
FITS: Modeling Time Series with $10k$ Parameters

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 In this paper, we introduce FITS, a lightweight yet powerful model for time series
2 analysis. Unlike existing models that directly process raw time-domain data, FITS
3 operates on the principle that time series can be manipulated through interpolation
4 in the complex frequency domain. By discarding high-frequency components with
5 negligible impact on time series data, FITS achieves performance comparable to
6 state-of-the-art models for time series forecasting and anomaly detection tasks,
7 while having a remarkably compact size of only approximately $10k$ parameters.
8 Such a lightweight model can be easily trained and deployed in edge devices,
9 creating opportunities for various applications. The anonymous code repo is
10 available in: <https://anonymous.4open.science/r/FITS>

11 1 Introduction

12 Time series analysis plays a crucial role in numerous domains, including finance, energy, weather
13 forecasting, and signal processing, where understanding and predicting temporal patterns are essential.
14 Existing time series analysis methods primarily focus on extracting features in the time domain (Zhou
15 et al., 2021; Liu et al., 2022; Zeng et al., 2022; Nie et al., 2023; Zhang et al., 2022). However, due to
16 the inherent complexity and dynamic nature of time series data, the information contained in the time
17 domain tends to be sparse and dispersed. Consequently, researchers design intricate methodologies
18 and complex models to capture and exploit this information, often relying on approaches such as
19 transformer architectures (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022a). However, these
20 sophisticated techniques often lead to the proliferation of large-scale and computationally demanding
21 models, posing challenges in terms of efficiency and scalability.

22 Conversely, the frequency domain representation of time series data offers a more concise and
23 compact representation of its underlying information. Recognizing this potential, previous studies
24 have explored the utilization of frequency domain information in time series analysis. For instance,
25 FEDformer (Zhou et al., 2022a) incorporates spectral information as a supplementary feature, en-
26 hancing the modeling capabilities of transformer-based time series models. Another approach,
27 FNet (Lee-Thorp et al., 2022), leverages frequency domain multiplication to replace convolution
28 operations, thereby reducing computational overhead. Moreover, LTSF-Linear (Zeng et al., 2022)
29 has demonstrated that highly accurate predictions can be achieved by solely learning the dominant
30 periodicity. Similarly, methods like TimesNet (Wu et al., 2023) segment the time series based on
31 frequencies with high amplitude and employ CNNs for multi-periodicity feature extraction.

32 However, existing methodologies often overlook the fundamental nature of the frequency domain
33 representation, which utilizes complex numbers to express both amplitude and phase information.
34 Motivated by the fact that longer time series segments provide a higher-resolution frequency rep-
35 resentation, we propose FITS (Frequency Interpolation Time Series Analysis Baseline). The core
36 component of FITS is a complex-valued linear layer that can explicitly learn amplitude scaling and
37 phase shift to perform interpolation in the complex frequency domain. Although FITS conducts
38 interpolation in the frequency domain, it remains an end-to-end time domain model incorporating

39 the rFFT (Brigham & Morrow, 1967). Specifically, we project the input segment to the complex
40 frequency domain for frequency interpolation using rFFT. We then project the interpolated frequency
41 representation back to the time domain as a longer segment for supervision. This end-to-end design
42 enables FITS to adapt to various downstream tasks with commonly-used time domain supervision,
43 such as forecasting and reconstruction.

44 Additionally, FITS incorporates a low-pass filter to obtain a compact representation with minimal
45 information loss, resulting in small model volume and minimal computational overhead while
46 maintaining state-of-the-art (SOTA) performance. Notably, under most settings, FITS achieves
47 SOTA performance with under **10k parameters**, which is **50 times smaller** than the lightweight
48 temporal linear model DLinear (Zeng et al., 2022) and approximately **10,000 times smaller** than
49 other mainstream models. The low memory and computation overhead make FITS suitable for
50 deploying or even training on edge devices for forecasting or anomaly detection.

51 To summarize, our contributions are twofold:

- 52 • We introduce FITS, a lightweight model containing merely **5k~10k** parameters for time
53 series analysis. Despite its compact size which is several orders of magnitude smaller than
54 mainstream models, FITS delivers exceptional performance in various tasks, including
55 long-term forecasting and anomaly detection, achieving state-of-the-art performance in
56 several datasets.
- 57 • FITS employs the complex-valued neural network for time series analysis, which provides a
58 novel perspective that simultaneously captures amplitude and phase information, leading to
59 more comprehensive and efficient modeling of time series data.

60 **2 Related Work and Motivation**

61 **2.1 Frequency-aware Time Series Analysis Models**

62 Recent advancements in time series analysis have witnessed the utilization of frequency domain
63 information to capture and interpret underlying patterns. FNet (Lee-Thorp et al., 2022) leverages a
64 pure attention-based architecture to efficiently capture temporal dependencies and patterns solely in
65 the frequency domain, eliminating the need for convolutional or recurrent layers. On the other hand,
66 FEDFormer (Zhou et al., 2022a) and FiLM (Zhou et al., 2022b) incorporate frequency information as
67 supplementary features to enhance the model’s capability in capturing long-term periodic patterns
68 and speed up computation.

69 The other line of work aims to capture the periodicity inherent in the data. For instance, DLinear (Zeng
70 et al., 2022) adopts a single linear layer to extract the dominant periodicity from the temporal domain
71 and surpasses a range of deep feature extraction-based methods. More recently, TimesNet (Wu et al.,
72 2023) achieves state-of-the-art results by identifying several dominant frequencies instead of relying
73 on a single dominant periodicity. Specifically, they use the Fast Fourier Transform (FFT) to find the
74 frequencies with the largest energy and reshape the original 1D time series into 2D images according
75 to their periods.

76 However, these approaches still rely on feature engineering to identify the dominant period set.
77 Selecting this set based on energy may only consider the dominant period and its harmonics, limiting
78 the information captured. Moreover, these methodologies are still considered inefficient and prone to
79 overfitting.

80 **2.2 Divide and Conquer the Frequency Components**

81 Treating a time series as a signal allows us to break it down into a linear combination of sinusoidal
82 components without any information loss. Each component possesses a unique frequency, initial
83 phase, and amplitude. Forecasting directly on the original time series can be challenging, but
84 forecasting each frequency component is comparatively straightforward, as we only need to apply a
85 phase bias to the sinusoidal wave based on the time shift. Subsequently, we linearly combine these
86 shifted sinusoidal waves to obtain the forecasting result.

87 This approach effectively preserves the frequency characteristics of the given look-back window
88 while maintaining semantic consistency between the look-back window and the forecasting horizon.

89 Specifically, the resulting forecasted values maintain the frequency features of the original time series
 90 with a reasonable time shift, ensuring that semantic consistency is maintained.

91 However, forecasting each sinusoidal component in the time domain can be cumbersome, as the
 92 sinusoidal components are treated as a sequence of data points. To address this, we propose conducting
 93 this manipulation in the complex frequency domain, which offers a more compact and information-
 94 rich representation, as described below.

95 3 Method

96 3.1 Preliminary: FFT and Complex Frequency Domain

97 The Fast Fourier Transform (FFT, (Brigham & Morrow, 1967)) is a widely used algorithm for
 98 efficiently computing the Discrete Fourier Transform (DFT) of a sequence of complex numbers. The
 99 DFT is a mathematical operation that converts a discrete-time signal from the time domain to the
 100 complex frequency domain. In cases where the input signal is real, such as in time series analysis,
 101 the Real FFT (rFFT) is commonly used to obtain a compact representation. With an input of N real
 102 numbers, the rFFT produces a sequence of $N/2 + 1$ complex numbers that represent the signal in the
 103 complex frequency domain.

104 Complex Frequency Domain

105 In Fourier analysis, the complex frequency domain is a representation of a signal in which each
 106 frequency component is characterized by a complex number. This complex number captures both
 107 the amplitude and phase of the component, providing a comprehensive description. The amplitude
 108 of a frequency component represents the magnitude or strength of that component in the original
 109 time-domain signal. In contrast, the phase represents the temporal shift or delay introduced by that
 110 component. Mathematically, the complex number associated with a frequency component can be
 111 represented as a complex exponential element with a given amplitude and phase:

$$X(f) = |X(f)|e^{j\theta(f)},$$

112 where $X(f)$ is the complex number associated with the frequency component at frequency f , $|X(f)|$
 113 is the amplitude of the component, and $\theta(f)$ is the phase of the component. As shown in Fig. 1(a), in
 114 the complex plane, the complex exponential element can be visualized as a vector with a length equal
 115 to the amplitude and angle equal to the phase:

$$X(f) = |X(f)|(\cos \theta(f) + j \sin \theta(f))$$

116 Therefore, the complex number in the complex frequency domain provides a concise and elegant
 117 means of representing the amplitude and phase of each frequency component in the Fourier transform.

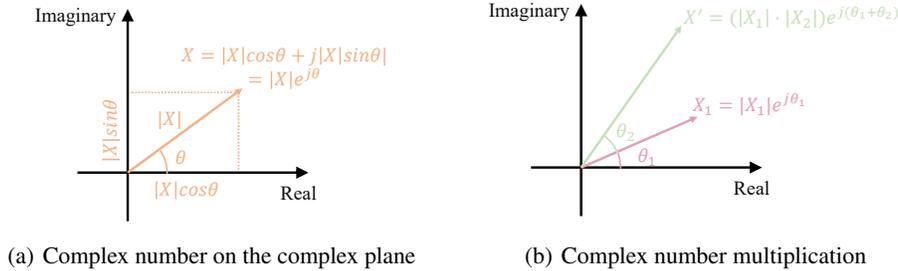


Figure 1: Illustration of Complex Number Visualization and Multiplication

118 **Time Shift and Phase Shift.** The time shift of a signal corresponds to the phase shift in the frequency
 119 domain. Especially in the complex frequency domain, we can express such phase shift by multiplying
 120 a unit complex exponential element with the corresponding phase. Mathematically, if we shift a
 121 signal $x(t)$ forward in time by a constant amount τ , resulting in the signal $x(t - \tau)$, the Fourier
 122 transform is given by:

$$X_\tau(f) = e^{-j2\pi f\tau} X(f) = |X(f)|e^{j(\theta(f)-2\pi f\tau)} = [\cos(-2\pi f\tau) + j\sin(-2\pi f\tau)]X(f)$$

123 The shifted signal still has an amplitude of $|X(f)|$, while the phase $\theta_\tau(f) = \theta(f) - 2\pi f\tau$ shows a
 124 shift which is linear to the time shift.

125 In summary, the amplitude scaling and phase shifting can be simultaneously expressed as the
 126 multiplication of complex numbers, as shown in Fig. 1(b).

127 3.2 FITS Pipeline

128 Motivated by the fact that a longer time series provides a higher frequency resolution in its frequency
 129 representation, we train FITS to generate an extended time series segment by interpolating the
 130 frequency representation of the input time series segment. We use a complex-valued linear layer to
 131 learn such interpolation. According to the fact that the amplitude scaling and phase shifting can be
 132 conveniently expressed as the multiplication of complex numbers, such complex linear combination
 133 allows FITS to effectively incorporate both the amplitude scaling and phase shift of frequency
 134 components during the interpolation process. As shown in Fig. 2, we use rFFT to project time series
 135 segments to the complex frequency domain. After the interpolation, the frequency representation is
 136 projected back with inverse rFFT (irFFT).

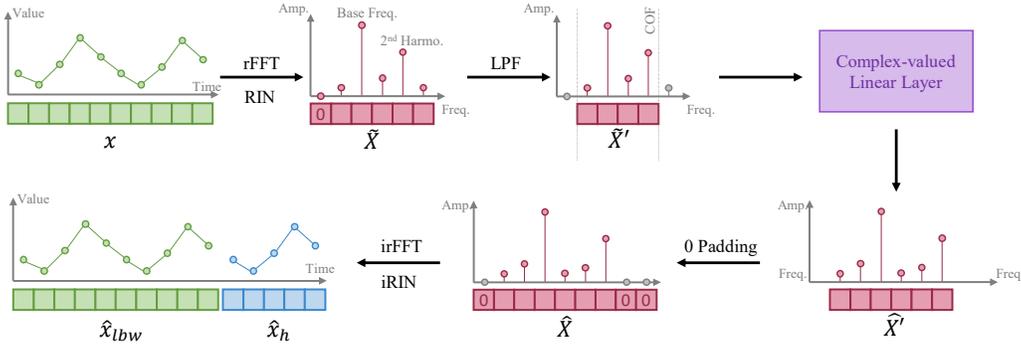


Figure 2: Pipeline of FITS, with a focus on the forecasting task. The reconstruction task follows the same pipeline, except for the reconstruction supervision loss.

137 However, we cannot directly use the frequency representation of the original input time series segment
 138 because the mean of such segments will result in a very large 0-frequency component in its complex
 139 frequency representation. To eliminate the 0-frequency component, we pass it through reversible
 140 instance-wise normalization (RIN) (Kim et al., 2022) to obtain a zero-mean instance. As a result,
 141 the normalized complex frequency representation now has a length of $N/2$, where N represents the
 142 original length of the time series.

143 Furthermore, we incorporate a low-pass filter (LPF) into the FITS model to further reduce its size.
 144 The LPF removes high-frequency components above a specified cutoff frequency, resulting in a more
 145 compact model representation while retaining the important information of the time series. The
 146 rationale behind this design will be elaborated in the subsequent section. Despite operating in the
 147 frequency domain, FITS is supervised in the time domain using common loss functions such as Mean
 148 Squared Error (MSE) after the irFFT, allowing for diverse supervision tailored to different time series
 149 downstream tasks.

150 In the case of forecasting tasks, we generate the look-back window along with the horizon as shown
 151 in Fig. 2. This allows us to provide supervision for forecasting and backcasting, where the model
 152 is encouraged to accurately reconstruct the look-back window. Our ablation study reveals that
 153 combining backcast and forecast supervision can yield improved performance in certain scenarios.

154 For reconstruction tasks, we downsample the original time series segment based on a specific
 155 downsampling rate. Subsequently, FITS is employed to perform frequency interpolation, enabling
 156 the reconstruction of the downsampled segment back to its original form. Thus, direct supervision
 157 is applied using reconstruction loss to ensure faithful reconstruction. The reconstruction tasks also
 158 follow the pipeline in Fig. 2 with the supervision replaced with reconstruction loss.

159 **3.3 Key Mechanisms of FITS**

160 **Complex Frequency Linear Interpolation.** To control the output length of the model, we introduce
 161 an interpolation rate denoted as η , which represents the ratio of the model’s output length L_o to its
 162 corresponding input length L_i .

163 It is worth noting that frequency interpolation operates on the normalized complex frequency repre-
 164 sentation, which has half the length of the original time series. Importantly, this interpolation rate can
 165 also be applied to the frequency domain, as indicated by the equation:

$$\eta_{freq} = \frac{L_o/2}{L_i/2} = \frac{L_o}{L_i} = \eta$$

166 Based on this formula, with an arbitrary frequency f , the frequency band $1 \sim f$ in the original
 167 signal is linearly projected to the frequency band $1 \sim \eta f$ in the output signal. As a result, we define
 168 the input length of our complex-valued linear layer as L and the interpolated output length as ηL .
 169 Notably, when applying the Low Pass Filter (LPF), the value of L corresponds to the cutoff frequency
 170 (COF) of the LPF. After performing frequency interpolation, the complex frequency representation is
 171 zero-padded to a length of $L_o/2$, where L_o represents the desired output length. Prior to applying the
 172 irFFT, an additional zero is introduced as the representation’s zero-frequency component.

173 **Low Pass Filter (LPF).** The primary objective of incorporating the LPF within FITS is to compress
 174 the model’s volume while preserving essential information. The LPF achieves this by discarding
 175 frequency components above a specified cutoff frequency (COF), resulting in a more concise fre-
 176 quency domain representation. The LPF retains the relevant information in the time series while
 177 discarding components beyond the model’s learning capability. This ensures that a significant portion
 178 of the original time series’ meaningful content is preserved. As demonstrated in Fig. 3, the filtered
 179 waveform exhibits minimal distortion even when only preserving a quarter of the original frequency
 180 domain representation. Furthermore, the high-frequency components filtered out by the LPF typically
 comprise noise and trends, which are inherently irrelevant for effective time series modeling.

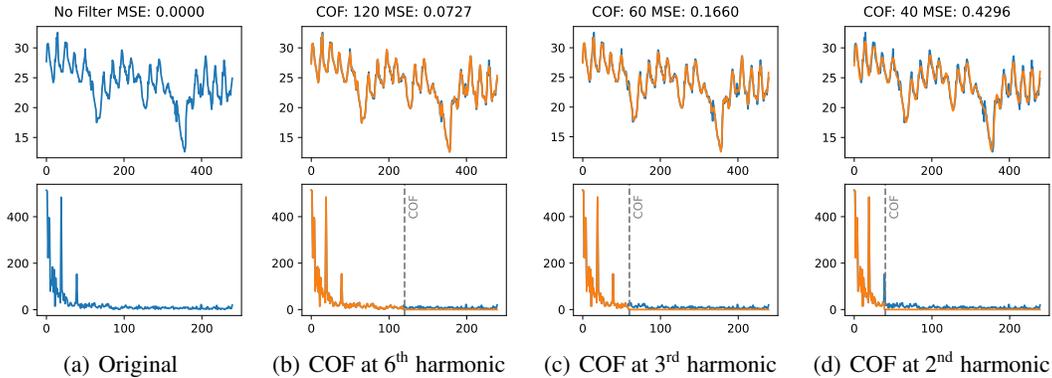


Figure 3: Waveform (1st row) and amplitude spectrum (2nd row) of a time series segment selected from the 'OT' channel of the ETTh1 dataset, spanning from the 1500th to the 1980th data point. The segment has a length of 480, and its dominant periodicity is 24, corresponding to a base frequency of 20. The blue lines represent the waveform/spectrum with no applied filter, while the orange lines represent the waveform/spectrum with the filter applied. The filter cutoff frequency is chosen based on a harmonic of the original time series.

181

182 Selecting an appropriate cutoff frequency (COF) remains a nontrivial challenge. To address this,
 183 we propose a method based on the harmonic content of the dominant frequency. Harmonics, which
 184 are integer multiples of the dominant frequency, play a significant role in shaping the waveform of
 185 a time series. By aligning the cutoff frequency with these harmonics, we keep relevant frequency
 186 components associated with the signal’s structure and periodicity. This approach leverages the
 187 inherent relationship between frequencies to extract meaningful information while suppressing noise
 188 and irrelevant high-frequency components. The impact of COF on different harmonics’ waveforms is
 189 shown in Fig. 3. We further elaborate on the impact of COF in our experimental results.

190 4 Experiments for Forecasting

191 4.1 Forecasting as Frequency Interpolation

192 Typically, the forecasting horizon is shorter than the given look-back window, rendering direct
193 interpolation unsuitable. Instead, we formulate the forecasting task as the interpolation of a look-back
194 window, with length L , to a combination of the look-back window and forecasting horizon, with
195 length $L + H$. This design enables us to provide more supervision during training. With this approach,
196 we can supervise not only the forecasting horizon but also the backcast task on the look-back window.
197 Our experimental results demonstrate that this unique training strategy contributes to the improved
198 performance of FITS. The interpolation rate of the forecasting task is calculated by:

$$\eta_{Fore} = 1 + \frac{H}{L},$$

199 where L represents the length of the look-back window and H represents the length of the forecasting
200 horizon.

201 4.2 Experiment Settings

202 **Datasets.** All datasets used in our experiments are widely-used and publicly available real-world
203 datasets, including, Traffic, Electricity, Weather, ETT (Zhou et al., 2021). We summarize the
204 characteristics of these datasets in Tab. 1. Apart from these datasets for long-term time series
205 forecasting, we also use the M4 dataset to test the short-term forecasting performance.

Table 1: The statistics of the seven used forecasting datasets.

| Dataset | Traffic | Electricity | Weather | ETTh1&ETTh2 | ETTm1 & ETTm2 |
|-----------------|---------|-------------|---------|-------------|---------------|
| Channels | 862 | 321 | 21 | 7 | 7 |
| Sampling Rate | 1hour | 1hour | 10min | 1hour | 15min |
| Total Timesteps | 17,544 | 26,304 | 52,696 | 17,420 | 69,680 |

206 **Baselines.** To evaluate the performance of FITS in comparison to state-of-the-art time series forecast-
207 ing models, including PatchTST (Nie et al., 2023), TimesNet (Wu et al., 2023), FEDFormer (Zhou
208 et al., 2022a), FiLM (Zhou et al., 2022b) and LTSF-Linear (Zeng et al., 2023), we directly refer to
209 the reported results in the original papers under the same settings. We report the comparison with
210 other transformer-based methods in the appendix.

211 **Evaluation metrics.** We follow the previous works (Zhou et al., 2022a; Zeng et al., 2022; Zhang
212 et al., 2022) to compare forecasting performance using Mean Squared Error (MSE) as the core
213 metrics. Moreover, to evaluate the short-term forecasting, we symmetric Mean Absolute Percentage
214 Error (SMAPE) following TimesNet (Wu et al., 2023).

215 **Implementation details.** Following the settings of LTSF-Linear (Zeng et al., 2023), we set the
216 look-back window of FITS as 720 for any forecasting horizon. Further experiments also show that a
217 longer look-back window can result in better performance. To avoid information leakage, We choose
218 the hyper-parameter based on the performance of the validation set.

219 4.3 Comparisons with SOTAs

220 Competitive Performance with High Efficiency

221 We present the results of our experiments on long-term forecasting in Tab. 2 and Tab. 3. The results
222 for short-term forecasting on the M4 dataset are provided in the Appendix. Remarkably, our FITS
223 consistently achieves comparable or even superior performance across all experiments.

224 Tab. 4 presents the number of trainable parameters for various TSF models using a look-back window
225 of 96 and a forecasting horizon of 720 on the Electricity dataset. The table clearly demonstrates the
226 exceptional efficiency of FITS compared to other models.

227 Among the listed models, the parameter counts range from millions down to thousands. Notably,
228 large models such as TimesNet and Pyraformer require a staggering number of parameters, with

Table 2: Long-term forecasting results on ETT dataset in MSE. The best result is highlighted in **bold**, and the second best is highlighted with underline. IMP is the improvement between FITS and the second best/ best result, where a larger value indicates a better improvement.

| Dataset | ETTh1 | | | | ETTh2 | | | | ETTm1 | | | | ETTm2 | | | |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | 96 | 192 | 336 | 720 | 96 | 192 | 336 | 720 | 96 | 192 | 336 | 720 | 96 | 192 | 336 | 720 |
| PatchTST | 0.370 | 0.413 | 0.422 | 0.447 | 0.274 | <u>0.341</u> | 0.329 | <u>0.379</u> | 0.293 | 0.333 | <u>0.369</u> | 0.416 | 0.166 | 0.223 | 0.274 | <u>0.362</u> |
| TimesNet | 0.384 | 0.436 | 0.491 | 0.521 | 0.340 | 0.402 | 0.452 | 0.462 | 0.338 | 0.374 | 0.410 | 0.478 | 0.187 | 0.249 | 0.321 | 0.408 |
| FEDFormer | 0.376 | 0.420 | 0.459 | 0.506 | 0.346 | 0.429 | 0.496 | 0.463 | 0.379 | 0.426 | 0.445 | 0.543 | 0.203 | 0.269 | 0.325 | 0.421 |
| FiLM | <u>0.371</u> | 0.414 | 0.442 | 0.465 | 0.284 | 0.357 | 0.377 | 0.439 | 0.302 | 0.338 | 0.373 | 0.420 | <u>0.165</u> | 0.222 | 0.277 | 0.371 |
| Dlinear | 0.374 | 0.405 | 0.429 | <u>0.440</u> | 0.338 | 0.381 | 0.400 | 0.436 | <u>0.299</u> | <u>0.335</u> | <u>0.369</u> | 0.425 | 0.167 | <u>0.221</u> | 0.274 | 0.368 |
| FITS | 0.375 | <u>0.408</u> | <u>0.429</u> | 0.427 | 0.274 | 0.333 | <u>0.340</u> | 0.374 | 0.305 | 0.339 | 0.367 | <u>0.418</u> | 0.164 | 0.217 | 0.269 | 0.347 |
| IMP | -0.005 | -0.003 | -0.007 | 0.013 | 0 | 0.008 | -0.011 | 0.005 | -0.012 | -0.006 | 0.002 | -0.002 | 0.002 | 0.004 | 0.005 | 0.015 |

Table 3: Long-term forecasting results on three popular datasets in MSE. The best result is highlighted in **bold** and the second best is highlighted with underline. IMP is the improvement between FITS and the second best/ best result, where a larger value indicates a better improvement.

| Dataset | Electricity | | | | Traffic | | | | Weather | | | |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | 96 | 192 | 336 | 720 | 96 | 192 | 336 | 720 | 96 | 192 | 336 | 720 |
| PatchTST | 0.129 | 0.147 | 0.163 | 0.197 | 0.360 | 0.379 | 0.392 | 0.432 | <u>0.149</u> | <u>0.194</u> | <u>0.245</u> | <u>0.314</u> |
| TimesNet | 0.168 | 0.184 | 0.198 | 0.220 | 0.593 | 0.617 | 0.629 | 0.640 | 0.172 | 0.219 | 0.280 | 0.365 |
| FEDFormer | 0.193 | 0.201 | 0.214 | 0.246 | 0.587 | 0.604 | 0.621 | 0.626 | 0.217 | 0.276 | 0.339 | 0.403 |
| FiLM | 0.154 | 0.164 | 0.188 | 0.236 | 0.416 | 0.408 | 0.425 | 0.520 | 0.199 | 0.228 | 0.267 | 0.319 |
| Dlinear | 0.140 | 0.153 | 0.169 | <u>0.203</u> | 0.410 | 0.423 | 0.435 | 0.464 | 0.176 | 0.218 | 0.262 | 0.323 |
| FITS | <u>0.138</u> | <u>0.152</u> | <u>0.166</u> | 0.205 | <u>0.401</u> | <u>0.407</u> | <u>0.420</u> | <u>0.456</u> | 0.145 | 0.188 | 0.236 | 0.308 |
| IMP | -0.009 | -0.005 | -0.003 | -0.008 | -0.041 | -0.028 | -0.028 | -0.024 | 0.004 | 0.006 | 0.009 | 0.006 |

229 300.6M and 241.4M, respectively. Similarly, popular models like Transformer, Informer, Autoformer,
 230 and FEDformer have parameter counts in the range of 13.61M to 20.68M. Even the lightweight yet
 231 state-of-the-art model PatchTST has a parameter count of over 1 million.

232 In contrast, FITS stands out as a highly efficient model
 233 with an impressively low parameter count. With only 4.5K
 234 to 16K parameters, FITS achieves comparable or even
 235 superior performance compared to these larger models.
 236 It is worth highlighting that FITS requires significantly
 237 fewer parameters compared to the next smallest model,
 238 Dlinear, which has 139.7K parameters. For instance, when
 239 considering a 720 look-back window and a 720 forecast-
 240 ing horizon, the Dlinear model requires over 1 million
 241 parameters, whereas FITS achieves similar performance
 242 with only 10k-50k parameters.

243 This analysis showcases the remarkable efficiency of FITS.
 244 Despite its small size, FITS consistently achieves compet-
 245 itive results, making it an attractive option for time series
 246 analysis tasks. FITS demonstrates that achieving state-of-
 247 the-art or close to state-of-the-art performance with a considerably reduced parameter footprint is
 248 possible, making it an ideal choice for resource-constrained environments.

249 Case Study on ETTh2 Dataset

250 We conduct a comprehensive case study on the performance of FITS using the ETTh2 dataset, which
 251 further highlights the impact of the look-back window and cutoff frequency on model performance.
 252 We provide a case study on other datasets in the Appendix. In our experiments, we observe that
 253 increasing the look-back window generally leads to improved performance, while the effect of
 254 increasing the cutoff frequency is minor.

255 Tab. 5 showcases the performance results obtained with different look-back window sizes and cutoff
 256 frequencies. Larger look-back windows tend to yield better performance across the board. On the
 257 other hand, increasing the cutoff frequency only results in marginal performance improvements.
 258 However, it is important to note that higher cutoff frequencies come at the expense of increased
 259 computational resources, as illustrated in Tab. 6.

Table 4: Number of trainable parameters and MACs of TSF models under look-back window=96 and forecasting horizon=720 on the Electricity dataset.

| Model | Parameters | MACs |
|-------------|-----------------|------------------|
| TimesNet | 301.7M | 1226.49G |
| Pyraformer | 241.4M | 0.80G |
| Transformer | 13.61M | 4.03G |
| Informer | 14.38M | 3.93G |
| Autoformer | 14.91M | 4.41G |
| FiLM | 14.91M | 5.97G |
| FEDformer | 20.68M | 4.41G |
| PatchTST | 1.5M | 5.07G |
| DLinear | 139.7K | 40M |
| FITS (Ours) | 4.5K~10K | 1.6M~8.9M |

Table 5: The results on the ETTh2 dataset. Values are visualized with a **green background**, where darker background indicates worse performance. The top-5 best results are highlighted with a **red background**, and the absolute best result is highlighted with **red bold** font. **F** represents supervision on the forecasting task, while **B+F** represents supervision on backcasting and forecasting tasks.

| Horizon | Look-back Window COF/nth Harmonic | 90 | | 180 | | 360 | | 720 | |
|---------|--------------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | | F | B+F | F | B+F | F | B+F | F | B+F |
| 96 | 2 | 0.297687 | 0.296042 | 0.291606 | 0.289387 | 0.278644 | 0.278403 | 0.277708 | 0.27696 |
| | 3 | 0.297796 | 0.297377 | 0.290061 | 0.288239 | 0.277512 | 0.277746 | 0.276537 | 0.277068 |
| | 4 | 0.297106 | 0.295624 | 0.290725 | 0.287993 | 0.27624 | 0.27693 | 0.274207 | 0.274498 |
| | 5 | 0.296168 | 0.296698 | 0.288518 | 0.287375 | 0.276367 | 0.277935 | 0.275989 | 0.275636 |
| 192 | 2 | 0.380163 | 0.379868 | 0.360591 | 0.359769 | 0.336552 | 0.337976 | 0.334854 | 0.335887 |
| | 3 | 0.37983 | 0.381802 | 0.359088 | 0.359498 | 0.336384 | 0.336358 | 0.334666 | 0.335507 |
| | 4 | 0.379657 | 0.380439 | 0.359087 | 0.358536 | 0.334803 | 0.349995 | 0.333522 | 0.333382 |
| | 5 | 0.378556 | 0.379883 | 0.358809 | 0.359376 | 0.335451 | 0.343227 | 0.33384 | 0.335053 |
| 336 | 2 | 0.402706 | 0.404805 | 0.373257 | 0.374678 | 0.344241 | 0.344414 | 0.341869 | 0.342549 |
| | 3 | 0.403238 | 0.404878 | 0.372231 | 0.373948 | 0.345578 | 0.344976 | 0.341436 | 0.342793 |
| | 4 | 0.402702 | 0.407712 | 0.376199 | 0.374435 | 0.343004 | 0.344167 | 0.340795 | 0.342245 |
| | 5 | 0.403484 | 0.409516 | 0.375102 | 0.37462 | 0.344333 | 0.342731 | 0.341043 | 0.342214 |
| 720 | 2 | 0.420072 | 0.424272 | 0.403985 | 0.407392 | 0.379822 | 0.38519 | 0.376871 | 0.37677 |
| | 3 | 0.418323 | 0.420538 | 0.400986 | 0.40686 | 0.379638 | 0.386397 | 0.376236 | 0.376004 |
| | 4 | 0.417485 | 0.420982 | 0.399987 | 0.408128 | 0.379096 | 0.386409 | 0.375865 | 0.375637 |
| | 5 | 0.419122 | 0.420355 | 0.400776 | 0.407871 | 0.378665 | 0.390754 | 0.377138 | 0.374586 |

260 Considering these observations, we find utiliz-
 261 ing a longer look-back window in combination
 262 with a low cutoff frequency to achieve near
 263 state-of-the-art performance with minimal com-
 264 putational cost. For instance, FITS surpasses
 265 other methods when employing a 720 look-back
 266 window and setting the cutoff frequency to the
 267 second harmonic. Remarkably, FITS achieves
 268 state-of-the-art performance with a parameter
 269 count of only around 10k. Moreover, by reduc-
 270 ing the look-back window to 360, FITS already
 271 achieves close-to-state-of-the-art performance
 272 by setting the cutoff frequency to the second
 273 harmonic, resulting in a further reduction of the
 274 model’s parameter count to under 5k (as shown
 275 in Tab. 6).

276 These results emphasize the lightweight nature
 277 of FITS, making it highly suitable for deploy-
 278 ment and training on edge devices with limited
 279 computational resources. By carefully selecting the look-back window and cutoff frequency, FITS can
 280 achieve excellent performance while maintaining computational efficiency, making it an appealing
 281 choice for real-world applications.

282 5 Experiment for Anomaly Detection

283 5.1 Reconstruction as Frequency Interpolation

284 As discussed before, we tackle the anomaly detection tasks in the self-supervised reconstructing
 285 approach. Specifically, we make a N time down-sampling on the input and train a FITS network with
 286 an interpolation rate of $\eta_{Rec} = N$ to up-sample it.

287 5.2 Experiment Settings

288 **Datasets.** We use five commonly used benchmark datasets: SMD (Server Machine Dataset (Su et al.,
 289 2019)), PSM (Polled Server Metrics (Abdulaal et al., 2021)), SWaT (Secure Water Treatment (Mathur
 290 & Tippenhauer, 2016)), MSL (Mars Science Laboratory rover), and SMAP (Soil Moisture Active
 291 Passive satellite) (Hundman et al., 2018).

Table 6: The number of parameters under different settings on ETTh1 & ETTh2 dataset.

| Horizon | COF/nth Harmonic | Look-back Window | | | |
|---------|------------------|------------------|-------|-------|-------|
| | | 90 | 180 | 360 | 720 |
| 96 | 2 | 703 | 1053 | 2279 | 5913 |
| | 3 | 1035 | 1820 | 4307 | 12064 |
| | 4 | 1431 | 2752 | 6975 | 20385 |
| | 5 | 1922 | 3876 | 10374 | 31042 |
| 192 | 2 | 1064 | 1431 | 2752 | 6643 |
| | 3 | 1564 | 2450 | 5192 | 13520 |
| | 4 | 2187 | 3698 | 8475 | 22815 |
| | 5 | 2914 | 5253 | 12558 | 34694 |
| 336 | 2 | 1615 | 1998 | 3483 | 7665 |
| | 3 | 2392 | 3395 | 6608 | 15704 |
| | 4 | 3321 | 5160 | 10725 | 26460 |
| | 5 | 4402 | 7293 | 15834 | 40006 |
| 720 | 2 | 3078 | 3510 | 5418 | 10512 |
| | 3 | 4554 | 5950 | 10266 | 21424 |
| | 4 | 6318 | 9030 | 16650 | 36180 |
| | 5 | 8370 | 12750 | 24570 | 54780 |

292 **Baselines.** We compare FITS with models such as TimesNet (Wu et al., 2023), Anomaly Trans-
 293 former (Xu et al., 2022), THOC (Shen et al., 2020), Omnianomaly (Su et al., 2019). Following
 294 TimesNet (Wu et al., 2023), we also compare the anomaly detection performance with other mod-
 295 els (Zeng et al., 2023; Zhang et al., 2022; Woo et al., 2022; Zhou et al., 2022a).

296 **Evaluation metrics.** Following the previous works (Xu et al., 2022; Shen et al., 2020; Wu et al.,
 297 2023), we use Precision, Recall, and F1-score as metrics.

298 **Implementation details.** We use a window size of 200 and downsample the time series segment by a
 299 factor of 4 to match the original segment during training with the FITS model. Anomaly detection
 300 follows the methodology of the Anomaly Transformer (Xu et al., 2022), where time points exceeding
 301 a certain reconstruction loss threshold are classified as anomalies. The threshold is selected based
 302 on the highest F1 score achieved on the validation set. To handle consecutive abnormal segments,
 303 we adopt a widely-used adjustment strategy (Su et al., 2019; Xu et al., 2018; Shen et al., 2020),
 304 considering all anomalies within a specific successive abnormal segment as correctly detected when
 305 one anomalous time point is identified. This approach aligns with real-world applications, where an
 306 abnormal time point often triggers the attention to the entire segment.

Table 7: Anomaly detection result of F1-scores on 5 datasets. The best result is highlighted in **bold**, and the second best is highlighted with underline. Full results are reported in the Appendix.

| Models | FITS | TimesNet | Anomaly Transformer | THOC | Omni Anomaly | Stationary Transformer | LightTS | Dlinear | IMP |
|--------|--------------|----------|---------------------|--------------|--------------|------------------------|---------|---------|--------|
| SMD | 99.95 | 85.81 | <u>92.33</u> | 84.99 | 85.22 | 84.72 | 82.53 | 77.1 | 7.62 |
| PSM | 93.96 | 97.47 | <u>97.89</u> | 98.54 | 80.83 | 97.29 | 97.15 | 93.55 | -3.93 |
| SWaT | 98.9 | 91.74 | <u>94.07</u> | 85.13 | 82.83 | 79.88 | 93.33 | 87.52 | 4.83 |
| SMAP | 70.74 | 71.52 | 96.69 | <u>90.68</u> | 86.92 | 71.09 | 69.21 | 69.26 | -25.95 |
| MSL | 78.12 | 85.15 | 93.59 | <u>89.69</u> | 87.67 | 77.5 | 78.95 | 84.88 | -15.47 |

307 5.3 Comparisons with SOTAs

308 As shown in Tab. 7, FITS achieves remarkable results on several datasets. Notably, on the SMD and
 309 SWaT datasets, FITS exhibits exceptional performance with F1-scores almost reaching perfection
 310 at around 99.95% and 98.9%, respectively. This demonstrates FITS’ ability to accurately detect
 311 anomalies and classify them correctly. In comparison, other models, such as TimesNet, Anomaly
 312 Transformer, and Stationary Transformer, struggle to match FITS’ performance on these datasets.

313 However, FITS shows comparatively lower performance on the SMAP and MSL datasets. These
 314 datasets present a challenge due to their binary event data nature, which may not be effectively
 315 captured by FITS’ frequency domain representation. While models specifically designed for anomaly
 316 detection, such as THOC and Omni Anomaly, achieve higher F1-scores on these datasets.

317 For a more comprehensive evaluation, waveform visualizations and detailed analysis can be found
 318 in the appendix, providing deeper insights into FITS’ strengths and limitations in different anomaly
 319 detection scenarios. It is important to note that the reported results are achieved with a parameter
 320 range of 1-4K and MACs (Multiply-Accumulate Operations) of 10-137K, which will be further
 321 detailed in the appendix.

322 6 Conclusions and Discussion

323 In this paper, we propose FITS for time series analysis, a low-cost model with 10^k parameters that can
 324 achieve performance comparable to state-of-the-art models that are often several orders of magnitude
 325 larger. As a frequency-domain modeling technique, FITS has difficulty handling binary-valued time
 326 series and time series with missing data. For the former category, time-domain modeling is preferable
 327 as the raw data format is sufficiently compact. For the latter category, we could first employ simple
 328 yet effective time-domain imputation techniques and then apply FITS for efficient analysis.

329 References

- 330 Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lincewicz. Practical approach to asynchronous
331 multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM*
332 *SIGKDD Conference on Knowledge Discovery; Data Mining*, KDD '21, pp. 2485–2494, New
333 York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi:
334 10.1145/3447548.3467174. URL <https://doi.org/10.1145/3447548.3467174>.
- 335 E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 4(12):63–70, 1967.
336 doi: 10.1109/MSPEC.1967.5217220.
- 337 Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soder-
338 strom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic threshold-
339 ing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Dis-*
340 *covery & Data Mining*. ACM, jul 2018. doi: 10.1145/3219819.3219845. URL <https://doi.org/10.1145/3219819.3219845>.
- 342 Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Re-
343 versible instance normalization for accurate time-series forecasting against distribution shift. In
344 *International Conference on Learning Representations, 2022*. URL [https://openreview.net/](https://openreview.net/forum?id=cGDAkQo1C0p)
345 [forum?id=cGDAkQo1C0p](https://openreview.net/forum?id=cGDAkQo1C0p).
- 346 James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with
347 fourier transforms, 2022.
- 348 Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet:
349 Time series modeling and forecasting with sample convolution and interaction. In *Advances in*
350 *Neural Information Processing Systems, 2022*.
- 351 Aditya P. Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training
352 on ics security. In *2016 International Workshop on Cyber-physical Systems for Smart Water*
353 *Networks (CySWater)*, pp. 31–36, 2016. doi: 10.1109/CySWater.2016.7469060.
- 354 Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth
355 64 words: Long-term forecasting with transformers. In *International Conference on Learning*
356 *Representations, 2023*.
- 357 Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hier-
358 archical one-class network. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin
359 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13016–13026. Cur-
360 ran Associates, Inc., 2020. URL [https://proceedings.neurips.cc/paper_files/paper/](https://proceedings.neurips.cc/paper_files/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf)
361 [2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf).
- 362 Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection
363 for multivariate time series through stochastic recurrent neural network. In *Proceedings of the*
364 *25th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD
365 '19, pp. 2828–2837, New York, NY, USA, 2019. Association for Computing Machinery. ISBN
366 9781450362016. doi: 10.1145/3292500.3330672. URL [https://doi.org/10.1145/3292500.](https://doi.org/10.1145/3292500.3330672)
367 [3330672](https://doi.org/10.1145/3292500.3330672).
- 368 Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential
369 smoothing transformers for time-series forecasting, 2022.
- 370 Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers
371 with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing*
372 *Systems*, 34:22419–22430, 2021.
- 373 Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet:
374 Temporal 2d-variation modeling for general time series analysis. In *International Conference on*
375 *Learning Representations, 2023*.

- 376 Haowen Xu, Yang Feng, Jie Chen, Zhaogang Wang, Honglin Qiao, Wenxiao Chen, Nengwen Zhao,
377 Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, and Dan Pei. Unsupervised anomaly
378 detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of*
379 *the 2018 World Wide Web Conference on World Wide Web - WWW '18*. ACM Press, 2018. doi:
380 10.1145/3178876.3185996. URL <https://doi.org/10.1145/2F3178876.3185996>.
- 381 Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series
382 anomaly detection with association discrepancy, 2022.
- 383 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series
384 forecasting? *arXiv preprint arXiv:2205.13504*, 2022.
- 385 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series
386 forecasting? 2023.
- 387 Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is
388 more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv*
389 *preprint arXiv:2207.01186*, 2022.
- 390 Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang.
391 Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings*
392 *of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.
- 393 Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency
394 enhanced decomposed transformer for long-term series forecasting. In *International Conference*
395 *on Machine Learning*, 2022a.
- 396 Tian Zhou, Ziqing Ma, xue wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong
397 Jin. FiLM: Frequency improved legendre memory model for long-term time series forecasting.
398 In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in*
399 *Neural Information Processing Systems*, 2022b. URL <https://openreview.net/forum?id=zTQdHSQUQWc>.
400