}{\tau} x} \\ &= k \sum_{m=-q}^{q} \sum_{n=-q}^{q} c_m c_n \sum_{x=0}^{\tau-1} e^{2\pi i \frac{m+n}{\tau} x} \end{aligned} \tag{17}$$

When $m + n \neq 0$, we have

$$\sum_{x=0}^{\tau-1} e^{2\pi i \frac{m+n}{\tau} x} = \frac{1 - e^{2\pi i (m+n)}}{1 - e^{2\pi i \frac{m+n}{\tau}}} = 0. \tag{18}$$

When $m + n = 0$, we have

$$\sum_{x=0}^{\tau-1} e^{2\pi i \frac{m+n}{\tau} x} = \tau. \tag{19}$$

Combining Eq.17, Eq.18 and Eq.19, we have

$$\begin{aligned} E_p &= k \sum_{x=0}^{\tau-1} \sum_{m=-q}^{q} \sum_{n=-q}^{q} c_m c_n e^{2\pi i \frac{m+n}{\tau} x} \\ &= k \sum_{m=-q}^{q} \tau |c_m|^2 \\ &= L \sum_{m=-q}^{q} |c_m|^2 \end{aligned} \tag{20}$$

Similarly, the energy of $f(x)$ is given by

$$\begin{aligned} E_t &= \sum_{x=0}^{L-1} \big[\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} + f_r(x) \big]^2 \\ &= \sum_{x=0}^{L-1} \big[\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} \big]^2 + \sum_{x=0}^{L-1} f_r^2(x) \\ &+ 2 \sum_{x=0}^{L-1} \sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x) \\ &= E_p + E_r + 2 \sum_{x=0}^{L-1} \sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x). \end{aligned} \tag{21}$$

For the Fourier component of basis frequency $k = L/\tau$, we have

$$\begin{aligned} F[k] &= \sum_{x=0}^{L-1} [\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} + f_r(x)] e^{-2\pi i \frac{k}{L} x} \\ &= c_1 L + \sum_{x=0}^{L-1} f_r(x) e^{-2\pi i \frac{1}{\tau} x}, \end{aligned} \tag{22}$$

where the last equality follows from performing simple geometric sequence summation. Similarly, the Fourier component of the remaining harmonics $nk(n = 2, \ldots, \tau - 1)$ is computed as:

$$F[nk] = c_n L + \sum_{x=0}^{L-1} f_r(x) e^{-2\pi i \frac{n}{\tau} x}. \tag{23}$$

Combining Eq. 20, Eq. 22, and Eq. 23, we achieve a lower bound of the total energy of this dominant harmonic series as follows

$$\begin{aligned} E_h = \sum_{n=0}^{\tau-1} |F[nk]|^2 &= \sum_{n=-q}^{q} |F[nk]|^2 \\ &= \sum_{n=-q}^{q} |c_n L + \sum_{x=0}^{L-1} f_r(x) e^{-2\pi i \frac{n}{\tau} x}|^2 \\ &= L^2 \sum_{n=-q}^{q} |c_n|^2 + 2L \sum_{x=0}^{L-1} \sum_{n=-q}^{q} c_n e^{2\pi i \frac{n}{\tau} x} f_r(x) + \sum_{n=-q}^{q} \Big| \sum_{x=0}^{L-1} f_r(x) e^{-2\pi i \frac{n}{\tau} x} \Big|^2 \\ &\geq L[E_p + 2 \sum_{x=0}^{L-1} \sum_{n=-q}^{q} c_n e^{2\pi i \frac{n}{\tau} x} f_r(x)], \end{aligned} \tag{24}$$

Applying Parseval's theorem (Parseval & Mémoire, 1806), we have

$$E_f = L E_t = L[E_p + E_r + 2 \sum_{x=0}^{L-1} \sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x)]. \tag{25}$$

Consequently, by Eq. 24 and Eq. 25, we can decuce that

$$\frac{E_h}{E_f} \geq \frac{E_p + 2 \sum_{x=0}^{L-1} \sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x)}{E_p + E_r + 2 \sum_{x=0}^{L-1} \sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x)}. \tag{26}$$

Applying the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned}
\big[\sum_{x=0}^{L-1}\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x)\big]^2 &\leq \sum_{x=0}^{L-1}\big[\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x}\big]^2 \sum_{x=0}^{L-1} f_r^2(x) \\
&= E_p E_r,
\end{aligned} \tag{27}$$

where the last equality follows from Eq. 15 and the definition of E_p and E_r. Therefore, we have

$$-\sqrt{E_p E_r} \leq \sum_{x=0}^{L-1}\sum_{m=-q}^{q} c_m e^{2\pi i \frac{m}{\tau} x} f_r(x) \leq \sqrt{E_p E_r}. \tag{28}$$

Combining Eq. 26 and Eq. 28, we obtain

$$\frac{E_h}{E_f} \geq \frac{E_p - 2\sqrt{E_p E_r}}{E_p + E_r - 2\sqrt{E_p E_r}} = \frac{\lambda - 2\sqrt{\lambda}}{\lambda - 2\sqrt{\lambda} + 1}. \tag{29}$$

Thus, we complete the proof of Theorem 2. □

B. Discussion on Complex-Valued Neural Network

Our model employs complex-valued neural network (CVNN) structures, with parameters defined on the complex plane. Several studies in frequency domain time series forecasting method mentioned in Section 2 have also naturally employed such structures, owing to the complex number format of spectrums derived from FFT. However, these studies provide little analysis for the effectiveness of CVNNs. It's important to note that while transforming complex number into real numbers for application in real-valued neural networks (RVNNs) is feasible, CVNNs offer superior capability in capturing complex representations within the frequency domain.

Our analysis begins with a complex linear layer that maps $w \in \mathbb{C}^{d_1}$ to $z = \boldsymbol{B}w \in \mathbb{C}^{d_2}$, where $\boldsymbol{B} \in \mathbb{C}^{d_2 \times d_1}$. The bias term can be integrated into $\boldsymbol{B}$ by appending a scaler 1 to w. A common RVNN approach, such as that used in Cao et al. (2020), treats the real and imaginary parts of w as distinct entities, $w_r \in \mathbb{R}^{d_1}$ and $w_c \in \mathbb{R}^{d_2}$. It employs two real weight matrices $\boldsymbol{B}_r, \boldsymbol{B}_c \in \mathbb{R}^{d_2 \times d_1}$, resulting in $z = \boldsymbol{B}_r w_r + i\boldsymbol{B}_c w_c$. This method relies solely on the real (imaginary) component of the input to produce the real (imaginary) component of the output. We use a scenario where $d_1 = 2$ with one dimension representing the bias and $d_2 = 1$ to show that the method is inadequate for representing even such simple transformations.

Consider a complex linear function $f(w) = \beta_0 + \beta_1 z$ with input $w \in \mathbb{C}$ and coefficients $\beta_0, \beta_1 \in \mathbb{C}$. Now we have observations $\{(w_j, z_j)\}_{j=1}^n$ with random noise, where $z_j = f(w_j) + \epsilon_j$. We can write the observations into the following matrix form:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \boldsymbol{z} = \boldsymbol{W}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \boldsymbol{\epsilon} = \begin{bmatrix} 1 & w_1 \\ 1 & w_2 \\ \vdots & \vdots \\ 1 & w_n \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}, \tag{30}$$

where $\boldsymbol{z}, \boldsymbol{\epsilon} \in \mathbb{C}^{n\times 1}, \boldsymbol{W} \in \mathbb{C}^{n \times 2}$. First, we can estimate the model directly using the complex method:

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0, \beta_1 \in \mathbb{C}}{\operatorname{argmin}} \left\| \boldsymbol{z} - \boldsymbol{W}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \right\|_2^2. \tag{31}$$

By a few steps of differentiation, we derive

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = (\boldsymbol{W}^* \boldsymbol{W})^{-1} \boldsymbol{W}^* \boldsymbol{z}, \tag{32}$$

where $\boldsymbol{W}^* = \bar{\boldsymbol{W}}^T$ represents conjugate transpose of $\boldsymbol{W}$. Then we have

$$\begin{aligned}
\underset{n\to\infty}{\operatorname{plim}} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} &= \underset{n\to\infty}{\operatorname{plim}} \left[(\boldsymbol{W}^* \boldsymbol{W})^{-1} \boldsymbol{W}^* (\boldsymbol{W}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \boldsymbol{\epsilon}) \right] \\
&= \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \boldsymbol{Q} \underset{n\to\infty}{\operatorname{plim}} \left[\boldsymbol{W}^* \boldsymbol{\epsilon} / n \right] = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix},
\end{aligned} \tag{33}$$

where $\boldsymbol{Q} = \mathrm{plim}_{n\to\infty}$ is the limit operator in probability. Here, $\boldsymbol{Q} = \mathrm{plim}_{n\to\infty} \boldsymbol{W}^*\boldsymbol{W}/n$ is assumed to be rank-2 matrix with finite elements, and we make another reasonable assumption $E(\boldsymbol{W}^*\boldsymbol{\epsilon}) = \mathbf{0}$ which ensures the last equality holds. Hence, the complex method is capable of obtaining coefficients that converge to the true parameters.

Conversely, consider the real method that handles the real and imaginary parts of $\boldsymbol{W}$ independently. We demonstrate that this method fails to ensure convergence to the true parameters. We write all the complex number in real format as follows

$$\begin{aligned} w_j &= u_j + iv_j \\ \epsilon_j &= \epsilon_j^r + i\epsilon_j^i, \\ \beta_0 &= \beta_0^r + i\beta_0^i, \quad \beta_1 = \beta_1^r + i\beta_1^i, \end{aligned} \tag{34}$$

where $u_j, v_j, \epsilon_j^r, \epsilon_j^i, \beta_0^r, \beta_0^i, \beta_1^i, \beta_1^i \in \mathbb{R}$. Applying these to the true model formulation, we have $x_j \in \mathbb{R}$ and $y_j \in \mathbb{R}$ take the forms

$$\begin{aligned} x_j &= \beta_0^r + \beta_1^r u_j - \beta_1^i v_j + \epsilon_j^r \\ y_j &= \beta_0^i + \beta_1^i u_j + \beta_1^r v_j + \epsilon_j^i, \end{aligned} \tag{35}$$

where $z_j = x_j + iy_j$. If we only use u_j to estimate x_j and only use v_j to estiamte y_j, as shown below

$$\begin{aligned} \hat{x_j} &= \hat{\gamma}_0 + \hat{\gamma}_1 u_j, \\ \hat{y_j} &= \hat{\delta}_0 + \hat{\delta}_1 v_j, \end{aligned} \tag{36}$$

this obviously leads to a loss of information. The estimates are obtained by

$$\begin{aligned} \hat{\gamma}_0, \hat{\gamma}_1 &= \underset{\gamma_0,\gamma_1 \in \mathbb{R}}{\mathrm{argmin}} \left\| \boldsymbol{x} - \boldsymbol{U} \left[\begin{array}{c} \gamma_0 \\ \gamma_1 \end{array} \right] \right\|_2^2, \\ \hat{\delta}_0, \hat{\delta}_1 &= \underset{\delta_0,\delta_1 \in \mathbb{R}}{\mathrm{argmin}} \left\| \boldsymbol{y} - \boldsymbol{V} \left[\begin{array}{c} \delta_0 \\ \delta_1 \end{array} \right] \right\|_2^2, \end{aligned} \tag{37}$$

where

$$\boldsymbol{x} = \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right] \in \mathbb{R}^{n\times 1}, \quad \boldsymbol{y} = \left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_n \end{array} \right] \in \mathbb{R}^{n\times 1}, \quad \boldsymbol{U} = \left[\begin{array}{cc} 1 & u_1 \\ 1 & u_2 \\ \vdots & \vdots \\ 1 & u_n \end{array} \right] \in \mathbb{R}^{n\times 2}, \quad \boldsymbol{V} = \left[\begin{array}{cc} 1 & v_1 \\ 1 & v_2 \\ \vdots & \vdots \\ 1 & v_n \end{array} \right] \in \mathbb{R}^{n\times 2}. \tag{38}$$

By simple derivations, we obtain

$$\begin{aligned} \left[\begin{array}{c} \hat{\gamma}_0 \\ \hat{\gamma}_1 \end{array} \right] &= \left[\begin{array}{c} \beta_0^r \\ \beta_1^r \end{array} \right] - (\boldsymbol{U}^T\boldsymbol{U})\boldsymbol{U}^T(\boldsymbol{V} \left[\begin{array}{c} 0 \\ \beta_1^i \end{array} \right] + \boldsymbol{\epsilon}^r), \\ \left[\begin{array}{c} \hat{\delta}_0 \\ \hat{\delta}_1 \end{array} \right] &= \left[\begin{array}{c} \beta_0^i \\ \beta_1^r \end{array} \right] + (\boldsymbol{V}^T\boldsymbol{V})\boldsymbol{V}^T(\boldsymbol{U} \left[\begin{array}{c} 0 \\ \beta_1^i \end{array} \right] + \boldsymbol{\epsilon}^i), \end{aligned} . \tag{39}$$

Applying the probability limit operator to Eq. 39 and assuming $\boldsymbol{Q}^r = \mathrm{plim}_{n\to\infty} \boldsymbol{U}^*\boldsymbol{U}/n, \boldsymbol{Q}^i = \mathrm{plim}_{n\to\infty} \boldsymbol{V}^*\boldsymbol{V}/n$ are rank-2 matrices with finite elements and $E(\boldsymbol{U}^T\boldsymbol{\epsilon}^r) = E(\boldsymbol{V}^T\boldsymbol{\epsilon}^i) = \mathbf{0}$, we have

$$\begin{aligned} \underset{n\to\infty}{\mathrm{plim}} \left[\begin{array}{c} \hat{\gamma}_0 \\ \hat{\gamma}_1 \end{array} \right] &= \left[\begin{array}{c} \beta_0^r \\ \beta_1^r \end{array} \right] - \boldsymbol{Q}^r \underset{n\to\infty}{\mathrm{plim}} \boldsymbol{U}^T\boldsymbol{V} \left[\begin{array}{c} 0 \\ \beta_1^i \end{array} \right] \Big/ n \neq \left[\begin{array}{c} \beta_0^r \\ \beta_1^r \end{array} \right], \\ \underset{n\to\infty}{\mathrm{plim}} \left[\begin{array}{c} \hat{\delta}_0 \\ \hat{\delta}_1 \end{array} \right] &= \left[\begin{array}{c} \beta_0^i \\ \beta_1^r \end{array} \right] + \boldsymbol{Q}^i \underset{n\to\infty}{\mathrm{plim}} \boldsymbol{V}^T\boldsymbol{U} \left[\begin{array}{c} 0 \\ \beta_1^i \end{array} \right] \Big/ n \neq \left[\begin{array}{c} \beta_0^i \\ \beta_1^r \end{array} \right]. \end{aligned} \tag{40}$$

Therefore, from Eq. 35, Eq. 36, and Eq. 40, we show that this method do not ensure convergence in probability. These derivations can be easily generalize to scenarios with arbitrary d_1 and d_2, thus reflecting the limitations of this RVNN method in representing complex transformations.

Another RVNN methodology concatenate the real and imaginary parts into a single real vector $\tilde{w} = (w_r^T, w_c^T)^T \in \mathbb{R}^{2d_1}$ and uses a single real weight matrix $\tilde{\boldsymbol{B}} \in \mathbb{R}^{2d_1 \times 2d_2}$ to produce an $2d_2$-dimensional vector $\tilde{z} = \tilde{\boldsymbol{B}}\tilde{w}$. The first d_2 elements of $\tilde{z}$ correspond to the real part of the output vector and the remaining correspond to imaginary part, i.e.,$z = \tilde{z}[: d_2] + i\tilde{z}[d_2 :]$. While this formulation can technically represent any complex linear layer W mentioned above, it increases the complexity of the network by doubling the number of parameters compared to the CVNN approach. This not only makes the learning process more challenging but also overlooks the intrinsic properties of complex number such as phase rotation and amplitude modulation. This aspect of RVNNs, along with the inherent advantages of CVNNs in handling complex number characteristics, is discussed in detail in Hirose & Yoshida (2012); Chiheb et al. (2017); Bassey et al. (2021).

C. Discussion on Fundamental Frequency Detection

Detecting the fundamental frequency constitutes a crucial aspect of Dominant Harmonic Group Energy Weighting. As discussed in Section 4.4, we can adopt either a straightforward approach that directly selects the frequency with the highest amplitude or refer to pitch detection algorithms documented in the audio processing literature to address this issue. It should be noted that the accuracy of fundamental frequency detection algorithms may vary depending on the characteristics of the time series, thereby posing a challenge in selecting an optimal choice. Through our investigation, we have discovered that the naive method surpasses other pitch detection algorithms in terms of both accuracy and computational efficiency, leading us to designate it as the default fundamental frequency detection algorithm.

Furthermore, we conducted experiments to assess the performance of distinct fundamental frequency detection algorithms, including YIN (De Cheveigné & Kawahara, 2002) and SWIPE (Camacho & Harris, 2008). Specifically, we employed a univariate time series exhibiting strong periodicity and divided it into several time frames of varying lengths. Our evaluation aimed to determine whether these algorithms accurately detected the fundamental frequency and to quantify their computational costs. We utilized the Traffic dataset, with the variable denoted as '8' and spanning indices 0-3000, exhibiting an evident period of 24. For each length, we sampled 2000 time frames using a sliding window with a stride of 1. The corresponding results are presented in Table 4.

Table 4. Results of distinct fundamental frequency detection algorithms.

Time Frame Length	96		192		336		720	
	Acc	Time(s)	Acc	Time(s)	Acc	Time(s)	Acc	Time(s)
Naive	100.0%	0.021	100.0%	0.020	100.0%	0.024	100.0%	0.031
YIN	88.8%	2.863	97.3%	10.909	99.2%	32.659	100.0%	148.357
SWIPE	97.2%	0.446	98.0%	1.043	100.0%	1.974	100.0%	4.935

The results clearly demonstrate that the naive method, which selects the frequency with the highest amplitude as the fundamental frequency, outperforms other algorithms in terms of both accuracy and computational efficiency.

D. Experiments Details

D.1. Datasets

We include 8 real-world datasets that are widely used for evaluation of long-term time series forecasting.

ETT: The ETT(Electricity Transformer Temperature) datasets are collected from two different electric transformers labeled with 1 and 2, recording loads and oil temperature of transformers. We employ 4 ETT sub-datasets: ETTh1, ETTh2, ETTm1, ETTm2 in different resolutions (*h* for 1-hour and *m* for 15-min).

Electricity: It records the hourly electricity consumption of 321 clients from 2012 to 2014.

Traffic: It records the road occupancy rates measured by different sensors on San Francisco Bay area freeways, provided by California Department of Transportation.

Weather: It collects 21 meteorological indicators, such as humidity and air temperature, from the Weather Station of the Max Planck Biogeochemistry Institute in Germany in 2020.

Solar: The Solar Energy dataset records the solar power production of 137 PV plants in 2006.

The detailed dataset descriptions are included in Table 5.

Table 5. Dataset descriptions. *Dim* stands for the number of variate in the dataset. *Time Points* represents the total number of time points. *Frequency* denotes the sampling interval of time points. *Information* denotes the domain that the dataset is from.

Dataset	Dim	Time Points	Frequency	Information
ETTh1	7	17420	1-hour	Electricity
ETTh2	7	17420	1-hour	Electricity
ETTm1	7	69680	15-min	Electricity
ETTm2	7	69680	15-min	Electricity
Electricity	321	26304	1-hour	Electricity
Traffic	862	17544	1-hour	Transportation
Weather	21	52696	10-min	Weather
Solar	137	52560	10-min	Energy

D.2. Baseline Models

Autoformer (Wu et al., 2021): Autoformer introduces time series decomposition into the Transformer architecture, and designs an Auto-Correlation mechanism to take the place of self-attention to improve computation efficiency and information utilization.

FEDformer (Zhou et al., 2022b): FEDformer designs a Frequency Enhanced Block with 2 versions based on DFT (Discrete Fourier Transform) and DWT (Discrete Wavelet Transform) respectively to capture features in frequency domain. It also utilizes mix-of-experts decomposition to deal with distribution shifting. In our experiments, we employs the Fourier version for FEDformer.

iTransformer (Liu et al., 2023): iTransformer reverses the roles of the attention mechanism and the feed-forward network by embedding the time points of each individual series into variate tokens, which are then utilized by the attention mechanism to capture multivariate correlations.

Crossformer (Zhang & Yan, 2022): Crossformer introduces the Dimension-Segment-Wise (DSW) embedding and Two-Stage Attention (TSA) techniques to enhance the capturing of cross-dimension dependencies in multi-variate time series forecasting.

PatchTST (Nie et al., 2022): PatchTST proposes an effective design of Transformer-based models for time series forecasting tasks by introducing two key components: patching and channel-independent structure. We take the default setting that Patch_len=16 and Stride=8.

TimesNet (Wu et al., 2022): TimesNet transforms 1D time series into 2D space according to the main periods detected from the topk frequency spectrum amplitude by FFT. It utilizes a parameter-efficient inception block to capture the intra-period and inter-period dependencies at the same time in the 2D space.

SCINet (Liu et al., 2022): SCINet employs a recursive downsample-convolve-interact architecture, where multiple convolutional filters are utilized in each layer to extract distinct and valuable temporal features from downsampled sub-sequences or features.

DLinear (Zeng et al., 2023): DLinear proposes a simple one layer MLP structure for forecasting. Specifically, it employs decomposition and applies 2 MLPs to the trend and seasonal parts, respectively.

FiLM (Zhou et al., 2022a): FilM employs the Legendre Polynomials projection technique to approximate historical information, utilizes FFT for frequency feature extraction by Frequency Enhanced Layer.

FreTS (Yi et al., 2023): FreTS designs a frequency domain MLP structure based on the operation of complex number, which is applied in the frequency domain. It contains 2 stages on both inter-series and intra-series scales, to better capture channel-wise and time-wise dependencies in the frequency domain simultaneously.

FITS (Xu et al., 2024): FITS proposes a lightweight model that utilizes complex-valued linear layer to capture frequency domain features. It also contains a low-pass filter to deal with the high frequency noise in the time series.

StemGNN (Cao et al., 2020): StemGNN effectively captures inter-series correlations and temporal dependencies simultaneously by leveraging the frequency domain. This is achieved through an end-to-end framework that combines the Graph Fourier Transform (GFT) to model inter-series correlations and the Discrete Fourier Transform (DFT) to model temporal dependencies.

We implement **Autoformer**, **FEDformer**, **iTransformer**, **Crossformer**, **PatchTST**, **TimesNet**, **DLinear**, **FiLM** based on the Time-Series-Library repository: https://github.com/thuml/Time-Series-Library.

For the rest models, we implement them based on their official code repository.

SCINet: https://github.com/cure-lab/SCINet.

FreTS: https://github.com/aikunyi/FreTS.

FITS: https://github.com/VEWOXIC/FITS.

StemGNN:https://github.com/microsoft/StemGNN/.

D.3. Implementation Details

We implement our model and all baseline models in PyTorch (Paszke et al., 2019), and train the models on NVIDIA A100-SXM 80GB GPUs. We choose ADAM (Kingma & Ba, 2014) as optimizer with a initial learning rate of $1e^{-4}$. The batch size is set differently to distinct datasets according to the number of variate. Specifically, for Traffic and Electricity dataset, we set the batch size as 32. For Solar dataset, we set the batch size as 128, and for the rest datasets, we set the batch size as 256. We employ an early stopping counter to stop the training after 3 epochs if there is no loss degradation on the valid set. By default, the F-Block contains 2 layers of complex-valued Transformer Encoder with the dimension of 512. We keep the same hyper-parameters setting for all models and all experiments.

We reproduce Autoformer, FEDformer, iTransformer, Crossformer, PatchTST, TimesNet, DLinear, FiLM based on the TimesNet (Wu et al., 2022) Repository, and SCINet, FreTS, FITS, StemGNN based on their official code repository.

D.4. Robustness of ATFNet performance

To test the robustness of ATFNet we report the standard deviation of ATFNet performance by repeating each experiment for 5 times under distinct random seeds. The results are listed in Table 6, through which we can learn that the performance of ATFNet is stable.

Table 6. Multivariate long-term forecasting results of ATFNet with 5 different random seeds.

Dataset		ETTh1		ETTh2		ETTm1		ETTm2	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ATFNet	96	0.409 ± 0.002	0.442 ± 0.000	0.172 ± 0.001	0.283 ± 0.000	0.324 ± 0.003	0.376 ± 0.001	0.114 ± 0.001	0.227 ± 0.001
	192	0.459 ± 0.005	0.481 ± 0.002	0.212 ± 0.003	0.318 ± 0.003	0.370 ± 0.005	0.411 ± 0.002	0.143 ± 0.002	0.255 ± 0.002
	336	0.509 ± 0.007	0.517 ± 0.004	0.242 ± 0.004	0.344 ± 0.005	0.411 ± 0.005	0.442 ± 0.002	0.176 ± 0.002	0.284 ± 0.002
	720	0.610 ± 0.005	0.585 ± 0.001	0.308 ± 0.008	0.409 ± 0.006	0.459 ± 0.004	0.481 ± 0.003	0.221 ± 0.002	0.320 ± 0.002

Dataset		Traffic		Electricity		Weather		Solar	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ATFNet	96	0.376 ± 0.001	0.264 ± 0.001	0.131 ± 0.001	0.228 ± 0.001	0.158 ± 0.002	0.208 ± 0.002	0.182 ± 0.001	0.242 ± 0.002
	192	0.393 ± 0.001	0.271 ± 0.001	0.148 ± 0.000	0.244 ± 0.000	0.199 ± 0.001	0.248 ± 0.002	0.198 ± 0.001	0.260 ± 0.002
	336	0.407 ± 0.000	0.278 ± 0.000	0.164 ± 0.000	0.262 ± 0.000	0.248 ± 0.001	0.287 ± 0.002	0.204 ± 0.001	0.266 ± 0.002
	720	0.438 ± 0.001	0.298 ± 0.001	0.199 ± 0.002	0.295 ± 0.001	0.312 ± 0.001	0.336 ± 0.001	0.209 ± 0.001	0.278 ± 0.003

D.5. Model efficiency comparison

We assess the computational and memory requirements of ATFNet in comparison to other baseline models. This evaluation is performed on the ETTh1 dataset using a input-96-predict-96 configuration. We maintain consistent configurations across all models, aligning them with the main experiment for multi-variate long-term forecasting. The results are presented in Table 8.

Table 7. Comparison of computational and memory costs

	ATFNet	Autoformer	Crossformer	DLinear	FEDformer	FITS	FiLM	FreTS	PatchTST	SCINet	StemGNN	TimesNet	iTransformer
Memory (GB)	0.974	0.884	0.402	0.001	1.409	0.001	0.423	0.146	0.320	0.004	1.305	0.054	0.123
Training Speed (ms/iter)	48.0	29.0	57.0	7.0	176.0	8.0	65.0	11.0	15.0	50.0	21.0	30.0	15.0

D.6. RevIN(Kim et al., 2021) in F-Block

As we discuss in section 4.2, we apply RevIN in F-Block to address distribution drift problem in frequency domain. In this experiment we test the effectiveness of RevIN in F-Block. Specifically, we train a new version of ATFNet without complex-valued RevIN in F-Block. To maintain fairness comparison, we apply origin RevIN on the whole model rather than only keep RevIN in T-Block. The results are listed in Table 8.

Table 8. Multi-variate long-term forecasting results of ATFNet with and without RevIN in F-Block. ATFNet (w/o) represents ATFNet without RevIN in F-Block.

Datasets		ETTh1		ETTh2		ETTm1		ETTm2		Traffic		Electricity		Weather		Solar	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ATFNet	96	0.405	0.442	0.171	0.282	0.327	0.375	0.114	0.227	0.376	0.264	0.131	0.227	0.156	0.206	0.180	0.241
	192	0.467	0.482	0.213	0.315	0.370	0.412	0.141	0.254	0.394	0.273	0.148	0.244	0.199	0.246	0.200	0.264
	336	0.514	0.521	0.236	0.344	0.402	0.438	0.172	0.281	0.406	0.278	0.164	0.262	0.249	0.286	0.202	0.266
	720	0.614	0.589	0.297	0.399	0.453	0.476	0.219	0.317	0.437	0.299	0.200	0.295	0.311	0.336	0.208	0.273
ATFNet(w/o)	96	0.429	0.456	0.171	0.284	0.329	0.380	0.115	0.229	0.405	0.297	0.145	0.249	0.152	0.202	0.189	0.255
	192	0.500	0.499	0.218	0.320	0.377	0.412	0.146	0.258	0.418	0.301	0.161	0.263	0.197	0.243	0.212	0.271
	336	0.552	0.532	0.248	0.345	0.416	0.441	0.175	0.287	0.432	0.308	0.176	0.278	0.248	0.284	0.222	0.281
	720	0.725	0.642	0.324	0.401	0.474	0.485	0.224	0.323	0.464	0.325	0.211	0.303	0.321	0.336	0.220	0.279

D.7. Forecasting Results Visualization

In order to facilitate a clear comparison between different models, we present supplementary prediction demonstrations for three representative datasets in Figures 7-9.

D.8. Full Results

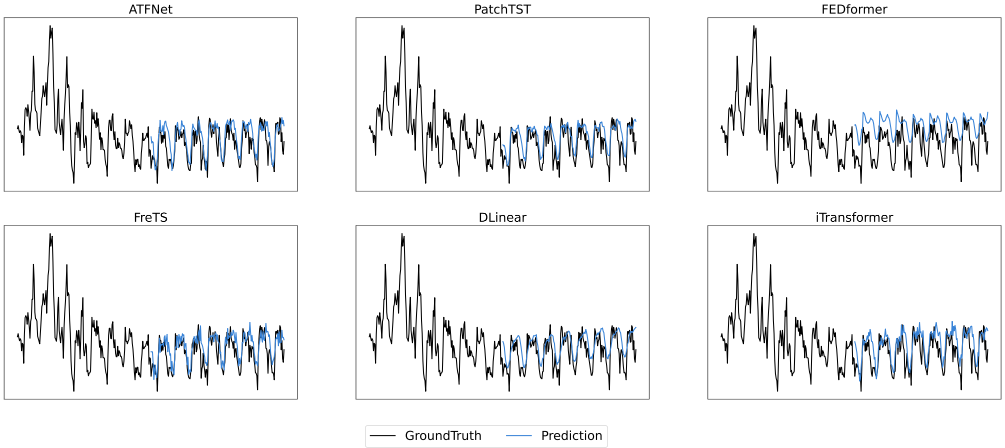


Figure 7. Visualization of the input-192-predict-192 forecasting results on ETTh1 dataset.

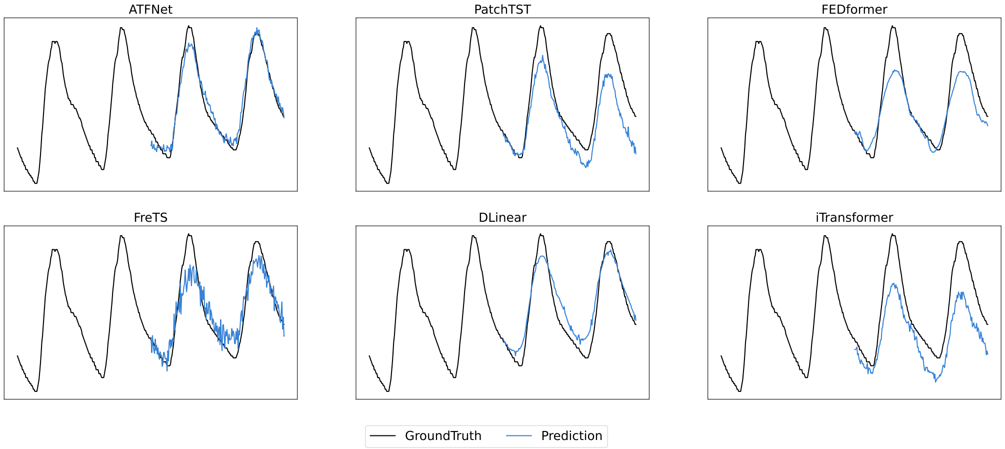


Figure 8. Visualization of the input-192-predict-192 forecasting results on ETTm2 dataset.

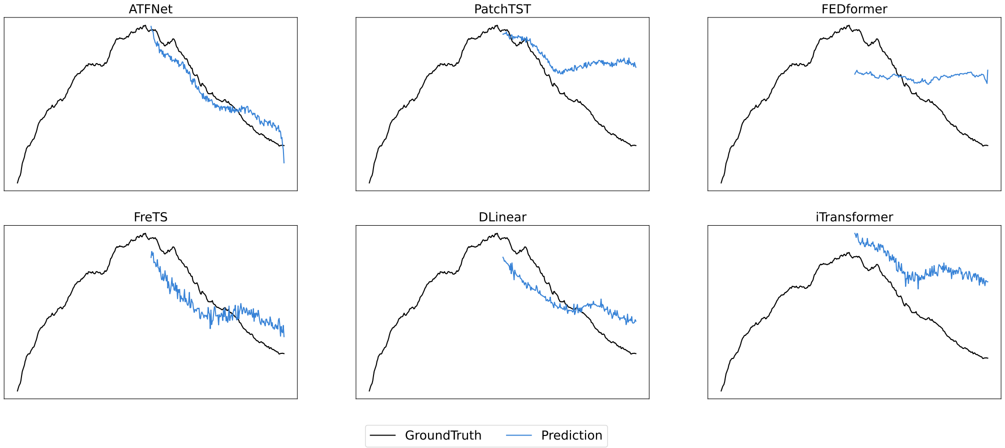


Figure 9. Visualization of the input-192-predict-192 forecasting results on Weather dataset.

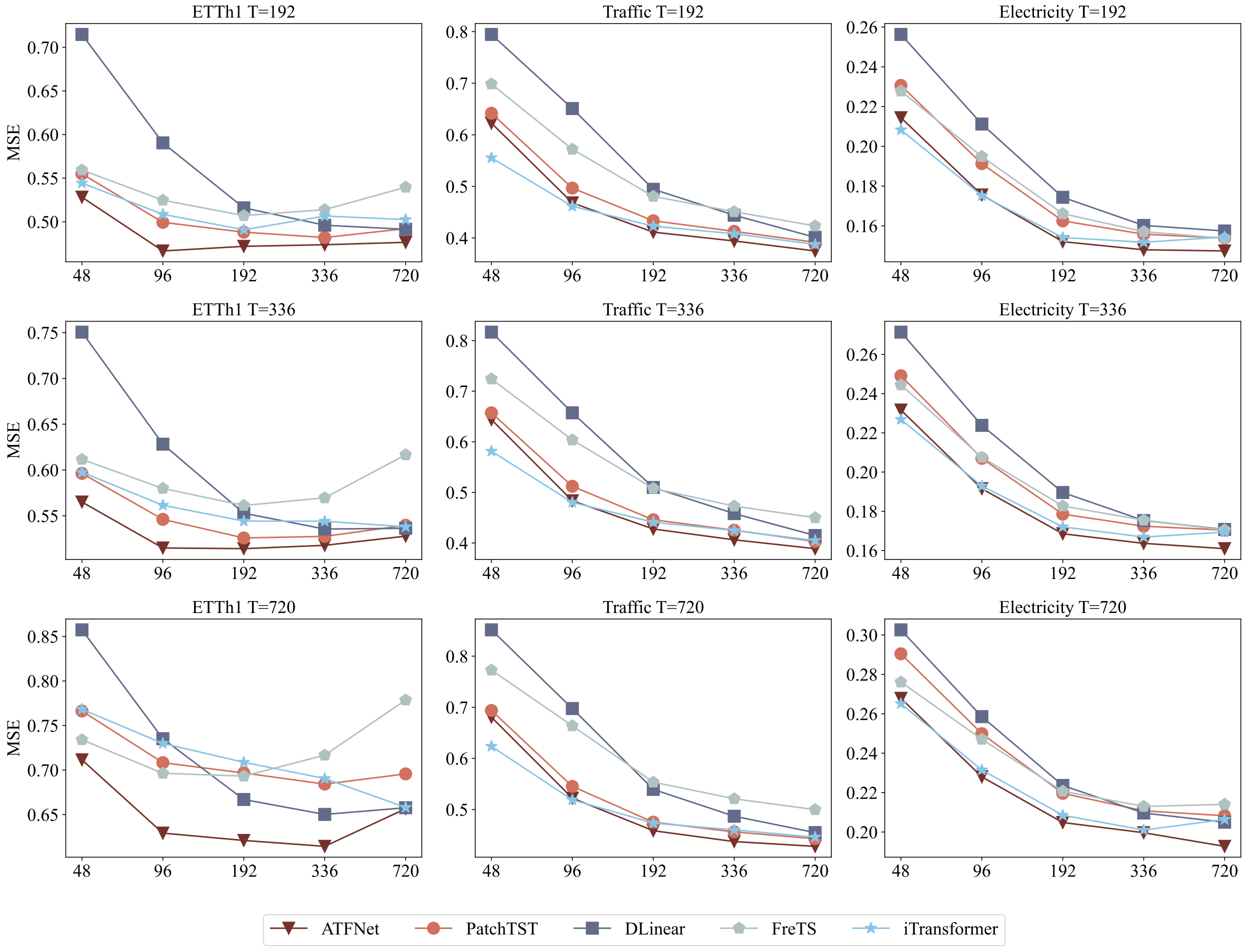


Figure 10. Multi-variate long-term time series forecasting results (MSE) with 5 different look-back window size $L \in \{48, 96, 192, 336, 720\}$. The prediction horizon is $T \in \{192, 336, 720\}$.

Table 9. Full Results of Multi-variate long-term time series forecasting.

Models		ATFNet		Autoformer		FEDformer		iTransformer		Crossformer		PatchTST		TimesNet		SCINet		DLinear		FiLM		FreTS		FITS		StemGNN	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.405**	**0.442**	0.573	0.546	0.551	0.544	0.426	0.453	0.414	0.453	0.419	0.448	0.544	0.536	0.744	0.623	0.451	0.475	0.474	0.471	0.437	0.454	1.084	0.736	0.609	0.590
	192	**0.467**	**0.482**	0.652	0.590	0.612	0.580	0.491	0.499	0.467	0.495	0.482	0.488	0.556	0.536	0.756	0.635	0.496	0.506	0.549	0.521	0.507	0.506	1.144	0.773	0.742	0.650
	336	**0.514**	0.521	0.647	0.605	0.643	0.600	0.544	0.533	0.555	0.556	0.526	**0.517**	0.612	0.581	0.820	0.669	0.536	0.535	0.569	0.539	0.561	0.546	2.035	1.014	0.685	0.635
	720	**0.614**	**0.589**	0.813	0.692	0.788	0.681	0.690	0.626	0.678	0.626	0.684	0.622	0.751	0.655	1.031	0.767	0.650	0.610	0.793	0.664	0.693	0.626	2.300	1.099	0.777	0.683
	Avg	**0.500**	**0.508**	0.672	0.608	0.649	0.601	0.538	0.528	0.529	0.533	0.528	0.519	0.616	0.577	0.837	0.674	0.533	0.532	0.596	0.549	0.550	0.533	1.641	0.905	0.703	0.639
ETTh2	96	**0.171**	**0.282**	0.211	0.316	0.207	0.313	0.181	0.293	0.196	0.315	0.173	0.285	0.203	0.312	0.263	0.367	0.185	0.297	0.201	0.303	0.175	0.288	0.286	0.379	0.287	0.391
	192	**0.211**	**0.315**	0.249	0.349	0.247	0.345	0.225	0.327	0.233	0.342	0.221	0.322	0.267	0.357	0.302	0.392	0.218	0.326	0.239	0.334	0.220	0.326	0.351	0.420	0.333	0.409
	336	**0.236**	0.344	0.266	0.364	0.262	0.363	0.257	0.353	0.259	0.362	0.249	**0.344**	0.279	0.365	0.317	0.402	0.239	0.347	0.263	0.347	0.243	0.348	0.425	0.462	0.394	0.456
	720	**0.297**	0.399	0.333	0.416	0.333	0.416	0.332	0.403	0.304	0.402	0.320	**0.392**	0.361	0.422	0.403	0.453	0.311	0.404	0.366	0.424	0.308	0.400	0.591	0.547	0.560	0.568
	Avg	**0.229**	**0.335**	0.265	0.361	0.262	0.359	0.249	0.344	0.248	0.355	0.241	0.336	0.277	0.364	0.321	0.403	0.238	0.344	0.267	0.352	0.236	0.341	0.413	0.452	0.393	0.456
ETTm1	96	0.327	**0.375**	0.770	0.603	0.456	0.490	0.338	0.385	0.333	0.386	**0.327**	0.382	0.382	0.427	0.396	0.426	0.339	0.380	0.351	0.384	0.332	0.380	0.441	0.460	0.528	0.536
	192	0.370	0.412	0.792	0.619	0.517	0.524	0.395	0.423	**0.365**	**0.409**	0.377	0.413	0.418	0.440	0.452	0.459	0.391	0.410	0.479	0.462	0.392	0.418	0.531	0.507	0.693	0.610
	336	**0.402**	**0.438**	0.855	0.656	0.570	0.557	0.440	0.454	0.421	0.453	0.411	0.441	0.474	0.488	0.496	0.484	0.433	0.438	0.483	0.458	0.441	0.453	0.658	0.574	0.667	0.625
	720	**0.453**	**0.476**	0.819	0.656	0.613	0.588	0.518	0.505	0.503	0.513	0.468	0.482	0.552	0.534	0.557	0.522	0.489	0.481	0.515	0.492	0.502	0.497	0.837	0.655	0.689	0.624
	Avg	**0.388**	**0.425**	0.809	0.633	0.539	0.540	0.423	0.442	0.405	0.440	0.396	0.429	0.457	0.473	0.475	0.473	0.413	0.427	0.457	0.449	0.417	0.437	0.617	0.549	0.644	0.599
ETTm2	96	0.114	0.227	0.138	0.258	0.122	0.239	0.120	0.233	0.113	0.235	**0.113**	**0.225**	0.122	0.238	0.131	0.251	0.116	0.230	0.121	0.235	0.116	0.232	0.136	0.258	0.169	0.298
	192	**0.141**	**0.254**	0.163	0.280	0.150	0.265	0.151	0.264	0.148	0.272	0.145	0.256	0.152	0.265	0.159	0.275	0.142	0.256	0.145	0.256	0.146	0.261	0.171	0.287	0.224	0.351
	336	0.172	**0.281**	0.194	0.303	0.186	0.293	0.187	0.292	0.181	0.299	0.178	0.286	0.187	0.291	0.190	0.299	**0.171**	0.282	0.180	0.285	0.176	0.289	0.210	0.316	0.252	0.364
	720	0.219	**0.317**	0.246	0.341	0.241	0.334	0.228	0.326	0.240	0.346	0.226	0.326	0.240	0.336	0.237	0.333	**0.216**	0.319	0.244	0.338	0.229	0.333	0.271	0.360	0.310	0.416
	Avg	0.162	**0.270**	0.185	0.296	0.175	0.283	0.171	0.279	0.171	0.288	0.165	0.273	0.176	0.283	0.180	0.290	**0.161**	0.272	0.173	0.278	0.167	0.279	0.197	0.305	0.239	0.357
traffic	96	**0.376**	**0.264**	0.611	0.370	0.593	0.369	0.388	0.277	0.537	0.306	0.397	0.281	0.838	0.487	0.505	0.373	0.433	0.308	0.478	0.343	0.419	0.305	0.654	0.463	0.843	0.480
	192	**0.394**	**0.273**	0.616	0.381	0.591	0.365	0.408	0.288	0.617	0.359	0.413	0.290	0.809	0.484	0.517	0.380	0.444	0.313	0.439	0.303	0.451	0.307	0.736	0.497	0.840	0.476
	336	**0.406**	**0.278**	0.641	0.397	0.616	0.377	0.425	0.299	1.307	0.794	0.425	0.296	0.775	0.498	0.531	0.388	0.458	0.321	0.461	0.319	0.473	0.315	0.931	0.610	0.828	0.468
	720	**0.437**	**0.299**	0.652	0.400	0.633	0.384	0.460	0.322	1.332	0.794	0.456	0.313	0.841	0.520	0.578	0.408	0.487	0.338	0.587	0.415	0.521	0.344	1.160	0.695	0.874	0.489
	Avg	**0.403**	**0.279**	0.630	0.387	0.608	0.374	0.420	0.296	0.948	0.563	0.423	0.295	0.816	0.497	0.533	0.387	0.456	0.320	0.491	0.345	0.466	0.318	0.870	0.566	0.847	0.478
electricity	96	**0.131**	**0.227**	0.200	0.312	0.202	0.314	0.133	0.229	0.144	0.246	0.140	0.244	0.270	0.360	0.165	0.272	0.148	0.250	0.148	0.246	0.145	0.248	0.200	0.313	0.195	0.309
	192	**0.148**	**0.244**	0.204	0.318	0.206	0.319	0.152	0.248	0.168	0.269	0.156	0.258	0.274	0.363	0.183	0.289	0.160	0.262	0.164	0.263	0.157	0.257	0.237	0.349	0.203	0.311
	336	**0.164**	**0.262**	0.219	0.331	0.232	0.346	0.167	0.263	0.183	0.284	0.172	0.274	0.287	0.375	0.184	0.290	0.175	0.278	0.187	0.288	0.175	0.276	0.368	0.459	0.238	0.338
	720	**0.200**	0.295	0.237	0.348	0.249	0.360	0.201	**0.295**	0.257	0.340	0.211	0.305	0.321	0.401	0.202	0.304	0.210	0.310	0.243	0.339	0.213	0.313	0.447	0.516	0.285	0.377
	Avg	**0.161**	**0.257**	0.215	0.327	0.222	0.335	0.163	0.259	0.188	0.285	0.170	0.270	0.288	0.375	0.184	0.289	0.173	0.275	0.185	0.284	0.173	0.273	0.313	0.409	0.230	0.334
weather	96	0.156	0.206	0.253	0.319	0.242	0.316	0.163	0.211	**0.146**	0.211	0.153	**0.202**	0.196	0.244	0.194	0.244	0.182	0.246	0.203	0.248	0.179	0.239	0.234	0.284	0.289	0.376
	192	0.199	0.246	0.303	0.353	0.303	0.362	0.207	0.251	**0.191**	0.256	0.197	**0.242**	0.242	0.281	0.241	0.283	0.219	0.278	0.241	0.282	0.219	0.276	0.273	0.311	0.268	0.350
	336	0.249	0.286	0.355	0.390	0.344	0.389	0.254	0.288	**0.245**	0.308	0.246	**0.283**	0.275	0.308	0.284	0.312	0.264	0.315	0.294	0.323	0.266	0.313	0.337	0.355	0.341	0.402
	720	**0.311**	0.335	0.401	0.420	0.393	0.411	0.327	0.339	0.336	0.377	0.318	**0.332**	0.348	0.351	0.349	0.358	0.322	0.362	0.341	0.349	0.330	0.366	0.430	0.411	0.381	0.433
	Avg	**0.228**	0.268	0.328	0.371	0.321	0.370	0.238	0.272	0.230	0.288	0.229	**0.265**	0.265	0.296	0.267	0.299	0.247	0.300	0.270	0.301	0.248	0.298	0.318	0.340	0.320	0.390
solar	96	0.180	0.241	0.687	0.593	0.251	0.359	0.193	0.243	**0.165**	**0.229**	0.185	0.246	0.312	0.348	0.249	0.319	0.232	0.304	0.260	0.305	0.191	0.262	0.324	0.391	0.255	0.303
	192	0.200	0.264	0.840	0.714	0.270	0.384	0.217	0.269	**0.187**	**0.244**	0.203	0.277	0.324	0.366	0.260	0.323	0.255	0.317	0.311	0.345	0.224	0.283	0.367	0.436	0.218	0.292
	336	**0.202**	0.266	0.609	0.572	0.299	0.412	0.227	0.280	0.205	**0.256**	0.218	0.286	0.344	0.377	0.275	0.334	0.272	0.331	0.322	0.354	0.245	0.301	0.420	0.473	0.236	0.309
	720	**0.208**	0.273	0.852	0.719	0.319	0.413	0.239	0.298	0.215	**0.267**	0.215	0.278	0.318	0.357	0.272	0.332	0.274	0.333	0.290	0.309	0.264	0.313	0.456	0.498	0.233	0.299
	Avg	0.198	0.261	0.747	0.649	0.285	0.392	0.219	0.272	**0.193**	**0.249**	0.205	0.272	0.325	0.362	0.264	0.327	0.258	0.321	0.296	0.328	0.231	0.290	0.392	0.450	0.235	0.301

Table 10. Full Results of Uni-variate long-term time series forecasting.

Models		ATFNet		Autoformer		FEDformer		PatchTST		TimesNet		SCINet		DLinear		FiLM		FreTS		FITS	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.093	0.235	0.118	0.272	0.110	0.264	0.098	0.241	0.098	0.241	0.136	0.291	0.099	0.247	0.107	0.257	**0.091**	**0.235**	0.161	0.316
	192	**0.110**	**0.259**	0.117	0.272	0.116	0.273	0.124	0.275	0.121	0.273	0.186	0.336	0.110	0.263	0.115	0.267	0.110	0.261	0.181	0.338
	336	0.118	0.270	0.119	0.275	0.126	0.283	0.124	0.279	0.117	0.271	0.162	0.319	**0.111**	**0.265**	0.129	0.287	0.116	0.271	0.213	0.365
	720	0.147	0.308	0.150	0.306	0.165	0.320	0.197	0.351	0.171	0.327	0.226	0.377	**0.140**	0.306	0.153	0.313	0.144	**0.305**	0.306	0.439
	Avg	0.117	0.268	0.126	0.281	0.129	0.285	0.136	0.287	0.126	0.278	0.178	0.331	0.115	0.270	0.126	0.281	**0.115**	**0.268**	0.215	0.364
ETTh2	96	**0.226**	**0.371**	0.273	0.422	0.259	0.407	0.233	0.374	0.252	0.394	0.328	0.455	0.255	0.402	0.277	0.413	0.250	0.389	0.509	0.561
	192	0.279	**0.418**	**0.278**	0.424	0.291	0.434	0.290	0.428	0.302	0.439	0.398	0.500	0.292	0.434	0.330	0.461	0.313	0.443	0.642	0.631
	336	0.298	0.433	**0.283**	**0.426**	0.306	0.443	0.345	0.469	0.314	0.451	0.432	0.525	0.305	0.443	0.313	0.451	0.351	0.470	0.691	0.655
	720	**0.349**	**0.477**	0.382	0.494	0.443	0.524	0.493	0.567	0.420	0.519	0.656	0.649	0.365	0.486	0.385	0.494	0.473	0.540	1.045	0.814
	Avg	**0.288**	**0.425**	0.304	0.442	0.324	0.452	0.340	0.460	0.322	0.450	0.454	0.532	0.304	0.441	0.326	0.455	0.347	0.460	0.722	0.665
ETTm1	96	**0.045**	**0.151**	0.068	0.202	0.056	0.178	0.045	0.154	0.045	0.152	0.051	0.169	0.049	0.162	0.051	0.167	0.049	0.161	0.070	0.202
	192	0.075	**0.202**	0.096	0.243	0.087	0.225	0.076	0.206	0.076	0.207	0.082	0.216	**0.075**	0.206	0.084	0.222	0.081	0.212	0.106	0.251
	336	**0.100**	**0.239**	0.130	0.281	0.127	0.276	0.105	0.245	0.106	0.249	0.109	0.253	0.102	0.244	0.114	0.260	0.109	0.254	0.135	0.286
	720	0.131	0.276	0.164	0.313	0.157	0.312	0.138	0.285	0.134	0.280	0.137	0.284	**0.126**	**0.272**	0.143	0.293	0.139	0.292	0.171	0.321
	Avg	**0.088**	**0.217**	0.115	0.260	0.107	0.248	0.091	0.223	0.090	0.222	0.095	0.231	0.088	0.221	0.098	0.236	0.095	0.230	0.121	0.265
ETTm2	96	**0.115**	0.236	0.202	0.343	0.136	0.273	0.116	**0.235**	0.122	0.251	0.137	0.272	0.132	0.262	0.157	0.288	0.123	0.246	0.229	0.362
	192	**0.180**	**0.307**	0.282	0.409	0.209	0.344	0.186	0.312	0.196	0.324	0.201	0.336	0.194	0.325	0.221	0.350	0.186	0.315	0.298	0.421
	336	**0.238**	**0.369**	0.337	0.442	0.287	0.403	0.252	0.381	0.265	0.392	0.272	0.396	0.266	0.386	0.298	0.413	0.257	0.384	0.418	0.505
	720	**0.285**	**0.416**	0.391	0.488	0.369	0.472	0.314	0.438	0.337	0.452	0.349	0.459	0.310	0.431	0.376	0.470	0.334	0.452	0.485	0.548
	Avg	**0.205**	**0.332**	0.303	0.421	0.250	0.373	0.217	0.342	0.230	0.355	0.240	0.366	0.226	0.351	0.263	0.380	0.225	0.349	0.358	0.459

Table 11. Full Results of Ablation Study.

Models		ATFNet		F-Block		T-Block		ATFNet v1		ATFNet v2		ATFNet v3	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.405**	0.442	0.413	0.445	0.419	0.448	0.414	**0.441**	0.414	0.441	0.417	0.444
	192	**0.467**	**0.482**	0.469	0.483	0.482	0.488	0.471	0.485	0.470	0.487	0.475	0.488
	336	0.514	0.521	0.517	0.524	0.526	0.517	0.517	0.520	**0.511**	**0.516**	0.529	0.530
	720	**0.614**	0.589	0.706	0.633	0.684	0.622	0.624	0.589	0.648	0.599	0.619	**0.588**
	Avg	**0.500**	**0.508**	0.526	0.521	0.528	0.519	0.506	0.509	0.511	0.511	0.510	0.512
ETTm1	96	0.327	0.375	0.324	**0.373**	0.327	0.382	0.327	0.378	**0.321**	0.376	0.326	0.380
	192	0.370	0.412	0.387	0.414	0.377	0.413	**0.370**	**0.411**	0.372	0.413	0.375	0.413
	336	**0.402**	0.438	0.432	0.450	0.411	0.441	0.412	0.441	0.412	0.438	0.407	**0.438**
	720	**0.453**	**0.476**	0.489	0.495	0.468	0.482	0.462	0.482	0.466	0.483	0.458	0.480
	Avg	**0.388**	**0.425**	0.408	0.433	0.396	0.429	0.393	0.428	0.393	0.427	0.392	0.427
traffic	96	0.376	0.264	**0.373**	0.265	0.397	0.281	0.377	**0.263**	0.375	0.267	0.386	0.268
	192	0.394	0.273	0.394	0.274	0.413	0.290	0.395	**0.272**	**0.393**	0.275	0.401	0.277
	336	**0.406**	**0.278**	0.410	0.283	0.425	0.296	0.409	0.280	0.407	0.283	0.412	0.281
	720	**0.437**	**0.299**	0.449	0.310	0.456	0.313	0.440	0.300	0.441	0.301	0.443	0.301
	Avg	**0.403**	**0.279**	0.406	0.283	0.423	0.295	0.405	0.279	0.404	0.282	0.410	0.282
weather	96	0.156	0.206	0.159	0.208	**0.153**	**0.202**	0.160	0.211	0.164	0.221	0.155	0.211
	192	0.199	0.246	0.204	0.252	**0.197**	**0.242**	0.201	0.249	0.202	0.260	0.198	0.249
	336	0.249	0.286	0.257	0.300	**0.246**	**0.283**	0.251	0.289	0.255	0.304	0.248	0.294
	720	**0.311**	0.336	0.327	0.373	0.318	**0.332**	0.318	0.346	0.318	0.352	0.314	0.343
	Avg	**0.228**	0.268	0.237	0.283	0.229	**0.265**	0.233	0.274	0.235	0.284	0.229	0.274