Efficiently Enhancing Long-term Series Forecasting via Ultra-long Lookback Windows

Suxin Tong[1,2], **Jingling Yuan**[1,2*]

[1]School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, China
[2]Hubei Key Laboratory of Transportation Internet of Things, Wuhan University of Technology, Wuhan 430070, China
suxin_tong@whut.edu.cn, yjl@whut.edu.cn

Abstract

Long-term series forecasting aims to predict future data over long horizons based on historical information. However, existing methods struggle to effectively utilize long lookback windows due to overfitting, computational resource constraints, or information extraction challenges, thereby limiting them to using limited lookback windows for predicting long-term future series. To address these issues, this paper introduces the Input Refinement and Prediction Auxiliary (IRPA) framework, a lightweight model consisting of four linear layers designed to extract key information from ultra-long lookback windows to enhance limited lookback windows and assist prediction processes. IRPA comprises an Input Refinement Module (IRM) and a Prediction Auxiliary Module (PAM), each constructed from two linear layer sub-modules. The IRM performs effective decomposition and patching of ultra-long series, refining seasonal and trend features to increase the information density in limited lookback windows and mitigate overfitting and parameter inflation. The PAM extracts historical similarities and seasonal patterns from ultra-long lookback windows to significantly improve prediction accuracy. IRPA substantially extends the utilization of lookback windows, offering a lightweight and efficient solution with broad applicability. Experimental results on eight datasets show IRPA reduces the Mean Squared Error (MSE) by an average of 16.1% for various models.

Code — https://github.com/SuxinTong/IRPA

Introduction

Long-term series forecasting, as a pivotal branch within the domain of time series analysis, plays an indispensable role across an array of critical fields. These include traffic flow prediction (Jiang et al. 2023), meteorological forecasting (Huang et al. 2023), and recommendation systems (Chen et al. 2024a), etc. Accurate long-term predictions not only arm policymakers with reliable data for decision-making but also facilitate long-term planning and enable early warning systems (Wu et al. 2021; Zhou et al. 2021; Cao et al. 2023). In recent years, the swift advancements in deep learning technologies have led to the emergence of various neural network-based time series forecasting models (Rasul et al. 2021; Nie et al. 2023; Zeng et al. 2023; Liu et al. 2024), which have demonstrated superior performance.

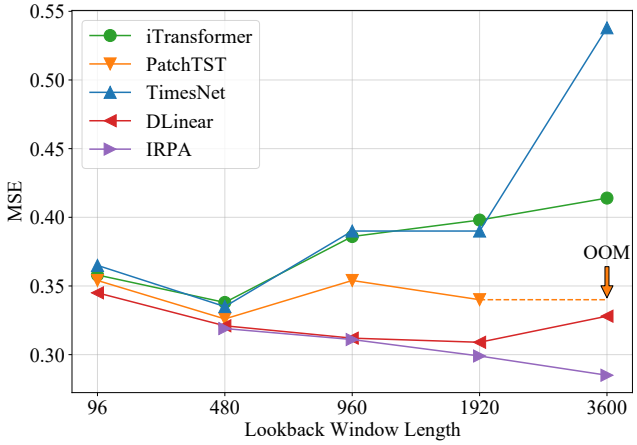

Figure 1: Existing models show overfitting and resource limitations with increasing lookback window length, while IRPA effectively mitigates these issues in the Weather dataset at prediction length k=720.

Despite these advancements, the effective utilization of long lookback windows in long-term series forecasting remains a significant challenge. Long lookback windows contain rich information about long-term trends, complex seasonal patterns, and rare events that can be crucial for accurate long-term predictions (Jia et al. 2024). However, leveraging this extensive historical data introduces several challenges, as shown in Figure 1:

1. **Overfitting Issues:** When processing long lookback windows, models often tend to overfit, capturing noise within the data and leading to unstable forecasting.
2. **Computational Resource Constraints:** Expanding historical lookback windows significantly increases computational costs and memory requirements, potentially exceeding practical resource limitations.
3. **Information Extraction Challenges:** Effectively distilling relevant information from long lookback windows poses a significant challenge. Models need to identify and leverage key patterns and trends while disregarding irrelevant or redundant information.

*Corresponding author.

To address these challenges, we propose an innovative solution centered on refining relevant information from ultra-long lookback windows. Our approach ensures that even limited historical windows capture richer content and leverage similar patterns within the historical data to enhance predictive accuracy. Compared to directly utilizing the entire ultra-long lookback windows, this approach effectively avoids overfitting due to excessive noise and prevents parameter proliferation. The motivation behind this approach lies in the fact that ultra-long lookback windows contain rich information about long-term trends and periodicities, while the most recent limited lookback window immediately preceding the future series often captures the most relevant temporal features and patterns for the prediction target period. If the association between these two can be effectively computed and historical information extracted to assist predictions, it can significantly enhance the accuracy and stability of forecasts.

Founded on this motivation, we craft a lightweight and universally applicable framework, the IRPA (Input Refinement and Prediction Auxiliary) framework. Comprising an Input Refinement Module (IRM) and a Prediction Auxiliary Module (PAM), this framework utilizes only four linear layers to facilitate efficient information extraction and predictive enhancement. Specifically, the IRM performs seasonal-trend decomposition and patching, leveraging the periodic and repetitive characteristics of seasonal components to guide the refinement of seasonal and trend features. This increases the information density in limited lookback windows and significantly mitigates parameter inflation. The PAM extracts historical similarities and seasonal patterns from ultra-long lookback windows, providing additional reference information for the prediction process and thus boosting the accuracy of predictions.

Our contributions are summarized as follows:

- We introduce the IRPA framework, an efficient plug-and-play solution that refines ultra-long historical time series for improved long-term series forecasting.
- We design an Input Refinement Module that efficiently extracts seasonal and trend-related information from ultra-long lookback windows, enhancing the information density in limited lookback windows and mitigating overfitting and parameter inflation.
- We propose a Prediction Auxiliary Module that leverages historical similarities and seasonal patterns to offer extra predictive guidance, improving forecasting outcomes.
- Experimental results demonstrate IRPA's effectiveness in enhancing various time series forecasting models, reducing MSE by 9.7% on iTransformer, 15.0% on PatchTST, 14.4% on TimesNet, and 25.2% on DLinear.

Related Work

Time Series Forecasting

Time series forecasting involves predicting future values using historical data. Traditional time series forecasting methods primarily rely on statistical models (Farlie 1964; Watson 1994; Makridakis and Hibon 1997). In recent years, machine learning methods (Kan et al. 2022; Sun and Yu 2023), especially deep learning models, have demonstrated formidable performance in time series forecasting tasks. Recurrent Neural Networks (RNNs) and their variants (Hochreiter and Schmidhuber 1997; Wen et al. 2017; Baytas et al. 2017; Cao et al. 2018; Salinas et al. 2020) were among the early deep learning approaches for handling sequential data, though they struggle with capturing long-term dependencies in long sequences. Convolutional Neural Networks (CNNs) (van den Oord et al. 2016; Bai, Kolter, and Koltun 2018; Chen et al. 2020; Wang et al. 2023) have also been applied to time series analysis, and they capture long-range temporal correlations through causal and dilated convolutions. Recently, the Transformer architecture (Vaswani et al. 2017; Lim et al. 2021; Yang et al. 2022) has excelled in capturing long-term dependencies through its self-attention mechanism. Enhanced attention models (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022) improve sequence handling but are relatively computationally intensive. Multilayer Perceptrons (MLPs), as a simple and efficient neural network structure, have demonstrated significant potential in time series forecasting. Recent research (Oreshkin et al. 2020; Challu et al. 2023; Olivares et al. 2023) has captured temporal patterns through stacked fully connected layers and residual branches, while DLinear and NLinear (Zeng et al. 2023) have exhibited remarkable performance in time series forecasting due to their lightweight and effective structures. This paper proposes a new framework based on the MLP architecture, aiming to effectively address the time series forecasting challenge with long lookback windows, leveraging the simplicity and computational efficiency of MLPs while enhancing their ability to model long-term dependencies.

Time Series Forecasting with Long Lookback Windows

Time series forecasting with long lookback windows involves using an extended period of historical data to predict future data points. Addressing this problem is critical for capturing long-term dependencies and intricate temporal patterns. However, handling long sequence data presents challenges such as high computational complexity, substantial memory consumption, and susceptibility to overfitting. Current solutions primarily focus on two directions: one approach entails segmenting the time series into overlapping or non-overlapping patches, as done in (Nie et al. 2023; Zhang and Yan 2023; Chen et al. 2024b), which utilize self-attention mechanisms to capture long-term dependencies. The other approach involves directly leveraging the entire time series to capture long-term dependencies, with models like iTransformer (Liu et al. 2024) introducing inverted dimensions for series embeddings, and another work (Wang et al. 2024b) employing both patch-level and series-level representations to capture endogenous and exogenous variables. Notwithstanding the progress made, these solutions could still face constraints like high computational complexity and overfitting tendencies. To tackle these challenges, this paper employs seasonal-trend decomposition, patching, and similarity calculation to efficiently and effectively extract information from long lookback windows.

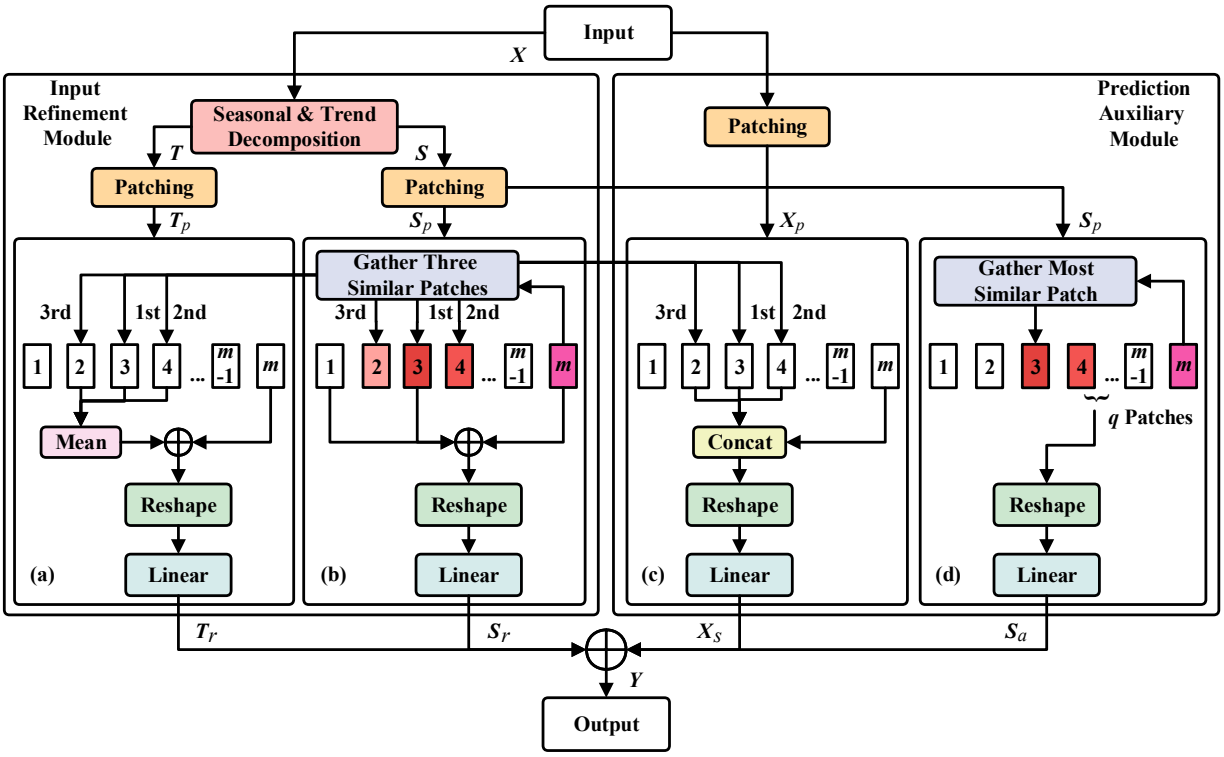

Figure 2: The overall structure of IRPA. (a) Trend Refinement Module. (b) Seasonal Refinement Module. (c) Historical Similarity Auxiliary Module. (d) Historical Seasonal Pattern Prediction Auxiliary Module.

Methodology

Problem Definition

Define the input historical time series data as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l\} \in \mathbb{R}^{b\times n\times l}$, where b is the batch size, l is the input lookback window length, and n is the number of variables. Our objective is to refine the historical time series of length l into a shorter time series of length r ($5r \leq l$), denoted as $\{\boldsymbol{x}_1^{'}, \boldsymbol{x}_2^{'}, \ldots, \boldsymbol{x}_r^{'}\} \in \mathbb{R}^{b\times n\times r}$, and then predict the future k timesteps $\boldsymbol{Y} = \{\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_{l+k}\} \in \mathbb{R}^{b\times n\times k}$ based on the refined series. Formally, it can be expressed as $\boldsymbol{Y} = f(\boldsymbol{X};\theta)$, where $\boldsymbol{Y}$ is the predicted future data, f is the prediction model, and θ represents the model parameters.

Overall Structure

Figure 2 shows the overall structure of IRPA, which consists of two key components: the Input Refinement Module (IRM) and the Prediction Auxiliary Module (PAM). The IRM is responsible for refining key features from the input data, while the PAM provides additional auxiliary information to improve prediction accuracy. Specifically, the Input Refinement Module includes seasonal and trend refinement modules; the input time series is first decomposed into trend and seasonal components, then these components are patched and input into the seasonal and trend refinement modules, which refine the components by identifying similar seasonal patterns in the long historical data. The Prediction Auxiliary Module includes historical similarity and historical seasonal pattern prediction auxiliary modules. The prediction auxiliary module identifies similar historical patches and their subsequent seasonal information from the ultra-long lookback window to provide additional reference information for the forecasting process. In the following sections, we will detail the structure and function of these two modules.

Input Refinement Module

The Input Refinement Module focuses on extracting critical information from raw input data and refining it to enhance prediction accuracy and efficiency. To effectively extract key information, we decompose the input time series data into seasonal and trend components and refine them separately. This decomposition reveals the underlying features of the data, allowing the model to focus on these distinct aspects. Moreover, the effectiveness of seasonal-trend decomposition has been demonstrated in numerous works (Wu et al. 2021; Zeng et al. 2023; Wang et al. 2023). Therefore, decomposing the input into seasonal and trend components for separate refinement can enhance the model's accuracy and stability. For the input time series data $\boldsymbol{X} \in \mathbb{R}^{b\times n\times l}$, it first undergoes a normalization process (Liu et al. 2022b), where $\boldsymbol{X}$ is subtracted by the mean and divided by the variance vector for all channels, which helps improve the model's stability and performance. We then decompose $\boldsymbol{X}$ into seasonal

component $\boldsymbol{S}$ and trend component $\boldsymbol{T}$ as follows:

$$\boldsymbol{T} = \text{AvgPool}(\text{Padding}(\boldsymbol{X})), \boldsymbol{S} = \boldsymbol{X} - \boldsymbol{T}, \quad (1)$$

where AvgPool denotes the moving average operation, and padding is applied to maintain a consistent series length. Subsequently, we perform the patching operation (Nie et al. 2023) on both the decomposed seasonal and trend components and the undecomposed series. The operation segments each of these components into patches of a specified length r with a stride of $r/2$, and the resulting m patches are reshaped. This patching process effectively divides the extended series into multiple shorter patches, facilitating subsequent similarity computations and information extraction. The formula for this process is articulated as:

$$\begin{aligned} \boldsymbol{S}_p = \text{Patching}(\boldsymbol{S}), \boldsymbol{T}_p = \text{Patching}(\boldsymbol{T}), \\ \boldsymbol{X}_p = \text{Patching}(\boldsymbol{X}), \end{aligned} \quad (2)$$

where $\boldsymbol{S}_p, \boldsymbol{T}_p, \boldsymbol{X}_p \in \mathbb{R}^{b \cdot n \times m \times r}$ represent the patched seasonal, trend, and undecomposed series, respectively.

Seasonal Refinement Module In the Seasonal Refinement Module, we choose to identify similar seasonal components from the extensive historical data and use these to guide the refinement of both seasonal and trend components. This approach is due to the cyclical and repetitive nature of seasonal components, which typically recur at fixed intervals, making it easier to find similar seasonal patterns within historical data. In contrast, trend components are often significantly influenced by external factors such as policy changes or climatic fluctuations. Directly searching for similar trend components may introduce unnecessary noise, reducing the model's predictive accuracy. To precisely locate similar historical patterns, we employ the Hellinger distance (Hellinger 1909) to quantify patch similarity, given its well-defined range and stability. Additionally, by sharing computational results across modules, we optimize efficiency. The formula is given by:

$$\boldsymbol{H} = \sqrt{1 - \sum_{i=0}^{r-1} \sqrt{\phi(\boldsymbol{S}_p[:,:,i]) \odot \phi(\boldsymbol{S}_p[:,-1,i])}}, \quad (3)$$

where $\boldsymbol{H} \in \mathbb{R}^{b \cdot n \times m}$ represents the computed seasonal similarity matrix, ϕ denotes the softmax activation function, and $\odot$ indicates the element-wise product. Selecting the last patch as the baseline for similarity calculation is primarily due to its proximity to the future series to be predicted, thus containing the temporal features and patterns most relevant to the target prediction period. This design not only effectively guides the refinement process of the historical series but also better captures the latest changes and trends that may impact future predictions.

We then select the most similar seasonal patch to guide the refinement of the seasonal component, leveraging the strong regularity of seasonal patterns to ensure stability and accuracy. Additionally, we incorporate the first patch as an anchor point, enabling the model to better capture the overall temporal evolution patterns from the start to the end of the series. The formula is as follows:

$$\boldsymbol{S}_r^{'} = (\text{Gather}(\boldsymbol{S}_p,\ \text{argmax}(\boldsymbol{H})) + \boldsymbol{S}_p[:,0,:]) \odot \sigma(\boldsymbol{W}_h), \quad (4)$$

where $\boldsymbol{W}_h \in \mathbb{R}^{b \cdot n \times r}$ denotes a sigmoid-gated weight matrix, employed to dynamically integrate the gathered seasonal component with the last seasonal patch. Subsequently, we apply a Reshape operation followed by a linear transformation to derive the refined seasonal output:

$$\boldsymbol{S}_r = \text{Reshape}(\boldsymbol{S}_r^{'} + \boldsymbol{S}_p[:,-1,:] \odot (1 - \sigma(\boldsymbol{W}_h)))\boldsymbol{W}_s + \boldsymbol{b}_s, \quad (5)$$

where $\boldsymbol{S}_r \in \mathbb{R}^{b \times n \times k}$ is the refined seasonality component, $\boldsymbol{W}_s \in \mathbb{R}^{b \times r \times k}$ is the weight matrix, and $\boldsymbol{b}_s$ is the bias. σ denotes the sigmoid activation function.

Trend Refinement Module For trend refinement, we gather the top three most similar trend patches: this approach is adopted because trend components may be influenced by various factors, and choosing multiple similar trend patches can capture more trend variation information, reducing the potential impact of anomalies or noise from a single component. Averaging multiple similar components also minimizes the uncertainty of individual trend components. The calculation of similar trend item $\boldsymbol{T}_r^{'} \in \mathbb{R}^{b \cdot n \times r}$ is given by:

$$\boldsymbol{T}_r^{'} = \text{Mean}(\sigma(\text{Gather}(\boldsymbol{S}_p,\ \text{topk}(\boldsymbol{H}, 3)))) \odot \sigma(\boldsymbol{W}_r), \quad (6)$$

Similarly to the seasonality refinement, the refined trend component is computed as:

$$\boldsymbol{T}_r = \text{Reshape}(\boldsymbol{T}_r^{'} + \boldsymbol{T}_p[:,-1,:] \odot (1 - \sigma(\boldsymbol{W}_r)))\boldsymbol{W}_t + \boldsymbol{b}_t, \quad (7)$$

where $\boldsymbol{T}_r \in \mathbb{R}^{b \times n \times k}$ is the refined trend component, $\boldsymbol{W}_r \in \mathbb{R}^{b \cdot n \times r}$, $\boldsymbol{W}_t \in \mathbb{R}^{b \times r \times k}$ are weight matrices, $\boldsymbol{b}_t$ is the bias.

Prediction Auxiliary Module

The Prediction Auxiliary Module aims to provide additional information to support the forecasting process, thereby improving prediction accuracy. The module consists of two key parts: the Historical Similarity Auxiliary Module and the Historical Seasonal Pattern Prediction Auxiliary Module.

Historical Similarity Auxiliary Module We opt to use the undecomposed original historical series to support predictions rather than the decomposed seasonal or trend historical series. This choice is based on the consideration that undecomposed similar items encapsulate the combined information of seasonality and trends, providing more references for the predictive process. Certain patterns may exist in the interaction between seasonality and trends; using the original series can capture these complex relationships. In contrast, using only decomposed seasonal and trend components might overlook some critical information. The calculation of historical similar item $\boldsymbol{X}_s^{'} \in \mathbb{R}^{b \cdot n \times 4 \times r}$ is:

$$\boldsymbol{X}_s^{'} = \text{Concat}(\text{Gather}(\boldsymbol{X}_p,\ \text{topk}(\boldsymbol{H}, 3)), \boldsymbol{X}_p[:,-1,:]). \quad (8)$$

Then, we perform a Reshape operation and linear transformation on the historical similar item:

$$\boldsymbol{X}_s = \text{Reshape}(\boldsymbol{X}_s^{'})\boldsymbol{W}_x + \boldsymbol{b}_x, \quad (9)$$

where the result of the historical similarity auxiliary $\boldsymbol{X}_s \in \mathbb{R}^{b \times n \times k}$, $\boldsymbol{W}_x \in \mathbb{R}^{b \times 4r \times k}$ is the weight matrix, and $\boldsymbol{b}_x$ represents the bias.

Historical Seasonal Pattern Prediction Auxiliary Module This module identifies the most similar seasonal patterns in historical data and uses their subsequent patterns to guide future predictions. We focus on seasonality components rather than trend components or the undecomposed original historical series because seasonality components have inherent periodicity and repeatability. This approach provides reliable and regular reference information while avoiding the introduction of uncertain trend information that could mislead predictions. To ensure that the selected similar seasonal patterns cover a sufficient subsequent time span, we define a threshold $q{=}\lceil k/r \rceil$, where k is the prediction length, ensuring that $q \cdot r \geq k$, meaning that the selected historical patterns include at least one complete subsequent development to guide future predictions. Thus, the most similar seasonal pattern auxiliary indices $\boldsymbol{S}_a^{'} \in \mathbb{R}^{b \cdot n \times q}$ are calculated as:

$$\boldsymbol{S}_a^{'}{=}\text{argmax}(\boldsymbol{H}[:,:(m-q)]) + [\mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{q}]. \quad (10)$$

Then, the subsequent k values of the most similar seasonal pattern are utilized to provide auxiliary prediction information:

$$\boldsymbol{S}_a = (\text{Reshape}(\text{Gather}(\boldsymbol{S}_p, \boldsymbol{S}_a^{'}))[:,:,:k])\boldsymbol{W}_a + \boldsymbol{b}_a, \quad (11)$$

where $\boldsymbol{S}_a \in \mathbb{R}^{b \times n \times k}$ represents the auxiliary prediction information, $\boldsymbol{W}_a \in \mathbb{R}^{b \times k \times k}$ is the weight matrix, and $\boldsymbol{b}_a$ denotes the bias.

The prediction result $\boldsymbol{Y} \in \mathbb{R}^{b \times n \times k}$ is the sum of the outputs from the four modules, then de-normalized to obtain the final prediction result. This combined prediction method aims to leverage the diverse features and information extracted by each module, potentially producing more accurate and robust prediction results. The formula is as follows:

$$\boldsymbol{Y} = \boldsymbol{S}_r + \boldsymbol{T}_r + \boldsymbol{X}_s + \boldsymbol{S}_a. \quad (12)$$

Experiments

Experimental Settings

Datasets To comprehensively evaluate the performance of the proposed IRPA framework, we have selected eight openly accessible datasets that are widely utilized in the realm of time series forecasting research, including Weather, Solar, ECL, Traffic, and four ETT datasets (Liu et al. 2024). They exhibit significant diversity in terms of temporal granularity, the number of variables, and data attributes, presenting a thorough and challenging testing environment for our model. Detailed statistics are in Table 1.

Baselines To thoroughly assess the effectiveness of the IRPA framework, we selected a suite of state-of-the-art time series forecasting models as baselines for comparison. These include Transformer-based models: iTransformer (Liu et al. 2024), PatchTST (Nie et al. 2023), Autoformer (Wu et al. 2021), and CNN-based models: TimesNet (Wu et al. 2023), SCINet (Liu et al. 2022a), as well as the linear model DLinear (Zeng et al. 2023).

Implementation Details All experiments were implemented using PyTorch 2.1.2 (Paszke et al. 2019) and conducted in a Linux environment equipped with two NVIDIA Tesla T4 16GB GPUs. We used L2 loss and the Adam (Kingma and Ba 2015) optimizer for model training, with an initial learning rate of 0.01. Each model was trained for 10 epochs with a batch size of 32. For the IRPA framework, the input series length was set to {480,1920,3600} based on the dataset's time sampling frequency. The series refinement length and patch length were both fixed at 96. For all baseline models, the input series length was set to 96. The prediction series length was set to {96,192,336,720}, covering various forecasting ranges. The data partitioning and baseline model experimental setup followed the configurations in (Liu et al. 2024), and the code utilized the implementation provided by (Wang et al. 2024a). To enhance reproducibility, we fixed random seeds. MSE and Mean Absolute Error (MAE) were used as primary evaluation metrics, providing insights into both variance and magnitude of prediction errors.

Datasets	Features	Timesteps	Frequency
ETTh1 & ETTh2	7	17,420	1 Hour
ETTm1 & ETTm2	7	69,680	15 Minutes
Weather	21	52,696	10 Minutes
Solar	137	52,560	10 Minutes
ECL	321	26,304	1 Hour
Traffic	862	17,544	1 Hour

Table 1: Statistics of the experimental datasets.

Main Results

Forecasting results Table 2 meticulously illustrates the long-term series forecasting performance of IRPA compared to six baseline models across eight datasets. The results indicate that IRPA achieved the best results in 54 out of 64 long-term forecasting tasks, significantly outperforming other models. Compared to the previous state-of-the-art method, iTransformer, IRPA reduced the average MSE by 9.4%. IRPA demonstrated outstanding performance across datasets with various characteristics, showcasing its strong generalization ability and adaptability. This suggests that the enhancement of input feature quality is as crucial as model improvements. As the prediction length increases, IRPA's advantage over baseline models becomes more pronounced, particularly in 720-time-step forecasts. This is primarily due to IRPA's series refinement mechanism, which more effectively captures historical long-term information for time series forecasting.

Framework generality To validate the generality of our framework, we applied the IRPA to baseline time series prediction models and analyzed the performance improvement. Specifically, the extended input series was refined by the Input Refinement Module before being fed into the subsequent baseline model, and the model's prediction output was augmented by adding the Prediction Auxiliary Module. As shown in Table 3, IRPA reduced the MSE by 9.7% on iTransformer, 15.0% on PatchTST, 14.4% on TimesNet, and 25.2% on DLinear, surpassing previous state-of-the-art levels. Overall, IRPA demonstrates strong generality and effectiveness across different models and datasets.

Model		IRPA		iTransformer		PatchTST		TimesNet		DLinear		SCINet		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.359**	**0.386**	0.386	0.405	0.414	0.419	0.384	0.402	0.386	0.400	0.654	0.599	0.449	0.459
	192	**0.391**	**0.408**	0.441	0.436	0.460	0.445	0.436	0.429	0.437	0.432	0.719	0.631	0.500	0.482
	336	**0.418**	**0.428**	0.487	0.458	0.501	0.466	0.491	0.469	0.481	0.459	0.778	0.659	0.521	0.496
	720	**0.454**	**0.464**	0.503	0.491	0.500	0.488	0.521	0.500	0.519	0.516	0.836	0.699	0.514	0.512
ETTh2	96	**0.270**	**0.334**	0.297	0.349	0.302	0.348	0.340	0.374	0.333	0.387	0.707	0.621	0.346	0.388
	192	**0.320**	**0.372**	0.380	0.400	0.388	0.400	0.402	0.414	0.477	0.476	0.860	0.689	0.456	0.452
	336	**0.345**	**0.395**	0.428	0.432	0.426	0.433	0.452	0.452	0.594	0.541	1.000	0.744	0.482	0.486
	720	**0.393**	**0.432**	0.427	0.445	0.431	0.446	0.462	0.468	0.831	0.657	1.249	0.838	0.515	0.511
ETTm1	96	**0.307**	**0.359**	0.334	0.368	0.329	0.367	0.338	0.375	0.345	0.372	0.418	0.438	0.505	0.475
	192	**0.329**	**0.369**	0.377	0.391	0.367	0.385	0.374	0.387	0.380	0.389	0.439	0.450	0.553	0.496
	336	**0.347**	**0.380**	0.426	0.420	0.399	0.410	0.410	0.411	0.413	0.413	0.490	0.485	0.621	0.537
	720	**0.395**	**0.414**	0.491	0.459	0.454	0.439	0.478	0.450	0.474	0.453	0.595	0.550	0.671	0.561
ETTm2	96	**0.162**	**0.255**	0.180	0.264	0.175	0.259	0.187	0.267	0.193	0.292	0.286	0.377	0.255	0.339
	192	**0.213**	**0.293**	0.250	0.309	0.241	0.302	0.249	0.309	0.284	0.362	0.399	0.445	0.281	0.340
	336	**0.264**	**0.330**	0.311	0.348	0.305	0.343	0.321	0.351	0.369	0.427	0.637	0.591	0.339	0.372
	720	**0.317**	**0.368**	0.412	0.407	0.402	0.400	0.408	0.403	0.554	0.522	0.960	0.735	0.433	0.432
Weather	96	**0.156**	**0.214**	0.174	0.214	0.177	0.218	0.172	0.220	0.196	0.255	0.221	0.306	0.266	0.336
	192	**0.198**	**0.253**	0.221	0.254	0.225	0.259	0.219	0.261	0.237	0.296	0.261	0.340	0.307	0.367
	336	**0.235**	**0.287**	0.278	0.296	0.278	0.297	0.280	0.306	0.283	0.335	0.309	0.378	0.359	0.395
	720	**0.285**	**0.326**	0.358	0.349	0.354	0.348	0.365	0.359	0.345	0.381	0.377	0.427	0.419	0.428
Solar	96	**0.196**	0.278	0.203	**0.237**	0.234	0.286	0.250	0.292	0.290	0.378	0.237	0.344	0.884	0.711
	192	**0.216**	0.292	0.233	**0.261**	0.267	0.310	0.296	0.318	0.320	0.398	0.280	0.380	0.834	0.692
	336	**0.235**	0.303	0.248	**0.273**	0.290	0.315	0.319	0.330	0.353	0.415	0.304	0.389	0.941	0.723
	720	**0.237**	0.295	0.249	**0.275**	0.289	0.317	0.338	0.337	0.356	0.413	0.308	0.388	0.882	0.717
ECL	96	**0.143**	**0.240**	0.148	0.240	0.195	0.285	0.168	0.272	0.197	0.282	0.247	0.345	0.201	0.317
	192	**0.156**	**0.250**	0.162	0.253	0.199	0.289	0.184	0.289	0.196	0.285	0.257	0.355	0.222	0.334
	336	**0.171**	**0.265**	0.178	0.269	0.215	0.305	0.198	0.300	0.209	0.301	0.269	0.369	0.231	0.338
	720	**0.205**	**0.293**	0.225	0.317	0.256	0.337	0.220	0.320	0.245	0.333	0.299	0.390	0.254	0.361
Traffic	96	0.429	0.296	**0.395**	**0.268**	0.544	0.359	0.593	0.321	0.650	0.396	0.788	0.499	0.613	0.388
	192	0.436	0.296	**0.417**	**0.276**	0.540	0.354	0.617	0.336	0.598	0.370	0.789	0.505	0.616	0.382
	336	0.438	0.296	**0.433**	**0.283**	0.551	0.358	0.629	0.336	0.605	0.373	0.797	0.508	0.622	0.337
	720	**0.454**	**0.302**	0.467	0.302	0.586	0.375	0.640	0.350	0.645	0.394	0.841	0.523	0.660	0.408

Table 2: Forecasting results with different input series lengths l (96 for baselines, $\{480,1920,3600\}$ for IRPA and then refined length to r=96), with prediction length $k \in \{96,192,336,720\}$. The best results are in **bold**, the second-best are underlined.

Model Analysis

Ablation Study We conducted extensive ablation studies on IRPA to investigate the impact of its various components on prediction performance. The results in Table 4 show that removing the Seasonal Refinement Module (w/o SRM), Trend Refinement Module (w/o TRM), Historical Similarity Auxiliary Module (w/o HSAM), and Historical Seasonal Pattern Prediction Auxiliary Module (w/o HSPPAM) leads to a decrease in model performance. This indicates that capturing the seasonality and long-term trends of the data is crucial for improving predictive accuracy, particularly for data with evident long-term variations, such as weather patterns or electrical load. Utilizing historical similar patterns and seasonal patterns can also improve predictive accuracy. These results validate the effectiveness of the IRPA framework design and demonstrate that the model can effectively leverage various patterns and features within time series data to enhance predictive performance.

Efficiency analysis To comprehensively evaluate the efficiency of the IRPA framework, we conducted a thorough analysis across four key dimensions: model parameter count, memory footprint, training time per iteration, and prediction performance. In our experimental setup, the lookback window length for baseline models was set to 96. The "+lookback" designation indicates an extension of the lookback window length to 3600, while "+IRPA" represents the integration of the IRPA framework into the base model with the same lookback window length. As shown in Figure 1 and Table 5, the IRPA framework demonstrates significant computational efficiency advantages compared to simply in-

Datasets	ETT-Avg	Weather	Solar	ECL	Traffic
iTransformer	0.383	0.258	0.233	0.178	0.428
+IRPA	**0.339**	**0.226**	**0.207**	**0.168**	**0.416**
Promotion	11.5%	12.4%	11.2%	5.6%	2.8%
PatchTST	0.381	0.259	0.270	0.216	0.555
+IRPA	**0.337**	**0.222**	**0.209**	**0.174**	**0.457**
Promotion	11.5%	14.3%	22.6%	19.4%	17.7%
TimesNet	0.391	0.259	0.301	0.193	0.620
+IRPA	**0.348**	**0.241**	**0.204**	**0.182**	**0.459**
Promotion	11.0%	6.9%	32.2%	5.7%	26.0%
DLinear	0.442	0.265	0.330	0.212	0.625
+IRPA	**0.330**	**0.219**	**0.221**	**0.169**	**0.439**
Promotion	25.3%	17.4%	33.0%	20.3%	29.8%

Table 3: The MSE improvement of baseline models with the IRPA framework. Results are averaged from all prediction lengths. Avg means further averaged by subsets.

Datasets	ETTm2		Weather		ECL	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
IRPA	**0.239**	**0.312**	**0.219**	**0.270**	**0.169**	**0.262**
w/o SRM	0.242	0.315	0.221	0.276	0.169	0.262
w/o TRM	0.243	0.313	0.229	0.279	0.176	0.273
w/o HSAM	0.242	0.313	0.220	0.274	0.170	0.264
w/o HSPPAM	0.240	0.314	0.224	0.279	0.176	0.269

Table 4: Average results from all prediction lengths of the ablation study conducted on three different time sampling frequency datasets.

creasing the lookback window of baseline models. IRPA achieved an average reduction in MSE of at least 20.22% while simultaneously reducing the model's parameter count by 69.25%. This not only effectively mitigates overfitting issues but also successfully addresses challenges related to resource constraints, as evidenced by the resolution of Out-of-Memory (OOM) problems encountered with the PatchTST model. For time complexity, from Equation 3, the complexity is $O(mr) = O(\frac{2l}{r}r) = O(l)$. With fully connected layers, IRPA maintains $O(l)$ complexity.

Varying Lookback Window We explored the influence of varying historical lookback window lengths on the forecasting efficacy of the IRPA model across six datasets, revealing a systematic relationship between sampling frequency and optimal lookback window length. Figure 3 shows that 60-minute interval datasets (ETTh1, ECL) perform best with a 480 length, 15-minute interval datasets (ETTm1, ETTm2) with a 1920 length, and 10-minute interval datasets (Weather, Solar) with a 2880/3600 length. The ratios between sampling frequencies (60/15 = 4, 60/10 = 6) correspond to the ratios of optimal lookback windows (1920/480 = 4, 2880/480 = 6), suggesting a predictable scaling principle for window size selection in time series forecasting.

Model	Parameters	Memory	Train. Time	MSE
iTransformer	5.15M	389.6MB	32.5ms	0.358
+ lookback	6.95M	590.6MB	40.6ms	0.414
+IRPA	5.97M	495.6MB	43.3ms	0.292
PatchTST	10.74M	3642.0MB	128.3ms	0.354
+ lookback	172.20M	OOM	-	-
+IRPA	11.55M	3723.4MB	133.9ms	0.289
TimesNet	1.25M	2148.9MB	894.9ms	0.365
+ lookback	16.73M	4420.4MB	1388.3ms	0.538
+IRPA	2.07M	2549.8MB	748.5ms	0.332
DLinear	139.68K	72.4MB	8.3ms	0.345
+ lookback	5.19M	270.6MB	20.9ms	0.328
+IRPA	936.04K	199.4MB	22.3ms	0.285

Table 5: Static and runtime metrics of IRPA and other baseline models on the Weather Dataset with a prediction length k=720.

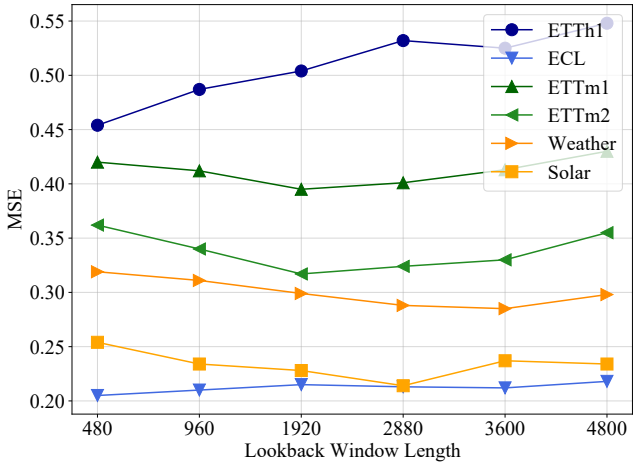

Figure 3: MSE results of IRPA with varying lookback window lengths at prediction length k=720.

Conclusion

In this paper, we introduce the IRPA framework, aimed at addressing critical challenges when using long lookback windows in long-term series forecasting, such as overfitting, inadequate information capture, and increasing demands on computational resources. IRPA innovatively refines inputs and auxiliary processes via specialized modules that efficiently extract crucial information from ultra-long lookback windows, thereby enhancing the quality of limited lookback windows and supporting the forecasting process. Experimental results indicate that IRPA significantly boosts the performance of various baseline models with only a minor increase in parameter count. This lightweight and generic framework provides a novel solution for long-term series forecasting, showcasing substantial advantages in enhancing predictive accuracy and computational efficiency.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62472332, No.62276196), the "Open Bidding for Selecting the Best Candidates" Project of Wuhan East Lake High-Tech Development Zone (No.2024KJB322), and the Hubei Provincial International Science and Technology Cooperation Project (No.2024EHA031).

References

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Baytas, I. M.; Xiao, C.; Zhang, X.; Wang, F.; Jain, A. K.; and Zhou, J. 2017. Patient Subtyping via Time-Aware LSTM Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 65–74.

Cao, H.; Huang, Z.; Yao, T.; Wang, J.; He, H.; and Wang, Y. 2023. Inparformer: Evolutionary decomposition transformers with interactive parallel attention for long-term time series forecasting. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, volume 37, 6906–6915.

Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; and Li, Y. 2018. BRITS: Bidirectional Recurrent Imputation for Time Series. *Advances in Neural Information Processing Systems*, 31: 6776–6786.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Ramírez, F. G.; Canseco, M. M.; and Dubrawski, A. 2023. NHITS: Neural Hierarchical Interpolation for Time Series Forecasting. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, volume 37, 6989–6997.

Chen, L.; Zhu, G.; Liang, W.; Cao, J.; and Chen, Y. 2024a. Keywords-enhanced Contrastive Learning Model for travel recommendation. *Information Processing & Management*, 61(6): 103874.

Chen, P.; Zhang, Y.; Cheng, Y.; Shu, Y.; Wang, Y.; Wen, Q.; Yang, B.; and Guo, C. 2024b. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Chen, Y.; Kang, Y.; Chen, Y.; and Wang, Z. 2020. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing*, 399: 491–501.

Farlie, D. 1964. Prediction and Regulation by Linear Least-Square Methods. *Journal of the Operational Research Society*, 15(4): 410–411.

Hellinger, E. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136): 210–271.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780.

Huang, C.; Bai, C.; Chan, S.; Zhang, J.; and Wu, Y. Q. 2023. MGTCF: multi-generator tropical cyclone forecasting with heterogeneous meteorological data. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, volume 37, 5096–5104.

Jia, Y.; Lin, Y.; Hao, X.; Lin, Y.; Guo, S.; and Wan, H. 2024. WITRAN: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting. *Advances in Neural Information Processing Systems*, 36: 12389–12456.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the Thirty-Seventh AAAI conference on artificial intelligence*, volume 37, 4365–4373.

Kan, K.; Aubet, F.-X.; Januschowski, T.; Park, Y.; Benidis, K.; Ruthotto, L.; and Gasthaus, J. 2022. Multivariate quantile function forecaster. In *International Conference on Artificial Intelligence and Statistics*, 10603–10621.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *The third International Conference on Learning Representations*.

Lim, B.; Arık, S. Ö.; Loeff, N.; and Pfister, T. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022a. SCINet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893.

Makridakis, S.; and Hibon, M. 1997. ARMA models and the Box–Jenkins methodology. *Journal of forecasting*, 16(3): 147–163.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.

Olivares, K. G.; Challu, C.; Marcjasz, G.; Weron, R.; and Dubrawski, A. 2023. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2): 884–900.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *The eighth International Conference on Learning Representations*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.;

Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32: 8024–8035.

Rasul, K.; Sheikh, A.-S.; Schuster, I.; Bergmann, U. M.; and Vollgraf, R. 2021. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *The ninth International Conference on Learning Representations*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191.

Sun, S. H.; and Yu, R. 2023. Copula Conformal prediction for multi-step time series prediction. In *The Twelfth International Conference on Learning Representations*.

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. WaveNet: A Generative Model for Raw Audio. In *The ninth ISCA Speech Synthesis Workshop*, 125.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. *Advances in Neural Information Processing Systems*, 30: 5998–6008.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting. In *The Eleventh International Conference on Learning Representations*.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Long, M.; and Wang, J. 2024a. Deep Time Series Models: A Comprehensive Survey and Benchmark. *arXiv preprint arXiv:2407.13278*.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Qiu, Y.; Zhang, H.; Wang, J.; and Long, M. 2024b. Timexer: Empowering transformers for time series forecasting with exogenous variables. *arXiv preprint arXiv:2402.19072*.

Watson, M. W. 1994. Vector autoregressions and cointegration. *Handbook of econometrics*, 4: 2843–2915.

Wen, R.; Torkkola, K.; Narayanaswamy, B.; and Madeka, D. 2017. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Yang, L.; Li, J.; Dong, R.; Zhang, Y.; and Smyth, B. 2022. NumHTML: Numeric-oriented hierarchical transformer model for multi-task financial forecasting. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, volume 36, 11604–11612.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In Williams, B.; Chen, Y.; and Neville, J., eds., *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, volume 37, 11121–11128.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *The Eleventh International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *International Conference on Machine Learning*, volume 162, 27268–27286.