ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11462613

FILTER THEN ATTEND: IMPROVING ATTENTION-BASED TIME SERIES FORECASTING WITH SPECTRAL FILTERING

Elisha Dayag, Nhat Thanh Tran, Jack Xin*

University of California, Irvine
Department of Mathematics
Irvine, CA 92697, United States

ABSTRACT

Transformer-based models are at the forefront in long time-series forecasting (LTSF). Despite their powerful modeling capacity, they are hindered in this domain by a bias toward high-energy, low-frequency features in the data. Recent work has established that learnable frequency filters can strengthen deep forecasting models by enhancing their spectral utilization. These works choose to use multilayer perceptrons to process their filtered signals and thus do not address the issues found with transformer-based models. In this paper, we demonstrate that applying learnable frequency filters to embedded signals enhance the performance of transformer-based forecasters across a number of architectures with negligible increases in memory and compute. We also conduct synthetic experiments that demonstrate learnable filters enhance transformer-based models by amplifying primarily middle and high-frequency features in the data.

***Index Terms*—** Deep Learning, Time Series Forecasting

1. INTRODUCTION

Time series forecasting is a critical task in domains such as energy usage [1], traffic prediction [2] and financial markets [3]. Given the resounding success of deep learning in the adjacent fields of computer vision and natural language processing, there has been much interest in applying deep learning to time series forecasting. The two most popular model paradigms for this task are multilayer perceptron (MLP) and transformer-based [4] methods. Naive applications of the transformer to forecasting have been deemed ineffective [5] in comparison to even simple linear models; adapting the transformer to be amenable to time series forecasting has required a merger with the techniques of classical signal processing and forecasting. Many of these mergers incorporate information from the frequency domain to help in forecasting [6] [7]. In this paper, we present a simple merger of transformer-based models with the well-established signal processing technique of frequency filtering. Despite being ubiquitous in classical signal processing, the use of learned frequency filters for deep learning based forecasting is nascent [8] [9]. Prior works have only investigated the use of learnable filters with MLP-based architectures. In this work we demonstrate across a variety of transformer-based models that adding a learned frequency filter into the embedding module enhances forecasting performance, especially on larger datasets with longer prediction horizons.

Our contributions can be summarized as follows:

- We conduct experiments on a variety of real-world datasets spanning different domains and characteristics across three different forms of attention and find that adding learnable filters provides significant improvements in forecasting performance at little added cost computationally.
- Via synthetic experiments, we provide evidence that filters improve the performance of transformer-based models by amplifying mid and high-frequency components, countering the low-pass filtering nature of transformers.

2. RELATED WORKS

In recent years a variety of transformer based approaches for forecasting have been proposed [10]. We specifically mention three models which form the baselines for our study and represent three distinct approaches to the inclusion of attention in time series forecasting.

PatchTST [11] segments time series into subseries-level patches which are used as tokens for the attention module. Additionally, this work utilizes channel-independence, meaning each token only contains information from a single channel/variate. On the other hand, iTransformer [12] embeds the raw time series of individual variates as tokens then applies the self-attention and feed-forward network on the inverted dimensions to obtain multivariate correlations and nonlinear representations of the data, respectively. Leddam [13] utilizes

Corresponding Author

The work was partially supported by NSF grants DMS-2151235, DMS-2219904, and a Qualcomm Gift Award

a dual attention module consisting of channel-wise self attention and autoregressive self-attention as well as a learnable convolution kernel to decompose the time series into seasonal and trend components.

Spectral analysis has a long history in time series forecasting [14]. Recent work has tried to combine spectral analysis with deep learning for improved forecasting. Fredformer [15] applies patching and channel wise attention in the frequency domain combined with frequency normalization techniques to force the transformer encoder to utilize low energy frequency features in the data. This is done with the intent of mitigating frequency bias in the transformer. FITS [16] explores the ability of frequency learning to develop a compact baseline for time series forecasting. In this work, the authors take the DFT of the signal, apply a low pass filter to it, then apply a learnable (complex) linear mapping in the frequency domain before recovering the signal using the IDFT. Part of the innovation of FITS is applying a frequency filter to the data before the learnable mappings. Rather than using a handcrafted filter, FilterNet [8] randomly initializes a learnable filter then performs multiplication with the input signal in the frequency domain. After converting back to a time signal using the IDFT, the filtered signal is processed using a MLP.

We believe that these works have demonstrated the efficacy of frequency learning as a technique in time series forecasting. Unlike some of these methods, we believe that the power of frequency filters is best utilized as a building block enabling the transformer.

3. METHODS

In our experiments, we are given a historical time series with lookback length L, where each timestamp has D variates: $X = \{X_1, \ldots, X_L\} \in \mathbb{R}^{D \times L}$. Our problem is to predict the next H timesteps, denoted by $\{X_{L+1}, \ldots, X_{L+H}\} \in \mathbb{R}^{D \times H}$.

3.1. Architecture

In our work, we attach learnable frequency filters to the architectures of PatchTST, iTransformer and Leddam giving us FilterFormer, iFilterFormer, and FilterLeddam respectively. Due to spatial constraints we omit the description of their respective attention modules, but we describe the embedding and frequency filtering modules that they all share in common below.

3.1.1. Embedding and Normalization

We begin by performing Reversible Instance Normalization [17] on our signal X. This normalization method helps our models overcome non-stationarity in the our dataset and can be formulated as

$$\text{Norm}(X) = \frac{X - \text{Mean}_L(X)}{\text{Std}_L(X)}, \tag{1}$$

where Mean_L and Std_L represent calculating the mean and standard deviation along the time dimension. is calculated along the time dimension for each channel. After obtaining our prediction P we apply inverse instance normalization, formulated as

$$\text{InverseNorm}(P) = P \times \text{Std}_L(X) + \text{Mean}_L(X), \tag{2}$$

to obtain our final prediction. After normalizing our signal we embed some representation of it (e.g. patches in FilterFormer, the raw time series in our other architectures) using a learnable linear layer then adding a positional encoding to obtain a latent space representation $Y = (XW + b) + Pos \in \mathbb{R}^{D \times N}$, where N is our embedding dimension.

3.1.2. Spectral Block

After performing batch normalization [18] on our embedded signals we pass them through our spectral filtering network. In this paper we consider a frequency filter P to be the Fourier transform of a fixed signal $p \in \mathbb{R}^n$. To realize our filter, we take the rFFT of an initially random set of real parameters which we denote by W. We then perform pointwise multiplication in the frequency domain before reverting our filtered signal back into the time domain using the IDFT. Because both our signal and the time representation of our filter are real-valued, we can be assured that their product is also the DFT of a real-valued signal. Thus, when we take the IDFT of our product, we are sure to obtain a real-valued signal again. The total operation of our spectral block can be written as:

$$\mathcal{F}^{-1}(\mathcal{F}(W) \cdot \mathcal{F}(Y)) = \mathcal{F}^{-1}(\mathcal{F}(W * Y)) \tag{3}$$

where * denotes circular convolution, and $\mathcal{F}$ denotes the DFT. During training, our filter learns which frequencies to amplify or attenuate in the embedded signal to help the following modules. In practice, our spectral block has $1000 - 2000$ parameters, which is negligible in comparison to our transformer backbones. After filtering our signal using the spectral network, we perform instance normalization once again before passing it along to our transformer for further processing.

4. EXPERIMENTS

We evaluate our models on nine popular datasets, namely the four ETT datasets (ETTm1, ETTm2, ETTh1, and ETTh2), Exchange-rate, Weather, Electricity, Solar-Energy, and Traffic. These datasets and their characteristics can be found at https://github.com/laiguokun/multivariate-time-series-data and https://github.com/zhouhaoyi/ETDataset/tree/main. We preprocess datasets using the scikit-learn StandardScaler. We split datasets into training, validation, and test sets in a 7:2:1 ratio and use the Adam optimizer with learning rate in $\{1e-3, 3e-4, 1e-4\}$ for 50 epochs with early stopping after 30 epochs without improved validation performance.

4.1. Baselines

We compare our models to iTransformer, PatchTST, and Leddam, the models used as a base to construct our FilterFormers and Filternet, which provides the baseline of using a learned frequency filter followed by a MLP (i.e. no self-attention). For iTransformer and Leddam we use the results and configurations of their respective authors, setting the lookback length to 96. For PatchTST and the corresponding FilterFormer, we set the lookback length to 336 on the traffic dataset to match the configuration of the original authors. In table 1 we present the multivariate forecasting results averaged over the prediction horizons $H \in \{96, 192, 336, 720\}$.

4.2. Implementation Details

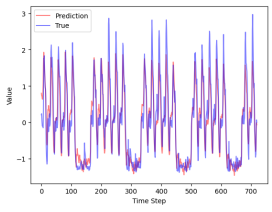
Our model is implemented with Pytorch 1.12 and all experiments were conducted on a cluster of four NVIDIA RTX 2080 Ti's. We use mean square error (MSE) as our loss function and report MSE as our evaluation metric. For baselines, we either use their reported numbers as stated or run the relevant training scripts found on their respective codebases.

4.3. Model Analysis

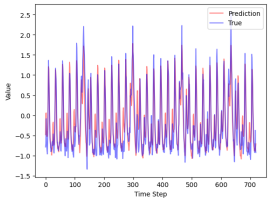
From table 1, we see that across a variety of datasets, our FilterFormers outperform most of our baseline models. The performance of the filtered model and the base model is highly correlated; filters are not a panacea and are ultimately limited by the forecasting power of the backbone they are attached to. Comparing all of our filtered models to Filternet (a data-driven filter followed by a multilayer perceptron) we find that while in general our models outperform this baseline, this improvement is most apparent on the larger ECL and Solar-Energy datasets. This implies that on complex datasets, forecasting models with filters need a larger capacity than a simple MLP, which for our models comes in various implementations of self-attention.

4.3.1. Visualization of Predictions

In Figure 1 we present some visualizations of our models on the electricity dataset. With a learned filter, these models are able to effectively capture high frequency patterns in the data over large prediction horizons, leading to the improvements in forecasting performance.



(a) FilterFormer

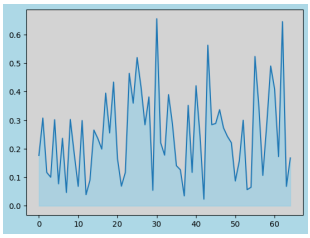


(b) iFilterFormer

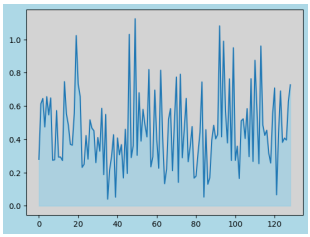
Fig. 1: Visualizations of our models' predictions on the input 96, prediction 720 task of the ECL dataset.

4.3.2. The Effect of Frequency Filters

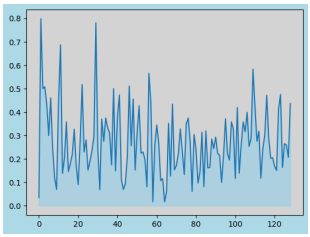
We examine the frequency filters learned by our three models on the electricity dataset for a prediction horizon of 720.



(a) FilterFormer



(b) iFilterFormer



(c) FilterLeddam

Fig. 2: Amplitude spectrums of the filters of our three models on the input 96, prediction horizon 720 task on the Electricity dataset.

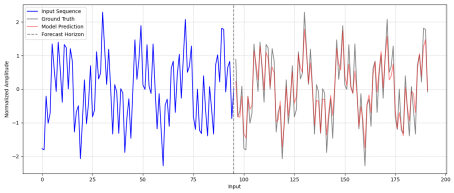
There is not a one-to-one correlation between the filters learned by each model. This is in part because our filters are applied to embedded versions of the time series. Thus, if each model settles on a different embedding of the dataset, we would expect different needs from their respective filters. We do note that FilterFormer and iFilterFormer, which are both based upon fairly "traditional" implementations of self-attention, do have a bias toward middle and high frequency components. More specifically, the most attentuated frequencies occur in the upper two-thirds of the spectrum under consideration by our filter. This is not the case in FilterLeddam, which utilizes different implementations of attention.

To gather insight into how frequency filters enable our forecasting models when we are applying them to embedded

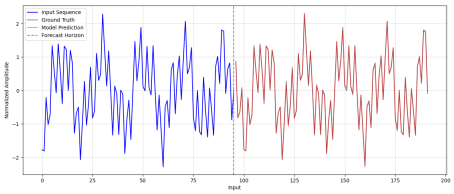
	FilterFormer	PatchTST	iFilterFormer	iTransformer	FilterLeddam	Leddam	Filternet
ETTm1	**0.378**	0.387	0.405	0.407	0.386	0.386	0.383
ETTm2	**0.275**	0.281	0.285	0.288	0.276	0.281	0.276
ETTh1	**0.428**	0.469	0.452	0.454	0.436	0.431	0.436
ETTh2	**0.374**	0.384	0.372	0.383	0.371	0.373	0.377
ECL	0.180	0.208	0.173	0.178	**0.168**	0.169	0.202
Exchange	0.358	0.367	**0.327**	0.360	0.341	0.354	0.361
Traffic	**0.396**	**0.396**	0.424	0.428	0.448	0.467	0.521
Weather	0.244	0.259	0.256	0.258	**0.241**	0.242	0.247
Solar-Energy	0.238	0.270	0.232	0.233	**0.229**	0.230	0.336

Table 1: Average MSE over all prediction lengths on our baseline datasets. Bold indicates best value.

signals, we conduct an experiment on a synthetic signal comprised of a low-frequency, mid-frequency, and high-frequency sine wave. In [8] it was noted that the base iTransformer struggles with signals of this nature. This difficulty is often chalked up to the frequency bias of transformers [15]: theoretical work suggests that self-attention acts as a low-pass filter [19]. Thus, we would expect a transformer-based model to attenuate the high and middle frequencies of its signal as it passes through the attention blocks (we refer to this effect as oversmoothing). We test both iTransformer and our iFilterFormer on our synthetic signal and plot the results in Figure 3. As expected, the model with filter is able to capture this signal more effectively. To examine the impact of our filter, we analyze the input signal in embedding space before and after the filter is applied to it. As we can see in Figure 4, while the filter amplifies many of the frequencies found in the original embedding, the most disproportionally amplified frequencies are those in the mid-high range. In this regard, the filter operates close to a high-pass filter. It follows that filtering can counteract the oversmoothing effect of self-attention, leading to better predictions.



(a) MSE = 1.0 e-1



(b) MSE = 2.6e-4

Fig. 3: Performance of iTransformer on a simple synthetic signal with low, mid, and high frequencies a) without filter and b) with filter.

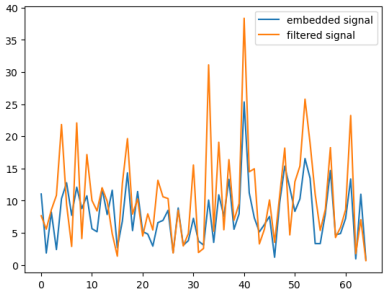


Fig. 4: The amplitude spectrum of the embedded signal in our synthetic experiment along with the amplitude spectrum of the embedding after we apply our spectral block.

5. CONCLUSIONS

In this paper, we demonstrated that adding a learnable filter improves the ability of transformer-based LTSF models. By incorporating the most basic implementation of a learnable filter into popular transformer LTSF models, we were able to improve forecasting performance with negligible increases in computational and memory requirements. We validated this effectiveness by testing our models on datasets spanning a variety of domains and complexities. We thus recommend adding a learned filter as a basic configuration for transformer-based forecasting models. It will rarely degrade performance, but it has the potential to drastically improve it. We intentionally stuck to a simple type of filter to demonstrate the efficacy of the method. We believe that further improvements can be made by developing more advanced data-driven filters. Along this line, in the future we will work develop to data-driven filters that are easier to interpret and with more expressivity and adaptivity. Overall, we hope that this work leads to further use of filters in conjunction with transformers in the field of long-term forecasting.

6. REFERENCES

[1] Neda Maleki, Oxana Lundström, Arslan Musaddiq, John Jeansson, Tobias Olsson, and Fredrik Ahlgren,

"Future energy insights: Time-series and deep learning models for city load forecasting," *Applied Energy*, vol. 374, pp. 124067, 2024.

[2] Bas Van Der Bijl, Bart Gijsbertsen, Stan Van Loon, Yorran Reurich, Tom De Valk, Thomas Koch, and Elenna Dugundji, "A comparison of approaches for the time series forecasting of motorway traffic flow rate at hourly and daily aggregation levels," *Procedia Computer Science*, vol. 201, pp. 213–222, 2022.

[3] Kelum Gajamannage, Yonggi Park, and Dilhani I Jayathilake, "Real-time forecasting of time series in financial markets using sequentially trained dual-lstms," *Expert Systems with Applications*, vol. 223, pp. 119879, 2023.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, 2023, vol. 37, pp. 11121–11128.

[6] Kun Yi, Qi Zhang, Longbing Cao, Shoujin Wang, Guodong Long, Liang Hu, Hui He, Zhendong Niu, Wei Fan, and Hui Xiong, "A survey on deep learning based time series analysis with frequency transformation," *arXiv preprint arXiv:2302.02173*, 2023.

[7] Nhat Thanh Tran and Jack Xin, "Fourier-mixed window attention for efficient and robust long sequence time-series forecasting," *Frontiers in Applied Mathematics and Statistics*, vol. 11, pp. 1600136, 2025.

[8] Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan, "FilterNet: Harnessing frequency filters for time series forecasting," *Advances in Neural Information Processing Systems*, vol. 37, pp. 55115–55140, 2024.

[9] Yunling Zheng, Zeyi Xu, Fanghui Xue, Biao Yang, Jiancheng Lyu, Shuai Zhang, Yingyong Qi, and Jack Xin, "Afidaf: Alternating fourier and image domain adaptive filters as an efficient alternative to attention in vits," in *International Symposium on Visual Computing*. Springer, 2024, pp. 17–30.

[10] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.

[11] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[12] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[13] Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang, "Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling," *arXiv preprint arXiv:2402.12694*, 2024.

[14] Robert H Shumway, David S Stoffer, and David S Stoffer, *Time series analysis and its applications*, vol. 3, Springer, 2000.

[15] Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai, "Fredformer: Frequency debiased transformer for time series forecasting," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2400–2410.

[16] Zhijian Xu, Ailing Zeng, and Qiang Xu, "FITS: Modeling time series with \$10k\$ parameters," in *The Twelfth International Conference on Learning Representations*, 2024.

[17] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International conference on learning representations*, 2021.

[18] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[19] Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park, "An attentive inductive bias for sequential recommendation beyond the self-attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, vol. 38, pp. 8984–8992.