Supplementary material: *Adaptive Normalization for Non-stationary Time Series Forecasting: A Temporal Slice Perspective*

Anonymous Author(s)
Affiliation
Address
email

1 Effects of SAN on Non-stationary Time Series Forecasting

1.1 Discussions

As illustrated in the main paper, our proposed SAN is a compact plug-and-play framework. We will first give a brief discussion in this section on how SAN can be effective.

It is of utmost importance that SAN can well alleviate the impact of the non-stationary nature of time series data. Forecasting models may encounter a non-i.i.d problem with non-stationary data, that is, the marginal distribution of each input instance can be different, which may lead to a huge difference between the distribution of the training set and the test set. Thus the models can not generalize well in future predictions. However, SAN will normalize all the input instances into a standard normal distribution and force the mean and variance of the training and test data distributions to be identical. In this way, all the data instances are from the same distribution, therefore the forecasting task is simplified as the models can get rid of the noises caused by non-stationary factors and only focus on mining the time-invariant patterns. Moreover, compared to existing normalization methods for forecasting, our modeling of the non-stationary property in a time slice view is more in-depth and realistic, so SAN can better remove the non-stationary factors in input sequences while keeping their instinct information in the normalization phase. Hence, SAN is theoretically expected to perform better in non-stationary time series forecasting.

Another part that contributes to the effectiveness of SAN is the statistics prediction module and the two-stage training schema. With the statistics prediction module independently modeling the evolving trends of statistical properties, SAN adopts more precise statistics for adaptive denormalization than existing solutions. Moreover, the proposed two-stage strategy actually simplifies the original forecasting task by divide and conquer: In the first stage we try to learn the general direction and dispersion of the future data, which is easy to fit and is conducted by the light statistics prediction module. Next, we utilize the powerful backbone model to discover the scale-free periodic-like features to estimate future values under the guidance of the well-trained statistics prediction module. Therefore, backbone models in SAN are actually responsible for an easier subtask. Considering that SAN can usually give reliable estimations on future distributions, SAN is expected to perform well on non-stationary time series forecasting by splitting the task into two simpler subtasks.

1.2 Theoretical Analysis

Using the same notation in the paper, we prove that all the inputs after SAN's normalization follow a standard normal distribution, validating SAN's capability to remove the non-stationary factors theoretically.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

In detail, for arbitrary input sequence $\boldsymbol{x}^i$, SAN first split it into M non-overlapping slices $\{\boldsymbol{x}_j^i\}_{j=1}^M$ and normalizes them according to their statistics. Therefore we will get:

$$\forall i, j \; \mathbb{E}[\bar{\boldsymbol{x}}_j^i] = 0, Var[\bar{\boldsymbol{x}}_j^i] = I \tag{1}$$

And as for the statistics of normalized input $\bar{\boldsymbol{x}}^i$, it satisfies the following equations:

$$\begin{aligned} \mathbb{E}[\bar{\boldsymbol{x}}^i] &= \mathbb{E}_j[\mathbb{E}[\bar{\boldsymbol{x}}_j^i]] \\ &= \mathbb{E}_j[0] \\ &= 0 \end{aligned} \tag{2}$$

$$\begin{aligned} Var[\bar{\boldsymbol{x}}^i] &= \frac{\sum_{t=0}^{L_{in}} (\bar{x}_{:,t}^i - \mathbb{E}[\bar{\boldsymbol{x}}^i])^2}{L_{in}} \\ &= \frac{\sum_{t=0}^{L_{in}} (\bar{x}_{:,t}^i)^2}{MT} \\ &= \frac{1}{M} * (\frac{\sum_{t=0}^{T} (\bar{x}_{:,t}^i)^2}{T} + \frac{\sum_{t=T}^{2T} (\bar{x}_{:,t}^i)^2}{T} +, \ldots, \frac{\sum_{t=(M-1)T}^{MT} (\bar{x}_{:,t}^i)^2}{T}) \\ &= \mathbb{E}_j[Var[\bar{\boldsymbol{x}}_j^i]] \\ &= I \end{aligned} \tag{3}$$

Here $\bar{x}_{:,t}^i \in R^{V*1}$ denotes all the normalized variables in time step t. From the above equations, we can learn that any input sequence follows a standard normal distribution after the normalization operation of SAN, which meets our expectations.

2 Supplementary Experiments

2.1 Full Benchmark on the ETT Dataset

Table 1: Multivariate forecasting results on full ETT dataset.

Methods		DLinear		+ SAN		FEDformer		+ SAN		Autoformer		+ SAN		SCINet		+ SAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.377**	**0.399**	0.383	**0.399**	**0.371**	0.411	0.383	**0.409**	**0.458**	**0.448**	0.488	0.464	0.470	0.479	**0.391**	**0.405**
	192	**0.417**	0.426	0.419	**0.419**	**0.420**	0.443	0.431	**0.438**	**0.481**	0.474	0.498	**0.472**	0.541	0.520	**0.438**	**0.433**
	336	0.464	0.461	**0.437**	**0.432**	**0.446**	0.459	0.471	**0.456**	**0.508**	**0.485**	0.530	0.498	0.643	0.587	**0.477**	**0.451**
	720	0.493	0.505	**0.446**	**0.459**	**0.482**	0.495	0.504	**0.488**	**0.525**	0.516	0.555	**0.514**	0.774	0.669	**0.489**	**0.474**
ETTh2	96	0.292	0.356	**0.277**	**0.338**	0.341	0.382	**0.300**	**0.355**	0.384	0.420	**0.316**	**0.366**	0.690	0.625	**0.294**	**0.347**
	192	0.383	0.418	**0.340**	**0.378**	0.426	0.436	**0.392**	**0.413**	0.457	0.454	**0.413**	**0.426**	0.991	0.742	**0.374**	**0.398**
	336	0.473	0.477	**0.356**	**0.398**	0.481	0.479	**0.459**	**0.462**	0.468	0.473	**0.446**	**0.457**	1.028	0.759	**0.412**	**0.430**
	720	0.708	0.599	**0.396**	**0.435**	**0.458**	0.477	0.462	**0.472**	0.473	0.485	**0.471**	**0.474**	1.363	0.885	**0.437**	**0.461**
ETTm1	96	0.301	0.344	**0.288**	**0.342**	0.362	0.408	**0.311**	**0.355**	0.493	0.470	**0.343**	**0.378**	0.444	0.464	**0.321**	**0.360**
	192	0.335	0.366	**0.323**	**0.363**	0.395	0.427	**0.351**	**0.383**	0.546	0.498	**0.390**	**0.400**	0.491	0.500	**0.347**	**0.380**
	336	0.370	0.387	**0.357**	**0.384**	0.441	0.454	**0.390**	**0.407**	0.658	0.543	**0.415**	**0.418**	0.572	0.556	**0.385**	**0.403**
	720	0.425	0.421	**0.409**	**0.415**	0.488	0.481	**0.456**	**0.444**	0.626	0.532	**0.476**	**0.453**	0.728	0.654	**0.450**	**0.441**
ETTm2	96	0.169	0.263	**0.166**	**0.258**	0.191	0.283	**0.175**	**0.266**	0.261	0.329	**0.236**	**0.317**	0.303	0.404	**0.176**	**0.267**
	192	0.232	0.310	**0.223**	**0.302**	0.261	0.326	**0.246**	**0.315**	0.282	0.339	**0.260**	**0.329**	0.568	0.569	**0.240**	**0.311**
	336	0.303	0.361	**0.272**	**0.330**	0.327	0.365	**0.315**	**0.362**	0.350	0.378	**0.330**	**0.376**	0.793	0.689	**0.300**	**0.351**
	720	0.403	0.424	**0.360**	**0.384**	0.428	0.423	**0.412**	**0.422**	0.438	**0.428**	**0.417**	**0.428**	1.200	0.851	**0.391**	**0.405**

We provide the full multivariate forecasting results on the ETT dataset in Table 1, which includes the hourly datasets ETTh1&ETTh2 and the 15-minutes datasets ETTm1&ETTm2. It is obvious that SAN also achieves significant improvements on these datasets on various backbone models.

2.2 Univariate Forecasting Results

Following the same settings of our main experiment, we provide the univariate forecasting results in Table 2. Similar to the results of multivariate forecasting, SAN can boost the performance of mainstream forecasting models in most cases. On average of all the benchmark settings, DLinear enhanced by SAN reduces MSE by **6.04%** (from 0.230 to 0.214). The improvements for FEDformer, Autoformer and SCINet are **15.40%**, **29.27%** and **36.29%** respectively.

Table 2: Univariate forecasting results. The bold values indicate better performance.

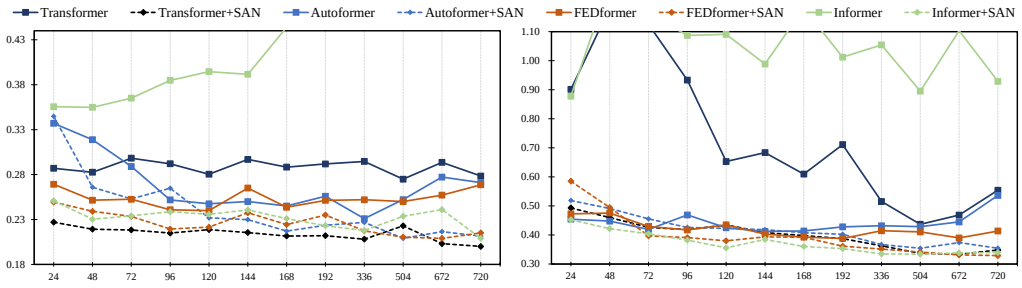
Methods		DLinear		+ SAN		FEDformer		+ SAN		Autoformer		+ SAN		SCINet		+ SAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	**0.203**	**0.315**	0.204	0.317	0.302	0.413	**0.248**	**0.363**	0.442	0.490	**0.283**	**0.386**	0.364	0.435	**0.321**	**0.412**
	192	**0.233**	**0.336**	0.238	0.341	0.377	0.459	**0.278**	**0.379**	0.555	0.550	**0.296**	**0.393**	0.345	0.419	**0.328**	**0.412**
	336	**0.268**	**0.363**	0.278	0.371	0.673	0.636	**0.324**	**0.411**	0.617	0.620	**0.359**	**0.440**	0.368	**0.435**	**0.363**	0.436
	720	0.330	0.425	**0.325**	**0.420**	0.575	0.575	**0.502**	**0.514**	0.645	0.624	**0.443**	**0.503**	0.420	0.478	**0.410**	**0.477**
Exchange	96	**0.108**	**0.254**	0.138	0.288	0.134	0.272	**0.113**	**0.252**	0.155	0.305	**0.097**	**0.233**	0.167	0.332	**0.090**	**0.226**
	192	**0.193**	**0.350**	0.287	0.436	**0.290**	0.418	0.307	**0.404**	0.405	0.495	**0.208**	**0.358**	0.486	0.552	**0.185**	**0.335**
	336	0.428	**0.511**	**0.416**	0.523	0.490	0.542	**0.431**	**0.501**	0.874	0.728	**0.401**	**0.495**	0.579	0.608	**0.396**	**0.484**
	720	1.137	0.848	**0.859**	**0.719**	1.302	0.883	**1.188**	**0.835**	1.193	0.845	**1.071**	**0.787**	**0.853**	**0.740**	1.106	0.797
Traffic	96	0.124	**0.197**	**0.123**	0.199	0.179	0.282	**0.144**	**0.236**	0.265	0.375	**0.172**	**0.273**	0.352	0.430	**0.267**	**0.364**
	192	0.125	**0.200**	**0.124**	**0.200**	0.211	0.316	**0.141**	**0.232**	0.266	0.372	**0.211**	**0.316**	0.291	0.377	**0.240**	**0.338**
	336	**0.126**	**0.206**	0.228	0.269	0.369	0.458	**0.207**	**0.318**	0.284	0.371	**0.164**	**0.259**	**0.298**	**0.387**	0.347	0.396
	720	0.141	0.226	**0.138**	**0.223**	**0.300**	**0.407**	0.477	0.526	0.260	0.369	**0.179**	**0.286**	0.339	0.417	**0.311**	**0.384**
Weather	96	0.004	0.047	**0.002**	**0.032**	**0.002**	**0.037**	0.003	0.042	0.004	0.047	**0.002**	**0.038**	0.005	0.060	**0.003**	**0.039**
	192	0.005	0.057	**0.002**	**0.037**	0.005	0.058	**0.004**	**0.049**	**0.003**	0.045	**0.003**	**0.047**	0.006	0.065	**0.002**	**0.036**
	336	0.006	0.068	**0.003**	**0.047**	**0.003**	**0.045**	0.004	0.052	0.008	0.068	**0.003**	**0.046**	0.007	0.068	**0.004**	**0.049**
	720	0.007	0.070	**0.004**	**0.050**	0.011	0.080	**0.004**	**0.048**	0.058	0.176	**0.004**	**0.049**	0.007	0.070	**0.003**	**0.045**
ILI	24	0.741	0.681	**0.663**	**0.626**	0.910	0.825	**0.798**	**0.688**	0.865	0.800	**0.765**	**0.721**	6.336	2.130	**0.707**	**0.665**
	36	0.570	0.634	**0.552**	**0.599**	0.873	0.823	**0.697**	**0.691**	0.984	0.855	**0.660**	**0.693**	6.159	1.998	**0.743**	**0.706**
	48	0.740	0.742	**0.647**	**0.669**	1.027	0.904	**0.820**	**0.761**	1.105	0.925	**0.753**	**0.752**	6.597	2.082	**0.783**	**0.744**
	60	0.911	0.848	**0.765**	**0.743**	1.221	1.002	**0.981**	**0.839**	1.222	0.982	**1.024**	**0.904**	7.556	2.418	**0.902**	**0.801**
ETTh1	96	0.058	**0.180**	**0.056**	0.181	0.097	0.241	**0.067**	**0.195**	0.093	0.241	**0.062**	**0.188**	0.110	0.262	**0.057**	**0.180**
	192	0.078	0.216	**0.076**	**0.212**	0.109	0.257	**0.081**	**0.215**	0.121	0.290	**0.082**	**0.216**	0.152	0.312	**0.075**	**0.209**
	336	0.099	0.246	**0.092**	**0.240**	0.103	0.251	**0.098**	**0.240**	0.115	0.271	**0.089**	**0.232**	0.183	0.350	**0.093**	**0.238**
	720	0.158	0.322	**0.092**	**0.240**	0.130	0.290	**0.103**	**0.248**	0.108	0.259	**0.106**	**0.249**	0.252	0.432	**0.096**	**0.245**
ETTh2	96	**0.132**	**0.280**	0.133	0.281	0.145	0.301	**0.141**	**0.286**	0.181	0.332	**0.141**	**0.288**	0.149	0.306	**0.129**	**0.274**
	192	0.177	0.330	**0.174**	**0.327**	0.188	0.339	**0.184**	**0.331**	0.213	0.371	**0.196**	**0.350**	0.187	0.340	**0.178**	**0.326**
	336	0.207	0.366	**0.200**	**0.359**	**0.220**	0.380	0.224	**0.371**	0.232	0.391	**0.221**	**0.370**	0.236	0.385	**0.222**	**0.374**
	720	0.301	0.447	**0.237**	**0.391**	0.279	0.427	**0.257**	**0.407**	0.267	0.417	**0.289**	**0.431**	0.326	0.468	**0.272**	**0.421**
ETTm1	96	0.027	**0.123**	**0.026**	**0.123**	0.060	0.193	**0.028**	**0.125**	0.059	0.193	**0.027**	**0.125**	0.065	0.204	**0.032**	**0.135**
	192	0.045	0.156	**0.040**	**0.151**	0.065	0.202	**0.044**	**0.159**	0.083	0.231	**0.042**	**0.155**	0.198	0.342	**0.049**	**0.168**
	336	0.059	0.178	**0.055**	**0.176**	0.066	0.199	**0.059**	**0.189**	0.069	0.205	**0.057**	**0.181**	0.221	0.382	**0.068**	**0.199**
	720	0.081	0.212	**0.077**	**0.208**	**0.084**	**0.230**	0.098	0.234	0.095	0.243	**0.081**	**0.213**	0.303	0.466	**0.093**	**0.231**
ETTm2	96	**0.063**	**0.183**	**0.063**	0.186	0.097	0.244	**0.060**	**0.183**	0.128	0.278	**0.068**	**0.195**	0.073	0.200	**0.069**	**0.193**
	192	**0.093**	**0.229**	**0.093**	0.230	0.129	0.281	**0.093**	**0.233**	0.145	0.298	**0.099**	**0.240**	0.107	0.248	**0.103**	**0.240**
	336	0.120	**0.263**	**0.119**	0.264	0.174	0.326	**0.129**	**0.276**	0.148	0.303	**0.123**	**0.269**	0.163	0.314	**0.135**	**0.281**
	720	0.173	**0.318**	**0.171**	0.319	0.201	0.354	**0.193**	**0.337**	0.208	0.359	**0.174**	**0.320**	0.325	0.441	**0.191**	**0.337**

2.3 Validation on Various Input Lengths

The input length plays an essential role in time series forecasting tasks as it determines how much historical temporal information the model can mine. One may hope that for powerful deep models, the longer the input length, the better the forecasting results. However, a recent study on this question reveals that deep Transformer-based models are not capable of capturing temporal dependencies in the long-term input sequences [6]. That is, the performance of these deep models stays stable or even degrades when the input length increases.

Apart from the design of these deep models, we hold that such a phenomenon can be raised by the non-stationary property of time series. As the input length increases, the variance among input sequences grows larger and ultimately makes it harder for deep models to discover the time-invariant patterns. Therefore, by removing the non-stationary factors in the input by SAN, deep models are expected to exhibit a steady decline in metrics with longer input lengths.

To evidence our thoughts, we conduct long-term forecasting experiments, i.e., $L_{out} = 720$, with various input lengths $L_{in} \in \{24, 48, 72, 96, 120, 144, 168, 192, 336, 504, 672, 720\}$ on the Transformer-based models. Here we choose Transformer [4], Informer [7], Autoformer [5] and FEDformer [8] as the backbone models. The MSE evaluations are plotted in Fig. 1. Note that we omit large values in the line chart to better demonstrate the trend of the overall results. From the figure, we can see that with the assistance of SAN, the performance of deep models with long sequence input is largely improved. When the input length is set to 720 on the Electricity dataset, the performance of Informer has been boosted by **77.83%** (from 0.9426 to 0.2090), and the average improvement on four backbones under the same setting is **52.55%**. Moreover, all of the backbones enhanced by SAN tend to produce more



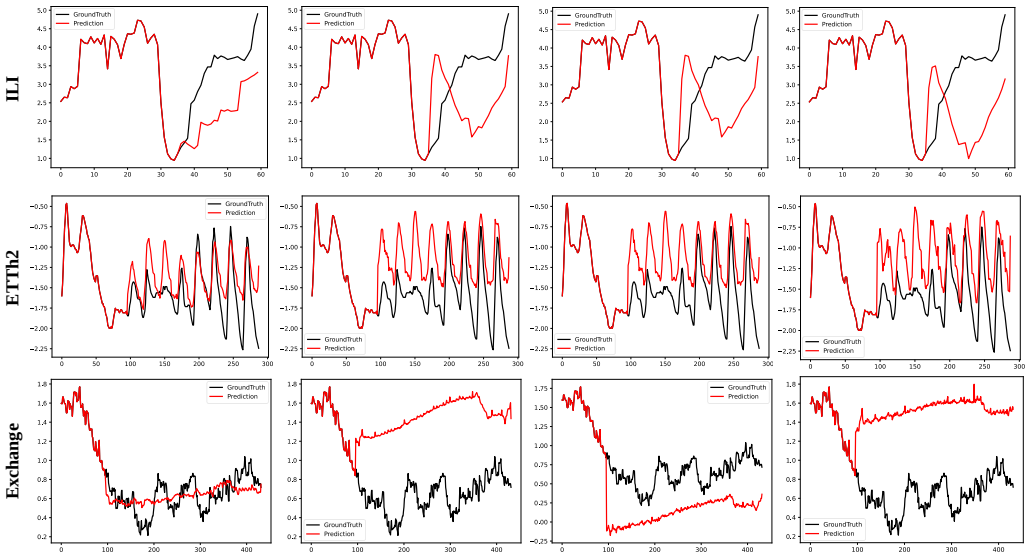
(a) 720 steps-Electricity (b) 720 steps-Weather

Figure 1: The long-term forecasting MSE evaluations of different Transformer-based models under various input lengths. Large values are discarded to illustrate the overall trend better.

accurate forecasting as the length increases. To be specific, on the Weather dataset, Transformer achieves a reduction on MSE of **29.40%** when prolonging input from 24 steps to 720 steps, and the average improvement on four backbones is **33.11%**. These results greatly meet our expectations and also validate the effectiveness of SAN on various input lengths.

2.4 Additional Prediction Showcases

We provide the additional comparison between SAN and other normalization methods in Fig. 2 with FEDformer [8] on various datasets. Clearly, SAN can better estimate the future distribution so as to help the backbone model to achieve superior performance, where the forecasting results are better aligned with the groundtruth.



(a) SAN (b) RevIN (c) NST (d) Dish-TS

Figure 2: Illustration of the additional prediction showcases comparing SAN and baseline models. The experiment is conducted on the ILI, ETTh2, and Exchange dataset. Following the same input sequence length setting in our main experiments, the target sequence length is set to 24, 192, and 336 respectively.

2.5 Ablation Study

Statistic Prediction Module In this section, we aim to analyze the effectiveness of our designs in the statistic prediction module. We instantiate our method and its variants on Autoformer and test their performance on two typical non-stationary datasets: Exchange and ETTh2. Similarly, we repeat the experiments three times with fixed seed and report the evaluations with standard deviation in Table 3.

Table 3: Forecasting errors under the multivariant setting with respect to variants of SAN. The best performance are highlighted in **bold**.

Variants		SAN		w/o individual		w/o residual		w/o SAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	**0.082±0.001**	**0.208±0.001**	0.089±0.005	0.209±0.007	0.135±0.003	0.264±0.001	0.152±0.006	0.283±0.007
	192	**0.157±0.001**	**0.296±0.003**	0.184±0.009	0.306±0.010	0.331±0.044	0.416±0.025	0.369±0.055	0.437±0.033
	336	**0.262±0.004**	**0.385±0.002**	0.340±0.001	0.422±0.001	0.658±0.044	0.593±0.024	0.534±0.130	0.544±0.066
	720	**0.689±0.043**	**0.629±0.020**	0.982±0.001	0.753±0.002	1.456±0.011	0.882±0.009	1.222±0.099	0.848±0.021
ETTh2	96	**0.316±0.001**	**0.366±0.001**	0.321±0.013	0.367±0.008	0.383±0.020	0.413±0.012	0.384±0.021	0.420±0.013
	192	**0.413±0.013**	0.426±0.007	0.414±0.023	**0.422±0.012**	0.463±0.030	0.469±0.020	0.457±0.020	0.454±0.014
	336	**0.446±0.004**	0.457±0.003	0.448±0.003	**0.453±0.001**	0.586±0.025	0.541±0.009	0.468±0.010	0.473±0.005
	720	**0.471±0.009**	**0.474±0.005**	0.483±0.012	0.477±0.005	0.889±0.007	0.682±0.001	0.473±0.005	0.485±0.005

Obviously, with the proposed two techniques combined, the statistic prediction module can achieve the best accuracy, leading to optimal forecasting performance. Besides, both *residual learning* and *individual preference* contribute positive effects and the former one is much more important, without which SAN can even bring negative effects to the backbone model. These results validate the rationality of our thoughts about the characteristics of the mean value and also reveal the importance of accurate modeling of future statistics to SAN. Besides, SAN without individual modeling performs well on the ETT2 dataset but performs poorly on the Exchange dataset. Such a phenomenon reveals that the evolving trends of different scenarios vary, and it is required to model the complex relationships among multiple variables individually. Moreover, since we only incorporate the properties of mean values into a simple MLP network, how to design a proper mechanism or network architecture for statistics modeling is a promising direction for optimizing our method, and we leave such explorations for future work.

Slicing Length The slicing length is a key parameter of SAN. We aim to study the effect of different slicing lengths on our method. Ablation experiments are conducted by using SCINet as the backbone model under the long-term forecasting setting ($L_{out} = 60$ for the ILI dataset and $L_{out} = 720$ for the rest datasets). Each experiment is conducted three times with a fixed random seed. The forecasting errors and the corresponding standard deviation are presented in Table 4.

Table 4: Forecasting errors under the multivariant setting with respect to different slicing lengths. The best performance are highlighted in **bold**.

Slicing Length	6		12		24		48	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	0.210±0.002	0.305±0.002	0.207±0.002	**0.305±0.002**	**0.206±0.004**	0.307±0.003	0.208±0.002	0.307±0.001
Exchange	**0.892±0.028**	**0.712±0.013**	0.895±0.037	**0.712±0.017**	0.901±0.005	0.715±0.002	0.898±0.037	0.714±0.015
Traffic	0.612±0.001	**0.376±0.001**	0.608±0.002	**0.373±0.001**	**0.607±0.001**	0.381±0.001	0.611±0.002	0.382±0.002
Weather	**0.338±0.002**	0.366±0.002	**0.338±0.001**	**0.365±0.002**	0.340±0.001	0.367±0.001	0.339±0.001	0.366±0.001
ILI	**2.487±0.034**	**1.063±0.008**	2.680±0.055	1.118±0.015	n/a	n/a	n/a	n/a
ETTh1	0.491±0.002	0.475±0.001	**0.488±0.001**	0.474±0.001	0.489±0.004	**0.473±0.001**	0.492±0.004	0.474±0.002
ETTh2	0.440±0.001	0.465±0.001	**0.435±0.002**	0.460±0.002	0.437±0.007	**0.459±0.006**	0.443±0.007	0.462±0.004
ETTm1	0.495±0.043	0.469±0.024	**0.450±0.001**	**0.441±0.001**	0.611±0.218	0.503±0.084	0.463±0.006	0.448±0.003
ETTm2	**0.391±0.001**	0.406±0.001	**0.391±0.001**	**0.405±0.001**	0.392±0.001	**0.405±0.001**	0.403±0.009	0.415±0.006

Our heuristic selection of slicing length appears to be effective among the candidates, indicating that both artificially defined and actual periods are useful in selecting the optimal setting. Additionally, there were no significant performance differences observed under various settings, suggesting that SAN is resilient to changes in slicing length.

2.6 Detailed Results of the Comparison between SAN and Normalization Methods

In Table 5, we provide the detailed experimental results of the comparison between SAN and state-of-the-art normalization methods for non-stationary time series forecasting: RevIN [2], NST [3] and Dish-TS [1]. We re-implement the former two methods and Dish-TS is implemented by its official code[1].

The table clearly shows that SAN outperforms existing approaches in most cases, except for the Weather dataset. Considering that the Weather dataset is the most stationary dataset, the results suggest that SAN can better remove the non-stationary factors in the raw data, even leading to an over-stationary issue that degrades the performance.

Besides, Dish-TS performs poorly in the benchmark. While it addresses the distribution shift between input and horizon series, it fails to optimize both the coefficient network and backbone network for overlooking the intrinsic bi-level optimization target of distribution estimation and forecasting tasks. By adopting a joint training schema, Dish-TS disturbs both networks and results in poor performance in certain cases. On the opposite, SAN benefits from the proposed two-stage schema which decouples the two tasks. This allows for proper optimization of each component and leads to improved overall performance.

Table 5: Detailed results of the comparison between SAN and normalization methods. The best results are highlighted in **bold**.

Methods		FEDformer								Autoformer							
		+ SAN		+ RevIN		+ NST		+Dish-TS		+ SAN		+ RevIN		+ NST		+Dish-TS	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	**0.164**	**0.272**	0.172	0.278	0.172	0.279	0.175	0.284	**0.172**	**0.281**	0.179	0.286	0.179	0.285	0.179	0.290
	192	**0.179**	**0.286**	0.185	0.289	0.187	0.291	0.188	0.296	**0.195**	**0.300**	0.216	0.316	0.209	0.309	0.215	0.318
	336	**0.191**	**0.299**	0.200	0.304	0.202	0.307	0.209	0.319	**0.211**	**0.316**	0.233	0.331	0.246	0.335	0.244	0.343
	720	**0.230**	0.334	0.243	0.337	**0.230**	**0.326**	0.239	0.343	**0.236**	**0.335**	0.246	0.341	0.252	0.345	0.286	0.370
Exchange	96	**0.079**	**0.205**	0.148	0.279	0.145	0.275	0.131	0.263	**0.082**	**0.208**	0.166	0.295	0.177	0.304	0.225	0.341
	192	**0.156**	**0.295**	0.266	0.377	0.274	0.383	0.538	0.523	**0.157**	**0.296**	0.299	0.404	0.275	0.385	0.760	0.610
	336	**0.260**	**0.384**	0.428	0.484	0.437	0.488	0.667	0.591	**0.262**	**0.385**	0.448	0.496	0.442	0.490	0.707	0.628
	720	**0.697**	**0.633**	1.056	0.789	1.064	0.787	1.480	0.954	**0.689**	**0.629**	1.068	0.791	1.049	0.784	2.341	1.063
Traffic	96	**0.536**	**0.330**	0.613	0.347	0.612	0.348	0.613	0.350	**0.569**	**0.350**	0.643	0.354	0.645	0.354	0.652	0.363
	192	**0.565**	**0.345**	0.637	0.356	0.641	0.357	0.644	0.362	**0.594**	**0.364**	0.659	0.373	0.643	0.367	0.669	0.374
	336	**0.580**	**0.354**	0.652	0.363	0.654	0.363	0.659	0.370	**0.591**	**0.363**	0.662	0.371	0.665	**0.363**	0.683	0.376
	720	**0.607**	**0.367**	0.686	0.382	0.688	0.380	0.693	0.388	**0.623**	0.380	0.700	0.384	0.667	**0.373**	0.703	0.392
Weather	96	**0.179**	0.239	0.187	**0.234**	0.187	**0.234**	0.244	0.317	**0.194**	0.256	0.212	0.257	0.211	**0.254**	0.268	0.338
	192	0.234	0.296	**0.235**	**0.272**	**0.235**	**0.272**	0.320	0.380	**0.258**	0.316	0.264	**0.300**	0.265	0.301	0.376	0.421
	336	0.304	0.348	**0.287**	**0.307**	0.289	0.308	0.424	0.452	0.329	0.367	0.309	0.329	**0.303**	**0.324**	0.476	0.486
	720	0.400	0.404	0.361	0.353	**0.359**	**0.352**	0.604	0.553	0.440	0.438	0.377	0.367	**0.366**	**0.357**	0.612	0.560
ILI	24	**2.461**	**1.075**	3.152	1.141	3.190	1.145	2.829	1.074	**2.548**	**1.098**	3.623	1.244	3.652	1.269	3.283	1.174
	36	**2.095**	**0.937**	2.498	0.990	2.615	1.026	2.595	0.972	**2.102**	**0.942**	2.767	1.074	2.394	1.005	2.792	1.054
	48	**2.107**	**0.936**	2.430	0.977	2.526	1.003	2.547	0.969	**2.103**	**0.932**	2.585	1.039	2.303	0.990	2.401	0.969
	60	**2.234**	**0.981**	2.822	1.084	2.891	1.109	2.866	1.066	**2.313**	**0.994**	2.693	1.068	2.489	1.008	2.681	1.042
ETTh1	96	**0.383**	**0.409**	0.392	0.413	0.394	0.414	0.390	0.424	0.522	0.474	0.491	0.463	0.550	0.503	**0.456**	**0.454**
	192	**0.431**	**0.438**	0.443	0.444	0.441	0.442	0.441	0.458	0.498	**0.472**	0.513	0.478	0.530	0.492	**0.495**	0.480
	336	**0.471**	**0.456**	0.495	0.467	0.485	0.466	0.495	0.486	0.571	0.509	0.528	0.485	**0.524**	**0.484**	0.539	0.496
	720	**0.504**	**0.488**	0.520	0.498	0.505	0.496	0.519	0.509	0.555	0.514	0.543	0.510	**0.510**	**0.491**	0.563	0.522
ETTh2	96	**0.300**	**0.355**	0.380	0.402	0.381	0.403	0.806	0.589	**0.316**	**0.366**	0.411	0.410	0.394	0.398	1.100	0.670
	192	**0.392**	**0.413**	0.457	0.443	0.478	0.453	0.936	0.659	**0.413**	**0.426**	0.478	0.450	0.473	0.450	0.976	0.672
	336	**0.459**	**0.462**	0.515	0.479	0.561	0.499	1.039	0.702	**0.446**	**0.457**	0.545	0.493	0.528	0.490	1.521	0.783
	720	**0.462**	**0.472**	0.507	0.487	0.502	0.481	1.237	0.759	**0.471**	**0.474**	0.523	0.490	0.524	0.498	1.105	0.745
ETTm1	96	**0.311**	**0.355**	0.340	0.385	0.336	0.382	0.348	0.397	**0.343**	**0.378**	0.458	0.446	0.468	0.448	0.477	0.460
	192	**0.351**	**0.383**	0.390	0.411	0.386	0.409	0.406	0.428	**0.390**	**0.400**	0.560	0.491	0.526	0.468	0.545	0.488
	336	**0.390**	**0.407**	0.432	0.436	0.438	0.441	0.438	0.450	**0.415**	**0.418**	0.607	0.508	0.786	0.559	0.650	0.533
	720	**0.456**	**0.444**	0.497	0.466	0.483	0.460	0.497	0.481	**0.476**	**0.453**	0.623	0.526	0.564	0.501	0.595	0.518
ETTm2	96	**0.175**	**0.266**	0.192	0.272	0.191	0.272	0.394	0.395	0.236	0.317	**0.233**	**0.307**	0.253	0.323	0.976	0.572
	192	**0.246**	**0.315**	0.270	0.320	0.270	0.321	0.552	0.472	**0.260**	**0.329**	0.288	0.337	0.289	0.335	0.532	0.485
	336	**0.315**	**0.362**	0.348	0.367	0.353	0.371	0.808	0.601	**0.330**	0.376	0.345	0.370	0.339	**0.365**	0.795	0.592
	720	**0.412**	0.422	0.430	**0.415**	0.445	0.422	1.282	0.771	**0.417**	0.428	0.434	**0.419**	0.426	0.432	1.271	0.768

[1] https://github.com/weifantt/Dish-TS

3 Implementation Details

3.1 Architecture of Statistic Prediction Module

The computation of MLP(x_1, x_2) in our paper can be summarized as follows:

$$\begin{aligned} x_1 &= act_1(W_1 * x_1) \\ x_2 &= act_1(W_2 * x_2) \\ x &= [x_1; x_2] \\ output &= act_2(W_3 * x) \end{aligned} \tag{4}$$

Here, the symbol [*;*] represents the concatenate operation. We set $act_1(), act_2() = Relu(), Relu()$ for standard deviation and the activate function of the mean is set to $Tanh(), Identity()$ respectively. W_1, W_2, W_3 are learnable transformation matrices with hidden sizes of {512,512,1024}.

3.2 Algorithm of The Two-stage Training Schema

To apply SAN to backbone forecasting models, we propose a two-stage training schema to tackle the challenge of the bi-level optimization target. The statistics prediction module is first trained into convergence, which is then frozen and treated as a plugin during the second stage of training the forecasting model. We provide the pseudo-code of such a procedure in Alg. 1.

Algorithm 1 Two-stage Training Schema.

Require: Input series $X = \{\boldsymbol{x}^i\}_{i=1}^N$; Horizon series $Y = \{\boldsymbol{y}^i\}_{i=1}^N$; Slicing length T

Initialize parameters ϕ, θ
while not converge **do**
  for all input $\boldsymbol{x}^i \in X$, horizon $\boldsymbol{y}^i \in Y$ **do**
    Compute input statistics μ_j^i, σ_j^i by Eq. 1 with T
    Predict future statistics $\hat{\boldsymbol{\mu}}^i, \hat{\boldsymbol{\sigma}}^i$ by Eq. 3 using $f_\phi(*)$
    Update ϕ using loss function l_{sp}
  end for
end while ▷ Training of the statistics prediction module

while not converge **do**
  for all input $\boldsymbol{x}^i \in X$, horizon $\boldsymbol{y}^i \in Y$ **do**
    Compute input statistics μ_j^i, σ_j^i by Eq. 1 with T
    Normalize input series to $\bar{\boldsymbol{x}}^i$ by Eq. 2
    Forecast $\bar{\boldsymbol{y}}^i = g_\theta(\bar{\boldsymbol{x}}^i)$
    Predict future statistics $\hat{\boldsymbol{\mu}}^i, \hat{\boldsymbol{\sigma}}^i$ by Eq. 3 using $f_\phi(*)$
    $\hat{\boldsymbol{\mu}}^i$.detach(), $\hat{\boldsymbol{\sigma}}^i$.detach() ▷ Stop-gradient, freeze the statistics prediction module
    Denormalize $\bar{\boldsymbol{y}}^i$ to $\hat{\boldsymbol{y}}^i$ by Eq. 4
    Update θ using loss function l_{fc}
  end for
end while ▷ Training of the forecasting model

4 Limitations

Though SAN shows promising performance on the benchmark dataset, there are still some limitations of this method. First is that we mainly select the slicing length heuristically or search in predefined candidates and the current design cannot handle indivisible length or the multi-period characteristic of time series. Such a solution works for the experiments but lacks generality in real-world applications. Second is that SAN may lead to an over-stationary issue, leading to sub-optimal performance. Therefore, a more flexible solution with automatic slicing length selection and normalization intensity control will be our exploring direction.

References

[1] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting. *arXiv preprint arXiv:2302.14829* (2023).

[2] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

[3] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary Transformers: Rethinking the Stationarity in Time Series Forecasting. *arXiv preprint arXiv:2205.14415* (2022).

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

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

[6] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*.

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

[8] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)* (Baltimore, Maryland).