SMARTformer: Semi-Autoregressive Transformer with Efficient Integrated Window Attention for Long Time Series Forecasting

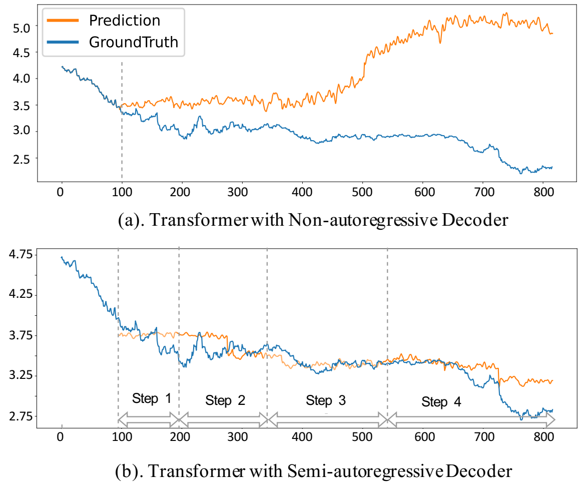
Yiduo Li[1] , **Shiyi Qi**[1] , **Zhe Li**[1] , **Zhongwen Rao**[2] , **Lujia Pan**[2] and **Zenglin Xu**[1,3]
[1]School of Computer Science and Technology, Harbin Institute of Technology Shenzhen, China
[2]Huawei Noah's Ark Lab, Shenzhen, China
[3]Department of Network Intelligence, Peng Cheng Lab, Shenzhen, China
{liyiduo5, syqi12138, plum271828,zenglin}@gmail.com, {raozhongwen, panlujia}@huawei.com

Abstract

The success of Transformers in long time series forecasting (LTSF) can be attributed to their attention mechanisms and non-autoregressive (NAR) decoder structures, which capture long-range dependencies. However, time series data also contain abundant local temporal dependencies, which are often overlooked in the literature and significantly hinder forecasting performance. To address this issue, we introduce SMARTformer, which stands for **S**e**M**i-**A**uto**R**egressive **T**ransformer. SMARTformer utilizes the Integrated Window Attention (IWA) and Semi-AutoRegressive (SAR) Decoder to capture global and local dependencies from both encoder and decoder perspectives. IWA conducts local self-attention in multi-scale windows and global attention across windows with linear complexity to achieve complementary clues in local and enlarged receptive fields. SAR generates subsequences iteratively, similar to autoregressive (AR) decoding, but refines the entire sequence in a NAR manner. This way, SAR benefits from both the global horizon of NAR and the local detail capturing of AR. We also introduce the Time-Independent Embedding (TIE), which better captures local dependencies by avoiding entanglements of various periods that can occur when directly adding positional embedding to value embedding. Our extensive experiments on five benchmark datasets demonstrate the effectiveness of SMARTformer against state-of-the-art models, achieving an improvement of **10.2%** and **18.4%** in multivariate and univariate long-term forecasting, respectively.

1 Introduction

Multivariate Time Series Forecasting, as an interdisciplinary research prevalent in scientific and engineering problems, has witnessed great advances in recent years, with a notable trend of predicting accurate series from short term [Li *et al.*, 2018; Liu *et al.*, 2018; Salinas *et al.*, 2020; Bai *et al.*, 2020; Deng *et al.*, 2021] to long term [Zhou *et al.*, 2021; Li *et al.*, 2023a; Li *et al.*, 2023b]. Recently, Transformer-based models [Vaswani *et al.*, 2017] have demonstrated great potentials on long time series forecasting (LTSF) for capturing long-range correlations.



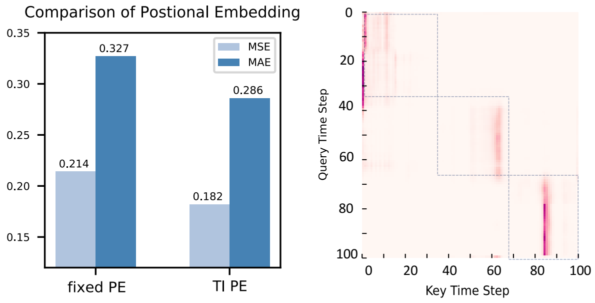
(a). Transformer with Non-autoregressive Decoder

(b). Transformer with Semi-autoregressive Decoder

Figure 1: Visualization on predicting 720 timesteps. (a) shows the prediction from Non-stationary Transformers [Liu *et al.*, 2022]. (b) shows the prediction by the equipment of our proposed Semi-Autoregressive Decoder, which generates non-overlapping segments through several steps recurrently. Segments from previous steps are utilized again to auxiliary the following forecast.

Despite these potentials, existing transformer-based models do not adequately consider the characteristics of time series data and are still suffering from ineffectiveness and inefficiency in capturing those local dependencies among complicated temporal patterns, impairing the ability to accurately model long time series. For instance, as shown in Figure 1a, when predicting a long time sequence with a size of 720 timesteps for the real-world Exchange-rate dataset [Lai *et al.*, 2018], the non-stationary transformer [Liu *et al.*, 2022] with a typical non-autoregressive decoder, leads to performance collapse in a single prediction step.

Moreover, positional embedding is also directly related to local dependencies. Due to the different scales in time series, directly adding positional embedding to the value embedding, as previous works [Zhou *et al.*, 2022a; Wu *et al.*, 2021], can result in entanglements of various periods, thus leading to



(a) (b)

Figure 2: (a) Experimental results demonstrates the effectiveness of Time-Independent Positional Embedding (TI PE) in LTSF. (b) Canonical self-attention scores from a 2-layer Transformer trained on the Exchange-rate dataset, showing that LTSF attentions tend to have strong local characteristics, and often be dominated by a few points.

confusion for future prediction. Therefore, we develop TIE to better capture the local dependencies.

We further investigate the vanilla attention mechanism's limitations by examining the attention maps among different timesteps computed on the Exchange rate dataset (Figure 2b). The attention map is highly sparse, which aligns with recent research [Zhou *et al.*, 2021]. They found that the attention feature map follows a long-tail distribution, where a few dot-product pairs contribute to attentions while most can be ignored. This limited attention mechanism hinders transformer models from learning better temporal correlations among timesteps, leading to significant computation costs that restrict the practical deployment of such models in real-world applications.

In summary, by investigating the architectures of Transformers for time series interaction, there are several major bottlenecks, i.e., the input embeddings that can incorporate more decoupled prior information, the attention mechanism that can efficiently and effectively model the interaction among different temporal patterns, and the decoder that can stably forecast with consistent long-range dependencies.

We propose the SMARTformer, which stands for **S**e**M**i-**A**uto**R**egressive **T**ransformer with Efficient Integrated Window Attention, to address issues in general LTSF tasks. The SMARTformer architecture is illustrated in Figure 3. To better decouple positional embedding from value embedding, we design a simple yet effective Time-Independent Embedding (TIE) method, which avoids entanglements of various periods and enhances the inductive bias of significant periodic variations. This improvement is shown on the Electricity Dataset in Figure 2a. To capture abundant latent correlations, we design the Integrated Window Attention (IWA), which separates an attention layer into two branches. One branch conducts local self-attention in non-overlapping multi-scale windows, while the other conducts global attention across windows. These designs achieve valuable temporal patterns (as shown in Table 7) and break the bottleneck of computational efficiency (as demonstrated in Figure 5). Furthermore, to enhance the decoder's power for outputting consistent long-range dependencies, we design a semi-autoregressive (SAR) decoder. The promising performance of SAR is demonstrated in Figure 1b, which enables the non-stationary Transformer to better fit the ground truth sequence. Extensive experiments on five public and commonly used multivariate time series datasets from different domains demonstrate the outstanding performance of the proposed SMARTformer.

2 Related Work

We review transformer models for LTSF according to the attention mechanism and decoder design.

Efficient Attentions in LTSF. To tackle LTSF tasks, designing effective self-attention mechanisms is crucial. According to attentions, Transformer variants can be roughly categorized into two types. The first type is Temporal Sparse Attention, which sparsifies attention with predefined patterns [Li *et al.*, 2019; Kitaev *et al.*, 2020; Liu *et al.*, 2021; Cirstea *et al.*, 2022]. Through reducing complexity, they are trapped by prefined structures, failing to accurately capture the correlations. The second type is Frequency Domain Attention, which fuses decomposition blocks with Fast Fourier Transform or other frequency analysis method to discover series-wise connections [Wu *et al.*, 2021; Zhou *et al.*, 2022a; Chen *et al.*, 2022]. However, converting data to frequency domain may inevitably lose fine temporal variations, thus leading to sub-optimal solutions. Based on the data distribution of the attention matrices (as illustrated in Figure 2a), the proposed Integrated Window Attention is designed to model local and global interaction at the same time, which is significantly different from existing methods.

Non-Autoregressive Decoding. Autoregressive (AR) decoding is widely used in NLP seq2seq models [Sutskever *et al.*, 2014] and Transformer-based pretraining models [Lewis *et al.*, 2019; Yang *et al.*, 2019]. Meanwhile, AR decoding dominates in short time series forecasting, [Qin *et al.*, 2017; Salinas *et al.*, 2020; Lai *et al.*, 2018; Li *et al.*, 2019]. However, for LTSF, AR decoding achieves unsatisfactory performance due to error accumulation, as mathematically proven in [Sun and Boning, 2022]. Thus, Informer initially adopted Non-autoregressive (NAR) decoding [Gu *et al.*, 2017] to avoid error accumulation and improve efficiency, which was also used in subsequent works (Autoformer [Wu *et al.*, 2021], FEDformer [Zhou *et al.*, 2022b], and Scaleformer [Shabani *et al.*, 2022]). Although these methods achieved encouraging results by injecting multi-scale or decomposition prior, they still neglected the disadvantages of NAR decoding itself. Our SAR decoder acts orthogonally to their contributions and can be easily adapted to them and other time series transformers to consistently enhance their performance.

3 Method

Given a D-variates time series $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T] \in \mathbb{R}^{T\times D}$ with T timesteps, the LTSF problem aims to predict $\tilde{W} = [\mathbf{w}_{T+1}, \mathbf{w}_{T+2}, \ldots, \mathbf{w}_{T+L}] \in \mathbb{R}^{L\times D}$ with L future timestep values. To tackle the LTSF tasks, we design the architecture of the proposed SMARTformer, as shown in Figure 3. In the following, we introduce the major components,

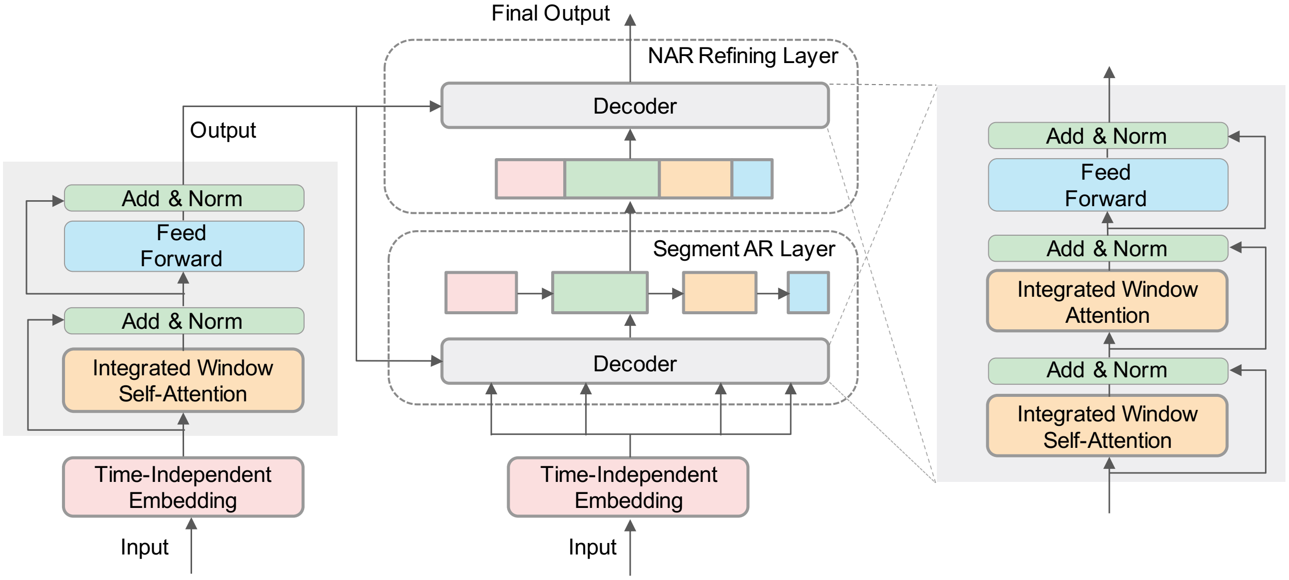


Figure 3: SMARTformer architecture. The left part is Encoder, where the Time-Independent Embedding is applied before it and the Integrated Window Attention is designed to capture comprehensive correlations. The middle part is the hierarchical Semi-Autoregressive (SAR) Decoder to handle reliable long sequences prediction, which stacks the Segment Auto-regressive (AR) Layer and the Non-Autoregressive (NAR) Refining Layer. The former focuses on decoding short sequences recurrently, while the latter is on refining the global context on top of it. Their detailed structure is shown in the right part.

i.e., the Time-Independent Embedding, the Integrated Window Attention, and the hierarchical Semi-autoregressive Decoder.

3.1 Time-Independent Embedding

In transformer-based time series forecasting, data embedding significantly impacts performance because attentions are insensitive to data order. To enhance the awareness of temporal variations, local positional and temporal information is often added [Zeng *et al.*, 2022]. Previous studies use fixed positional encoding or date-specific embedding to maintain ordering information. However, time series data often contain waveforms with different periods. Adding positional encoding composed of periodic waveforms may destroy original values by causing phase cancellation. We propose that positional encoding should be decoupled from values in time series embedding to reduce the distortion of original data.

To this end, we design a Time-Independent (TI) embedding method. Raw historical data are used as the encoder input $W_{1:T} = \{\boldsymbol{w}_{i,d} \mid 1 \le i \le T, 1 \le d \le D\}$, where $\boldsymbol{w_{i,d}}$ denotes the i-th timestep in dimension d. For each timestep, we embed them through two parts: the Value Embedding with a simple 1-D convolutional layer to embed each timestep of raw data to a C_v dimension vector, and the Time-Independent Positional Embedding denoted by F_i, calculated as the concatanation of three types of temporal features, i.e.,

$$F_i = (\mathbf{E}_i^{(m/h)} + \mathbf{E}_i^{(wk)} + \mathbf{E}_i^{(mth)}). \tag{1}$$

Here $\mathbf{E}^{(m/h)}, \mathbf{E}^{(wk)}, \mathbf{E}^{(mth)}$ denote three learnable projection matrices for positional embeddings of minute/hour, weekday and month, respectively. $F_i \in \mathbb{R}^{C_p}$ denotes the vector after positional embedding. Thus, the positional information is shared among timesteps of the same order on a daily/weekly/monthly basis to enhance the understanding of global temporal changes.

For each token $\boldsymbol{w}_i$, we concatenate and normalize these two embeddings (following [Ba *et al.*, 2016]) as

$$E_i = \text{norm}\left(\text{Conv}\left(\mathbf{w}_i\right) || \;\; F_i\right), \tag{2}$$

where $\text{Conv}\left(\mathbf{w}_i\right) \in \mathbb{R}^{C_v}$ denotes the latent variable after Value Embedding and $\mathbf{E}_i \in \mathbb{R}^C$ ($C = C_v + C_p$) denotes the input embedding. In this way, we leverage the positional information without destroying data semantics, acquiring global awareness of ultra-long temporal changes.

3.2 Integrated Window Attention

First, the input feature $X \in \mathbb{R}^{T \times C}$ is linearly projected to K heads, and our proposed attention splits them into two groups — one with S heads (for Intra-window attention) and the other with K-S heads (for Inter-window attention).

Intra-window Attention. For attention in windows, X is evenly partitioned into non-overlapping windows of equal length w along the time dimension. Here, we assume w is divisible by the whole length T. w can be adjusted to balance the learning capacity and computation complexity. Therefore, intra-window attention can be defined as

$$\begin{aligned} X &= \left[X^1, X^2, \ldots, X^M\right], \\ Y_k^i &= \text{Attention}\left(X^i W_k^Q, X^i W_k^K, X^i W_k^V\right), \\ \text{Intra-Attention}(X) &= \left[Y_k^1, Y_k^2, \ldots, Y_k^M\right], \end{aligned} \tag{3}$$

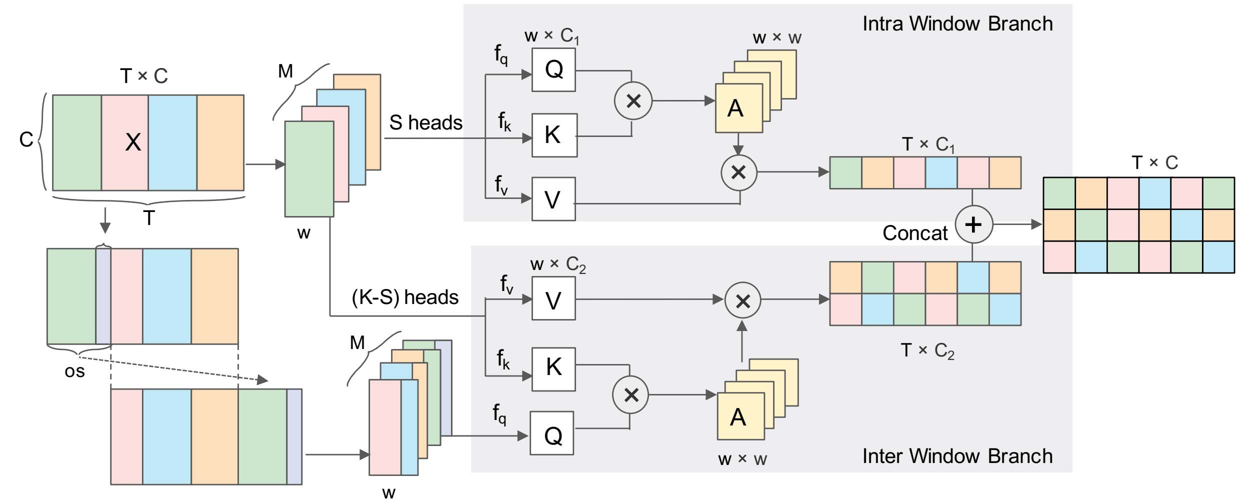


Figure 4: Framework of Integrated Window Attention with two branches where one branch (with S heads) conducts local self-attention in non-overlapping windows, while the other branch (with $K - S$ heads) conducts global attention inter windows.

where $X^i \in \mathbb{R}^{\frac{T}{M} \times C}$ and $M = T/w$ $(i = 1, 2, \ldots, M)$. $W_k^Q \in \mathbb{R}^{C \times d_k}$, $W_k^K \in \mathbb{R}^{C \times d_k}$, and $W_k^V \in \mathbb{R}^{C \times d_k}$ represent the projection matrices of queries, keys, and values for the k-th head respectively where d_k is set as C/K. Besides, the window length w is an important parameter because it achieves strong modeling capability while limiting the computation cost, where we adjust by applying small lengths for shallow layers and larger lengths for deep layers respectively to capture multi-scale characteristics.

Inter-window Attention. For remaining $K - S$ heads, we perform Inter-Attention, where we shift the original sequence X into X_s with os timesteps, and split it through the same window size w as the former branch to acquire $\hat{X}$ as

$$\begin{aligned} &X_s = (X[os : T, 0 : C] \;||\; X[0 : os, 0 : C]), \\ &\hat{X} = \left[X_s^1, X_s^2, \ldots, X_s^M\right]. \end{aligned} \tag{4}$$

The shifted sequence $\hat{X}$ is projected as Queries and X is projected as Keys and Values. Then, we acquire the k-th head attention scores Y_k^i for the i-th window as

$$\begin{aligned} &Y_k^i = \text{Attention}\left(\hat{X}^i W_k^Q, X^i W_k^K, X^i W_k^V\right), \\ &\text{Inter-Attention}(X) = \left[Y_k^1, Y_k^2, \ldots, Y_k^M\right], \end{aligned} \tag{5}$$

where $e \cdot w < os < (e+1) \cdot w, 0 < e < M$. os is an offset to establish powerful connections across different windows and enhance interactions among tokens at the edge of windows. Then, we integrate two branches to acquire the final output as

$$\begin{aligned} &\text{IntWin-Attention } (X) = (\mathrm{Y}_1 \,||\, \ldots || \,\mathrm{Y}_k), \\ &\mathrm{Y}_k = \begin{cases} \text{Intra-Attention}_k(X) & 1 \leq k \leq S \\ \text{Inter-Attention}_k(X) & S+1 \leq k \leq K. \end{cases} \end{aligned} \tag{6}$$

By doing so, the computation complexity of the Integrated Window Attention in two branches is both $\mathcal{O}(w \times L) = \mathcal{O}(L)$, scaling linearly with the input length L, reducing greatly to a lower complexity than the standard MSA and ensuring high throughput on GPUs. Moreover, another benefit of head-splitting is that the learnable parameters Q, K, and V are decomposed into two smaller matrices, which helps to reduce model parameters. And we apply different w window sizes for different layers, small w for early stages and larger w for later. Adjusting the length of w provides the flexibility to enlarge the attention area of each token. Thus, we reconcile the global and local self-attention in a single layer to acquire ample variations in an efficient and effective way.

3.3 Semi-Autoregressive Decoder

We describe our hierarchical SAR decoder to handle robust long sequence prediction, as shown in Figure 3. It stacks the Segment AR layer and the NAR refining layer. These layers share the identical structure of a typical Transformer decoder, except for using the IWA instead.

Segment AR Layer. Despite focusing on modeling a long sequence, local contexts also play a crucial role in knowledge propagation. Therefore, in the lower section of our network, we adopt a Segment AR Decoder Layer, which predicts each subsequence iteratively by applying the same decoder layer. For a long sequence with a length of L to predict in the M-th decoder layer, it generates non-overlapping segments through k steps with a length l_k. And the output of the M-th decoder layer $Z_{de}^M \in \mathbb{R}^{L \times C}$ is acquired by concatenating all the k-step representations along the time dimension as

$$Y_{de}^M = (\hat{Y}_{de(1)}^M || \; \hat{Y}_{de(2)}^M || \ldots || \; \hat{Y}_{de(k)}^M), \tag{7}$$

where $\hat{Y}_{de(j)}^M$ denotes the j-th step output.

In the j-th step, we assume L_j for the predicted length. A local representation $H_{(j)}^M \in \mathbb{R}^{L_j \times C}$ takes as the input, which comes from the predictions of previous steps and positional information of current step to capture local fine variations. $\bar{Y}_{de(j-1)}^M \in \mathbb{R}^{L_j \times C}$ denotes a slice from the previous predic-

Method		SMARTformer		Stationary		DLinear		FEDformer		Autoformer		Informer		LogTrans		LSSL		LSTM	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.171**	**0.224**	0.190	0.237	0.196	0.255	0.217	0.296	0.266	0.336	0.300	0.384	0.689	0.596	0.174	0.252	0.369	0.406
	192	**0.224**	**0.268**	0.241	0.281	0.237	0.296	0.276	0.336	0.307	0.367	0.598	0.544	0.658	0.589	0.238	0.313	0.416	0.435
	336	**0.283**	**0.313**	0.309	0.328	**0.283**	0.335	0.339	0.380	0.359	0.395	0.578	0.523	0.797	0.652	0.287	0.355	0.455	0.454
	720	**0.361**	**0.358**	0.406	0.394	0.375	0.381	0.403	0.428	0.419	0.428	1.059	0.741	0.869	0.675	0.384	0.415	0.535	0.520
Traffic	96	**0.583**	**0.317**	0.612	0.338	0.650	0.396	0.587	0.366	0.613	0.388	0.719	0.391	0.684	0.384	0.798	0.436	0.843	0.453
	192	**0.592**	**0.322**	0.613	0.340	0.598	0.370	0.604	0.373	0.616	0.382	0.696	0.379	0.685	0.390	0.849	0.481	0.847	0.453
	336	**0.608**	**0.333**	0.618	0.328	0.609	0.373	0.621	0.383	0.622	0.337	0.777	0.420	0.734	0.408	0.828	0.476	0.853	0.455
	720	**0.638**	**0.346**	0.653	0.355	0.645	0.394	0.641	0.382	0.660	0.408	0.864	0.472	0.717	0.396	0.854	0.489	1.500	0.805
Electricity	96	**0.163**	**0.269**	0.177	0.284	0.197	0.282	0.193	0.297	0.201	0.317	0.274	0.368	0.258	0.357	0.300	0.392	0.375	0.437
	192	**0.171**	**0.277**	0.205	0.304	0.196	0.285	0.195	0.308	0.222	0.334	0.298	0.389	0.266	0.368	0.297	0.390	0.442	0.473
	336	**0.191**	**0.292**	0.221	0.324	0.209	0.301	0.212	0.313	0.231	0.338	0.307	0.399	0.280	0.380	0.317	0.403	0.439	0.473
	720	**0.203**	**0.306**	0.244	0.341	0.245	0.333	0.246	0.355	0.254	0.361	0.373	0.439	0.283	0.376	0.338	0.417	0.980	0.814
Exchange	96	0.109	0.233	0.111	0.237	**0.088**	**0.218**	0.139	0.276	0.197	0.323	0.847	0.752	0.968	0.812	0.395	0.474	1.453	1.049
	192	0.186	**0.314**	0.239	0.345	**0.176**	0.315	0.195	0.308	0.222	0.334	1.204	0.895	1.040	0.851	0.776	0.698	1.846	1.179
	336	0.348	**0.421**	0.421	0.476	**0.320**	0.427	0.426	0.464	0.509	0.524	1.672	1.036	1.659	1.081	1.029	0.797	2.136	1.231
	720	**0.789**	**0.649**	1.082	0.804	0.839	0.695	1.090	0.800	1.447	0.941	2.478	1.310	1.941	1.127	2.283	1.222	2.984	1.427
ILI	24	**2.284**	**0.933**	2.494	1.065	2.398	1.040	3.228	1.260	3.483	1.287	5.764	1.677	4.480	1.444	4.381	1.425	5.914	1.734
	36	**1.770**	**0.845**	1.877	0.883	2.646	1.088	2.679	1.080	3.103	1.148	4.755	1.467	4.799	1.467	4.442	1.416	6.631	1.845
	48	**1.897**	**0.897**	2.010	0.900	2.614	1.086	2.622	1.078	2.669	1.085	4.763	1.469	4.800	1.468	4.559	1.443	6.736	1.857
	60	**1.877**	**0.891**	2.178	0.963	2.804	1.146	2.857	1.157	2.770	1.125	5.264	1.564	5.278	1.560	4.651	1.474	6.870	1.879

Table 1: Multivariate long-term series forecasting results on five datasets with a fixed input length I = 96 and prediction length $O \in \{96, 192, 336, 720\}$ (For ILI dataset, input length I = 36 and prediction length $O \in \{24, 36, 48, 60\}$). A lower MSE/MAE indicates better forecasting performance. The best results are highlighted in bold and the second best are underlined.

Method		SMARTformer		Stationary		DLinear		FEDformer		Autoformer		Informer		LogTrans		LSSL		LSTM	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.0012**	**0.026**	0.0013	0.027	0.0057	0.063	0.0035	0.046	0.0110	0.081	0.0038	0.044	0.0046	0.052	0.0067	0.065	0.0051	0.060
	192	**0.0014**	**0.028**	0.0016	0.030	0.0061	0.066	0.0054	0.059	0.0075	0.067	0.0023	0.040	0.0056	0.060	0.0061	0.067	0.0054	0.056
	336	**0.0015**	**0.029**	0.0016	0.030	0.0064	0.068	0.0041	0.050	0.0063	0.062	0.0041	0.049	0.0060	0.054	0.0034	0.038	0.0063	0.065
	720	**0.0019**	**0.033**	0.0021	0.034	0.0067	0.070	0.0150	0.091	0.0085	0.070	0.0031	0.042	0.0071	0.063	0.0072	0.073	0.0054	0.060
Traffic	96	**0.152**	**0.247**	0.188	0.281	0.242	0.317	0.170	0.263	0.246	0.346	0.257	0.353	0.226	0.317	0.194	0.290	0.542	0.531
	192	**0.154**	**0.248**	0.192	0.287	0.205	0.279	0.173	0.265	0.266	0.370	0.299	0.376	0.314	0.408	0.172	0.272	0.551	0.535
	336	**0.161**	**0.267**	0.197	0.295	0.199	0.276	0.178	0.266	0.263	0.371	0.312	0.387	0.387	0.453	0.178	0.278	0.555	0.536
	720	**0.162**	**0.262**	0.217	0.311	0.221	0.296	0.187	0.286	0.269	0.372	0.366	0.436	0.491	0.437	0.263	0.386	0.989	0.801

Table 2: Univariate long-term series forecasting results on two typical datasets with a fixed input length I = 96 and prediction length $O \in \{96, 192, 336, 720\}$. A lower MSE/MAE indicates better forecasting performance. The best results are highlighted in bold and the second best are underlined.

tion $\hat{Y}^{M}_{de(j-1)} \in \mathbb{R}^{L_{j-1} \times C}$ which we pad or slice to a proper length when $j = 1$ or $l_{j-1} \neq l_j$ to align them in the time dimension. The positional embeddings $\boldsymbol{F}_{(j)} \in \mathbb{R}^{L_j \times C_p}$ utilize TIE to enhance the awareness of complicated periodic variations. We concatenate them along the channel dimension and learn through an MLP as

$$H^{M}_{(j)} = \text{MLP}([F_{(j)} || \ \bar{Y}^{M}_{de(j-1)}]). \tag{8}$$

The equation to perform in the Segment AR decoder can be summarized as

$$\hat{Y}^{M}_{de(j)} = \text{Decoder}(H^{M}_{(j)}, Y^{N}_{en}), \tag{9}$$

where Y^{N}_{en} denotes the final output after all the N layers encoder. As for the predicted length for each step, it can be determined based on the sampling frequency of the dataset. Selecting a length that is an integer multiple of the dataset period (day/ week) can help with model prediction.

NAR Refining Layer. We introduce a NAR Refining Layer to add global representational power beyond the Segment AR Layer. The input of the M+1-th Refining Layer comes from the hidden states of the prior layer $Y^{M}_{de} \in \mathbb{R}^{L \times C}$. And the equations of the M+1-th decoder layer can be summarized as

$$Y^{M+1}_{de} = \text{Decoder}(Y^{M}_{de}, Y^{N}_{en}). \tag{10}$$

We stack these two types of layers interchangeably to capture both global and local contexts efficiently. Such design benefits from both the global horizon of the NAR decoding and the local detail capturing of the AR decoding to enhance the power of the decoder for capturing short and long dependencies steadily.

4 Experiments

4.1 Main Results

We conduct extensive experiments to evaluate the performance of SMARTformer and further perform ablation studies to justify how each component contributes to the final results.

Method		Informer		+SAR decoder		Promotion	Autoformer		+SAR decoder		Promotion	FEDformer		+SAR decoder		Promotion
Metric		MSE	MAE	MSE	MAE	Ratio	MSE	MAE	MSE	MAE	Ratio	MSE	MAE	MSE	MAE	Ratio
Electricity	96	0.274	0.368	**0.268**	**0.361**	2.04%	0.201	0.317	**0.184**	**0.299**	7.09%	0.193	0.297	**0.174**	**0.279**	7.95%
	192	0.298	0.389	**0.295**	**0.384**	1.30%	0.222	0.334	**0.195**	**0.308**	11.14%	0.195	0.308	**0.187**	**0.291**	4.81%
	336	0.307	0.399	**0.301**	**0.391**	2.05%	0.231	0.338	**0.204**	**0.318**	9.76%	0.212	0.313	**0.198**	**0.295**	6.17%
	720	0.373	0.439	**0.305**	**0.385**	14.03%	0.254	0.361	**0.229**	**0.338**	8.86%	0.246	0.355	**0.217**	**0.319**	10.96%
	960	0.334	0.405	**0.324**	**0.394**	2.79%	0.272	0.369	**0.246**	**0.350**	8.04%	0.250	0.354	**0.236**	**0.334**	5.64%
	1200	0.348	0.420	**0.329**	**0.403**	4.22%	0.283	0.382	**0.261**	**0.362**	6.98%	0.265	0.370	**0.249**	**0.354**	5.18%
Traffic	96	0.719	0.391	**0.651**	**0.356**	9.20%	0.613	0.388	**0.583**	**0.358**	6.31%	0.587	0.366	**0.583**	**0.317**	7.03%
	192	0.696	0.379	**0.663**	**0.363**	4.98%	0.616	0.382	**0.592**	**0.358**	5.37%	0.604	0.373	**0.593**	**0.320**	8.01%
	336	0.777	0.420	**0.693**	**0.382**	12.12%	0.622	0.337	**0.618**	**0.335**	0.62%	0.621	0.383	**0.611**	**0.333**	7.33%
	720	0.864	0.472	**0.728**	**0.393**	19.39%	0.660	0.408	**0.637**	**0.372**	5.94%	0.641	0.382	**0.637**	**0.353**	4.10%
	960	0.799	0.434	**0.748**	**0.409**	6.81%	0.649	0.395	**0.645**	**0.394**	0.44%	0.647	0.394	**0.639**	**0.369**	3.79%
	1200	0.901	0.492	**0.819**	**0.447**	10.01%	0.653	0.396	**0.642**	**0.392**	1.36%	0.651	0.400	**0.641**	**0.378**	3.51%

Table 3: Performance promotion by applying our proposed Semi-Autoregressive Decoder to Transformers. We report the averaged MSE/MAE of all prediction lengths and the relative promotion ratios by our decoder.

Method		T		T+SAR		T+TIE		T+TIE+SAR		T+TIE+IWA		SMARTformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.201	0.250	0.185	0.233	0.183	0.234	0.176	0.228	0.181	0.231	**0.171**	**0.224**
	192	0.260	0.297	0.234	0.279	0.233	0.277	0.228	0.271	0.231	0.277	**0.224**	**0.268**
	336	0.321	0.340	0.302	0.325	0.304	0.328	0.289	0.315	0.303	0.319	**0.283**	**0.313**
	720	0.405	0.388	0.377	0.370	0.381	0.370	0.368	0.366	0.380	0.373	**0.361**	**0.358**

Table 4: Ablation study of the components in SMARTformer: Time-Independent Embedding (TIE), Integrated Window Attention (IWA), and Semi-Autoregressive decoder (SAR); T denotes Transformer.

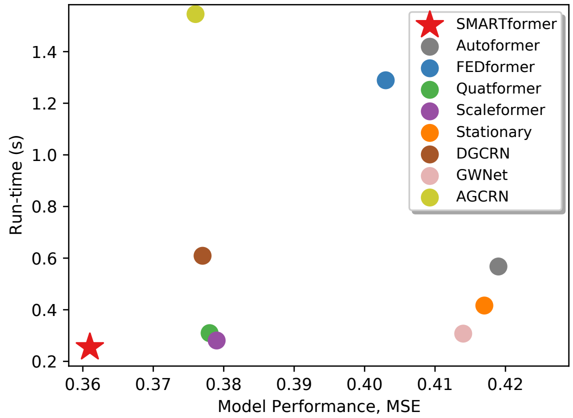


Figure 5: Running Time Efficiency Analysis. We compare the mean running time per iteration in the training phase and their forecasting performance, evaluated by MSE.

Datasets. We perform empirical studies on five real-world benchmark datasets as follows: (1) Electricity. (2) Weather[2]. (3) Traffic[3]. (4) Exchange[Lai *et al.*, 2018]. (5) ILI [4]. The train/val/test splits for the first three datasets are the same as [Zhou *et al.*, 2021], the last two are split by the ratio of 7:1:2 following [Wu *et al.*, 2021].

Baselines. We select nine strong recent baselines, including: Transformer-based models: Non-stationary Transformers [Liu *et al.*, 2022], FEDformer [Zhou *et al.*, 2022a], Autoformer [Wu *et al.*, 2021], Informer [Zhou *et al.*, 2021]and LogTrans [Li *et al.*, 2019]; MLP-based models: DLinear [Zeng *et al.*, 2022]; RNN-based model LSSL [Gu *et al.*, 2022] and LSTM [Hochreiter and Schmidhuber, 1997].

Forecasting Results. As shown in Table 1, for multivariate time series forecasting, SMARTformer outperforms other deep models impressively in all benchmarks, with 37 top-1 and 40 top-2 cases out of 40 in total. Compared with SOTA works (Non-stationary Transformers, Dlinear and FEDformer), ours yields an overall **10.2%**, **10.1%** and **17.7%** relative MSE reduction. We also list the univariate results of two typical datasets in Table 2. Compared with SOTA works (Non-stationary Transformers and Dlinear), ours yields an overall **13.3%** and **51.3%** relative MSE reduction. And on the Weather dataset, the improvement can reach more than **70%**. It proves the effectiveness of SMARTformer in long-term forecasting.

Implementation details. All the experiments are implemented in PyTorch 1.9.1 [Paszke *et al.*, 2019] and conducted for three runs on a single NVIDIA 3090 GPU. Following previous works, the input sequence length is set to 36 for ILI and 96 for the other four datasets. We also use a data normalization layer RevIN [Kim *et al.*, 2022] as a pre-processing block to further enhance the model's robustness. Each model is trained by ADAM [Kingma and Ba, 2015] using L2 loss and batch size of 32. The model contains 3 encoder layers and 2 decoder layers. And we have an additional set of hyperparameters. In IWA, the shift length $os = (M/2+1/2) \times w$, the window size $w \in \{24, 36, 48\}$. And in TIE, the dim of value

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

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

[4] ILI. https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html.

Method		fixed PE		w/o PE		TIE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.193	0.308	0.368	0.430	**0.163**	**0.269**
	192	0.201	0.315	0.379	0.437	**0.171**	**0.277**
	336	0.214	0.329	0.388	0.445	**0.191**	**0.292**
	720	0.246	0.355	0.396	0.462	**0.203**	**0.306**

Table 5: Comparisons of Positional Embedding methods.

Method		NAR		Segment AR		SAR	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.172	0.276	0.167	0.271	**0.163**	**0.269**
	192	0.195	0.301	0.180	0.286	**0.171**	**0.277**
	336	0.218	0.322	0.208	0.312	**0.191**	**0.292**
	720	0.233	0.334	0.230	0.331	**0.203**	**0.306**

Table 6: Comparisons of components in SAR.

Method		Intra-Window		Inter-Window		IWA	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.164	0.269	0.167	0.271	**0.163**	**0.269**
	192	0.177	0.283	0.178	0.286	**0.171**	**0.277**
	336	0.201	0.307	0.196	0.299	**0.191**	**0.292**
	720	0.223	0.321	0.208	0.310	**0.203**	**0.306**

Table 7: Comparisons of components in IWA.

Param		e			s		
		no shift	w/4	w/2	0	1	M/2
192	MSE	0.234	0.227	**0.224**	0.239	0.226	**0.224**
	MAE	0.282	0.271	**0.268**	0.287	0.270	**0.268**
720	MSE	0.366	0.363	**0.361**	0.368	0.367	**0.361**
	MAE	0.364	0.362	**0.358**	0.366	0.365	**0.358**

Table 8: Impact of shift (os) in IWA, $os = e \times w + s$. w is the window size and M is the number of windows. When e varies, s is fixed to $M/2$. When s varies, e is $w/2$.

embedding $C_v = 3/4 \times D$, the dim of Positional Embedding $C_p = D/4$, D denotes the dim of the model. In the segment AR layer of SAR, the subsequence length is $L/4$ each time.

Semi-Autoregressive Decoder Generity. We apply our SAR decoder to mainstream Transformers (Informer, Autoformer, and FEDformer) and report the performance promotion of each model (Table 3). Our method consistently improves the forecasting ability of different benchmarks. Overall, it achieves averaged **7.7%**, **6.2%**, and **6.3%** promotion on Informer, Autoformer, and FEDformer, making each of them surpass previous state-of-the-art, which validates that our SAR decoder is an effective framework that can be widely applied to Transformer-based models in long-term forecasting to enhance their performance. It is worth noting that with the predicted length increasing, the performance of the SAR decoder changes quite steadily, implying its robustness in extremely long-term forecasting.

4.2 Ablation Study and Analysis

Ablation Study. We perform an ablation study on the Weather dataset to study the effects of our three main modules: TIE, IWA, and SAR. We consider Transformer as the baseline. As shown in Table 4, we observe that each module is essential for accurate prediction where the accuracy drops more significantly when without SAR or TIE, which justifies our design choices. And we note that when replacing IWA with canonical attention, the accuracy loss is relatively smaller than removing other components. Another benefit of IWA is greatly reducing the computations thus improving efficiency.

Impact of SAR. As shown in Table 6, we evaluate the components of our SAR decoder. Using only the Segment AR Layer performs better than using only the NAR decoder, for providing more dependable local features. And combining both achieves the best results stably, proving the importance of modeling global and local features respectively.

Impact of TIE. In Table 5, we compare TIE with fixed positional embedding (fixed PE) and without positional embedding in SMARTformer. As can be observed, the forecasting performance declines rapidly without positional embedding. And our method outperforms fixed PE, which is well aligned with our design considerations for decoupling PE from value embedding.

Impact of IWA. We study the effects of two branches (Inter-Window and Intra-Window) in IWA. As shown in Table 7, we observe the necessity of both branches. When the predicted length is relatively small, the Intra-Window branch contributes more, and the Inter-Window branch is quite essential, especially in the relatively long predicted length.

Impact of os in IWA. As shown in Table 8, we evaluate the impact of shift os under different e and w situations on the Weather dataset. $s = w/2, e = M/2$ performs the best, because the design of e and w is essential to enhance interactions among tokens across windows.

Running Time Efficiency Analysis. As shown in Figure 5, we evaluate SMARTformer with 8 strong baselines, Transformer based models: Autoformer, FEDformer, Quatformer [Chen *et al.*, 2022], Scaleformer[Shabani *et al.*, 2022], Non-stationary Transformers (represented as Stationary) and GNN based models: DGCRN [Li *et al.*, 2021], GWNet [Wu *et al.*, 2019] and AGCRN [Bai *et al.*, 2020]. Experiments are conducted on the Weather Dataset with a 96 length input and a 720 predicted length output. Our SMARTformer is the fastest and best accurate among the 9 models.

5 Conclusion

Long time series forecasting is notably associated with the ability to stably capture local dependencies. Thus, we propose SMARTformer, consisting of three effective mechanisms, where Time-Independent Embedding enhances periodic variations, Integrated Window Attention achieves complementary clues in various receptive fields, and Semi-Autoregressive Decoder identifies solid local and global characteristics in two stages. Extensive experiments show that our framework achieves superior forecasting performance.

Ethical Statement

There are no ethical issues.

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China (No. 2018AAA0100204), a key program of fundamental research from Shenzhen Science and Technology Innovation Commission (No. JCYJ20200109113403826), the Major Key Project of PCL (No. 2022ZD0115301), and an Open Research Project of Zhejiang Lab (NO.2022RC0AB04).

References

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[Bai *et al.*, 2020] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

[Chen *et al.*, 2022] Weiqi Chen, Wenwei Wang, Bingqing Peng, Qingsong Wen, Tian Zhou, and Liang Sun. Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 146–156, 2022.

[Cirstea *et al.*, 2022] Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 1994–2001. ijcai.org, 2022.

[Deng *et al.*, 2021] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 269–278, 2021.

[Gu *et al.*, 2017] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

[Gu *et al.*, 2022] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Kim *et al.*, 2022] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jangho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2022.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Kitaev *et al.*, 2020] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Lai *et al.*, 2018] 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.

[Lewis *et al.*, 2019] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[Li *et al.*, 2018] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[Li *et al.*, 2019] 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.

[Li *et al.*, 2021] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2021.

[Li *et al.*, 2023a] Zhe Li, Zhongwen Rao, Lujia Pan, Pengyun Wang, and Zenglin Xu. Ti-mae: Self-supervised masked time series autoencoders. *CoRR*, abs/2301.08871, 2023.

[Li *et al.*, 2023b] Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *CoRR*, abs/2302.04501, 2023.

[Liu *et al.*, 2018] Hao Liu, Lirong He, Haoli Bai, Bo Dai, Kun Bai, and Zenglin Xu. Structured inference for recurrent hidden semi-markov model. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2447–2453, 2018.

[Liu *et al.*, 2021] 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 *International Conference on Learning Representations*, 2021.

[Liu *et al.*, 2022] 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.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[Qin *et al.*, 2017] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2627–2633. ijcai.org, 2017.

[Salinas *et al.*, 2020] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[Shabani *et al.*, 2022] Amin Shabani, Amir Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. *CoRR*, abs/2206.04038, 2022.

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

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

[Vaswani *et al.*, 2017] 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.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1907–1913. ijcai.org, 2019.

[Wu *et al.*, 2021] 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.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[Zeng *et al.*, 2022] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *CoRR*, abs/2205.13504, 2022.

[Zhou *et al.*, 2021] 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.

[Zhou *et al.*, 2022a] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR, 2022.

[Zhou *et al.*, 2022b] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR, 2022.