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. Towards these challenges, we propose Autoformer as a novel decomposition architecture with an Auto-Correlation mechanism. We go beyond 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.

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. Thus, in this paper, we study the long-term forecasting problem of time series, characterizing itself by the large length of predicted series, which can be severalfold of the length of the input series. Recent deep forecasting models 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. 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 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. It can be used to process the complex time series and
Time Series Forecasting

- Energy Consumption
- Traffic Flow
- Economic Changes
- Weather Variations
- Disease Propagation

Future Time Series

Past Observations

Forecasting

Long-term Planning and Early Warning
- Longer Forecasting Horizon
- More Accurate Prediction
Long-Term Time Series Forecasting

Energy Consumption
Traffic Flow
Economic Changes
Weather Variations
Disease Propagation

Extend The Forecast Time Into The Far Future
Scaled Dot-Product Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, which we compute as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

Multi-Head Attention

Transformers

Autoformer is for long-term Forecasting.
Transformers For Time series Forecasting

- Modeling the **temporal dependencies** with **point-wise Self-Attention**
- Aggregate the representations for forecasting

Informer[Zhou et al. AAAI2021], Reformer[Kitaev et al. ICLR2020], Log Trans[Li et al. NeurIPS19]
Long-Term Time Series Forecasting

Longer Forecasting Horizon → Intricate Temporal Patterns
Deal with Long Series (complexity)

Uptrend  Plateau  Downtrend  Steep Drop  Fluctuation  Uptrend...

Past Observations → Future Time Series
## Transformers for Long-Term Series Forecasting

| Intricate Temporal Patterns | Transformers | Autoformer |
|-----------------------------|--------------|------------|
| Hard to directly find reliable temporal dependencies from series $X$ | Point-wise Self-Attention is $O(L^2)$ | Decomposition architecture to ravel out the entangled temporal patterns |
| Point-wise Self-Attention is $O(L^2)$ | Adopt sparse version for efficiency | Series-wise Auto-Correlation based on stochastic process theory with inherent $O(L \log L)$ complexity |
| • Adopt sparse version for efficiency | • Loss information and cause the information utilization bottleneck $X$ | |
| Deal with Long Series | | |

### Transformer: Autoformer

- **Decomposition** architecture to ravel out the entangled temporal patterns.
- **Series-wise** Auto-Correlation based on stochastic process theory with inherent $O(L \log L)$ complexity.
Overall Architecture

Autoformer Encoder

Encoder Input

Auto-Correlation

Series Decomp

Feed Forward

Series Decomp

N x

Time Series

Seasonal Part

Trend-cyclical Part

Zero

Data Mean

Seasonal Init

Auto-Correlation

Series Decomp

Auto-Correlation

Series Decomp

Feed Forward

Series Decomp

M x

Trend-cyclical Init

Input Data Mean

Autoformer Decoder

Prediction
Overall Architecture

Decomposition architecture for intricate temporal patterns.
Overall Architecture

Series-wise Auto-Correlation for information utilization bottleneck.
Decomposition: Pre-processing Convention

- Limited by the capabilities of decomposition
- Overlooks the potential future interactions among components

![Time Series Components](image)

- Trend-cyclical component
- Seasonal component
- Trend-cyclical Prediction
- Seasonal Prediction

**Results**

- Highlight the inherent properties of time series

**Past**

**Future**
Deep Decomposition Architecture

Decomposition with Moving Average

\[ \mathcal{X}_t = \text{AvgPool}(\text{Padding}(\mathcal{X})) \]

\[ \mathcal{X}_s = \mathcal{X} - \mathcal{X}_t, \]

Progressive decomposition capacity

Decompose the trend from the intermediate “future” and refine it during the decoder.
Deep Decomposition Architecture: Input

\[ \mathcal{X}_{\text{ens}}, \mathcal{X}_{\text{ent}} = \text{SeriesDecomp}(\mathcal{X}_{\text{en}}[:l]) \]

\[ \mathcal{X}_{\text{des}} = \text{Concat}(\mathcal{X}_{\text{ens}}, \mathcal{X}_0) \]

\[ \mathcal{X}_{\text{det}} = \text{Concat}(\mathcal{X}_{\text{ent}}, \mathcal{X}_\text{Mean}) \]
Deep Decomposition Architecture: Encoder

Focus on seasonal part modeling,
Provide cross information for decoder

\[ S_{en,1}^{l,1} = \text{SeriesDecomp} \left( \text{AutoCorrelation} \left( x_{en}^{l-1} \right) + x_{en}^{l-1} \right) \]

\[ S_{en,2}^{l,2} = \text{SeriesDecomp} \left( \text{FeedForward} \left( S_{en}^{l,1} + S_{en}^{l,1} \right) \right) \]
Deep Decomposition Architecture: Decoder

- **Autoformer Encoder**: \(N \times x\)
- **Autoformer Decoder**: \(M \times x\)
- **Encoder Input**: To Predict
- **Seasonal Init**: Zero
- **Trend-cyclical Init**: Input Data Mean
- **Time Series**: Trend Part, Seasonal Part, Trend-cyclical Part

**Adopt the Auto-Correlation for the seasonal prediction**

**Accumulation for the trend-cyclical prediction**

Mathematical equations:

\[
S_{de}^{l,1}, T_{de}^{l,1} = \text{SeriesDecomp} \left( \text{AutoCorrelation} \left( S_{de}^{l-1}, X_{de}^{l-1} \right) \right)
\]

\[
S_{de}^{l,2}, T_{de}^{l,2} = \text{SeriesDecomp} \left( \text{AutoCorrelation} \left( S_{de}^{l,1}, X_{en}^{N} \right) + S_{de}^{l,1} \right)
\]

\[
S_{de}^{l,3}, T_{de}^{l,3} = \text{SeriesDecomp} \left( \text{FeedForward} \left( S_{de}^{l,2} + S_{de}^{l,2} \right) \right)
\]

\[
T_{de}^{l} = T_{de}^{l-1} + W_{l,1} \cdot T_{de}^{l,1} + W_{l,2} \cdot T_{de}^{l,2} + W_{l,3} \cdot T_{de}^{l,3}
\]
Auto-Correlation Mechanism

Period-based dependencies
The same phase position of different periods

Benefited from the deep decomposition, the **seasonal part** is highlighted with **periodicity**.

Conducts the **dependencies discovery** and **representation aggregation** at the series level.
Auto-Correlation Mechanism

Discover period-based dependencies with autocorrelation in stochastic process:

$$R_{\mathcal{X}\mathcal{X}}(\tau) = \lim_{L \to \infty} \frac{1}{L} \sum_{t=0}^{L-1} \mathcal{X}_t \mathcal{X}_{t-\tau}.$$  

Autocorrelation reflects the time delay similarity, and corresponds to the confidence of period estimation.

Larger autocorrelation $R(\tau)$ means

• stronger time delay similarity w.r.t. $\tau$
• more confidence of period length as $\tau$
Auto-Correlation Mechanism

① Discover period-based dependencies

② Aggregate similar sub-processes from different periods
Auto-Correlation Mechanism

Efficient computation of autocorrelation with Wiener–Khinchin theorem by FFT

$$S_{XX}(f) = \mathcal{F}(X_t) \mathcal{F}^*(X_t) = \int_{-\infty}^{\infty} X_t e^{-i2\pi ft} dt \int_{-\infty}^{\infty} X_t e^{-i2\pi ft} dt$$

$$R_{XX}(\tau) = \mathcal{F}^{-1}(S_{XX}(f)) = \int_{-\infty}^{\infty} S_{XX}(f) e^{i2\pi f\tau} df,$$

Discover period-based dependencies with inherent $O(L \log L)$ complexity
Auto-Correlation Mechanism

Align the delayed series,
Aggregate sub-series representations

Select the Top $k$ period lengths

Aggregate representations from similar sub-processes

Normalization

$$\tau_1, \cdots, \tau_k = \arg \text{Topk} (\mathcal{R}_{Q, K}(\tau))_{\tau \in \{1, \cdots, L\}}$$

$$\hat{\mathcal{R}}_{Q, K}(\tau_1), \cdots, \hat{\mathcal{R}}_{Q, K}(\tau_k) = \text{SoftMax} (\mathcal{R}_{Q, K}(\tau_1), \cdots, \mathcal{R}_{Q, K}(\tau_k))$$

$$\text{AutoCorrelation}(Q, K, \mathcal{V}) = \sum_{i=1}^{k} \text{Roll}(\mathcal{V}, \tau_k) \hat{\mathcal{R}}_{Q, K}(\tau_k),$$

Align the delayed series,
Aggregate sub-series representations
Auto-Correlation vs Self-Attention Family

Auto-Correlation can extend the point-wise aggregation to series-wise.

Informer[Zhou et al. AAAI2021], Reformer[Kitaev et al. ICLR2020], Log Trans[Li et al. NeurIPS19]
## Experiments: Multivariate setting

| Models | Transformers | LSTMs | TCN |
|--------|--------------|-------|-----|
|          | Informer[41] | LogTrans[20] | Reform[17] | LSTNet[19] | LSTM[13] | TCN[3] |
| **Metric** | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Energy | 96 | 0.255 | 0.339 | 0.365 | 0.453 | 0.768 | 0.642 | 0.658 | 0.619 | 3.142 | 1.365 | 2.041 | 1.073 | 3.041 | 1.330 |
|        | 192 | 0.281 | 0.340 | 0.533 | 0.563 | 0.989 | 0.757 | 1.078 | 0.827 | 3.154 | 1.369 | 2.249 | 1.112 | 3.072 | 1.339 |
|        | 336 | 0.339 | 0.372 | 1.363 | 0.887 | 1.334 | 0.872 | 1.549 | 0.972 | 3.160 | 1.369 | 2.568 | 1.238 | 3.105 | 1.348 |
|        | 720 | 0.422 | 0.419 | 3.379 | 1.388 | 3.048 | 1.328 | 2.631 | 1.242 | 3.171 | 1.368 | 2.720 | 1.287 | 3.135 | 1.354 |
| Electricity | 96 | 0.201 | 0.317 | 0.274 | 0.368 | 0.258 | 0.357 | 0.312 | 0.402 | 0.680 | 0.645 | 0.375 | 0.437 | 0.985 | 0.813 |
|        | 192 | 0.222 | 0.334 | 0.296 | 0.386 | 0.266 | 0.368 | 0.348 | 0.433 | 0.725 | 0.676 | 0.442 | 0.473 | 0.996 | 0.821 |
|        | 336 | 0.231 | 0.338 | 0.300 | 0.394 | 0.280 | 0.380 | 0.350 | 0.433 | 0.828 | 0.727 | 0.439 | 0.473 | 1.000 | 0.824 |
|        | 720 | 0.254 | 0.361 | 0.373 | 0.439 | 0.283 | 0.376 | 0.340 | 0.420 | 0.957 | 0.811 | 0.980 | 0.814 | 1.438 | 1.784 |
| Economics | 96 | 0.197 | 0.323 | 0.847 | 0.752 | 0.968 | 0.812 | 1.065 | 0.829 | 1.551 | 1.058 | 1.453 | 1.049 | 3.004 | 1.432 |
|        | 192 | 0.300 | 0.369 | 1.204 | 0.895 | 1.040 | 0.851 | 1.188 | 0.906 | 1.477 | 1.028 | 1.846 | 1.179 | 3.048 | 1.444 |
|        | 336 | 0.509 | 0.524 | 1.672 | 1.036 | 1.659 | 1.081 | 1.357 | 0.976 | 1.507 | 1.031 | 2.136 | 1.231 | 3.113 | 1.459 |
|        | 720 | 1.447 | 0.941 | 2.478 | 1.310 | 1.941 | 1.127 | 1.510 | 1.016 | 2.285 | 1.243 | 2.984 | 1.427 | 3.150 | 1.458 |
| Traffic | 96 | 0.613 | 0.388 | 0.719 | 0.391 | 0.684 | 0.384 | 0.732 | 0.423 | 1.107 | 0.685 | 0.843 | 0.453 | 1.438 | 0.784 |
|        | 192 | 0.616 | 0.382 | 0.696 | 0.379 | 0.685 | 0.390 | 0.733 | 0.420 | 1.157 | 0.706 | 0.847 | 0.453 | 1.463 | 0.794 |
|        | 336 | 0.622 | 0.337 | 0.777 | 0.420 | 0.733 | 0.408 | 0.742 | 0.420 | 1.216 | 0.730 | 0.853 | 0.455 | 1.479 | 0.799 |
|        | 720 | 0.660 | 0.408 | 0.864 | 0.472 | 0.717 | 0.396 | 0.755 | 0.423 | 1.481 | 0.805 | 1.500 | 0.805 | 1.499 | 0.804 |
| Weather | 96 | 0.266 | 0.336 | 0.300 | 0.384 | 0.458 | 0.490 | 0.689 | 0.596 | 0.594 | 0.587 | 0.369 | 0.406 | 0.615 | 0.589 |
|        | 192 | 0.307 | 0.367 | 0.598 | 0.544 | 0.658 | 0.589 | 0.752 | 0.638 | 0.560 | 0.565 | 0.416 | 0.435 | 0.629 | 0.600 |
|        | 336 | 0.359 | 0.395 | 0.578 | 0.523 | 0.797 | 0.652 | 0.639 | 0.596 | 0.597 | 0.587 | 0.455 | 0.454 | 0.639 | 0.608 |
|        | 720 | 0.419 | 0.428 | 1.059 | 0.741 | 0.869 | 0.675 | 1.130 | 0.792 | 0.618 | 0.599 | 0.535 | 0.520 | 0.639 | 0.610 |
| Disease | 24 | 3.483 | 1.287 | 5.764 | 1.677 | 4.480 | 1.444 | 4.400 | 1.382 | 6.026 | 1.770 | 5.914 | 1.734 | 6.624 | 1.830 |
|        | 36 | 3.103 | 1.148 | 4.755 | 1.467 | 4.799 | 1.467 | 4.783 | 1.448 | 5.340 | 1.668 | 6.631 | 1.845 | 6.858 | 1.879 |
|        | 48 | 2.669 | 1.085 | 4.763 | 1.469 | 4.800 | 1.468 | 4.832 | 1.465 | 6.080 | 1.787 | 6.736 | 1.857 | 6.968 | 1.892 |
|        | 60 | 2.770 | 1.125 | 5.264 | 1.564 | 5.278 | 1.560 | 4.882 | 1.483 | 5.548 | 1.720 | 6.870 | 1.879 | 7.127 | 1.918 |

* ETT means the ETTm2. See supplementary materials for the full benchmark of ETT1, ETT2, ETTm1.

Prediction Accuracy

| Relative Promotion (In MSE) |
|-----------------------------|
| Input-96-predict-336        |
| ↑ 74%                      |
| Input-96-predict-336        |
| ↑ 18%                      |
| Input-96-predict-336        |
| ↑ 61%                      |
| Input-96-predict-336        |
| ↑ 15%                      |
| Input-96-predict-336        |
| ↑ 21%                      |
| Input-24-predict-48        |
| ↑ 43%                      |
### Experiments: ETT benchmark

| Models  | Metric | Autoformer | Informer [14] | LogTrans [9] | Reformer [7] | LSTNet [8] | LSTMa [11] |
|---------|--------|------------|----------------|----------------|---------------|-------------|-------------|
|         |        | MSE        | MAE        | MSE            | MAE            | MSE         | MAE         | MSE         | MAE         |
| ETTl    | 24     | 0.384      | 0.425      | 0.577          | 0.549          | 0.686       | 0.604       | 0.991       | 0.754       |
|         | 48     | 0.392      | 0.419      | 0.685          | 0.625          | 0.766       | 0.757       | 1.313       | 0.906       |
|         | 168    | 0.490      | 0.481      | 0.931          | 0.752          | 1.002       | 0.846       | 1.824       | 1.138       |
|         | 336    | 0.505      | 0.484      | 1.128          | 0.873          | 1.362       | 0.952       | 2.117       | 1.280       |
|         | 720    | 0.498      | 0.500      | 1.215          | 0.896          | 1.397       | 1.291       | 2.415       | 1.520       |
| ETTl2   | 24     | 0.261      | 0.341      | 0.720          | 0.665          | 0.828       | 0.750       | 1.531       | 1.613       |
|         | 48     | 0.312      | 0.373      | 1.457          | 1.001          | 1.806       | 1.034       | 1.871       | 1.735       |
|         | 168    | 0.457      | 0.455      | 3.489          | 1.515          | 4.070       | 1.681       | 4.660       | 1.846       |
|         | 336    | 0.471      | 0.475      | 2.723          | 1.340          | 3.875       | 1.763       | 4.028       | 1.688       |
|         | 720    | 0.474      | 0.484      | 3.467          | 1.473          | 3.913       | 1.552       | 5.381       | 2.015       |
| ETTm1   | 24     | 0.383      | 0.403      | 0.323          | 0.369          | 0.419       | 0.412       | 0.724       | 0.607       |
|         | 48     | 0.454      | 0.453      | 0.494          | 0.503          | 0.507       | 0.583       | 1.098       | 0.777       |
|         | 96     | 0.481      | 0.463      | 0.678          | 0.614          | 0.768       | 0.792       | 1.433       | 0.945       |
|         | 288    | 0.634      | 0.528      | 1.056          | 0.786          | 1.462       | 1.320       | 1.820       | 1.094       |
|         | 672    | 0.606      | 0.542      | 1.192          | 0.926          | 1.669       | 1.461       | 2.187       | 1.232       |
| ETTm2   | 24     | 0.153      | 0.261      | 0.173          | 0.301          | 0.211       | 0.332       | 0.333       | 0.429       |
|         | 48     | 0.178      | 0.280      | 0.303          | 0.409          | 0.427       | 0.487       | 0.558       | 0.571       |
|         | 96     | 0.255      | 0.339      | 0.365          | 0.453          | 0.768       | 0.642       | 0.658       | 0.619       |
|         | 288    | 0.342      | 0.378      | 1.047          | 0.804          | 1.090       | 0.806       | 2.441       | 1.190       |
|         | 672    | 0.434      | 0.430      | 3.126          | 1.302          | 2.397       | 1.214       | 3.090       | 1.328       |

**Prediction Accuracy**

- **Relative Promotion (In MSE)**
  - ETTl: 55%
  - Input-96-predict-336: 80%
  - Input-96-predict-288: 40%
  - Input-96-predict-288: 66%
## Experiments: Univariate setting

| Models       | Autoformer | N-BEATS[23] | Informer[41] | LogTrans[20] | Reformer[17] | DeepAR[28] | Prophet[33] | ARIMA[1] |
|--------------|------------|-------------|--------------|--------------|--------------|-------------|-------------|---------|
| Metric       | MSE        | MAE         | MSE          | MAE          | MSE          | MAE          | MSE         | MAE      | MSE     | MAE     | MSE     | MAE     | MSE     | MAE     | MSE     | MAE     | MSE     | MAE     |
| ETT          | 96         | 0.065       | 0.189        | 0.082        | 0.219        | 0.088       | 0.225       | 0.082     | 0.217     | 0.131   | 0.288   | 0.099   | 0.237   | 0.287   | 0.456   | 0.211   | 0.362   |
|              | 192        | 0.118       | 0.256        | 0.120        | 0.268        | 0.132       | 0.283       | 0.133     | 0.284     | 0.186   | 0.354   | 0.154   | 0.310   | 0.312   | 0.483   | 0.261   | 0.406   |
|              | 336        | 0.154       | 0.305        | 0.226        | 0.370        | 0.180       | 0.336       | 0.201     | 0.361     | 0.220   | 0.381   | 0.277   | 0.428   | 0.331   | 0.474   | 0.317   | 0.448   |
|              | 720        | 0.182       | 0.335        | 0.188        | 0.338        | 0.300       | 0.435       | 0.268     | 0.407     | 0.267   | 0.430   | 0.332   | 0.468   | 0.534   | 0.593   | 0.366   | 0.487   |
| Exchange     | 96         | 0.241       | 0.387        | 0.156        | 0.299        | 0.591       | 0.615       | 0.279     | 0.441     | 1.327   | 0.944   | 0.417   | 0.515   | 0.828   | 0.762   | 0.112   | 0.245   |
|              | 192        | 0.273       | 0.403        | 0.669        | 0.665        | 1.183       | 0.912       | 1.950     | 1.048     | 1.258   | 0.924   | 0.813   | 0.735   | 0.909   | 0.974   | 0.304   | 0.404   |
|              | 336        | 0.508       | 0.539        | 0.611        | 0.605        | 1.367       | 0.984       | 2.438     | 1.262     | 2.179   | 1.296   | 1.331   | 0.962   | 1.304   | 0.988   | 0.736   | 0.598   |
|              | 720        | 0.991       | 0.768        | 1.111        | 0.860        | 1.872       | 1.072       | 2.010     | 1.247     | 1.280   | 0.953   | 1.894   | 1.181   | 3.238   | 1.566   | 1.871   | 0.935   |

**Competitive baseline**

**N-BEATS**
Showcases

(1) ETT dataset with input-96-predict-336 (Energy, with obvious periodicity)

(2) Exchange dataset with input-96-predict-192 (Economics, without obvious periodicity)
Ablation of decomposition architecture

Progressive decomposition architecture outperforms separately forecasting convention, especially the long-term setting.

The decomposition architecture can be generalized to other Transformers with remarkable promotion.

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.
Visualization of decomposition architecture

Figure 4: Visualization of learned seasonal $X_{de}^M$ and trend-cyclical $T_{de}^M$ 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.

- With the increasing of decomposition blocks, the predictions of both seasonal and trend part get better and better progressively.
Auto-Correlation vs. Self-Attention Family

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.

| Prediction Length \( O \) | 336  | 720  | 1440 | 336  | 720  | 1440 | 336  | 720  | 1440 |
|----------------------------|------|------|------|------|------|------|------|------|------|
| **Input Length \( I \)**   |      |      |      |      |      |      |      |      |      |
| 96                         | 0.339| 0.422| 0.555| 0.355| 0.429| 0.503| 0.361| 0.425| 0.574|
| 192                        | 0.372| 0.419| 0.496| 0.392| 0.430| 0.484| 0.406| 0.440| 0.534|
| 336                        | 0.375| 0.537| 0.667| 0.450| 0.554| -    | 0.501| 0.647| -    |
| **Auto-Correlation**       |      |      |      |      |      |      |      |      |      |
| MSE                        | 0.425| 0.502| 0.589| 0.470| 0.533| -    | 0.485| 0.491| -    |
| MAE                        |      |      |      |      |      |      |      |      |      |
| Full Attention[35]         | 0.362| 0.539| 0.582| 0.420| 0.552| 0.958| 0.474| 0.601| -    |
| LogSparse Attention[20]    | 0.413| 0.522| 0.529| 0.450| 0.513| 0.736| 0.474| 0.524| -    |
| LSH Attention[17]          | 0.366| 0.502| 0.663| 0.407| 0.636| 1.069| 0.442| 0.615| -    |
| ProbSparse Attention[41]   | 0.481| 0.822| 0.715| 0.404| 1.148| 0.732| 0.417| 0.631| 1.133|
| MSE                        | 0.472| 0.559| 0.586| 0.425| 0.654| 0.602| 0.434| 0.528| 0.691|
| MAE                        |      |      |      |      |      |      |      |      |      |

Under various input-predict settings, Auto-Correlation outperforms the self-attention and their variants.
Visualization of learned dependencies

Figure 5: Visualization of learned dependencies. For clearness, we select the top-6 time delay sizes $\tau_1, \cdots, \tau_6$ of Auto-Correlation and mark them in raw series (red lines). For self-attentions, top-6 similar points with respect to the last time step (red stars) are also marked by orange points.

Auto-Correlation can discover the relevant information more sufficiently and precisely.

Transformers[Vaswani et al. NeurIPS17], Informer[Zhou et al. AAAI2021], Reformer[Kitaev et al. ICLR2020]
Visualization of learned lags

Learned lags can reflect the human-interpretable prediction.
Efficiency Analysis

Figure 6: Efficiency Analysis. For memory, we replace Auto-Correlation with self-attention family in Autoformer and record the memory with input 96. For running time, we run the Auto-Correlation or self-attentions $10^3$ times to get the execution time per step. The output length increases exponentially.

Auto-Correlation presents remarkable $O(L \log L)$ complexity in both memory and computation.
Autoformer achieves the remarkable state-of-the-art on extensive benchmarks.
Open Source

https://github.com/thuml/Autoformer

Well-organized code and pre-processed dataset
Thank You!
whx20@mails.tsinghua.edu.cn