First De-Trend then Attend: Rethinking Attention for Time-Series Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer-based models have gained large popularity and demonstrated promising results in long-term time-series forecasting in recent years. In addition to learning attention in time domain, recent works also explore learning attention in frequency domains (e.g., Fourier domain, wavelet domain), given that seasonal patterns can be better captured in these domains. In this work, we seek to understand the relationships between attention models in different time and frequency domains. Theoretically, we show that attention models in different domains are equivalent under linear conditions (i.e., linear kernel to attention scores). Empirically, we analyze how attention models of different domains show different behaviors through various synthetic experiments with seasonality, trend and noises, with emphasis on the role of softmax operation therein. Both these theoretical and empirical analyses motivate us to propose a new method: TDformer (Trend Decomposition Transformer), that first applies seasonal-trend decomposition, and then additively combines an MLP which predicts the trend component with Fourier attention which predicts the seasonal component to obtain the final prediction. Extensive experiments on benchmark time-series forecasting datasets demonstrate that TDformer achieves state-of-the-art performance against existing attention-based models.

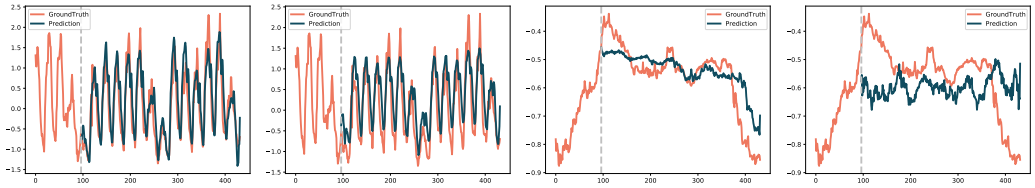
1 Introduction

Transformer [18] recently gains wide popularity in time-series forecasting, inspired by its success in natural language processing and its ability to capture long-range dependencies [19]. Apart from the vanilla Transformer that calculates attention in time domain, recently variants of Transformer which calculate attention in frequency domains (e.g., Fourier domain or wavelet domain) (Figure 2) [22, 21, 24, 20, 14] have also been proposed to better model global characteristics of time series.

Despite the progress made by Transformer-based methods for time series forecasting, there lacks a rule of thumb to select the domain in which attention is best learned. Our work is driven by better understanding the following research question: *Does learning attention in one domain offer better representation ability than the other? If so, how?* We show mathematically that under linear conditions, learning attention in time or frequency domains leads to equivalent representation power. We then show that due to the softmax non-linearity used for normalization, this theoretical linear equivalence does not hold empirically. In particular, attention models in different domains demonstrate different empirical advantages. This finding sheds light on how to best apply attention models under different practical scenarios. We propose TDformer based on these insights and demonstrate that we achieve state-of-the-art performance against current attention-based models.

More specifically, we find that (1) for *data with strong seasonality*, frequency-domain attention models are more *sample-efficient* compared with time-domain attention models, as softmax with exponential terms correctly amplify the dominant frequency modes in Fourier space. (2) For *data with trend*,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) ECL TDformer (b) ECL FEDformer (c) Weather TDformer (d) Weather FEDformer

Figure 1: Prediction comparison between TDformer and FEDformer on electricity dataset ((a),(b)) and weather dataset ((c), (d)). We predict the future 336 steps given context 96 steps (the gray dash line). Orange line represents the ground truth, and blue line represents the prediction.

attention models generally show inferior *generalizability*, as attention models by nature interpolate rather than extrapolate the context. This finding of difference in performances of attention models on various types of time series data emphasizes the importance of seasonal-trend decomposition module in the attention model framework. (3) For *data with noisy spikes*, frequency-domain attention models are more *robust* to such spiky data, as large-value spikes in the time domain correspond to small-amplitude high-frequency modes, whose attention would be filtered out by softmax operations.

Due to the different performances of the various attention modules on data with seasonality and trend, we propose TDformer that first decomposes the context time series into trend and seasonal components. We use a MLP for predicting the future trend, Fourier attention to predict the future seasonal part, and add these two components to obtain the final prediction. Extensive experiments on benchmark forecasting datasets demonstrate the effectiveness of our proposed approach. As a motivating example, we visualize predictions of TDformer and one of the best performing baselines FEDformer in Figure 1. On data with strong seasonality (Figure 1a and Figure 1b) TDformer preserves both the seasonality and trend of the original data, while FEDformer [24] deviates from the trend of the ground truth. On data with strong trend (Figure 1c and Figure 1d), TDformer generates predictions that better follow the trend of the original data.

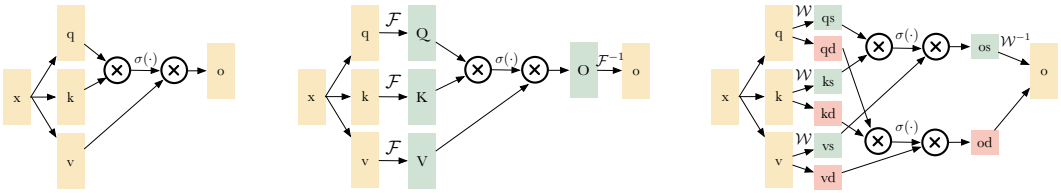
In summary, our contributions are:

- We theoretically show that under linear conditions, attention models in time domain, Fourier domain and wavelet domain have the same representation power;
- We empirically analyze attention models in different domains with synthetic data of different characteristics, given the non-linearity of softmax. We show that frequency-domain attention performs the best on data with seasonality, and attention models in general have inferior generalizability on trend data, which motivates the design of a hybrid model based on seasonal trend decomposition;
- We propose TDformer that separately models the trend with MLP and seasonality with Fourier attention, and shows state-of-the-art performance against current attention models on time-series forecasting benchmarks.

2 Related Work

Time-Domain Attention Forecasting Models. Informer [22] proposes efficient ProbSparse self-attention mechanism. Autoformer [21] renovates time-series decomposition as a basic inner block and designs Auto-Correlation mechanism for dependencies discovery. Non-stationary Transformer [14] proposes Series Stationarization and De-stationary Attention to address over-stationarization.

Frequency-Domain Attention Forecasting Models. FEDformer [24] proposes Fourier and wavelet enhanced blocks based on Multiwavelet-based Neural Operator Learning [4] to capture important structures in time series through frequency domain mapping. ETSformer [20] selects top-K largest amplitude modes as frequency attention and combines with exponential smoothing attention. Adaptive Fourier Neural Operator (AFNO) [3] builds upon FNO [13] and proposes an efficient token mixer that learns to mix in the Fourier domain. FNet [11] replaces the self-attention with Fourier Transform and promotes efficiency without much loss of accuracy on NLP benchmarks. T-WaveNet [15] constructs a tree-structured network with each node built with invertible neural network (INN) based wavelet transform unit for iterative decomposition. Adaptive Wavelet Transformer Network (AWT-Net) [5]



(a) (b) (c)

Figure 2: Comparison between (a) time attention, (b) Fourier attention and (c) wavelet attention. For simplicity, we only draw one layer of multiwavelet decomposition/reconstruction, and similar analysis follows for multiple layers. See precise notations in Section 3.

generates wavelet coefficients to classify each point into high or low sub-bands components and exploits Transformer to enhance the original shape features.

Decomposition-Based Forecasting Models decompose time series into trend and seasonality (with i.e., STL decomposition [2]). Apart from attention-based Autoformer and FEDformer, N-BEATS [16] models trend with small-degree polynomials and seasonality with Fourier series. N-HiTS [1] redefines N-BEATS by enhancing its input decomposition via multi-rate data sampling and its output synthesizer via multi-scale interpolation. FreDo [17] incorporates frequency-domain features into AverageTile model that averages history sub-series. FiLM [23] applies Legendre Polynomials projections to approximate historical information and Fourier projection to remove noise. DeepFS [6] encodes temporal patterns with self-attention and predicts Fourier series parameters and trend with MLP.

Despite the success of attention models in time, Fourier, and wavelet domains, there is still a lack of notion for understanding their relationships and respective advantages. Decomposition-based methods also adopt decomposition layers without giving strong reasoning for their necessity. We propose to fill this gap from both theoretical and empirical perspectives, and based on these analysis build a new framework that shows better forecasting performance.

3 Linear Equivalence of Attention in Various Domains

3.1 Formulation of Attention Models

We first briefly introduce the canonical Transformers. Denote input queries, keys and values as $\mathbf{q} \in \mathbb{R}^{L\times D}, \mathbf{k} \in \mathbb{R}^{L\times D}, \mathbf{v} \in \mathbb{R}^{L\times D}$, which are transformed from input $\mathbf{x}$ through linear embeddings. Denote output of attention module as $\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) \in \mathbb{R}^{L\times D}$. As shown in Figure 2 (a), The attention operation in canonical attention is formulated as

$$\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \sigma\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d_q}}\right)\mathbf{v}, \tag{1}$$

where d_q is the dimension for queries that serves as normalization term in attention operation, and $\sigma(\cdot)$ represents activation function. When $\sigma(\cdot) = \mathrm{softmax}(\cdot)$[1], we have *softmax attention*: $\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathrm{softmax}\left(\mathbf{q}\mathbf{k}^T/\sqrt{d_q}\right)\mathbf{v}$. When $\sigma(\cdot) = \mathrm{Id}(\cdot)$ (identity mapping), we have *linear attention*: $\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathbf{q}\mathbf{k}^T\mathbf{v}$ (we ignore the normalization term $\sqrt{d_q}$ for simplicity).

Definition 3.1 (Time Attention)**.** Equation 1 refers to time domain attention where $\mathbf{q}, \mathbf{k}, \mathbf{v}$ are all in original time domain, shown in Figure 2 (a).

Definition 3.2 (Fourier Attention)**.** Fourier attention first converts queries, keys, and values with Fourier Transform, performs a similar attention mechanism in the frequency domain, and finally converts the results back to the time domain using inverse Fourier transform, shown in Figure 2 (b). Let $\mathcal{F}(\cdot), \mathcal{F}^{-1}(\cdot)$ denote Fourier transform and inverse Fourier transform, then Fourier attention is $\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathcal{F}^{-1}\Big(\sigma\big(\mathcal{F}(\mathbf{q})\overline{\mathcal{F}(\mathbf{k})}^T/\sqrt{d_q}\big)\mathcal{F}(\mathbf{v})\Big)$.

Definition 3.3 (Wavelet Attention)**.** Wavelet transform applies wavelet decomposition and reconstruction to obtain signals of different scales. Wavelet attention performs attention calculation to

[1] $\mathrm{softmax}(\mathbf{x}) = \frac{e^{x_i}}{\sum_i e^{x_i}}$

decomposed queries, keys, and values in each scale, and reconstructs the output from attention results in each scale, illustrated in Figure 2 (c). Let $\mathcal{W}(\cdot), \mathcal{W}^{-1}(\cdot)$ denote wavelet decomposition and wavelet reconstruction, then wavelet attention is $\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathcal{W}^{-1}\Big(\sigma\left(\mathcal{W}(\mathbf{q})\mathcal{W}(\mathbf{k}^T)/\sqrt{d_q}\right)\mathcal{W}(\mathbf{v})\Big)$.

3.2 Linear Equivalence of Time, Fourier and Wavelet Attention

In this section we formally prove that time, Fourier and wavelet attention models are equivalent under linear attention case.

Lemma 3.1. When $\sigma(\cdot) = \mathrm{Id}(\cdot)$ (linear attention), time, Fourier and wavelet attention are equivalent.

Proof. Let $\mathbf{W} = (\frac{\omega^{jk}}{\sqrt{L}}) \in \mathbb{C}^{L\times L}, \omega = e^{-\frac{2\pi j}{L}}$ denote the Fourier matrix, then Fourier transform to signal $\mathbf{x} \in \mathbb{R}^{L\times D}$ can be expressed as $\mathbf{X} = \mathbf{W}\mathbf{x}, \mathbf{X} \in \mathbb{C}^{L\times D}$, and inverse Fourier transform can be expressed as $\mathbf{x} = \mathbf{W}^H\mathbf{X}$, where $\mathbf{W}^H$ is the Hermitian (conjugate transpose) of $\mathbf{W}$. Given properties of Fourier matrix, we could easily show that

$$\mathbf{W}^{-1} = \mathbf{W}^H, \mathbf{W}^T = \mathbf{W}. \tag{2}$$

Following this expression, Fourier domain linear attention can be written as

$$\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathbf{W}^H[(\mathbf{W}\mathbf{q})\overline{(\mathbf{W}\mathbf{k})}^T(\mathbf{W}\mathbf{v})] = \mathbf{q}\mathbf{k}^T\mathbf{v}. \tag{3}$$

Therefore, calculating attention in Fourier domain is equivalent to time-domain attention.

For wavelet attention, we take single-scale wavelet decomposition and reconstruction as an example, and multi-scale wavelet transform follows the same analysis. Using the same notation, let $\mathbf{W} \in \mathbb{R}^{L\times\frac{L}{2}}, \mathbf{W}^{-1} \in \mathbb{R}^{\frac{L}{2}\times L}$ denote the wavelet decomposition and reconstruction matrix, then wavelet decomposition to signal $\mathbf{x} \in \mathbb{R}^{L\times D}$ can be expressed as $\mathbf{X} = \mathbf{W}\mathbf{x}, \mathbf{X} \in \mathbb{R}^{\frac{L}{2}\times D}$, and wavelet reconstruction can be expressed as $\mathbf{x} = \mathbf{W}^{-1}\mathbf{X}$. Since wavelet matrix is orthogonal, we have the property that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. Wavelet linear attention is

$$\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathbf{W}^{-1}[(\mathbf{W}\mathbf{q})(\mathbf{W}\mathbf{k})^T(\mathbf{W}\mathbf{v})] = \mathbf{q}\mathbf{k}^T\mathbf{v}, \tag{4}$$

which is again equivalent to time-domain attention. Therefore, we show that mathematically, time, Fourier and wavelet attention models are equivalent given linear assumptions. □

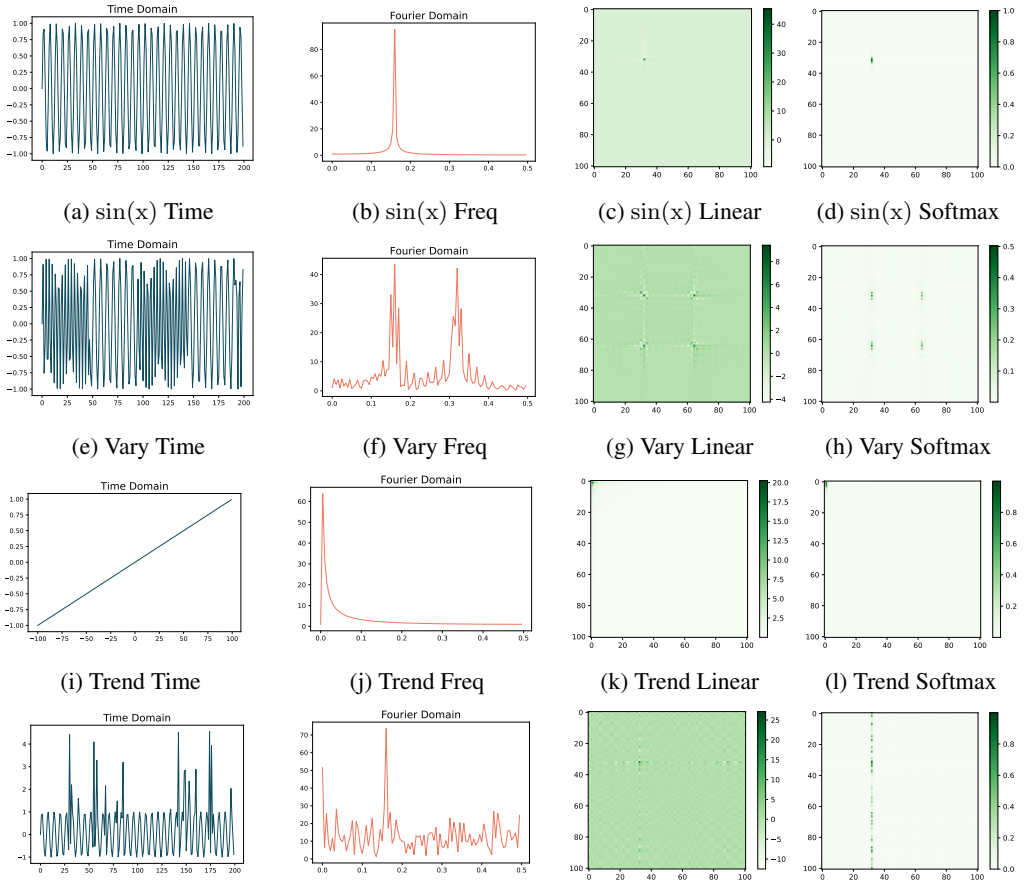
4 Investigation on the Role of Softmax

Although these attention models are equivalent given linear assumptions, in practice we apply softmax as normalization, which changes the behavior of different attention models. In this section, we empirically analyze how softmax causes such performance gaps on datasets with three different representative properties: seasonality, trend and noises. For all experiments in this section, the task is to predict the next 96 time steps given history 96 time steps. We implement the wavelet-domain attention model based on multiwavelet transform model [4].

4.1 Data with Seasonality

For data with fixed seasonality, Fourier attention is the most sample-efficient. We use $\sin(\mathrm{x})$ as an example of seasonal data (visualized in Figure 3a and Figure 3b). There exist dominant frequency modes for data with seasonality. We visualize linear attention (Figure 3c) and softmax attention (Figure 3d) in Fourier space. Attention scores are concentrated on the dominant frequency mode. As softmax with exponential terms has the "polarization" effect (increasing the gap between large and small values), softmax attention further concentrates the scores on the dominant frequency, helping the model to better capture seasonal information. Therefore, we find that frequency-domain attention models are capable of quickly recognizing the dominant frequency modes (more sample efficient) compared with time-domain models (Figure 4a).

To further illustrate such polarization effect, we also compare softmax attention with polynomial kernels $\sigma(x) = x_i^d/\sum_i x_i^d$, where d is the degree of polynomials (without loss of generality we assume $x_i > 0, \forall i$). Polarization effect increases with respect to polynomial degrees. As shown in Figure 4c, the performance also increases as we increase the polarization effect and approaches



(a) $\sin(x)$ Time (b) $\sin(x)$ Freq (c) $\sin(x)$ Linear (d) $\sin(x)$ Softmax

(e) Vary Time (f) Vary Freq (g) Vary Linear (h) Vary Softmax

(i) Trend Time (j) Trend Freq (k) Trend Linear (l) Trend Softmax

(m) Spike Time (n) Spike Freq (o) Spike Linear (p) Spike Softmax

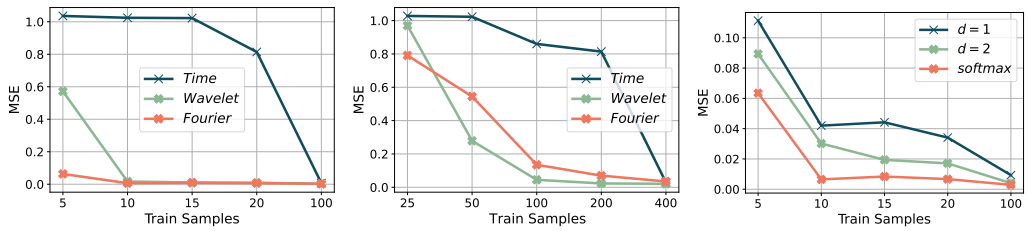
Figure 3: (a)-(d): Data with fixed seasonality: $\sin(x)$. Fourier softmax attention amplifies the correct frequency modes compared with Fourier linear attention. (e)-(h): Data with varying seasonality. Fourier softmax attention amplifies the dominant frequency modes, but also neglects the small-amplitude modes that embed the localized frequency information. (i)-(l): Data with linear trend. Fourier softmax attention incorrectly amplifies the low-frequency modes compared with Fourier linear attention. (m)-(p): Data with spikes as noises. Fourier softmax attention filters out the noisy components and emphasizes the correct frequency modes compared with Fourier linear attention.

Table 1: MSE and MAE of attention models and MLP with linear-trend data.

Metric	Time	Fourier	Wavelet	MLP
MSE	3.157 ± 0.435	8.567 ± 0.487	2.327 ± 0.689	$\mathbf{0 \pm 0}$
MAE	1.741 ± 0.121	2.880 ± 0.073	1.477 ± 0.239	$\mathbf{0.006 \pm 0.003}$

the performance of softmax operations. We also notice that apart from the polarization effect from exponential terms, normalization itself also introduces performance gaps between different attention models. The possible reason is that it's easier to optimize in the sparse Fourier domain compared with time domain. We leave this as our future explorations.

For data with varying seasonality, wavelet attention is the most effective. We use alternating $\sin(x)$ and $\sin(2x)$ as an example of varying seasonal data (visualized in Figure 3e and Figure 3f). The Fourier representation has both dominant modes as well as small-amplitude modes, where the latter embeds the varying-seasonality information. The Fourier softmax attention correctly amplifies the dominant frequency modes, but at the same time neglects the small-amplitude modes that convey the information of varying seasonality. By contrast, wavelet attention combines multi-scale time-frequency representation, and provides better localized frequency information. As shown in Figure 4b, wavelet attention is the most effective for varying-seasonality data.



(a) Attention Models (b) Attention Models (c) Polynomial Kernels

Figure 4: (a): Sample efficiency comparison of time, Fourier, wavelet attention models on data with fixed seasonality ($\sin(x)$). Fourier attention models are more sample-efficient. (b): Sample efficiency comparison on data with varying seasonality (alternating $\sin(x)$ and $\sin(2x)$). Wavelet attention models are more sample-efficient. (c): Sample efficiency comparison between polynomial kernels and softmax. Polarization effect increases with respect to the degree of polynomial kernels and approaches the softmax performance.

Table 2: MSE and MAE of different attention models with spiky data.

Metric	Time	Fourier	Wavelet
MSE	0.303 ± 0.002	$\mathbf{0.019 \pm 0.003}$	0.030 ± 0.008
MAE	0.495 ± 0.001	$\mathbf{0.111 \pm 0.010}$	0.137 ± 0.021

4.2 Data with Trend

For data with trend, all attention models show inferior generalizability, especially Fourier attention. We take linear trend data as an example (Figure 3i and Figure 3j) and evaluate different attention models. The first several frequency modes in Fourier space carry large values; the attention scores hence mostly focus on the first few frequency modes (top-left corner of Figure 3k). With the polarization effect of softmax, attention scores emphasize even more on these low-frequency components (Figure 3l) and generate misleading reconstruction results. We evaluate different attention models in Table 1. Fourier attention, with inappropriate polarization, leads to the largest errors.

Moreover, all these attention models fail to extrapolate linear trend well and suffer from large errors, since attention mechanism by nature works through interpolating the context history. By contrast, MLP perfectly predicts such trend signals, as shown in Table 1. This motivates us to decompose the time series into trend and seasonality [21, 24], apply attention mechanism only for seasonality, and use MLP for modeling trend.

4.3 Data with Spikes

For data carrying noises, Fourier attention is the most robust. We randomly inject large-value spikes into the training set of $\sin(x)$ as a motivating example (Figure 3m and Figure 3n). Spikes which have large values in time domain result in small-amplitude frequency components after Fourier transforms. With the polarization effect of softmax, time-domain softmax attention focuses incorrectly on large-value spikes while Fourier-domain softmax attention correctly filters out the noisy components and attends to the dominant frequency modes induced by $\sin(x)$. Comparing Figure 3o and Figure 3p, linear attention still distributes attention to the noisy frequency modes, while softmax attention mostly focuses correctly on the dominant frequency modes. Therefore, frequency-domain attention models are more robust to spikes, as shown in Table 2. All these analysis on datasets with different characteristics help guide our model design in the next section.

5 Our Method: TDformer

The performance difference in data with various characteristics motivates our model design. For data with seasonality, Fourier softmax attention amplifies dominant frequency modes and demonstrates the best performance. For data with trend, Fourier softmax incorrectly attends to only the low-frequency modes and produces large errors. Meanwhile, all attention models which work through interpolating the historical context, do not generalize well on trend data compared with MLP. These analyses

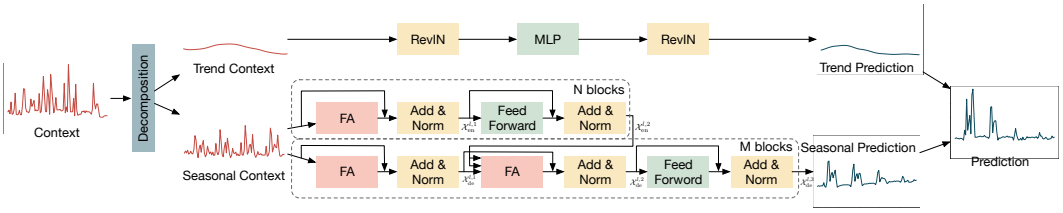


Figure 5: TDformer. We first apply seasonal trend decomposition to decompose the context time series into trend part and seasonal part. We adopt MLP to predict the trend part, and Fourier Attention (FA) model to predict the seasonal part, and add two parts together for final prediction.

motivate us to decompose time series into trend and seasonality, use Fourier attention to predict the seasonal part and MLP to predict the trend part. Figure 5 overviews our proposed model architecture.

We first decompose the time series into trend parts and seasonal parts following FEDformer [24]. More specifically, we apply multiple average filters with different sizes to extract different trend patterns, and apply adaptive weights to combine these patterns into the final trend component. The seasonal component is acquired by subtracting trend from the original time series:

$$\mathbf{x_{trend}} = \sigma(w(\mathbf{x})) * f(\mathbf{x}), \mathbf{x_{seasonal}} = \mathbf{x} - \mathbf{x_{trend}}, \tag{5}$$

where $\sigma, w(x), f(x)$ denote the softmax operation, data-dependent weights and average filters.

For the trend component, we use a three-layer MLP to predict the future trend. As reversible instance normalization (RevIN) proves to be effective to remove and restore the non-stationary information [7, 23] which mainly resides in trend, we also add RevIN layers before and after MLP: $\mathcal{X}_{\text{trend}} = \text{RevIN}(\text{MLP}(\text{RevIN}(\mathbf{x_{trend}})))$. For the seasonal component, we adopt Transformer architecture but replace time-domain attention with Fourier-domain attention. More specifically, we first feed the seasonal part to N layers of encoder:

$$\mathcal{X}_{\text{en}}^{l,1} = \text{Norm}(\text{FA}(\mathcal{X}_{\text{en}}^{l-1}) + \mathcal{X}_{\text{en}}^{l-1}), \mathcal{X}_{\text{en}}^{l,2} = \text{Norm}(\text{FF}(\mathcal{X}_{\text{en}}^{l,1}) + \mathcal{X}_{\text{en}}^{l,1}), \mathcal{X}_{\text{en}}^{l} = \mathcal{X}_{\text{en}}^{l,2}, l = 1, \cdots, N, \tag{6}$$

where $\mathcal{X}_{\text{en}}^{0} = \mathbf{x_{seasonal}}$, FA and FF are short for Fourier Attention and Feed Forward network. Fourier Attention computes the attention in Fourier space and converts the output to time domain at the end (Definition 3.2) with $\sigma(\cdot) = \text{softmax}(\cdot)$:

$$\mathbf{o}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathcal{F}^{-1}\{\text{softmax}(\mathcal{F}\{\mathbf{q}\}\overline{\mathcal{F}\{\mathbf{k}\}}^{T})\mathcal{F}\{\mathbf{v}\}\}. \tag{7}$$

The seasonal part is also zero-padded for the future part and fed into M layers of decoder to obtain the final seasonal output:

$$\mathcal{X}_{\text{de}}^{l,1} = \text{Norm}(\text{FA}(\mathcal{X}_{\text{de}}^{l-1}) + \mathcal{X}_{\text{de}}^{l-1}), \mathcal{X}_{\text{de}}^{l,2} = \text{Norm}(\text{FA}(\mathcal{X}_{\text{en}}^{N}, \mathcal{X}_{\text{de}}^{l,1}) + \mathcal{X}_{\text{de}}^{l,1}), \tag{8}$$

$$\mathcal{X}_{\text{de}}^{l,3} = \text{Norm}(\text{FF}(\mathcal{X}_{\text{de}}^{l,2}) + \mathcal{X}_{\text{de}}^{l,2}), \mathcal{X}_{\text{de}}^{l} = \mathcal{X}_{\text{de}}^{l,3}, l = 1, \cdots, M, \tag{9}$$

where $\mathcal{X}_{\text{de}}^{0} = \text{Padding}(\mathbf{x_{seasonal}})$. We add the trend prediction from MLP and seasonal prediction from Transformer to obtain the final output prediction, i.e., $\mathcal{X}_{\text{final}} = \mathcal{X}_{\text{trend}} + \mathcal{X}_{\text{de}}^{M}$. Optimization is based on a reconstruction MSE loss between predicted and ground truth future time series.

Remark. While FEDformer [24] and Autoformer [21] also have seasonal-trend decomposition, their trend and seasonal components are not disentangled; the trend prediction still comes from the attention module, which is sub-optimal based on our analysis in Section 4 and our empirical results in Section 6. By contrast, we apply seasonal-trend decomposition in the beginning, and apply Fourier attention only on seasonality components. This seemingly simple different way of decomposition brings significant performance gains to see in the experiment section, with even less model complexity. Non-stationary Transformer also computes attention for trend data. Moreover, with RevIN, TDformer has the similar effect of stationarization.

Table 3: MSE and MAE of different attention models with real-world seasonal and trend data.

Method	Metric	Traffic				Weather			
		96	192	336	720	96	192	336	720
Time	MSE	0.659	0.671	0.691	0.691	0.332	0.556	0.743	0.888
	MAE	0.358	0.358	0.368	0.363	0.395	0.533	0.622	0.702
Fourier	MSE	0.631	0.629	0.655	0.667	0.774	0.743	0.833	1.106
	MAE	0.338	0.336	0.345	0.350	0.648	0.632	0.659	0.769
Wavelet	MSE	0.622	0.629	0.640	0.655	0.358	0.564	0.815	1.312
	MAE	0.337	0.334	0.338	0.346	0.413	0.535	0.664	0.841

6 Experiments

6.1 Dataset and Baselines

We conduct experiments on benchmark time-series forecasting datasets: ETTm2 [22], electricity[2], exchange [10], traffic[3], weather[4]. We quantify the strength of seasonality for each dataset (details in Appendix). Electricity and traffic are strongly seasonal data, while exchange rate and weather demonstrate less seasonality and more trend. We compare TDformer with state-of-the-art attention models: Non-stationary Transformer [14], FEDformer [24], Autoformer [21], Informer [22], LogTrans [12], Reformer [9]. As classical models (e.g., ARIMA), RNN-based models and CNN-based models generate large errors as shown in previous papers [22, 21], here we do not include their performance in the comparison. We use Adam [8] optimizer with a learning rate of $1e^{-4}$ and batch size of 32. We split the dataset with $7:2:1$ into training, validation and test set, use validation set for hyperparameter tuning and report the results on the test set. For all real-world experiments, we feed the past 96 timesteps as context to predict the next $96, 192, 336, 720$ timesteps following previous works [24, 21]. All experiments are repeated 5 times and we report the mean MSE and MAE. We implement in Pytorch on NVIDIA V100 16GB GPUs.

6.2 Comparing Attention Models on Real-World Datasets

As an extension to experiments on synthetic data (Section 4), we also compare attention models on real-world datasets, and observe consistent results as on synthetic datasets. Note that for a fair comparison, we directly compare the attention models without additional components like decomposition blocks or additional learnable transformation kernels [21, 24]. We choose traffic dataset as data with seasonality and weather dataset as data with trend. As shown in Table 3, frequency-domain attention models demonstrate better performance with seasonal data, which aligns with our observations on synthetic datasets. For trend data, Fourier-attention models show larger errors compared with time and wavelet attention models, which is also consistent with our observations on synthetic datasets. Compared with the reported performance after seasonal-trend decomposition as in FEDformer [24] and Autoformer [21], the errors on seasonal data remain similar, while errors increase significantly on trend data. This emphasizes the importance of seasonal-trend decomposition.

6.3 Main Results

We compare TDformer with the state-of-the-art baselines and report on MSE and MAE in Table 4. TDformer consistently demonstrates better performance across different datasets and forecasting horizons. On average, TDformer reduces the MSE by 9.14% compared with Non-stationary Transformer and by 14.69% compared with FEDformer, and we attribute such improvement to our separate modeling of trend and seasonality with MLP and Fourier attention. As we mention in Remark 5, trend prediction of FEDformer and Non-stationary Transformer still come from attention modules, while TDformer decouples the modeling of trend and seasonality, and demonstrates better forecasting results. See Figure 1 for qualitative comparison.

6.4 Ablation Study

To separately understand the effect of trend and seasonal modules, we conducted ablation studies. TDformer-MLP-TA(WA) replaces Fourier attention with time (wavelet) attention for seasonality,

[2] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`

[3] `http://pems.dot.ca.gov`

[4] `https://www.bgc-jena.mpg.de/wetter/`

Table 4: MSE and MAE of multivariate time-series forecasting on benchmark datasets with input context length 96 and forecasting horizon $\{96, 192, 336, 720\}$. We **bold** the best performing results.

Methods		TDformer		Non-stat TF		FEDformer		Autoformer		Informer		LogTrans		Reformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	**0.160**	**0.263**	0.169	0.273	0.193	0.308	0.201	0.317	0.274	0.368	0.258	0.357	0.312	0.402
	192	**0.172**	**0.275**	0.182	0.286	0.201	0.315	0.222	0.334	0.296	0.386	0.266	0.368	0.348	0.433
	336	**0.186**	**0.290**	0.200	0.304	0.214	0.329	0.231	0.338	0.300	0.394	0.280	0.380	0.350	0.433
	720	**0.215**	**0.313**	0.222	0.32	0.246	0.355	0.254	0.361	0.373	0.439	0.283	0.376	0.340	0.420
Exchange	96	**0.089**	**0.208**	0.111	0.237	0.148	0.278	0.197	0.323	0.847	0.752	0.968	0.812	1.065	0.829
	192	**0.183**	**0.305**	0.219	0.335	0.271	0.380	0.300	0.369	1.204	0.895	1.040	0.851	1.188	0.906
	336	**0.353**	**0.429**	0.421	0.476	0.460	0.500	0.509	0.524	1.672	1.036	1.659	1.081	1.357	0.976
	720	**0.932**	**0.725**	1.092	0.769	1.195	0.841	1.447	0.941	2.478	1.310	1.941	1.127	1.510	1.016
Traffic	96	**0.545**	**0.320**	0.612	0.338	0.587	0.366	0.613	0.388	0.719	0.391	0.684	0.384	0.732	0.423
	192	**0.571**	**0.329**	0.613	0.340	0.604	0.373	0.616	0.382	0.696	0.379	0.685	0.390	0.733	0.420
	336	**0.589**	**0.331**	0.618	0.328	0.621	0.383	0.622	0.337	0.777	0.420	0.733	0.408	0.742	0.420
	720	**0.606**	**0.337**	0.653	0.355	0.626	0.382	0.660	0.408	0.864	0.472	0.717	0.396	0.755	0.423
Weather	96	0.177	**0.215**	**0.173**	0.223	0.217	0.296	0.266	0.336	0.300	0.384	0.458	0.490	0.689	0.596
	192	**0.224**	**0.257**	0.245	0.285	0.276	0.336	0.307	0.367	0.598	0.544	0.658	0.589	0.752	0.638
	336	**0.278**	**0.290**	0.321	0.338	0.339	0.359	0.380	0.395	0.578	0.523	0.797	0.652	0.639	0.596
	720	**0.368**	**0.351**	0.414	0.410	0.403	0.428	0.419	0.428	1.059	0.741	0.869	0.675	1.130	0.792
ETTm2	96	**0.174**	**0.256**	0.192	0.274	0.203	0.287	0.255	0.339	0.365	0.453	0.768	0.642	0.658	0.619
	192	**0.243**	**0.302**	0.280	0.339	0.269	0.328	0.281	0.340	0.533	0.563	0.989	0.757	1.078	0.827
	336	**0.308**	**0.344**	0.334	0.361	0.325	0.366	0.339	0.372	1.363	0.887	1.334	0.872	1.549	0.972
	720	**0.400**	**0.400**	0.417	0.413	0.421	0.415	0.422	0.419	3.379	1.338	3.048	1.328	2.631	1.242

Table 5: MSE and MAE of our model ablations.TDformer-MLP-TA replaces Fourier Attention by Time Attention (TA) for seasonality; TDformer-MLP-WA replaces Fourier Attention by Wavelet Attention (WA) for seasonality; TDformer-TA-FA replaces MLP with Time Attention (TA) for trend. TDformer w/o RevIN removes RevIN normalization.

Method	Metric	Traffic				Exchange			
		96	192	336	720	96	192	336	720
TDformer	MSE	0.545	0.571	0.589	0.606	0.089	0.183	0.353	0.932
	MAE	0.320	0.329	0.331	0.337	0.208	0.305	0.429	0.725
TDformer-MLP-TA	MSE	0.573	0.592	0.605	0.630	0.086	0.181	0.340	0.923
	MAE	0.334	0.336	0.340	0.351	0.205	0.303	0.422	0.721
TDformer-MLP-WA	MSE	0.552	0.583	0.599	0.629	0.088	0.185	0.348	0.925
	MAE	0.322	0.330	0.337	0.347	0.208	0.307	0.426	0.721
TDformer-TA-FA	MSE	0.590	0.590	0.617	0.642	0.242	0.349	0.629	0.908
	MAE	0.338	0.336	0.349	0.357	0.327	0.419	0.558	0.720
TDformer w/o RevIN	MSE	0.577	0.595	0.607	0.636	0.093	0.201	0.392	1.042
	MAE	0.320	0.325	0.328	0.339	0.222	0.330	0.474	0.763

and shows larger errors especially on seasonal data (traffic), as Fourier attention is more capable of capturing seasonality. Exchange data is mainly composed of trend, so different attention variants demonstrate similar performance. We also replace MLP with time attention for trend (TDformer-TA-FA) and observe large errors, as attention models show inferior generalization ability on trend data. TDformer w/o RevIN removes RevIN normalization and displays larger errors, which shows the importance of normalization for non-stationary data.

7 Conclusion

In this work we are driven by better understanding the relationships and separate benefits of attention models in time, Fourier and wavelet domains. We show that theoretically these three attention models are equivalent given linear assumptions. However, empirically due to the role of softmax, these models have respective benefits when applied to datasets with specific properties. Moreover, all attention models show inferior generalizability on data with trend. Based on these analyses of performance differences, we propose TDformer which separately models trend and seasonality with MLP and Fourier attention models after seasonal trend decomposition. TDformer achieves state-of-the-art performance against current attention models on time-series forecasting benchmarks. In the future, we plan to explore more complicated models to predict trend (e.g., autoregressive models) and explore other seasonal-trend decomposition methods.

References

[1] Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

[2] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73, 1990.

[3] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

[4] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in Neural Information Processing Systems*, 34:24048–24062, 2021.

[5] Hao Huang and Yi Fang. Adaptive wavelet transformer network for 3d shape representation learning. In *International Conference on Learning Representations*, 2021.

[6] Song Jiang, Tahin Syed, Xuan Zhu, Joshua Levy, Boris Aronchik, and Yizhou Sun. Bridging self-attention and time series decomposition for periodic forecasting. 2022.

[7] 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.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[10] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[11] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

[12] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[13] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[14] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Rethinking the stationarity in time series forecasting. *arXiv preprint arXiv:2205.14415*, 2022.

[15] LIU Minhao, Ailing Zeng, LAI Qiuxia, Ruiyuan Gao, Min Li, Jing Qin, and Qiang Xu. T-wavenet: A tree-structured wavelet neural network for time series signal analysis. In *International Conference on Learning Representations*, 2021.

[16] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[17] Fan-Keng Sun and Duane S Boning. Fredo: Frequency domain-based long-term time series forecasting. *arXiv preprint arXiv:2205.12301*, 2022.

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

[19] 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.

[20] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

[21] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[22] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[23] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *arXiv preprint arXiv:2205.08897*, 2022.

[24] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We focus on promoting understanding of existing models, and propose method that improves on forecasting benchmarks.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 6
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 6
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 6
   (b) Did you mention the license of the assets? [Yes] See Section 6
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We didn't introduce new datasets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use public benchmark forecasting datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use public benchmark forecasting datasets.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We didn't use crowdsourcing or involve human subjects in this paper.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We didn't use crowdsourcing or involve human subjects in this paper.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We didn't use crowdsourcing or involve human subjects in this paper.

A Appendix

We first apply STL decomposition [2] for each dataset

$$X_t = T_t + S_t + R_t, \tag{10}$$

where T_t, S_t, R_t respectively represent the trend, seasonal and remainder component. For data with strong seasonality, the seasonal component would have much larger variation than the remainder component; while for data with little seasonality, the two variances should be similar. Therefore, we can quantify the strength of seasonality as

$$\mathrm{S} = \max(0, 1 - \frac{\mathrm{Var}(R_t)}{\mathrm{Var}(S_t) + \mathrm{Var}(R_t)}). \tag{11}$$

Following this equation, we summarize the seasonality strength of each dataset in Table 6. Electricity and traffic are strongly seasonal data, while exchange rate and weather demonstrate less seasonality and more trend.

Table 6: Seasonality strength of benchmark datasets.

Dataset	Electricity	Exchange	Traffic	Weather	ETTm2
Seasonality Strength	0.982	0.284	0.967	0.317	0.711