PREFORMER: PREDICTIVE TRANSFORMER WITH MULTI-SCALE SEGMENT-WISE CORRELATIONS FOR LONG-TERM TIME SERIES FORECASTING

Dazhao Du\textsuperscript{1,2}, Bing Su\textsuperscript{3,4,*}, Zhewei Wei\textsuperscript{3,4}

Institute of Software, Chinese Academy of Science\textsuperscript{1}
University of Chinese Academy of Sciences\textsuperscript{2}
Gaoling School of Artificial Intelligence, Renmin University of China\textsuperscript{3}
Beijing Key Laboratory of Big Data Management and Analysis Methods\textsuperscript{4}

ABSTRACT
In long-term time series forecasting, most Transformer-based methods adopt the standard point-wise attention mechanism, which not only has high complexity but also cannot explicitly capture the predictive dependencies from contexts since the corresponding key and value are transformed from the same point. This paper proposes a predictive Transformer-based model called Preformer. Preformer introduces a novel efficient 	extit{Multi-Scale Segment-Correlation} mechanism that divides time series into segments and utilizes segment-wise correlation-based attention to replace point-wise attention. A multi-scale structure is developed to aggregate dependencies at different temporal scales and facilitate the selection of segment length. Preformer further designs a predictive paradigm for decoding, where the key and value come from two successive segments rather than the same segment. Experiments demonstrate that Preformer outperforms other Transformer-based models. The codes are available at \url{https://github.com/ddz16/Preformer}.

Index Terms— Time series forecasting, temporal modeling, Transformer

1. INTRODUCTION
Long-term time series forecasting has a wide range of real-world applications. Existing deep learning-based methods can be divided into four categories, i.e., RNN [1, 2, 3, 4, 5], TCN [6, 7, 8, 9], MLP [10, 11] and Transformer-based models [12, 13]. Among them, transformer-based methods have recently received attention from the time series forecasting community, especially for long-term forecasting [14, 15, 16], because the self-attention in Transformer can directly model the relationships between any element pairs. However, the standard self-attention in Transformer calculates similarities between all time point pairs, where the time and space complexities increase quadratically with the length of the time series. Recent works [17, 18, 14, 16] explore different sparse attention mechanisms to suppress the contribution of irrelevant time steps and ease the computational pressure. These models still perform dot-product attention to time steps individually and utilize the point-wise connections to capture temporal dependencies. However, a single point may have limited influence on predicting the future. Autoformer [15] conducts the series-wise dependencies discovery by performing Auto-Correlation where the aggregation operation acts on the whole delayed series and complicated Fourier transforms are required.

These methods perform the correlation either at the point level or at the overall series level, which not only requires high computational redundancy to intensively tackle point pairs or perform time-frequency domain transformations but also does not directly reflect the true dependencies within the time series. Time series tend to have strong continuity and internal dynamics locally (within a segment). Therefore, there exist stronger correlations at the segment level. To this end, we propose a novel attention mechanism called \textit{Multi-Scale Segment-Correlation (MSSC)}, which calculates the cross-correlation [19] between two segments as similarity measurement. The output of Segment-Correlation is obtained by weighting the aggregation of all segments, with the segment being the smallest unit of attention, which actually preserves the continuity within each segment. Since there are fewer segments than points, less calculation is required when performing segment-wise attention. The segment length is a critical hyperparameter. Long segments ignore fine-grained information while short segments have high computational redundancy. To tackle this issue, MSSC performs correlation calculations and fusion on multiple segment lengths, i.e., multi-scale resolutions, while maintaining low complexity.

Besides, the standard attention paradigm [12] can be applied to MSSC for time series encoding, but it is not well suited for forecasting since the unknown prediction segment generates the query by itself to predict itself. For prediction tasks, it is more reasonable to utilize the query of the previous segment to generate the prediction of the unknown segment. In this case, if the given query for prediction is similar to the keys of some segments, their next segments rather than these segments themselves should contribute more to the prediction. Motivated by this, we further propose a \textit{Predictive Multi-Scale Segment-Correlation (PreMSSC)}, where current segment output \(Y_t\) can be obtained via using the previous segment \(Q_{t-1}\) to query all segments \(\{K_1, K_2, \ldots\}\) and weighting the values of their next segments \(\{V_2, V_3, \ldots\}\) by the calculated correlations. We derive our predictive model, namely \textit{Predictive Transformer (Preformer)}, via replacing the standard self-attention and cross-attention in the original Transformer model with MSSC and PreMSSC, respectively.

2. METHOD

2.1. Problem Definition
Multi-horizon time series forecasting aims to predict values at multiple future time steps [20]. Typically, given the previous time series \(X_{1:t_0} = \{x_1, x_2, \ldots, x_{t_0}\}\), where \(x_t \in \mathbb{R}^{d_x}\) and \(d_x\) is
The overall architecture of Preformer. We utilize the series decomposition blocks to decompose the time series into trend and seasonal components. Our proposed MSSC and PreMSSC are the key modules to model the seasonal component.

2.2. The Preformer Model

As shown in Figure 1, the overall architecture of Preformer is similar to AutoFormer [15]. Though model architectures are similar, the ideas of core modules are completely different. Considering the strong locality (continuity between adjacent points) of the time series, we treat the segment as the smallest unit of attention calculation and maintain the continuity within each segment when aggregating.

**Model inputs.** The way the original time series is input to the Preformer is consistent with Autoformer [15]. We utilize additional time-dependent features (e.g., hour-of-the-day, day-of-the-week) called covariates as parts of inputs. Before being fed to the MSSC module, inputs to the encoder and decoder are transformed into the feature dimension through an embedding layer.

**Encoder and decoder.** The encoder consists of $N$ identical layers, where each layer consists of a MSSC module and a feed-forward network each followed by a series decomposition module with residual connections. The input of the encoder includes the past time series values and covariates. The series decomposition modules decompose the inputs into the seasonal and trend components through average pooling layers [15]. All decomposition modules in the encoder eliminate the trend component, which makes the encoder focus on seasonal pattern modeling. The decoder consists of $M$ identical layers. Different from the encoder, there is an additional PreMSSC module where key and value matrices are transformed from the encoder outputs in each decoder layer. The inputs of the decoder include two components: seasonal and trend components. The decomposition modules in the decoder extract the trend part from hidden variables progressively, which is finally added to the seasonal part to derive the output. A fully connected layer takes the output of the decoder as input, and generates final predictions $\hat{Y}_{t_0 + 1:t_0 + \tau} \in \mathbb{R}^{\tau \times d_y}$.

2.3. Segment-Correlation

Segment-Correlation is the key module in Preformer, which performs segment-wise attention instead of point-wise attention. We denote the input of each Segment-Correlation module as $H \in \mathbb{R}^{L \times d}$, where $L$ and $d$ are the length and dimension of the input respectively. Formally, for the single head situation, the input series $H$ will be projected by three projection matrices to obtain the query, key, and value, i.e., $Q = HW_q$, $K = HW_k$, $V = HW_v$. Then all the $Q$, $K$ and $V$ are segmented into several segments having the same length $L_{seg}$: $\{Q_1, Q_2, \ldots, Q_m\}, \{K_1, K_2, \ldots, K_n\}, \{V_1, V_2, \ldots, V_n\}$, where $Q_i \in \mathbb{R}^{L_{seg} \times d}$, $K_i \in \mathbb{R}^{L_{seg} \times d}$, $V_i \in \mathbb{R}^{L_{seg} \times d}$, $m, n$ denote the number of segments, and $L_{seg}$ is a hyperparameter that determines the computational complexity by controlling the segment length.

The correlation measurement $c_{ij}$ between any pair of query segment $Q_i$ and key segment $K_j$ is computed by the function:

$$c_{ij} = Correlation(Q_i, K_j) = \frac{1}{d \times L_{seg}} Q_i \odot K_j,$$

where $\odot$ is the dot product operator between two matrices of the same size. For each query segment $Q_i$, whose correlation measurements with all the key segments will be normalized by the Softmax function to obtain the aggregation weight:

$$\hat{c}_{i1}, \hat{c}_{i2}, \ldots, \hat{c}_{in} = \text{Softmax}(c_{i1}, c_{i2}, \ldots, c_{in}).$$

The output at the position of the $i$-th segment $Y_i$ is the weighted sum of all the value segments $\{V_j\}_{j = 1, \ldots, n}$:

$$Y_i = \sum_{j=1}^{n} \hat{c}_{ij} V_j.$$

Lastly, output of the Segment-Correlation module is obtained by concatenating all the $Y_i$ along length dimension:

$$\text{SC}(H; L_{seg}) = \text{Concat}(Y_1, \ldots, Y_m),$$

where SC is the abbreviation of Segment-Correlation.

**Multi-Scale Segment-Correlation.** Since $L_{seg}$ determines the resolution of the smallest unit involved in the Segment-Correlation measurement, we introduce the multi-scale segment correlation module with or without the predictive paradigm, representing MSSC and PreMSSC respectively.
calculation, a large $L_{seg}$ means that the coarse-grained temporal dependencies in the time series can be captured, while Segment-Correlation with a small $L_{seg}$ can capture fine-grained dependencies. To attenuate the impact of hyperparameter $L_{seg}$ selection on performance, we propose Multi-Scale Segment-Correlation (MSSC) by fusing the output of multiple Segment-Correlation with different $L_{seg}$. Considering the inefficiency of iterating over all choices to find an optimal combination, we use exponentially growing segment lengths to cover as many scale levels as possible. Specifically, as the scale level increases, we start with a small initial segment length $L_0$ and increase the segment length exponentially, i.e., $L_i = 2^i L_0$, where $i \in \{0, 1, \ldots, I_{max}\}$ denotes the scale level and $I_{max} = \lfloor \log_2 \left( \frac{L}{L_0} \right) \rfloor$. The input $H$ is passed through these Segment-Correlation layers of different scale levels to obtain the outputs of the corresponding scale levels. Considering that the fine-grained inputs contain more information, we aggregate the outputs of all scale levels to get the output of the entire MSSC module, where the weight of the $l$-th level is set to decrease exponentially as the scale level increases. Therefore, MSSC is formulated as:

$$
MSSC(H) = \sum_{l=0}^{I_{max}} \frac{2^i}{\sum_{l=0}^{I_{max}} 2^i} SC(H; 2^i L_0) .
$$

**Predictive paradigm.** In the decoding phase, the query for the period to be predicted is its preceding segment rather than itself. Therefore, if some segments with respect to keys are highly relevant to the query, their future segments rather than themselves should contribute more to the prediction of the query. Inspired by this intuition, we propose a predictive paradigm by introducing a segment delay between the keys and their corresponding values. For Segment-Correlation without the predictive paradigm, the output at the position of the $i$-th segment $Y_i$ is obtained according to eq. (3). If we reformulate $\hat{c}_{ij}$ as $\hat{c}(Q, K_j)$, then we have:

$$
Y_i = \sum_{j=1}^{K} \hat{c}(Q, K_j) V_j .
$$

There are two differences between the predictive paradigm and the non-predictive paradigm. Firstly, to get the output of the current segment $Y_i$, the query of the previous segment $Q_{i-1}$ is used to calculate correlations with the keys of all segments $\{K_1, \ldots, K_{i-1}\}$. Secondly, the correlation $\hat{c}(Q_{i-1}, K_j)$ corresponding to the key of a certain segment $K_j$ is regarded as the weight of the next segment $V_{j+1}$ to aggregate values. That is, the segment of value is offset forward by one segment relative to the segment of the corresponding key. Therefore, we can obtain $Y_i$ by the following equation:

$$
Y_i = \sum_{j=1}^{K} \hat{c}(Q_{i-1}, K_j) V_{j+1} .
$$

### 2.4. Complexity Analysis

For Single-Scale Segment-Correlation, if the segment length $L_{seg} = L_0$, the computational complexity is $O(L^2 / L_0)$. For Multi-Scale Segment-Correlation, the computational complexity is the sum of all scales, which is still at the same computational level as $O(L^2 / L_0)$ benefiting from the exponential increase in segment length. Most other sparse mechanisms introduce some extra operations, such as selections of dominant queries in Informer and fast Fourier transform in Autoformer. Although their theoretical complexity is $O(L \log L)$, the actual operation efficiency is not even as good as our Multi-Scale Segment-Correlation. Please see section 3.4 for more details.

### 3. EXPERIMENTS

#### 3.1. Experimental Setup

We conduct experiments on five datasets. (1) ETT [14] contains data related to electricity which is collected from two stations in 2 years. (2) Electricity [14] contains the hourly electricity consumption of 321 clients in 2 years. (3) Exchange [3] collects the daily exchange rates of eight countries ranging from 1990 to 2016. (4) Traffic [14] collects the hourly road occupancy rates from the California Department of Transportation in 2 years. (5) Weather [15] records climatological data of the Max-Planck-Institute every 10 minutes in 2020 year. We split the datasets following Autoformer [15]. We train the model on train set, tune the hyper-parameters on validation set, and evaluate the performance on test set.

We use the ADAM [21] optimizer with an initial learning rate of 1e-4 and the learning rate decay to train our model. We utilize an early stop training strategy to avoid overfitting. The number of training epochs is set to 10 and the batch size is set to 32. Preformer contains two encoder layers and one decoder layer for all experimental settings. Previous work has demonstrated that Transformer outperforms RNN and TCN in long-term forecasting [14], so we only compare with four Transformer-based models, i.e., Autoformer [15], Informer [14], LogTrans [17], Pyraformer [16].

#### 3.2. Main Results

To comprehensively compare Preformer and baselines, we mainly conduct multivariate forecasting experiments on various datasets under multiple settings, i.e., $d_q > 1$. Preformer is also suitable for univariate forecasting. For all datasets, the input length is fixed to 96 and the prediction length is chosen from $\{96, 192, 336, 720\}$. From table 1, we find that Preformer is better than other models except Autoformer in all cases and performs slightly worse than Autoformer in only a few settings. For example, under the input-96-predict-192 setting, compared to previous state-of-the-art results, Preformer has achieved 4.3% (0.281 → 0.269) relative MSE improvement in ETTm2, 14.9% (0.222 → 0.189) in Electricity, 10.7% (0.242 → 0.216) in Exchange, 6.6% (0.222 → 0.210) in Traffic, and 10.7% (0.182 → 0.166) in Weather.
(0.300 → 0.268) in Exchange, 8.3% (0.616 → 0.565) in Traffic, 10.4% (0.307 → 0.275) in Weather. Moreover, Preformer shows stable performance in cases of long prediction horizons, which shows that it is suitable for long-term forecasting. Autoformer is the best prediction model besides our Preformer. Therefore, we plot the forecasting results of Preformer and Autoformer in fig. 3 for further comparison. Our Preformer can accurately predict the periodicity, trend and even some small fluctuations. Though under a very long prediction horizon, Preformer can also perform well.

3.3. Ablation Study

Table 2. Ablations of predictive paradigm and multi-scale structure.

| Models       | MS+Predictive (ours) | Only Predictive (without MS) | Only MS (without Predictive) |
|--------------|----------------------|------------------------------|-------------------------------|
|              | MSE      | MAE   | MSE      | MAE   | MSE      | MAE   |
| ETT1         | 96       | 0.414 | 0.439 | 0.480   | 0.472 | 0.416   | 0.436 |
|              | 192      | 0.445 | 0.455 | 0.493   | 0.484 | 0.472   | 0.472 |
|              | 336      | 0.466 | 0.468 | 0.511   | 0.497 | 0.471   | 0.475 |
|              | 720      | 0.471 | 0.487 | 0.605   | 0.553 | 0.484   | 0.499 |
| Exchange     | 96       | 0.148 | 0.282 | 0.186   | 0.305 | 0.187   | 0.305 |
|              | 192      | 0.268 | 0.378 | 0.359   | 0.457 | 0.280   | 0.386 |
|              | 336      | 0.447 | 0.499 | 0.704   | 0.659 | 0.520   | 0.542 |
|              | 720      | 1.092 | 0.812 | 1.400   | 0.931 | 1.232   | 0.883 |
| Traffic      | 96       | 0.560 | 0.349 | 0.561   | 0.352 | 0.567   | 0.351 |
|              | 192      | 0.565 | 0.349 | 0.573   | 0.356 | 0.583   | 0.358 |
|              | 336      | 0.577 | 0.351 | 0.577   | 0.353 | 0.581   | 0.354 |
|              | 720      | 0.597 | 0.358 | 0.599   | 0.363 | 0.598   | 0.362 |

Impact of the multi-scale structure. The multi-scale structure can effectively extract dependencies at different temporal resolutions, which is important for time series forecasting. To illustrate this, we remove the multi-scale structure from all PreMSSC and MSSC modules in Preformer to get the model without the multi-scale structure. As shown in table 2, the prediction performance of Preformer with the multi-scale structure (MS+Pre) is better than that without the multi-scale structure (Only Pre) in all cases. Especially on the ETT1 and Exchange datasets, the multi-scale structure can lead to considerable performance improvements.

Impact of the predictive paradigm. To explore whether the predictive paradigm is really effective for prediction tasks, we conduct ablations of it on three datasets with different settings. As shown in table 2, MS+Pre represents the standard Preformer model, while Only MS means that replacing the PreMSSC module in Preformer’s decoder with the MSSC module. The experimental results show that Preformer with the predictive paradigm achieves better performance in almost all cases, which proves that the proposed predictive paradigm is helpful for the prediction tasks.

The sensitivity analysis of the segment length. The segment length $L_{seg}$ is a critical hyperparameter in Segment-Correlation. The multi-scale structure can facilitate the selection of segment length. We choose different initial segment lengths $L_0$ from {2, 4, 8, 12, 24, 48} on four datasets for our experiments. As shown in fig. 4, Preformer with the multi-scale structure performs more stably and better on all datasets, i.e., the mean and variance of MSE scores are smaller.

3.4. Efficiency analysis

We compare the running memory and time of models with different sparse attention mechanisms during the training phase in fig. 5. It is worth noting that Multi-Scale Segment-Correlation is more efficient in time and memory than other sparse mechanisms whose theoretical complexity $O(L \log L)$ in practical applications since the Fast Fourier Transform in Autoformer and the selections of dominant queries in Informer both require additional time-space complexity. Extensive experiments show the superiority of Preformer.

4. CONCLUSION

In this paper, we propose a Transformer-based model called Preformer for long-term time series forecasting. In Preformer, we introduce an efficient attention mechanism called Multi-Scale Segment-Correlation (MSSC) which utilizes correlations between segment pairs to discover dependencies and aggregate information in time series. Further, we design a novel predictive paradigm and combine it with MSSC to get Predictive Multi-Scale Segment-Correlation (PreMSSC), which can discover predictive dependencies from contexts. Extensive experiments show the superiority of Preformer.
5. REFERENCES

[1] Jerome Connor, Les E Atlas, and Douglas R Martin, “Recurrent networks and narma modeling,” in *Advances in Neural Information Processing Systems*, 1992, pp. 301–308.

[2] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” in *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

[3] 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*, 2018, pp. 95–104.

[4] Huan Song, Deepta Rajan, Jayaraman J Thiagarajan, and Andreas Spanias, “Attend and diagnose: Clinical time series analysis using attention models,” in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[5] David Salinas, Valentín Flunkert, Jan Gasthaus, and Tim Januschowski, “Deepar: Probabilistic forecasting with autoregressive recurrent networks,” *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[6] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

[7] Rajat Sen, Hsiang-Fu Yu, and Inderjit Dhillon, “Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting,” in *Advances in Neural Information Processing Systems*, 2019, pp. 5998–6008.

[8] Kin G Olivares, Cristian Challu, Grzegorz Marcjansz, Rafal Weron, and Artur Dubrawski, “Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx,” *International Journal of Forecasting*, 2022.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30.