Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting

Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long (✉)
School of Software, BNRist, Tsinghua University, China
{whx20,xjh20}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

Abstract

Extending the forecasting time is a critical demand for real applications, such as extreme weather early warning and long-term energy consumption planning. This paper studies the long-term forecasting problem of time series. Prior Transformer-based models adopt various self-attention mechanisms to discover the long-range dependencies. However, intricate temporal patterns of the long-term future prohibit the model from finding reliable dependencies. Also, Transformers have to adopt the sparse versions of point-wise self-attentions for long series efficiency, resulting in the information utilization bottleneck. Going beyond Transformers, we design Autoformer as a novel decomposition architecture with an Auto-Correlation mechanism. We break with the pre-processing convention of series decomposition and renovate it as a basic inner block of deep models. This design empowers Autoformer with progressive decomposition capacities for complex time series. Further, inspired by the stochastic process theory, we design the Auto-Correlation mechanism based on the series periodicity, which conducts the dependencies discovery and representation aggregation at the sub-series level. Auto-Correlation outperforms self-attention in both efficiency and accuracy. In long-term forecasting, Autoformer yields state-of-the-art accuracy, with a 38% relative improvement on six benchmarks, covering five practical applications: energy, traffic, economics, weather and disease. Code is available at this repository: https://github.com/thuml/Autoformer.

1 Introduction

Time series forecasting has been widely used in energy consumption, traffic and economics planning, weather and disease propagation forecasting. In these real-world applications, one pressing demand is to extend the forecast time into the far future, which is quite meaningful for the long-term planning and early warning. Thus, in this paper, we study the long-term forecasting problem of time series, characterizing itself by the large length of predicted time series. Recent deep forecasting models [41, 17, 20, 28, 23, 29, 19, 35] have achieved great progress, especially the Transformer-based models. Benefiting from the self-attention mechanism, Transformers obtain great advantage in modeling long-term dependencies for sequential data, which enables more powerful big models [7, 11].

However, the forecasting task is extremely challenging under the long-term setting. First, it is unreliable to discover the temporal dependencies directly from the long-term time series because the dependencies can be obscured by entangled temporal patterns. Second, canonical Transformers with self-attention mechanisms are computationally prohibitive for long-term forecasting because of the quadratic complexity of sequence length. Previous Transformer-based forecasting models [41, 17, 20] mainly focus on improving self-attention to a sparse version. While performance is significantly improved, these models still utilize the point-wise representation aggregation. Thus, in the process of efficiency improvement, they will sacrifice the information utilization because of the sparse point-wise connections, resulting in a bottleneck for long-term forecasting of time series.

To reason about the intricate temporal patterns, we try to take the idea of decomposition, which is a standard method in time series analysis [11,27]. It can be used to process the complex time series and extract more predictable components. However, under the forecasting context, it can only be used as the pre-processing of past series because the future is unknown [15]. This common usage limits the capabilities of decomposition and overlooks the potential future interactions among decomposed components. Thus, we attempt to go beyond pre-processing usage of decomposition and propose a generic architecture to empower the deep forecasting models with immanent capacity of progressive decomposition. Further, decomposition can unravel the entangled temporal patterns and highlight the inherent properties of time series [15]. Benefiting from this, we try to take advantage of the series periodicity to renovate the point-wise connection in self-attention. We observe that the sub-series at the same phase position among periods often present similar temporal processes. Thus, we try to construct a series-level connection based on the process similarity derived by series periodicity.

Based on the above motivations, we propose an original Autoformer in place of the Transformers for long-term time series forecasting. Autoformer still follows residual and encoder-decoder structure but renovates Transformer into a decomposition forecasting architecture. By embedding our proposed decomposition blocks as the inner operators, Autoformer can progressively separate the long-term trend information from predicted hidden variables. This design allows our model to alternately decompose and refine the intermediate results during the forecasting procedure. Inspired by the stochastic process theory [8,24], Autoformer introduces an Auto-Correlation mechanism in place of self-attention, which discovers the sub-series similarity based on the series periodicity and aggregates similar sub-series from underlying periods. This series-wise mechanism achieves $O(L \log L)$ complexity for length-$L$ series and breaks the information utilization bottleneck by expanding the point-wise representation aggregation to sub-series level. Autoformer achieves the state-of-the-art accuracy on six benchmarks. The contributions are summarized as follows:

- To tackle the intricate temporal patterns of the long-term future, we present Autoformer as a decomposition architecture and design the inner decomposition block to empower the deep forecasting model with immanent progressive decomposition capacity.
- We propose an Auto-Correlation mechanism with dependencies discovery and information aggregation at the series level. Our mechanism is beyond previous self-attention family and can simultaneously benefit the computation efficiency and information utilization.
- Autoformer achieves a 38% relative improvement under the long-term setting on six benchmarks, covering five real-world applications: energy, traffic, economics, weather and disease.

2 Related Work

2.1 Models for Time Series Forecasting

Due to the immense importance of time series forecasting, various models have been well developed. Many time series forecasting methods start from the classic tools [32,9]. ARIMA [6,5] tackles the forecasting problem by transforming the non-stationary process to stationary through differencing. The filtering method is also introduced for series forecasting [18,10]. Besides, recurrent neural networks (RNNs) models are used to model the temporal dependencies for time series [36,26,40,22]. DeepAR [28] combines autoregressive methods and RNNs to model the probabilistic distribution of future series. LSTNet [19] introduces convolutional neural networks (CNNs) with recurrent-skip connections to capture the short-term and long-term temporal patterns. Attention-based RNNs [39,30,31] introduce the temporal attention to explore the long-range dependencies for prediction. Also, many works based on temporal convolution networks (TCN) [34,4,3,29] attempt to model the temporal causality with the causal convolution. These deep forecasting models mainly focus on the temporal relation modeling by recurrent connections, temporal attention or causal convolution.

Recently, Transformers [35,38] based on the self-attention mechanism shows great power in sequential data, such as natural language processing [11,27], audio processing [14] and even computer vision [12,21]. However, applying self-attention to long-term time series forecasting is computationally prohibitive because of the quadratic complexity of sequence length $L$ in both memory and time. LogTrans [20] introduces the local convolution to Transformer and proposes the LogSparse attention to select time steps following the exponentially increasing intervals, which reduces the complexity to $O(L(\log L)^2)$. Reformer [17] presents the local-sensitive hashing (LSH) attention and reduces the complexity to $O(L \log L)$. Informer [41] extends Transformer with KL-divergence based ProbSparse
We use the Transformer to a deep decomposition architecture (Figure 1), including the inner decomposition architecture (Figure 1), which can extract the long-term stationary trend from predicted intermediate results. As a standard method in time series analysis, time series decomposition deconstructs a time series into several components, each representing one of the underlying categories of patterns that are more predictable. It is primarily useful for exploring historical changes over time. For the forecasting tasks, decomposition is always used as the pre-processing of historical series before predicting future series, such as Prophet with trend-seasonality decomposition and N-BEATS with basis expansion and DeepGLO with matrix decomposition. However, such pre-processing is limited by the plain decomposition effect of historical series and overlooks the hierarchical interaction between the underlying patterns of series in the long-term future. This paper takes the decomposition idea from a new progressive dimension. Our Autoformer harnesses the decomposition as an inner block of deep models, which can progressively decompose the hidden series throughout the whole forecasting process, including both the past series and the predicted intermediate results.

3. Autoformer

The time series forecasting problem is to predict the most probable length-$O$ series in the future given the past length-$I$ series, denoting as input-$I$-predict-$O$. The long-term forecasting setting is to predict the long-term future, i.e. larger $O$. As aforementioned, we have highlighted the difficulties of long-term series forecasting: handling intricate temporal patterns and breaking the bottleneck of computation efficiency and information utilization. To tackle these two challenges, we introduce the decomposition as a built-in block to the deep forecasting model and propose Autoformer as a decomposition architecture. Besides, we design the Auto-Correlation mechanism to discover the period-based dependencies and aggregate similar sub-series from underlying periods.

3.1 Decomposition Architecture

We renovate Transformer to a deep decomposition architecture (Figure 1), including the inner series decomposition block, Auto-Correlation mechanism, and corresponding Encoder and Decoder.

**Series decomposition block** To learn with the complex temporal patterns in long-term forecasting context, we take the idea of decomposition, which can separate the series into trend-cyclical and seasonal parts. These two parts reflect the long-term progression and the seasonality of the series respectively. However, directly decomposing is unrealizable for future series because the future is just unknown. To tackle this dilemma, we present a series decomposition block as an inner operation of Autoformer (Figure 1), which can extract the long-term stationary trend from predicted intermediate hidden variables progressively. Concretely, we adapt the moving average to smooth out periodic fluctuations and highlight the long-term trends. For length-$L$ input series $X \in \mathbb{R}^{L \times d}$, the process is:

$$X_t = \text{AvgPool}(\text{Padding}(X))$$

$$X_s = X - X_t,$$

(1)

where $X_s, X_t \in \mathbb{R}^{L \times d}$ denote the seasonal and the extracted trend-cyclical part respectively. We adopt the AvgPool($\cdot$) for moving average with the padding operation to keep the series length unchanged. We use $X_s, X_t = \text{SeriesDecomp}(X)$ to summarize above equations, which is a model inner block.

**Model inputs** The inputs of encoder part are the past $I$ time steps $X_{\text{en}} \in \mathbb{R}^{I \times d}$. As a decomposition architecture (Figure 1), the input of Autoformer decoder contains both the seasonal part $X_{\text{des}} \in \mathbb{R}^{(I + O) \times d}$ and trend-cyclical part $X_{\text{det}} \in \mathbb{R}^{(I + O) \times d}$ to be refined. Each initialization consists of two parts: the component decomposed from the latter half of encoder’s input $X_{\text{en}}$ with length $I/2$ to provide recent information, placeholders with length $O$ filled by scalars. It’s formulized as follows:

$$X_{\text{ens}}, X_{\text{ent}} = \text{SeriesDecomp}(X_{\text{en}}_{I/2})$$

$$X_{\text{des}} = \text{Concat}(X_{\text{ens}}, X_0)$$

$$X_{\text{det}} = \text{Concat}(X_{\text{ent}}, X_{\text{Mean}}),$$

(2)
where $\mathbf{X}_{\text{en}}$, $\mathbf{X}_{\text{en}} \in \mathbb{R}^{l \times d}$ denote the seasonal and trend-cyclical parts of $\mathbf{X}_{\text{en}}$ respectively, and $\mathbf{X}_0$, $\mathbf{X}_{\text{Mean}} \in \mathbb{R}^{l \times d}$ denote the placeholders filled with zero and the mean of $\mathbf{X}_{\text{en}}$ respectively.

**Encoder** As shown in Figure 1, the encoder focuses on the seasonal part modeling. The output of the encoder contains the past seasonal information and will be used as the cross information to help the decoder refine prediction results. Suppose we have $N$ encoder layers. The overall equations for $l$-th encoder layer are summarized as $\mathbf{X}^{l-1}_{\text{en}} = \text{Encoder}(\mathbf{X}^{l-1}_{\text{en}})$. Details are shown as follows:

$$
\begin{align*}
\mathbf{S}^{l,1}_{\text{en}} &= \text{SeriesDecomp} \left( \text{Auto-Correlation}(\mathbf{X}^{l-1}_{\text{en}}) + \mathbf{X}^{l-1}_{\text{en}} \right) \\
\mathbf{S}^{l,2}_{\text{en}} &= \text{SeriesDecomp} \left( \text{FeedForward}(\mathbf{S}^{l,1}_{\text{en}}) + \mathbf{S}^{l,1}_{\text{en}} \right),
\end{align*}
$$

where “-” is the eliminated trend part. $\mathbf{X}^{l}_{\text{en}} = \mathbf{S}^{l,2}_{\text{en}}, l \in \{1, \cdots , N\}$ denotes the output of $l$-th encoder layer and $\mathbf{X}^{0}_{\text{en}}$ is the embedded $\mathbf{X}_{\text{en}}$. $\mathbf{S}^{l,i}_{\text{en}}, i \in \{1, 2\}$ represents the seasonal component after the $i$-th series decomposition block in the $l$-th layer respectively. We will give detailed description of Auto-Correlation($\cdot$) in the next section, which can seamlessly replace the self-attention.

**Decoder** The decoder contains two parts: the accumulation structure for trend-cyclical components and the stacked Auto-Correlation mechanism for seasonal components (Figure 1). Each decoder layer contains the inner Auto-Correlation and encoder-decoder Auto-Correlation, which can refine the prediction and utilize the past seasonal information respectively. Note that the model extracts the potential trend from the intermediate hidden variables during the decoder, allowing Autoformer to progressively refine the trend prediction and eliminate interference information for period-based dependencies discovery in Auto-Correlation. Suppose there are $M$ decoder layers. With the latent variable $\mathbf{X}^{N}_{\text{en}}$ from the encoder, the equations of $l$-th decoder layer can be summarized as $\mathbf{X}^{l}_{\text{de}} = \text{Decoder}(\mathbf{X}^{l-1}_{\text{de}}, \mathbf{X}^{N}_{\text{en}})$. The decoder can be formalized as follows:

$$
\begin{align*}
\mathbf{S}^{l,1}_{\text{de}}, \mathbf{T}^{l,1}_{\text{de}} &= \text{SeriesDecomp} \left( \text{Auto-Correlation}(\mathbf{X}^{l-1}_{\text{de}}) + \mathbf{X}^{l-1}_{\text{de}} \right) \\
\mathbf{S}^{l,2}_{\text{de}}, \mathbf{T}^{l,2}_{\text{de}} &= \text{SeriesDecomp} \left( \text{Auto-Correlation}(\mathbf{S}^{l,1}_{\text{de}}, \mathbf{X}^{N}_{\text{en}}) + \mathbf{S}^{l,1}_{\text{de}} \right) \\
\mathbf{S}^{l,3}_{\text{de}}, \mathbf{T}^{l,3}_{\text{de}} &= \text{SeriesDecomp} \left( \text{FeedForward}(\mathbf{S}^{l,2}_{\text{de}}) + \mathbf{S}^{l,2}_{\text{de}} \right) \\
\mathbf{T}^{l}_{\text{de}} &= \sum_{i=1}^{3} \mathbf{W}_{i} \cdot \mathbf{T}^{l-1}_{\text{de}} + \mathbf{W}_{i} \cdot \mathbf{T}^{l,2}_{\text{de}} + \mathbf{W}_{i} \cdot \mathbf{T}^{l,3}_{\text{de}},
\end{align*}
$$

where $\mathbf{X}^{l}_{\text{de}} = \mathbf{S}^{l,3}_{\text{de}}, l \in \{1, \cdots , M\}$ denotes the output of $l$-th decoder layer. $\mathbf{X}^{0}_{\text{de}}$ is embedded from $\mathbf{X}_{\text{des}}$ for deep transform and $\mathbf{T}^{0}_{\text{de}} = \mathbf{X}_{\text{det}}$ is for accumulation. $\mathbf{S}^{l,i}_{\text{de}}, \mathbf{T}^{l,i}_{\text{de}}, i \in \{1, 2, 3\}$ represent the seasonal component and trend-cyclical component after the $i$-th series decomposition block in the $l$-th layer respectively. $\mathbf{W}_{i} \cdot \mathbf{T}^{l,i}_{\text{de}}, i \in \{1, 2, 3\}$ represents the projector for the $i$-th extracted trend $\mathbf{T}^{l,i}_{\text{de}}$.

Figure 1: Autoformer architecture. The encoder eliminates the long-term trend-cyclical part by series decomposition blocks (blue blocks) and focuses on seasonal patterns modeling. The decoder accumulates the trend part extracted from hidden variables progressively. The past seasonal information from encoder is utilized by the encoder-decoder Auto-Correlation (center green block in decoder).
The final prediction is the sum of the two refined decomposed components, as $W_S \ast X_{de}^M + T_{de}^M$, where $W_S$ is to project the deep transformed seasonal component $X_{de}^M$ to the target dimension.

### 3.2 Auto-Correlation Mechanism

As shown in Figure 2, we propose the Auto-Correlation mechanism with series-wise connections to expand the information utilization. Auto-Correlation discovers the period-based dependencies by calculating the series autocorrelation and aggregates similar sub-series by time delay aggregation.

**Period-based dependencies** It is observed that the same phase position among periods naturally provides similar sub-processes. Inspired by the stochastic process theory [8, 24], for a real discrete-time process $\{X_t\}$, we can obtain the autocorrelation $R_{X,X}(\tau)$ by the following equations:

$$R_{X,X}(\tau) = \lim_{L \to \infty} \frac{1}{L} \sum_{t=1}^{L} X_t X_{t-\tau}. \quad (5)$$

$R_{X,X}(\tau)$ reflects the time-delay similarity between $\{X_t\}$ and its $\tau$ lag series $\{X_{t-\tau}\}$. As shown in Figure 2, we use the autocorrelation $R(\tau)$ as the unnormalized confidence of estimated period length $\tau$. Then, we choose the most possible $k$ period lengths $\tau_1, \cdots, \tau_k$. The period-based dependencies are derived by the above estimated periods and can be weighted by the corresponding autocorrelation.

**Time delay aggregation** The period-based dependencies connect the sub-series among estimated periods. Thus, we present the time delay aggregation block (Figure 2), which can roll the series based on selected time delay $\tau_1, \cdots, \tau_k$. This operation can align similar sub-series that are at the same phase position of estimated periods, which is different from the point-wise dot-product aggregation in self-attention family. Finally, we aggregate the sub-series by softmax normalized confidences.

For the single head situation and time series $X$ with length-$L$, after the projector, we get query $Q$, key $K$ and value $V$. Thus, it can replace self-attention seamlessly. The Auto-Correlation mechanism is:

$$\tau_1, \cdots, \tau_k = \arg \text{Topk} \,(R_{Q,K}(\tau))$$

$$\tilde R_{Q,K}(\tau_1), \cdots, \tilde R_{Q,K}(\tau_k) = \text{SoftMax} \,(R_{Q,K}(\tau_1), \cdots, R_{Q,K}(\tau_k))$$

$$\text{Auto-Correlation}(Q,K,V) = \sum_{i=1}^{k} \text{Roll}(V, \tau_k) \tilde R_{Q,K}(\tau_k), \quad (6)$$

where $\arg \text{Topk}(\cdot)$ is to get the arguments of the Topk autocorrelations and let $k = \lfloor c \times \log L \rfloor$. $c$ is a hyper-parameter. $R_{Q,K}$ is autocorrelation between series $Q$ and $K$. Roll($X$, $\tau$) represents the operation to $X$ with time delay $\tau$, during which elements that are shifted beyond the first position are re-introduced at the last position. For the encoder-decoder Auto-Correlation (Figure 1), $K$, $V$ are from the encoder $X_{en}^N$ and will be resized to length-$O$, $Q$ is from the previous block of the decoder.
Auto-Correlation vs. self-attention family

Different from the point-wise self-attention family, Auto-Correlation presents the series-wise connections. Concretely, for the temporal dependencies, we find the dependencies among sub-series based on the periodicity. In contrast, the self-attention family only calculates the relation between scattered points. Though some self-attentions consider the local information, they only utilize this to help point-wise dependencies discovery. For the information aggregation, we adopt the time delay block to aggregate the similar sub-series from underlying periods. In contrast, self-attentions aggregate the selected points by dot-product. Benefiting from the inherent sparsity and sub-series-level representation aggregation, Auto-Correlation can simultaneously benefit the computation efficiency and information utilization.

4 Experiments

We extensively evaluate the proposed Autoformer on six real-world benchmarks, covering five mainstream time series forecasting applications: energy, traffic, economics, weather and disease.

Datasets

Here is a description of the six experiment datasets: (1) ETT dataset contains the data collected from electricity transformers, including load and oil temperature that are recorded every...
15 minutes between July 2016 and July 2018. (2) Electricity dataset contains the hourly electricity consumption of 321 customers from 2012 to 2014. (3) Exchange [19] records the daily exchange rates of eight different countries, including Australia, Britain, Canada, Switzerland, China, Japan, New Zealand and Singapore, ranging from 1990 to 2016. (4) Traffic is a collection of hourly data from California Department of Transportation, which describes the road occupancy rates measured by different sensors on San Francisco Bay area freeways. (5) Weather is recorded every 10 minutes for 2020 whole year, which contains 21 meteorological indicators, such as air temperature, humidity, etc. (6) ILI includes the weekly recorded influenza-like illness (ILI) patients data from Centers for Disease Control and Prevention of the United States between 2002 and 2021, which describes the ratio of patients seen with ILI and the total number of the patients. We follow standard protocol and split all datasets into training, validation and test set in chronological order by the ratio of 7:1:2.

### Implementation details
Our method is trained with the L2 loss, using the ADAM [16] optimizer with an initial learning rate of $10^{-4}$. Batch size is set to 32. The training process is stopped after 10 epochs. All experiments are repeated three times, implemented in PyTorch [25] and conducted on a single NVIDIA TITAN RTX 24GB GPUs. The hyper-parameter $c$ of Auto-Correlation is in the range of 1 to 3 to trade off performance and efficiency. See supplementary materials for standard deviations and sensitivity analysis. Autoformer contains 2 encoder layers and 1 decoder layer.

### Baselines
We include 10 baseline methods. For the multivariate setting, we select three latest state-of-the-art transformer-based models: Informer [41], Reformer [17], LogTrans [20], two RNN-based models: LSTMNet [19], LSTM [13] and CNN-based TCN [3] as baselines. For the univariate setting, we include more competitive baselines: N-BEATS [23], DeepAR [28], Prophet [33] and ARMIA [11].
Table 2: Univariate results with different prediction lengths $O \in \{96, 192, 336, 720\}$ on typical datasets. We set the input length $I$ as 96. A lower MSE or MAE indicates a better prediction.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>MSE</td>
<td>MAE</td>
<td>MSE</td>
<td>MAE</td>
<td>MSE</td>
<td>MAE</td>
<td>MSE</td>
<td>MAE</td>
</tr>
<tr>
<td>ETT</td>
<td>96</td>
<td>0.065</td>
<td>0.189</td>
<td>0.882</td>
<td>0.219</td>
<td>0.088</td>
<td>0.225</td>
<td>0.082</td>
</tr>
<tr>
<td></td>
<td>192</td>
<td>0.118</td>
<td>0.256</td>
<td>0.520</td>
<td>0.268</td>
<td>0.132</td>
<td>0.283</td>
<td>0.133</td>
</tr>
<tr>
<td></td>
<td>336</td>
<td>0.154</td>
<td>0.305</td>
<td>0.570</td>
<td>0.326</td>
<td>0.180</td>
<td>0.336</td>
<td>0.201</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>0.182</td>
<td>0.335</td>
<td>0.610</td>
<td>0.388</td>
<td>0.238</td>
<td>0.435</td>
<td>0.268</td>
</tr>
<tr>
<td>Exchange</td>
<td>96</td>
<td>0.241</td>
<td>0.387</td>
<td>0.596</td>
<td>0.299</td>
<td>0.156</td>
<td>0.299</td>
<td>0.279</td>
</tr>
<tr>
<td></td>
<td>192</td>
<td>0.273</td>
<td>0.403</td>
<td>0.669</td>
<td>0.665</td>
<td>1.183</td>
<td>0.912</td>
<td>1.950</td>
</tr>
<tr>
<td></td>
<td>336</td>
<td>0.508</td>
<td>0.539</td>
<td>0.665</td>
<td>0.605</td>
<td>1.367</td>
<td>0.984</td>
<td>2.438</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>0.991</td>
<td>0.768</td>
<td>1.111</td>
<td>0.860</td>
<td>1.872</td>
<td>1.072</td>
<td>2.010</td>
</tr>
</tbody>
</table>

4.1 Main Results

To compare performances under different future horizons, we fix the input length and evaluate models with a wide range of prediction lengths: 96, 192, 336, 720. This setting precisely meets the definition of long-term forecasting. Here are results on both the multivariate and univariate settings.

Multivariate results As for the multivariate setting, Autoformer achieves the consistent state-of-the-art performance in all benchmarks and all prediction length settings (Table 1). Especially, under the input-96-predict-336 setting, compared to previous state-of-the-art results, Autoformer gives 74% (1.334→0.339) MSE reduction in ETT, 18% (0.280→0.231) in Electricity, 61% (1.357→0.509) in Exchange, 15% (0.733→0.622) in Traffic and 21% (0.455→0.359) in Weather. For the input-36-predict-60 setting of ILI, Autoformer makes 43% (4.882→2.770) MSE reduction. Overall, Autoformer yields a 38% averaged MSE reduction among above settings. Note that Autoformer still provides remarkable improvements in the Exchange dataset that is without obvious periodicity. See supplementary materials for detailed showcases. Besides, we can also find that the performance of Autoformer changes quite steadily as the prediction length $O$ increases. It means that Autoformer retains better long-term robustness, which is meaningful for real-world practical applications, such as weather early warning and long-term energy consumption planning.

Univariate results We list the univariate results of two typical datasets in Table 2. Under the comparison with extensive baselines, our Autoformer still achieves state-of-the-art performance for the long-term forecasting tasks. In particular, for the input-96-predict-336 setting, our model achieves 14% (0.180→0.145) MSE reduction on the ETT dataset with obvious periodicity. For the Exchange dataset without obvious periodicity, Autoformer surpasses other baselines by 17% (0.611→0.508) and shows greater long-term forecasting capacity. Also, we find that ARIMA [1] performs best in the input-96-predict-96 setting of the Exchange dataset but fails in the long-term setting. This situation of ARIMA can be benefited from its inherent capacity for non-stationary economic data but is limited by the intricate temporal patterns of real-world series.

4.2 Ablation studies

Table 3: Ablation of decomposition in multivariate ETT with MSE metric. Ours adopts our progressive architecture into other models. Sep employs two models to forecast pre-decomposed seasonal and trend-cyclical components separately. Promotion is the MSE reduction compared to Origin.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Predict-O</td>
<td>Sep</td>
<td>Origin</td>
<td>Sep</td>
<td>Ours</td>
<td>Sep</td>
</tr>
<tr>
<td>96</td>
<td>0.604</td>
<td>0.311</td>
<td><strong>0.204</strong></td>
<td>0.365</td>
<td>0.490</td>
</tr>
<tr>
<td>192</td>
<td>1.060</td>
<td>0.760</td>
<td><strong>0.266</strong></td>
<td>0.533</td>
<td>0.658</td>
</tr>
<tr>
<td>336</td>
<td>1.413</td>
<td>0.665</td>
<td><strong>0.375</strong></td>
<td>1.363</td>
<td>1.469</td>
</tr>
<tr>
<td>720</td>
<td>2.672</td>
<td>3.200</td>
<td><strong>0.537</strong></td>
<td>3.379</td>
<td>2.766</td>
</tr>
</tbody>
</table>

Decomposition architecture With our proposed progressive decomposition architecture, other models can gain consistent promotion, especially as the prediction length $O$ increases (Table 3). This
verifies that our method can generalize to other models and release the capacity of other dependencies learning mechanisms, alleviate the distraction caused by intricate patterns. Besides, our architecture outperforms the pre-processing, although the latter employs a bigger model and more parameters. Especially, pre-decomposing may even bring negative effect because it neglects the interaction of components during long-term future, such as Transformer [35] predict-720, Informer [41] predict-336.

**Auto-Correlation vs. self-attention family** As shown in Table 4, our proposed Auto-Correlation achieves the best performance under various input- I-predict- O settings, which verifies the effectiveness of series-wise connections comparing to point-wise self-attentions (Figure 3). Furthermore, we can also observe that Auto-Correlation is memory efficiency from the last column of Table 4, which can be used in long sequence forecasting, such as input-336-predict-1440.

Table 4: Comparison of Auto-Correlation and self-attention in the multivariate ETT. We replace the Auto-Correlation in Autoformer with different self-attentions. The “-” indicates the out-of-memory.

<table>
<thead>
<tr>
<th>Input Length I</th>
<th>96</th>
<th>192</th>
<th>336</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction Length O</td>
<td>336</td>
<td>720</td>
<td>1440</td>
</tr>
<tr>
<td>Auto-Correlation</td>
<td>MSE</td>
<td>0.339</td>
<td>0.422</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.372</td>
<td>0.419</td>
</tr>
<tr>
<td>Full Attention</td>
<td>MSE</td>
<td>0.375</td>
<td>0.537</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.425</td>
<td>0.502</td>
</tr>
<tr>
<td>LogSparse Attention</td>
<td>MSE</td>
<td>0.362</td>
<td>0.539</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.413</td>
<td>0.522</td>
</tr>
<tr>
<td>LSH Attention</td>
<td>MSE</td>
<td>0.366</td>
<td>0.502</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.404</td>
<td>0.475</td>
</tr>
<tr>
<td>ProbSparse Attention</td>
<td>MSE</td>
<td>0.481</td>
<td>0.822</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.472</td>
<td>0.559</td>
</tr>
</tbody>
</table>

### 4.3 Model Analysis

**Time series decomposition** As shown in Figure 4 without our series decomposition block, the forecasting model cannot capture the increasing trend and peaks of the seasonal part. By adding the series decomposition blocks, Autoformer can aggregate and refine the trend-cyclical part from series progressively. This design also facilitates the learning of the seasonal part, especially the peaks and troughs. This verifies the necessity of our proposed progressive decomposition architecture.

![Figure 4: Visualization of learned seasonal $X^{M}$ and trend-cyclical $T^{M}_{\text{de}}$ of the last decoder layer. We gradually add the decomposition blocks in decoder from left to right. This case is from ETT dataset under input-96-predict-720 setting. For clearness, we add the linear growth to raw data additionally.](image)

**Dependencies learning** The marked time delay sizes in Figure 5 (a) indicate the most likely periods. Our learned periodicity can guide the model to aggregate the sub-series from the same or neighbor phase of periods by Roll$(X, \tau_i) \in \{1, \cdots, 6\}$. For the last time step (declining stage), Auto-Correlation fully utilizes all similar sub-series without omissions or errors compared to self-attentions. This verifies that Autoformer can discover the relevant information more sufficiently and precisely.

**Complex seasonality modeling** As shown in Figure 6, the lags that Autoformer learns from deep representations can indicate the real seasonality of raw series. For example, the learned lags of the
This paper studies the long-term forecasting problem of time series, which is a pressing demand for real-world applications. However, the intricate temporal patterns prevent the model from learning reliable dependencies. We propose the Autoformer as a decomposition architecture by embedding the series decomposition block as an inner operator, which can progressively aggregate the long-term trend part from intermediate prediction. Besides, we design an efficient Auto-Correlation mechanism to conduct dependencies discovery and information aggregation at the series level, which contrasts clearly from the previous self-attention family. Autoformer can naturally achieve $\mathcal{O}(L \log L)$ complexity and yield consistent state-of-the-art performance in extensive real-world datasets.

5 Conclusions

Efficiency analysis We compare the running memory and time among Auto-Correlation-based and self-attention-based models (Figure 7) during the training phase. The proposed Autoformer shows $\mathcal{O}(L \log L)$ complexity in both memory and time and achieves better long-term sequences efficiency.
Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grants 62022050 and 62021002, Beijing Nova Program under Grant Z201100006820041, China’s Ministry of Industry and Information Technology, the MOE Innovation Plan and the BNRist Innovation Fund.

References


[38] Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. In *NeurIPS*, 2020.

