Supplemental Material: *Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting*

Yong Liu[*], **Haixu Wu**[*], **Jianmin Wang, Mingsheng Long**[✉]
School of Software, BNRist, Tsinghua University, China
{liuyong21,whx20}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

1 Proof of De-stationary Attention

Definition. Self-Attention [13] is defined as:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1}$$

where $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V} \in \mathbb{R}^{S\times d_k}$ are length-S query, key and value, where S is the length of input sequence and d_k is the feature dimension, and $\mathrm{Softmax}(\cdot)$ is conducted on each row.

Assumption 1. *The embedding layer and feed forward layer are functions conducted separately at each time point of the input and hold the linear property.*

For example, the query $\mathbf{Q}$ as the input of the first $\mathrm{Attn}(\cdot)$ layer is obtained by feeding the input $\mathbf{x} = [x_1, x_2, \cdots, x_S]^\top \in \mathbb{R}^{S\times C}$ into the embedding layer $f : \mathbb{R}^{C\times 1} \to \mathbb{R}^{d_k\times 1}$, where C is the number of series variables. And each of the query token in $\mathbf{Q} = [q_1, q_2, ..., q_S]^\top$ can be calculated as $q_i = f(x_i)$ w.r.t. each time point in $\mathbf{x} = [x_1, x_2, \cdots, x_S]^\top$. Function f holds the linear property means that $f(ax + by) = af(x) + bf(y)$, where a, b are scalars and x, y are vectors.

Assumption 2. *Each variable of the input series has the same variance.*

For each input time series $\mathbf{x}$, we calculate its mean and variance as follows:

$$\mu_{\mathbf{x}} = \frac{1}{S}\sum_{i=1}^{S} x_i,\ \sigma_{\mathbf{x}}^2 = \frac{1}{S}\sum_{i=1}^{S}(x_i - \mu_{\mathbf{x}})^2,$$

where $\mu_{\mathbf{x}}, \sigma_{\mathbf{x}} \in \mathbb{R}^{C\times 1}$ is the mean and standard deviation of all x_is. Since it is a convention to conduct normalization on each series variable to avoid certain variable that dominates the scale, we can assume that each variable shares the same variance, and thus $\sigma_{\mathbf{x}}$ is reduced to a scalar.

Theorem.

$$\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) = \mathrm{Softmax}\left(\frac{\sigma_{\mathbf{x}}^2\,\mathbf{Q}'\mathbf{K}'^\top + \mathbf{1}\mu_{\mathbf{Q}}^\top\mathbf{K}^\top}{\sqrt{d_k}}\right). \tag{2}$$

Equation 2 means the $\mathrm{Softmax}\left(\mathbf{Q}\mathbf{K}^\top/\sqrt{d_k}\right)$ learned from raw series $\mathbf{x}$ can be calculated by current $\mathbf{Q}', \mathbf{K}'$ learned from stationarized series $\mathbf{x}'$, and the calculation also requires the non-stationary information $\sigma_{\mathbf{x}}, \mu_{\mathbf{Q}}, \mathbf{K}$ that are eliminated during stationarization.

[*]Equal Contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Proof 1. (First layer analysis) After our stationarization, the model receives the normalized input $\mathbf{x}' = [x'_1, x'_2, ..., x'_S]^\top$ and each $x'_i = (1/\sigma_\mathbf{x}) \odot (x_i - \mu_\mathbf{x})$. Based on Assumption 2, $\sigma_\mathbf{x}$ is reduced to a scalar and we can simplify the normalized input of each time point to $x'_i = (x_i - \mu_\mathbf{x})/\sigma_\mathbf{x}$. Then $\mathbf{x}'$ is fed into the embedding layer f. Based on Assumption 1, we get current query $\mathbf{Q}' = [q'_1, ..., q'_S]^\top$ of the first $\mathrm{Attn}(\cdot)$ layer:

$$q'_i = f(\frac{x_i - \mu_\mathbf{x}}{\sigma_\mathbf{x}}) = \frac{f(x_i) - f(\mu_\mathbf{x})}{\sigma_\mathbf{x}} = \frac{q_i - f(\frac{1}{S}\sum_{i=1}^S x_i)}{\sigma_\mathbf{x}} = \frac{q_i - \frac{1}{S}\sum_{i=1}^S f(x_i)}{\sigma_\mathbf{x}} = \frac{q_i - \mu_\mathbf{Q}}{\sigma_\mathbf{x}},$$

where $\mu_\mathbf{Q} = \frac{1}{S}\sum_{i=1}^S q_i \in \mathbb{R}^{d_k \times 1}$. Then $\mathbf{Q}' = [q'_1, ..., q'_S]^\top$ can be written as $(\mathbf{Q} - \mathbf{1}\mu_\mathbf{Q}^\top)/\sigma_\mathbf{x}$ and $\mathbf{1} \in \mathbb{R}^{S\times 1}$ is an all-ones vector. And so is the corresponding transformed $\mathbf{K}'$. Without the stationarization, the input of $\mathrm{Softmax}(\cdot)$ in Self-Attention should be $(\mathbf{Q}\mathbf{K}^\top/\sqrt{d_k})$, while now the attention is calculated based on $\mathbf{Q}', \mathbf{K}'$. And we have the following equations:

$$\mathbf{Q}'\mathbf{K}'^\top = \frac{1}{\sigma_\mathbf{x}^2}\left(\mathbf{Q}\mathbf{K}^\top - \mathbf{1}(\mu_\mathbf{Q}^\top\mathbf{K}^\top) - (\mathbf{Q}\mu_\mathbf{K})\mathbf{1}^\top + \mathbf{1}(\mu_\mathbf{Q}^\top\mu_\mathbf{K})\mathbf{1}^\top\right),$$

$$\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) = \mathrm{Softmax}\left(\frac{\sigma_\mathbf{x}^2\,\mathbf{Q}'\mathbf{K}'^\top + \mathbf{1}(\mu_\mathbf{Q}^\top\mathbf{K}^\top) + (\mathbf{Q}\mu_\mathbf{K})\mathbf{1}^\top - \mathbf{1}(\mu_\mathbf{Q}^\top\mu_\mathbf{K})\mathbf{1}^\top}{\sqrt{d_k}}\right).$$

We find that $\mathbf{Q}\mu_\mathbf{K} \in \mathbb{R}^{S\times 1}$ and $\mu_\mathbf{Q}^\top\mu_\mathbf{K} \in \mathbb{R}$, and they are repeatedly operated on each column and element of $\sigma_\mathbf{x}^2\mathbf{Q}'\mathbf{K}'^\top \in \mathbb{R}^{S\times S}$. Since $\mathrm{Softmax}(\cdot)$ is invariant to the same translation on the row dimension of input, we have the following equation:

$$\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) = \mathrm{Softmax}\left(\frac{\sigma_\mathbf{x}^2\,\mathbf{Q}'\mathbf{K}'^\top + \mathbf{1}\mu_\mathbf{Q}^\top\mathbf{K}^\top}{\sqrt{d_k}}\right).$$

Proof 2. (Multiple layers analysis) We have deduced an equivalent expression of the output of the first $\mathrm{Softmax}(\cdot)$. If we can successfully approximate the attention map that related to $\mathbf{Q}$ and $\mathbf{K}$, we only need to consider $\mathrm{Attn}(\cdot)$ with respect to the change of $\mathbf{V}$. Fortunately, $\mathrm{Attn}(\cdot)$ as the function of $\mathbf{V}$ gives each time point of the output $\mathbf{E} = [e_1, ..., e_S]^\top \in \mathbb{R}^{S\times d_k}$ as a simplex:

$$e_j = \left\{\sum_{i=1}^S w_i v_i | \mathbf{V} = [v_1, v_2, ..., v_S]^\top, \sum_{i=1}^S w_i = 1, w_i \geq 0\right\},$$

which also holds the linear property $f(a\mathbf{V_1} + b\mathbf{V_2}) = af(\mathbf{V_1}) + bf(\mathbf{V_2})$. Therefore, the $\mathrm{Attn}(\cdot)$ layer is also a function that satisfies our Assumption 1. We will have each time point of the output $\mathbf{E}$ varies linearly with each time point of the input $\mathbf{x}$, and then $\mathbf{E}$ will become the next block's input. As the feed forward layer, residual adding and $\mathrm{Attn}(\cdot)$ layer are the repeating building blocks of Transformer, they also compose a function with linear property as stated in Assumption 1. By the first layer analysis and induction on each layer, Equation 2 will holds for $\mathrm{Softmax}(\cdot)$ of all layers under our assumptions.

Attention design Based on the analysis, we develop De-stationary Attention as:

$$\log\tau = \mathrm{MLP}(\sigma_\mathbf{x}, \mathbf{x}), \mathbf{\Delta} = \mathrm{MLP}(\mu_\mathbf{x}, \mathbf{x}),$$
$$\mathrm{Attn}(\mathbf{Q}', \mathbf{K}', \mathbf{V}', \tau, \mathbf{\Delta}) = \mathrm{Softmax}\left(\frac{\tau\,\mathbf{Q}'\mathbf{K}'^\top + \mathbf{1}\mathbf{\Delta}^\top}{\sqrt{d_k}}\right)\mathbf{V}', \tag{3}$$

where $\tau \in \mathbb{R}^+$ and $\mathbf{\Delta} \in \mathbb{R}^{S\times 1}$ is defined as the scaling and shifting de-stationary factors respectively to approximate $\sigma_\mathbf{x}^2$ and $\mathbf{K}\mu_\mathbf{Q}$ under the real scenario. Since the key to making Equation 2 established is to approximate the attention map successfully, we apply a direct deep learning implementation. To be concisely, we use a multi-layer perceptron as the projector to learn de-stationary factors $\tau, \mathbf{\Delta}$ from the statistics $\mu_\mathbf{x}, \sigma_\mathbf{x}$ and unstationarized $\mathbf{x}$. De-stationary Attention learns the temporal dependencies from both stationarized series $\mathbf{Q}', \mathbf{K}'$ and non-stationary series $\mathbf{x}$, $\mu_\mathbf{x}$, $\sigma_\mathbf{x}$, and multiplies by the stationarized values $\mathbf{V}'$ to keep the linear property. It can benefit from the predictability of stationarized series and re-incorporate the inherent non-stationarity of raw series simultaneously.

2 Hyperparameter Sensitivity

We verify the robustness of the proposed Non-stationary Transformers framework with respect to hyper-parameter dim, which is the hidden layer dimension of the MLP projector that learns de-stationary factors. Considering the efficiency of hyperparameters search, we fix the number of hidden layers, and the hidden layer dimension varies in $\{64, 128, 256\}$. The results are shown in Table 1. For datasets with relatively high non-stationarity (Exchange and ILI), large dim would be a better choice, which indicates that non-stationary information entangled with unstationarized input should be learned by a projector with big capacity. Besides, as the dataset presents higher non-stationarity, the influence of de-stationary project design becomes more significant.

Table 1: The performance of Non-stationary Transformers under different choices of the hidden layer dimension (dim) in the projector. We adopt the forecasting setting as input-36-predict-48 for the ILI dataset and input-96-predict-336 for the other datasets.

Dataset	Exchange		ILI		ETTm2		Electricity		Traffic		Weather	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$dim = 64$	0.448	0.493	2.067	0.908	**0.334**	**0.361**	**0.200**	**0.304**	0.629	0.345	**0.321**	**0.338**
$dim = 128$	0.432	0.477	**2.010**	**0.900**	0.370	0.388	0.201	0.301	**0.618**	**0.328**	0.340	0.354
$dim = 256$	**0.421**	**0.476**	2.223	0.928	0.367	0.381	0.201	0.304	0.631	0.351	0.333	0.347

3 Supplementary of Main Results

3.1 Multivariable Forecasting Results

As shown in Table 2, we list additional benchmark on the ETT datasets [16], which includes the hourly recorded ETTh1/ETTh2 and 15-minutely recorded ETTm1. Non-stationary Transformer also achieves remarkable improvement over the state-of-the-art on various forecasting horizons. For the input-96-predict-336 long-term setting, Non-stationary Transformer surpasses previous best results by **4.4%**$(0.615 \to 0.588)$ in ETTh1, **3.5%**$(0.572 \to 0.552)$ in ETTh2 and **26.7%**$(0.675 \to 0.495)$ MSE reduction in ETTm1. The overall results show averaged **11.5%** MSE reduction over previous state-of-the-art deep forecasting models.

We also list additional model comparison in Table 3, including the concurrent work FEDformer [17], and non-Transformer models LSSL [6] and GRU [5]. Our method still outperforms these models in most cases (83%). Notably, LSSL [6] achieves good performance on Weather [3] dataset with the highest stationarity but poorly performs on others, especially non-stationary datasets.

Table 2: Forecasting results comparison under different prediction lengths $O \in \{96, 192, 336, 720\}$ on ETT dataset. The input sequence length is set to 96.

Models		**Ours**		Autoformer[15]		Pyraformer[11]		Informer[16]		LogTrans[10]		Reformer[8]		LSTNet[9]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.513**	**0.491**	0.536	0.548	0.783	0.657	0.984	0.786	0.767	0.758	0.773	0.640	1.457	0.961
	192	**0.534**	**0.504**	0.543	0.551	0.863	0.709	1.027	0.791	1.003	0.849	0.910	0.704	1.998	1.215
	336	**0.588**	**0.535**	0.615	0.592	0.941	0.753	1.032	0.774	1.362	0.952	1.000	0.760	2.655	1.369
	720	0.643	0.616	**0.599**	**0.600**	1.042	0.819	1.169	0.858	1.397	1.291	1.242	0.860	2.143	1.380
ETTh2	96	**0.476**	**0.458**	0.492	0.517	1.380	0.943	2.826	1.330	0.829	0.751	1.595	1.031	3.568	1.688
	192	**0.512**	**0.493**	0.556	0.551	3.809	1.634	6.186	2.070	1.807	1.036	2.671	1.300	3.243	2.514
	336	**0.552**	**0.551**	0.572	0.578	4.282	1.792	5.268	1.942	3.875	1.763	2.596	1.297	2.544	2.591
	720	**0.562**	**0.560**	0.580	0.588	4.252	1.790	3.667	1.616	3.913	1.552	2.647	1.304	4.625	3.709
ETTm1	96	**0.386**	**0.398**	0.523	0.488	0.536	0.506	0.615	0.556	0.588	0.593	0.778	0.623	2.003	1.218
	192	**0.459**	**0.444**	0.543	0.498	0.539	0.520	0.723	0.620	0.769	0.793	0.929	0.707	2.764	1.544
	336	**0.495**	**0.464**	0.675	0.551	0.720	0.635	1.300	0.908	1.462	1.320	1.016	0.733	1.257	2.076
	720	**0.585**	**0.516**	0.720	0.573	0.940	0.740	0.972	0.744	1.669	1.461	1.122	0.793	1.917	2.941

Table 3: Forecasting results comparison with additional baseline forecasting models.

Models		Ours		FEDformer [17]		LSSL [6]		GRU [5]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	**0.111**	**0.237**	0.148	0.271	0.395	0.474	1.453	1.049
	192	**0.219**	**0.335**	0.271	0.380	0.776	0.698	1.846	1.179
	336	**0.421**	**0.476**	0.460	0.500	1.029	0.797	2.136	1.231
	720	**1.092**	**0.769**	1.195	0.841	2.283	1.222	2.984	1.427
ILI	24	**2.294**	**0.945**	3.228	1.260	4.381	1.425	5.914	1.734
	36	**1.825**	**0.848**	2.679	1.080	4.442	1.416	6.631	1.845
	48	**2.010**	**0.900**	2.622	1.078	4.559	1.443	6.736	1.857
	60	**2.178**	**0.963**	2.857	1.157	4.651	1.474	6.870	1.879
ETTm2	96	**0.192**	**0.274**	0.203	0.287	0.243	0.342	2.041	1.073
	192	0.280	0.339	**0.269**	**0.328**	0.392	0.448	2.249	1.112
	336	0.334	**0.361**	**0.325**	0.366	0.932	0.724	2.568	1.238
	720	**0.417**	**0.413**	0.421	0.415	1.372	0.879	2.720	1.287
Electricity	96	**0.169**	**0.273**	0.193	0.308	0.300	0.392	0.375	0.437
	192	**0.182**	**0.286**	0.201	0.315	0.297	0.390	0.442	0.473
	336	**0.200**	**0.304**	0.214	0.329	0.317	0.403	0.439	0.473
	720	**0.222**	**0.321**	0.246	0.355	0.338	0.417	0.980	0.814
Traffic	96	0.612	**0.338**	**0.587**	0.366	0.798	0.436	0.843	0.453
	192	0.613	**0.340**	**0.604**	0.373	0.849	0.481	0.847	0.453
	336	**0.618**	**0.328**	0.621	0.383	0.828	0.476	0.853	0.455
	720	0.653	**0.355**	**0.626**	0.382	0.854	0.489	1.500	0.805
Weather	96	**0.173**	**0.223**	0.217	0.296	0.174	0.252	0.369	0.406
	192	0.245	**0.285**	0.276	0.336	**0.238**	0.313	0.416	0.435
	336	0.321	**0.338**	0.339	0.380	**0.287**	0.355	0.455	0.454
	720	0.414	**0.410**	0.403	0.428	**0.384**	0.415	0.535	0.520

3.2 Performance of Non-stationary Transformer and Variants

We apply our proposed Non-stationary Transformers framework to six Transformer variants: Transformer [13], Informer [16], Reformer [8], Autoformer [15], ETSformer [14] and FEDformer [17]. The averaged results are shown in Table 4 of the main text due to the limited pages. We provide supplementary forecasting results in Table 4 and Table 5. The experimental results demonstrate that our Non-stationary Transformers framework can consistently promotes these Transformer variants, even on the concurrent work ETSformer and FEDformer.

3.3 Comparison with Stationarization Methods

We provide full comparison among Non-stationary Transformers and two stationarization methods: Revin[7] and Series Stationarization. The averaged results are shown in Table 5 of the main text due to the limited pages. As is listed in Table 6, our framework achieves the state-of-the-art performance especially on datasets with high non-stationarity. For Transformer, the proposed method achieves **25.6%**($1.467 \rightarrow 1.092$) MSE reduction on Exchange under the predict-720 settings, **10.8%**($2.572 \rightarrow 2.294$) on ILI under the predict-24 settings, and **30.3%**($0.598 \rightarrow 0.417$) on ETTm2 under the predict-720 settings. As for Reformer, since De-stationary Attention is not directly deduced from the LSH attention [8], current approximation as stated in Equation 2 may not be the optimal solution, but the introducing of De-stationary Attention still achieves relative **11.6%**($0.632 \rightarrow 0.559$) promotion on ETTm2 and **4.4%**($2.834 \rightarrow 2.770$) on ILI under the predict-336 setting. The comparison demonstrates De-stationary Attention mechanism can further benefit the predictive ability of Transformers.

Table 4: Detailed forecasting performance of Non-stationary Transformers. We report the MSE/MAE of different prediction lengths $O \in \{96, 192, 336, 720\}$ and $\{24, 36, 48, 60\}$ for comparison. The input sequence length is set to 36 for ILI and 96 for the others.

Models		Transformer + Ours		Informer + Ours		Reformer + Ours		Autoformer + Ours	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.111±0.015	0.237±0.010	0.129±0.018	0.258±0.012	0.128±0.019	0.258±0.012	0.171±0.005	0.276±0.006
	192	0.219±0.031	0.335±0.020	0.251±0.042	0.354±0.035	0.246±0.045	0.356±0.037	0.273±0.005	0.365±0.007
	336	0.421±0.032	0.476±0.022	0.373±0.047	0.434±0.032	0.422±0.039	0.478±0.030	0.481±0.010	0.573±0.009
	720	1.092±0.027	0.769±0.024	1.229±0.035	0.795±0.049	1.050±0.050	0.781±0.047	1.024±0.012	0.751±0.012
ILI	24	2.294±0.152	0.945±0.041	2.856±0.177	1.071±0.067	3.206±0.277	1.131±0.079	3.029±0.116	1.166±0.028
	36	1.825±0.128	0.848±0.033	1.805±0.143	0.860±0.051	2.750±0.161	1.018±0.074	2.648±0.134	1.023±0.032
	48	2.010±0.134	0.900±0.035	1.780±0.194	0.849±0.054	2.710±0.184	1.017±0.050	2.202±0.161	0.965±0.038
	60	2.178±0.146	0.963±0.037	2.058±0.173	0.933±0.058	2.792±0.153	1.095±0.047	2.302±0.088	1.003±0.024
ETTm2	96	0.192±0.023	0.274±0.016	0.241±0.035	0.312±0.025	0.209±0.040	0.287±0.028	0.236±0.022	0.319±0.019
	192	0.280±0.021	0.339±0.013	0.433±0.036	0.420±0.025	0.435±0.037	0.421±0.026	0.263±0.026	0.316±0.025
	336	0.334±0.011	0.361±0.017	0.507±0.032	0.464±0.023	0.559±0.033	0.475±0.024	0.320±0.019	0.349±0.014
	720	0.417±0.009	0.413±0.011	0.659±0.019	0.539±0.028	0.769±0.021	0.582±0.021	0.402±0.015	0.396±0.010
Electricity	96	0.169±0.008	0.273±0.002	0.195±0.008	0.302±0.003	0.190±0.007	0.293±0.004	0.193±0.009	0.295±0.003
	192	0.182±0.007	0.286±0.003	0.215±0.007	0.321±0.006	0.199±0.009	0.301±0.008	0.211±0.006	0.310±0.007
	336	0.200±0.005	0.304±0.005	0.235±0.006	0.339±0.006	0.208±0.005	0.310±0.005	0.220±0.005	0.316±0.004
	720	0.222±0.016	0.321±0.013	0.260±0.014	0.358±0.014	0.226±0.015	0.326±0.018	0.241±0.019	0.337±0.017
Traffic	96	0.612±0.019	0.338±0.014	0.649±0.028	0.370±0.016	0.669±0.037	0.364±0.020	0.604±0.027	0.342±0.012
	192	0.613±0.028	0.340±0.018	0.689±0.035	0.393±0.019	0.680±0.036	0.369±0.022	0.607±0.034	0.383±0.020
	336	0.618±0.018	0.328±0.012	0.755±0.055	0.431±0.054	0.688±0.038	0.371±0.033	0.611±0.019	0.353±0.010
	720	0.653±0.014	0.355±0.003	0.783±0.026	0.440±0.004	0.692±0.019	0.385±0.014	0.653±0.014	0.376±0.013
Weather	96	0.173±0.006	0.223±0.004	0.186±0.017	0.235±0.014	0.195±0.020	0.242±0.013	0.215±0.024	0.263±0.019
	192	0.245±0.014	0.285±0.015	0.259±0.024	0.292±0.019	0.255±0.027	0.289±0.023	0.257±0.027	0.296±0.018
	336	0.321±0.016	0.338±0.023	0.295±0.026	0.317±0.018	0.306±0.030	0.323±0.025	0.307±0.009	0.321±0.011
	720	0.414±0.032	0.410±0.031	0.361±0.020	0.362±0.022	0.388±0.024	0.376±0.026	0.364±0.006	0.357±0.007

Table 5: Performance promotion by applying the proposed framework to concurrent ETSformer and FEDformer. We report the averaged MSE/MAE of all prediction lengths (stated in Table 2 of the main text) and the relative MSE reduction ratios (Promotion) by our framework.

Dataset	Exchange		ILI		ETTm2		Electricity		Traffic		Weather	
Model	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETSformer [14]	0.410	0.427	2.619	1.034	0.293	0.342	0.208	0.323	0.629	0.403	0.271	0.334
+ Ours	0.369	0.407	2.353	1.017	0.290	0.334	0.203	0.314	0.618	0.380	0.254	0.293
Promotion	**10.00%**		**10.16%**		0.77%		**2.17%**		1.75%		**6.18%**	
FEDformer [17]	0.519	0.500	2.847	1.144	0.305	0.349	0.214	0.327	0.610	0.376	0.309	0.360
+ Ours	0.500	0.487	2.728	1.046	0.312	0.346	0.198	0.300	0.604	0.362	0.268	0.292
Promotion	**3.66%**		**4.18%**		-2.38%		**7.38%**		0.86%		**13.36%**	

Table 6: Detailed forecasting results obtained by applying different methods to Transformer and Reformer. We report the MSE/MAE of different prediction lengths for comparison.

Base Models		Transformer						Reformer					
Methods		+ RevIN [7]		+ Series Stationarization		**+ Ours**		+ RevIN [7]		+ Series Stationarization		**+ Ours**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.136	0.258	0.136	0.258	**0.111**	**0.237**	0.133	0.263	0.139	0.265	**0.128**	**0.258**
	192	0.239	0.348	0.239	0.348	**0.219**	**0.335**	0.256	0.363	0.257	0.364	**0.246**	**0.356**
	336	0.425	0.479	0.425	0.479	**0.421**	**0.476**	0.426	0.477	0.426	0.477	**0.422**	**0.478**
	720	1.467	0.862	1.475	0.865	**1.092**	**0.769**	1.059	0.786	1.059	0.786	**1.050**	**0.781**
ILI	24	2.572	0.980	2.573	0.980	**2.294**	**0.945**	3.399	1.170	3.399	1.170	**3.206**	**1.131**
	36	1.955	0.870	1.955	0.870	**1.825**	**0.848**	2.909	1.049	2.909	1.048	**2.750**	**1.018**
	48	2.056	0.902	2.057	0.902	**2.010**	**0.900**	2.834	1.067	2.832	1.067	**2.710**	**1.017**
	60	2.238	0.982	2.238	0.982	**2.178**	**0.963**	2.954	1.099	2.952	1.098	**2.792**	**1.095**
ETTm2	96	0.267	0.317	0.253	0.311	**0.192**	**0.274**	0.211	0.295	0.212	0.297	**0.209**	**0.287**
	192	0.456	0.405	0.453	0.404	**0.280**	**0.339**	0.478	0.426	0.477	0.426	**0.435**	**0.421**
	336	0.528	0.455	0.546	0.461	**0.334**	**0.361**	0.632	0.485	0.613	0.483	**0.559**	**0.475**
	720	0.589	0.487	0.593	0.489	**0.417**	**0.413**	0.845	0.631	0.846	0.630	**0.769**	**0.582**
Electricity	96	0.172	0.275	0.171	0.275	**0.169**	**0.273**	0.188	0.291	**0.184**	**0.289**	0.190	0.293
	192	0.192	0.296	0.192	0.296	**0.182**	**0.286**	0.198	0.301	0.199	0.302	**0.198**	**0.301**
	336	0.207	0.306	0.208	0.306	**0.200**	**0.304**	0.212	0.314	0.212	0.314	**0.208**	**0.310**
	720	0.217	0.316	**0.216**	**0.315**	0.222	0.321	0.232	0.331	0.231	0.330	**0.226**	**0.326**
Traffic	96	0.620	0.341	0.614	0.337	**0.612**	**0.338**	**0.650**	**0.364**	0.655	0.366	0.669	0.364
	192	0.630	0.348	0.637	0.351	**0.613**	**0.340**	0.688	0.374	0.683	0.377	**0.680**	**0.369**
	336	0.656	0.360	0.653	0.359	**0.634**	**0.348**	0.708	0.383	0.704	0.383	**0.688**	**0.371**
	720	0.666	0.360	0.661	0.360	**0.653**	**0.355**	0.700	0.392	0.722	0.395	**0.692**	**0.385**
Weather	96	0.175	0.225	0.175	0.225	**0.173**	**0.223**	**0.189**	**0.236**	0.190	0.237	0.195	0.242
	192	0.273	0.298	0.273	0.297	**0.245**	**0.285**	0.269	0.294	0.269	0.294	**0.255**	**0.289**
	336	0.333	0.326	0.333	0.325	**0.321**	**0.338**	0.312	0.328	0.313	0.329	**0.306**	**0.323**
	720	0.424	0.415	0.436	0.420	**0.414**	**0.410**	0.395	0.376	0.395	0.376	**0.388**	**0.376**

3.4 Prediction Showcases

We provide supplementary showcases of predictions given by three models: vanilla Transformer, Transformer with Series Stationarization, and Non-stationary Transformer. We plot the last dimension of forecasting results that comes from the *test set* of ETTm1 for qualitative comparison.

As is shown in Figures 1, 2, 3, and 4, we find that vanilla Transformer is inclined to output predictions with scale and level far from the ground truth, but its ability to capture local series variation remains strong. While Series Stationarization benefits Transformer by aligning the statistics among each series, the base model neglects the intrinsic non-stationarity of time series and becomes more likely to output stationary but uneventful series. With the help of our framework, the equipped model will be free from the disturbance caused by data non-stationarity and fulfill the potential to capture local variations.

Table 7: Parameters increment and performance promotion of Non-stationary Transformers.

Models	Transformer	Informer	Reformer	Autoformer	FEDformer	ETSformer
Param increment	0.10%	0.09%	0.21%	0.10%	0.06%	0.19%
Performance gain	49.43%	47.34%	46.89%	10.57%	4.51%	5.17%

3.5 Efficiency of Non-stationary Transformers

As is shown in Table 7, we list the parameters increment and the performance gain of our proposed method. It is obvious that Non-stationary Transformers significantly boosts the forecasting performance by a large margin with hardly any additional parameters.

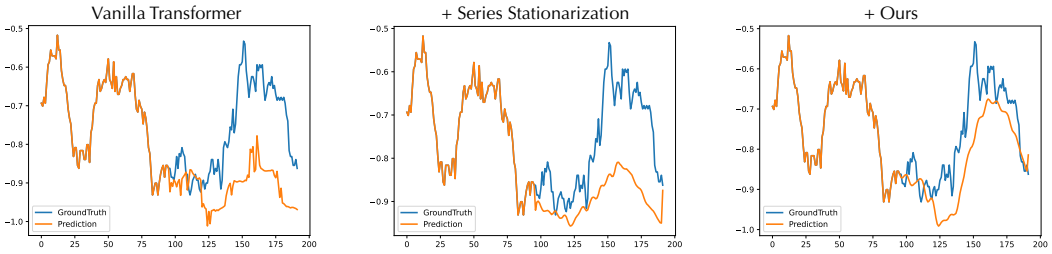

Figure 1: Visualization of ETTm1 predictions given by different models under the input-96-predict-96 setting. Blue lines stand for the ground truth and orange lines stand for predictions of the model. The first shared part is the time series input with length 96.

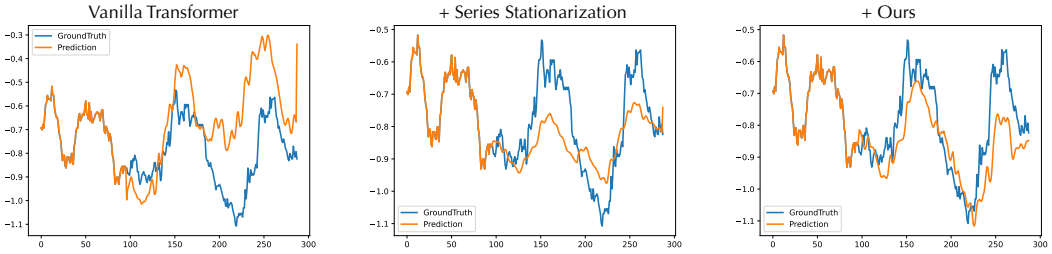

Figure 2: Visualization of predictions given by models under the input-96-predict-192 setting.

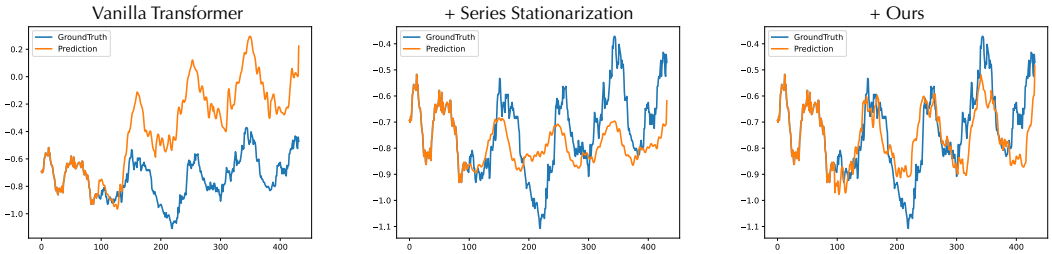

Figure 3: Visualization of predictions given by models under the input-96-predict-336 setting.

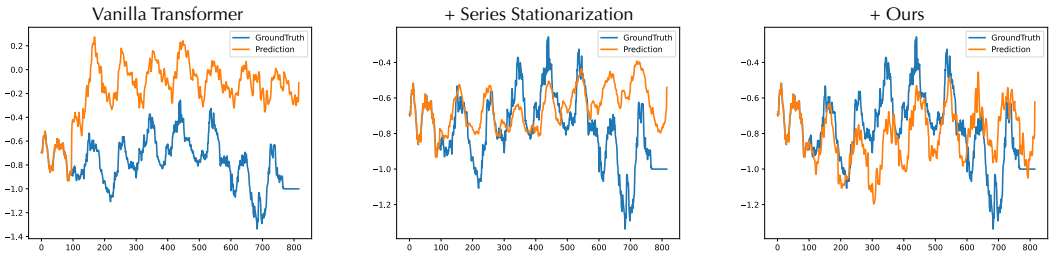

Figure 4: Visualization of predictions given by models under the input-96-predict-720 setting.

Table 8: ADF test statistic of raw series and series processed by our normalization.

Dataset	Exchange	ILI	ETTm2	Electricity	Traffic	Weather
Raw series	-1.889	-5.406	-6.225	-8.483	-15.046	-26.661
After Normalization	-9.937	-10.313	-33.485	-20.888	-18.946	-35.010

Algorithm 1 Series Stationarization - Normalization.

Require: Input past time series $\mathbf{x} \in \mathbb{R}^{S \times C}$; Input Length S; Variables number C.

$\mu_{\mathbf{x}} = \texttt{Mean}(\mathbf{x}, \texttt{dim=0})$ ▷ $\mu_{\mathbf{x}} \in \mathbb{R}^{1 \times C}$

$\sigma_{\mathbf{x}} = \texttt{Std}(\mathbf{x}, \texttt{dim=0})$ ▷ $\sigma_{\mathbf{x}} \in \mathbb{R}^{1 \times C}$

$\mathbf{x}' = \texttt{Repeat}\big((1/\sigma_{\mathbf{x}}), \texttt{dim=0}\big) \odot \big(\mathbf{x} - \texttt{Repeat}(\mu_{\mathbf{x}}, \texttt{dim=0})\big)$ ▷ Normalize to $\mathbf{x}' \in \mathbb{R}^{S \times C}$

Return $\mathbf{x}', \mu_{\mathbf{x}}, \sigma_{\mathbf{x}}$ ▷ Return stationarized input and original statistics

Algorithm 2 Series Stationarization - De-normalization.

Require: Predicted time series $\mathbf{y}' \in \mathbb{R}^{O \times C}$ by the base model; original statistics of input $\mu_{\mathbf{x}}, \sigma_{\mathbf{x}} \in \mathbb{R}^{1 \times C}$; Output Length O; Variables number C.

$\mathbf{y} = \texttt{Repeat}\big(\sigma_{\mathbf{x}}, \texttt{dim=0}\big) \odot \mathbf{y}' + \texttt{Repeat}(\mu_{\mathbf{x}}, \texttt{dim=0})$ ▷ De-normalize to $\mathbf{y} \in \mathbb{R}^{O \times C}$

Return $\mathbf{y}$ ▷ Return de-normalized output

Algorithm 3 De-stationary Attention.

Require: Queries $\mathbf{Q}' \in \mathbb{R}^{S \times d_k}$; Keys $\mathbf{K}' \in \mathbb{R}^{S \times d_k}$; Values $\mathbf{V}' \in \mathbb{R}^{S \times d_k}$; De-stationary factors $\tau \in \mathbb{R}^{+}, \mathbf{\Delta} \in \mathbb{R}^{S \times 1}$; Input Length S; Feature dimension d_k.

$\texttt{Output} = \texttt{Softmax}\left(\big(\tau\, \mathbf{Q}'{\mathbf{K}'}^{\top} + \texttt{Repeat}(\mathbf{\Delta}, \texttt{dim=1})\big)/\sqrt{d_k}\right)\mathbf{V}'$ ▷ rescaling by τ and $\mathbf{\Delta}$

Return $\texttt{Output}$ ▷ Return de-stationary attention output

Algorithm 4 Non-stationary Transformers - Overall Architecture.

Require: Input past time series $\mathbf{x} \in \mathbb{R}^{S \times C}$; Input Length S; Predict length O; Variables number C; Feature dimension d_k; Encoder layers number N; Decoder layers number M. Technically, we set d_k as 512, N as 2, M as 1.

$\mathbf{x}', \mu_{\mathbf{x}}, \sigma_{\mathbf{x}} = \texttt{Normalization}(\mathbf{x})$ ▷ $\mathbf{x}' \in \mathbb{R}^{S \times C}, \mu_{\mathbf{x}} \in \mathbb{R}^{1 \times C}, \sigma_{\mathbf{x}} \in \mathbb{R}^{1 \times C}$

$\log \tau, \mathbf{\Delta} = \texttt{MLP}(\mathbf{x}, \mu_{\mathbf{x}}, \sigma_{\mathbf{x}})$ ▷ $\tau \in \mathbb{R}^{+}, \mathbf{\Delta} \in \mathbb{R}^{S \times 1}$

$\mathbf{x}'_{\text{enc}}, \mathbf{x}'_{\text{dec}} = \mathbf{x}', \texttt{ConCat}\big(\mathbf{x}'_{\frac{S}{2}:S}, \texttt{Zeros}(O, C)\big)$ ▷ $\mathbf{x}'_{\text{enc}} \in \mathbb{R}^{S \times C}, \mathbf{x}'_{\text{dec}} \in \mathbb{R}^{(\frac{S}{2}+O) \times C}$

$\mathbf{x}^{0\prime}_{\text{enc}} = \texttt{Embed}(\mathbf{x}'_{\text{enc}})$ ▷ $\mathbf{x}^{0\prime}_{\text{enc}} \in \mathbb{R}^{S \times d_k}$

for l **in** $\{1, \cdots, N\}$**:** ▷ Non-stationary Encoder

  $\mathbf{x}^{l-1\prime}_{\text{enc}} = \texttt{LayerNorm}\big(\mathbf{x}^{l-1\prime}_{\text{enc}} + \texttt{Attn}(\mathbf{x}^{l-1\prime}_{\text{enc}}, \tau, \mathbf{\Delta})\big)$ ▷ $\mathbf{x}^{l-1\prime}_{\text{enc}} \in \mathbb{R}^{S \times d_k}$

  $\mathbf{x}^{l\prime}_{\text{enc}} = \texttt{LayerNorm}\big(\mathbf{x}^{l-1\prime}_{\text{enc}} + \texttt{FFN}(\mathbf{x}^{l-1\prime}_{\text{enc}})\big)$ ▷ $\mathbf{x}^{l\prime}_{\text{enc}} \in \mathbb{R}^{S \times d_k}$

End for

$\mathbf{x}^{0\prime}_{\text{dec}} = \texttt{Embed}(\mathbf{x}'_{\text{dec}})$ ▷ $\mathbf{x}^{0\prime}_{\text{dec}} \in \mathbb{R}^{(\frac{S}{2}+O) \times d_k}$

for l **in** $\{1, \cdots, M\}$**:** ▷ Non-stationary Decoder

  $\mathbf{x}^{l-1\prime}_{\text{dec}} = \texttt{LayerNorm}\big(\mathbf{x}^{l-1\prime}_{\text{dec}} + \texttt{Attn}(\mathbf{x}^{l-1\prime}_{\text{dec}}, \tau, \mathbf{\Delta} = 0)\big)$ ▷ $\mathbf{x}^{l-1\prime}_{\text{dec}} \in \mathbb{R}^{(\frac{S}{2}+O) \times d_k}$

  $\mathbf{x}^{l-1\prime}_{\text{dec}} = \texttt{LayerNorm}\big(\mathbf{x}^{l-1\prime}_{\text{dec}} + \texttt{Attn}(\mathbf{x}^{l-1\prime}_{\text{dec}}, \mathbf{x}^{N\prime}_{\text{enc}}, \tau, \mathbf{\Delta})\big)$ ▷ $\mathbf{x}^{l-1\prime}_{\text{dec}} \in \mathbb{R}^{(\frac{S}{2}+O) \times d_k}$

  $\mathbf{x}^{l\prime}_{\text{dec}} = \texttt{LayerNorm}\big(\mathbf{x}^{l-1\prime}_{\text{dec}} + \texttt{FFN}(\mathbf{x}^{l-1\prime}_{\text{dec}})\big)$ ▷ $\mathbf{x}^{l\prime}_{\text{dec}} \in \mathbb{R}^{(\frac{S}{2}+O) \times d_k}$

End for

$\mathbf{y}' = \texttt{MLP}(\mathbf{x}^{M\prime}_{\text{dec}})_{\frac{S}{2}:\frac{S}{2}+O}$ ▷ $\mathbf{y}' \in \mathbb{R}^{O \times d_k}$

$\mathbf{y} = \texttt{De-normaliztion}(\mathbf{y}', \mu_{\mathbf{x}}, \sigma_{\mathbf{x}})$ ▷ $\mathbf{y} \in \mathbb{R}^{O \times d_k}$

Return $\mathbf{y}$ ▷ Return the prediction results

4 Ablations

4.1 Effects of Series Stationarization

We propose Series Stationarization, which has no additional learnable parameters, to increase the degree of stationarity and make the time series distribution more stable. As is shown in Table 8, after our normalization module processing, the ADF test statistic of the time series gets obviously smaller, which verifies normalization as an effective design to attenuate the non-stationarity of real world time series.

4.2 Ablation of De-stationary Factors

To explore the influence of de-stationary factors, we compare the forecasting results obtained by three variants: only using τ, only using $\mathbf{\Delta}$, and using both. We conduct experiments on two typical datasets: Exchange (8 variables) and Electricity (321 variables). As is shown in Table 9, the forecasting performance will degrade in all cases if we only employ single one of τ and $\mathbf{\Delta}$, especially without τ ($0.196 \rightarrow 0.212, 0.441 \rightarrow 0.550$ under the predict-336 setting), which validates the complete form as stated in Equation 3 is a better choice.

Table 9: Ablation on de-stationary factors: Column (only τ) means only use the scaling de-stationary factor in Equation 3, Column (only $\mathbf{\Delta}$) means only use the shifting de-stationary factors, and Column (τ and $\mathbf{\Delta}$) means use both.

Models		Only τ		Only $\mathbf{\Delta}$		τ and $\mathbf{\Delta}$	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.177	0.279	0.186	0.287	**0.169**	**0.273**
	192	0.191	0.297	0.196	0.299	**0.185**	**0.289**
	336	0.197	0.300	0.212	0.310	**0.196**	**0.297**
	720	0.221	0.320	0.227	0.326	**0.217**	**0.317**
Exchange	96	0.128	0.253	0.128	0.253	**0.120**	**0.247**
	192	0.263	0.369	0.263	0.370	**0.250**	**0.353**
	336	0.446	0.491	0.550	0.553	**0.441**	**0.488**
	720	1.348	0.847	1.621	0.911	**1.338**	**0.847**

5 Non-stationary Transformers: Experimental Details

5.1 Detailed Experiment Configurations

We compare each Transformers with and without our framework using the same training strategy. The only hyperparameters for our framework come from the projector design which learns de-stationary factors. We search the hyperparameters as stated in Section 2. The best hyperparameter is selected on the *validation set*.

As for other forecast models for the baseline comparison, most of the results are from Autoformer [15]. By contacting the authors of Autoformer, we obtain the hyper-parameter selection strategy as follows: for N-BEATS [12], we conduct a grid search for hidden channel in $\{256, 512, 768\}$, number of layers in $\{2, 3, 4, 5\}$, learning rate in $\{5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$. For LSTNet [9], since the paper also experiments on the Traffic [1], Electricity [2] and Exchange [9] datasets, the hyper-parameter setting is following the experimental details of the original paper. For N-HiTs [4], ETSformer [14], and FEDformer [17], as these methods share the same benchmark, we use their official code with three random seeds.

5.2 Implementation Details of Non-stationary Transformer and Variants

We provide the pseudo-code of Series Stationarization, De-stationary Attention and Non-stationary Transformers in Algorithms 1, 2, 3 and 4. All Transformers have two-layer encoder and one-layer decoder with the feature dimension d_k=512, including Transformer [13], Informer [16], Reformer [8], Autoformer [15], ETSformer [14] and FEDformer [17]. Besides, we adopt embedding method and

one-step generation strategy of Informer [16]. It is worth noting that for the row length of attention map differs from $S \times S$, where S is the initial input sequence length, we omit the shifting de-stationary factor $\mathbf{\Delta}$ in Equation 3 (i.e., the Self-Attention layer of Transformer decoder, and the Self-Attention layer of the Informer encoder where the shape of attention map is changed over layers), since the performance of only use τ will not degenerate a lot as shown in Table 9. For the cross attention, we first conduct the rescaling operation with de-stationary factors and then multiply by the corresponding mask. For Transformer variants, we conduct the rescaling operation on the pre-$\mathrm{Softmax}$ scores.

6 Broader Impact

6.1 Impact on Real-world Applications

We focus on real-world time series forecasting, which is challenging for Transformers because of data non-stationarity. Our method goes beyond previous studies that only stationarize the time series. We fully utilize the predictive capability of attention mechanism that captures essential temporal dependencies associated with inherent non-stationarity. Our proposed method achieves state-of-the-art performance in five real-world applications, which makes it more promising for Transformers to tackle real-world forecasting applications, and helps our society make better decisions and prevent risks in advance for various fields. And our paper mainly focuses on scientific research and has no obvious negative social impact.

6.2 Impact on Future Research

In this paper, we analyze the generalization difficulty of Transformers in distribution-varying time series forecasting. We propose a general framework to fulfill the potential of Transforms constrained by data non-stationary. Our work introduces an essential and promising direction to improve forecasting performance: to increase the stationarity of time series towards better predictability and mitigate the over-stationarization problem for the predictive capability of deep models simultaneously. The remarkable generality and effectiveness of the proposed framework can be instructive for future research.

7 Limitation

Our De-stationary Attention is deduced by analyzing the vanilla Self-Attention, which may not be the optimal solution for advanced attention mechanisms. There also remains room for re-incorporating non-stationarity on other classical stationarization methods, like differencing and quantile. Besides, the proposed framework is currently limited to the Transformer-based models, while the over-stationarization problem can appear on any deep time forecasting models if using stationarization methods inappropriately. Therefore, a more model-agnostic solution for the over-stationarization problem will be our exploring direction.

References

[1] Traffic Dataset. http://pems.dot.ca.gov/.

[2] UCI Electricity Load Time Series Dataset. https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

[3] Weather Dataset. https://www.bgc-jena.mpg.de/wetter/.

[4] 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.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state-space layers. *NeurIPS*, 2021.

[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 *ICLR*, 2022.

[8] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020.

[9] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*, 2018.

[10] 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. In *NeurIPS*, 2019.

[11] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *ICLR*, 2021.

[12] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ICLR*, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[14] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:1406.1078*, 2022.

[15] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *NeurIPS*, 2021.

[16] 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 *AAAI*, 2021.

[17] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, 2022.