HMformer: Unleashing Transformer's Potential for Time Series Forecasting via Hierarchical Multi-Scale Modeling

Renjun Huang[1,2], **Han Xiao**[1], **Bingqing Li**[1], **Baili Zhang**[1,3*], **Jianhua Lyu**[1]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Zhongguancun Academy, Beijing, China
[3]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Nanjing, China
hrj011128@seu.edu.cn, hxia0019@student.monash.edu, 220246410@seu.edu.cn
zhangbl@seu.edu.cn, lujianhua@seu.edu.cn

Abstract

Time series forecasting plays a critical role across a wide range of domains. Recently, an increasing number of Transformer-based forecasting models have emerged, achieving remarkably competitive performance. However, real-world time series data often exhibit complex multi-scale periodicities, which are not well-suited for modeling by the original Transformer architecture originally developed for NLP tasks. To address this limitation, we propose the Hierarchical Multi-scale Time Series Transformer (HMformer), employing a novel and sophisticated framework specifically designed for multi-scale time series forecasting. Specifically, HMformer incorporates a hierarchical cross-scale mixing mechanism that progressively aggregates temporal information from fine to coarse granularities, a scale-adaptive feature expansion design enhancing the extraction of high-level temporal semantics, and a multi-branch complementary prediction strategy for effectively integrating diverse temporal patterns. Collectively, these components enable HMformer to capture intricate, multi-scale temporal dynamics while retaining the Transformer's inherent strength in modeling long-range dependencies. Extensive experiments conducted on multiple real-world benchmark datasets—encompassing both long-term and short-term forecasting tasks—demonstrate that HMformer achieves state-of-the-art performance.

Code — https://github.com/dantian123121/HMformer

Introduction

Researchers and scholars from various fields have long been paying extensive attention to the breakthroughs and advancements in time series forecasting tasks (Wang et al. 2024b), owing to their critical role and broad application prospects in numerous domains such as weather prediction (Wu et al. 2023b), power dispatching (Muttaqi, Sutanto et al. 2021), energy management (Martín et al. 2010), and traffic flow control (Ma, Dai, and Zhou 2021). With the rapid development of deep learning methods over the past decade, a multitude of models have emerged to further enhance the accuracy of time series forecasting. However, in practical applications, the complex and non-stationary nature of time series data—characterized by deeply intertwined and latent temporal patterns such as trend-seasonality interactions (Wu et al. 2021) and multiple periodicities (Wu et al. 2023a)—poses significant challenges and difficulties for time series forecasting tasks.

In recent years, influenced and inspired by its remarkable success in the fields of NLP and CV (Vaswani et al. 2017; Devlin et al. 2019; Dosovitskiy et al. 2020), Transformer model has been progressively introduced into the domain of time series analysis and forecasting (Zhou et al. 2021; Nie et al. 2023; Zhang and Yan 2023; Xue et al. 2023). Unlike words in natural language processing, where each token carries rich semantic information, individual timesteps in time series data contain very limited meaningful patterns. As a result, using single-point observations as tokens in Transformer-based models has not yielded particularly impressive results. Notably, it has been demonstrated that even a simple linear model with minimal structural design can outperform previous Transformer-based approaches across multiple public time series forecasting benchmarks (Zeng et al. 2023). Fortunately, the issue has been successfully resolved by implementing a patch-based embedding method, where time series data is segmented into patches to improve the model's capacity for extracting local semantic features (Nie et al. 2023). However, this is far from sufficient, as real-world time series data often exhibit complex characteristics of multiple overlapping and mixed periodicities at different scales (Wang et al. 2024a). These characteristics cannot be directly captured by models that rely on a single scale (i.e., fixed patch size) for modeling. The data at any given time point in a time series are typically associated with multiple periodic patterns of varying lengths. For instance, traffic flow at a given moment is shaped by diurnal (day–night), weekly (weekday–weekend), and annual (off-peak–peak season) cycles. Therefore, a recent trend in time series forecasting tasks is the growing popularity of multiscale modeling approaches. The core idea of this methodology involves decomposing the input time series data into multiple branches at different resolution scales through downsampling operations such as pooling or convolution. Each branch, representing a specific scale, is then analyzed, modeled, and predicted independently. Finally, the predictions from all scales are integrated using a fusion strategy, such

*Corresponding author.

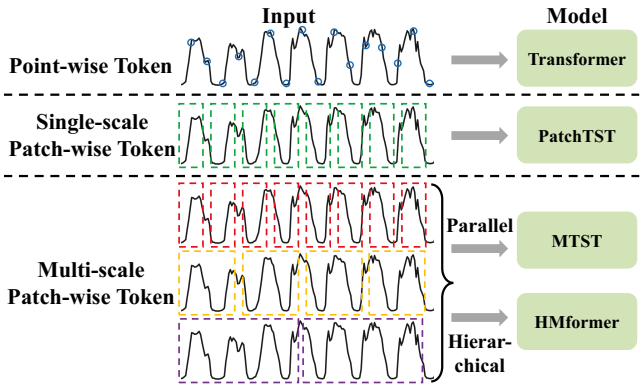

Figure 1: Comparison of input formats between our proposed HMformer and classical Transformer-based time series forecasting models.

as averaging. Coarse-scale time series branches, which have larger sampling intervals, are more suitable for observing macroscopic temporal modal information, such as the overall trend or average direction of the sequence. In contrast, fine-scale branches, with smaller sampling intervals, can capture more microscopic and high-frequency details, such as local fluctuations within the sequence. This complementary nature of multi-scale time series branches, when integrated, is conducive to achieving more accurate and robust forecasting performance. This multi-scale analysis approach has been explored in time series forecasting models with MLP- or CNN-based backbone structures, demonstrating notable progress. Examples include models such as TimeMixer (Wang et al. 2024a), AMD (Hu et al. 2025), MTST (Zhang et al. 2024), and MICN (Wang et al. 2023). However, since its original architecture was designed for NLP and lacks inherent mechanisms for multi-scale analysis, few Transformer-based models are capable of effectively modeling multiple mixed temporal patterns across different scales, despite attempts in MTST to incorporate parallel multi-branch architectures corresponding to different resolutions.

Based on the methods and challenges discussed above, in this paper, we attempt to adapt the model framework in an ingenious and novel manner while retaining the main structure of the Transformer, enabling it to hierarchically model multi-scale time series, as shown in Figure 1. Building on this, we propose HMformer for time series forecasting. Specifically, HMformer consists of multiple time series branches associated with different hierarchical levels, each of which is composed of several basic Transformer blocks. Each branch is responsible for processing its assigned time series data and predicting future values. The processed data from each branch is downsampled and subsequently fed into the next branch at a coarser scale to facilitate cross-scale information mixing. Finally, the complementary prediction results from each branch are aggregated to constitute the model output. Additionally, the deeper blocks in the HMformer, which correspond to coarser-scale time branches, are designed with larger feature dimensions to improve the extraction of advanced and comprehensive temporal features. Experiments demonstrate that HMformer achieves state-of-the-art performance on multiple publicly available benchmark datasets for real-world time series forecasting tasks. Our main contributions can be summarized as follows:

- Through an ingenious and novel framework adjustment, we incorporate a hierarchical multi-scale time series analysis approach into the model design without altering the main structure of the Transformer.
- We propose HMformer, which effectively models deeply mixed multi-scale temporal patterns, further unlocking the potential of the Transformer in time series forecasting.
- Extensive experiments on real-world datasets demonstrate that HMformer achieves state-of-the-art performance in time series forecasting tasks, surpassing other existing Transformer-based models.

Related Works

In the field of time series forecasting, methods based on neural network modeling have been widely adopted and significantly developed by researchers in recent years. Among the more popular models, they can be categorized into the following four classes based on their backbone architectures: RNN-based, CNN-based, MLP-based, and Transformer-based. RNN-based models (Lai et al. 2018; Salinas et al. 2020) capture state transitions in time series through recurrent modules; however, due to inherent structural issues such as susceptibility to gradient explosion or vanishing, as well as poor learning of long-range dependencies, their usage has become less frequent. CNN-based models (Liu et al. 2022a; Wu et al. 2023a) are capable of conveniently and effectively capturing local temporal features, which has led researchers to focus more on how to comprehensively analyze global temporal patterns. Meanwhile, MLP-based models (Oreshkin et al. 2019; Zeng et al. 2023), with their simple architecture and competitive forecasting performance, have also attracted attention and exploration from some scholars.

Given its outstanding performance in the fields of NLP and CV, Transformer-based models have been increasingly adopted for time series forecasting tasks in recent research (Ilbert et al. 2024). Initial attempts and explorations primarily focused on modifying or adapting the internal structure of the Transformer to fully leverage its inherent ability to model long-term dependencies, as exemplified by models such as Autoformer (Wu et al. 2021), Informer (Zhou et al. 2021), and Crossformer (Zhang and Yan 2023). Recently, PatchTST (Nie et al. 2023), iTransformer (Liu et al. 2024), and TimeXer (Wang et al. 2024c) have broken through the limitations of point-wise input—which struggles to effectively capture local semantic patterns in time series—by employing three distinct data formats as model inputs: patch-wise, series-wise, and a hybrid of patch-wise and series-wise representations. These advancements further unlock the potential of Transformer-based models in this field. Moreover, significant progress and breakthroughs have also been achieved in large-scale model training tailored for time series forecasting tasks (Zhou et al. 2023; Jin et al. 2024;

Chang et al. 2025).

Additionally, in the context of time series forecasting tasks, numerous innovative and more interpretable model architectures or data processing methods have emerged for decoding complex and chaotic temporal patterns. Autoformer (Wu et al. 2021) employs a Series Decomposition Block to progressively decompose time series into seasonal and trend components throughout the forecasting process, thereby facilitating the learning of more intricate temporal patterns. FEDformer (Zhou et al. 2022) enhances this module through iterative decomposition, while DLinear (Zeng et al. 2023) utilizes it to decompose raw data and model the components separately. Another approach leveraging Fourier transform-based multi-period decomposition is adopted by TimesNet (Wu et al. 2023a) and PDF (Dai et al. 2024) to capture both intra-cycle local variation characteristics and inter-cycle long-term dependencies in time series. Furthermore, the multi-scale analysis approach discussed in this work involves generating time series branches through iterative downsampling, followed by individual modeling and integration to achieve complementary multi-scale forecasting. While this method has been successfully implemented in MLP-based (e.g., TimeMixer (Wang et al. 2024a) and AMD (Hu et al. 2025)) and CNN-based (e.g., MICN (Wang et al. 2023)) temporal prediction models with demonstrated efficacy, its combination with Transformer-based frameworks like MTST (Zhang et al. 2024) has achieved limited success, primarily owing to the architectural constraints of the original Transformer. To unleash transformer's potential for time series forecasting, we propose HMformer, which incorporates an innovative hierarchical multi-scale branch design. This architecture effectively integrates progressively mixed multi-scale temporal patterns, thereby achieving superior forecasting performance.

HMformer

Problem Statement. The multivariate time series forecasting task can be formally described as follows: Given an input sequence of length T with multiple channels $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T\}$, the objective is to predict the future F time steps of data $\{\mathbf{X}_{T+1}, \mathbf{X}_{T+2}, \ldots, \mathbf{X}_{T+F}\}$. Here, $\mathbf{X}_i$ represents the C-dimensional vector at any time point i, where C denotes the number of variables (i.e., channels). In HMformer, we adopt the Channel-independence strategy from PatchTST (Nie et al. 2023), as it indirectly learns cross-variable correlations through weight sharing, demonstrating greater robustness to inter-channel noise and easier convergence during training. Since the data from each channel is processed and predicted independently, the aforementioned forecasting process can be formulated as:

$$\mathbf{x}^{(i)}_{T+1:T+F} = f\left(\mathbf{x}^{(i)}_{1:T}\right), \tag{1}$$

where $\mathbf{x}^{(i)}_{1:T} = \left\{x^{(i)}_1, x^{(i)}_2, \ldots, x^{(i)}_T\right\} \in \mathbb{R}^{T\times 1}$, $\mathbf{x}^{(i)}_{T+1:T+F} = \left\{x^{(i)}_{T+1}, x^{(i)}_{T+2}, \ldots, x^{(i)}_{T+F}\right\} \in \mathbb{R}^{F\times 1}$ denote the input and forecast sequences for the i-th variable, respectively. The function f represents the mapping that the neural network model aims to approximate as closely as possible.

Overall Framework. HMformer is a Transformer-based time series forecasting model with hierarchical multi-scale branches, and its specific architecture is shown in Figure 2. The input $\mathbf{x}^{(i)}_{1:T}$ is processed by K Patch Embedding operations with different patch sizes to generate multiple time series components at various scales. The temporal information input to each branch is further analyzed for local features and global dependencies by multiple Transformer blocks corresponding to the feature dimensions of that branch. Subsequently, the output from the final block of each branch is not only passed through a predictor to forecast the future sequence but also downsampled and used as part of the input for the next branch, enabling the hierarchical extraction of high-level temporal semantics through cross-scale fusion. Finally, complementary prediction information from multiple branches is aggregated to produce the final model output.For simplicity, we will use $\mathbf{x} = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T\times 1}$ to denote the input sequence, in place of the previous notation $\mathbf{x}^{(i)}_{1:T}$.

Patch Embedding. Corresponding to the K temporal branches, there are K Patch Embedding layers. In the k-th Patch Embedding (where $k \in \{1, \ldots, K\}$, the input $\mathbf{x}$ is repetitively padded at the end with s_k repetitions of x_T, and then divided into multiple equal-length vectors $\mathbf{x}_k \in \mathbb{R}^{n_k\times p_k}$ through an overlapping sliding window with a stride of s_k and a patch size of p_k. Here, s_k, p_k, and n_k are equal to $S \times 2^{k-1}$, $P \times 2^{k-1}$, and $\frac{N}{2^{(k-1)}}$, respectively, where S, P, and N represent the stride, patch size, and number of patches in the first branch, with $P = 2S$. Consequently, using the formula $N = \left\lfloor \frac{T-P}{S} \right\rfloor + 2$, it can be observed that the number of patches in each branch decreases by half successively. Thus, we obtain K time series at different scales: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$. Similar to the classical Transformer model, $\mathbf{x}_k$ is mapped to a latent space of dimension d_k corresponding to the feature dimensions of the blocks in its branch via a 1D convolutional layer. Absolute positional encoding is appended to compensate for the inherent limitation of the attention mechanism in perceiving sequential information, thereby forming the input $\mathbf{z}_k \in \mathbb{R}^{n_k\times d_k}$ to the blocks of this branch. Here, d_k equals $D \times 2^{k-1}$, where D represents the feature dimension size in the first branch.

Transformer Block. Each branch comprises M Transformer blocks designed to analyze and compute time series data at their respective scales. Analogous to convolutional neural networks, where the channel dimension of convolutional layers typically increases as the network depth grows, in HMformer, the local receptive field of a single patch in deeper temporal branches becomes larger, while the number of patches decreases. Consequently, we expand the latent feature dimension d_k of the Transformer blocks in deeper branches—those that incorporate cross-scale temporal information aggregated from earlier stages—referred to as Scale-Adaptive Feature Expansion (SAFE), aiming to strengthen the model's ability to capture sophisticated, highly overlapping multi-scale temporal patterns. Apart from the latent feature dimension d_k varying across branches, all other hyperparameters of the Transformer blocks are identically con-

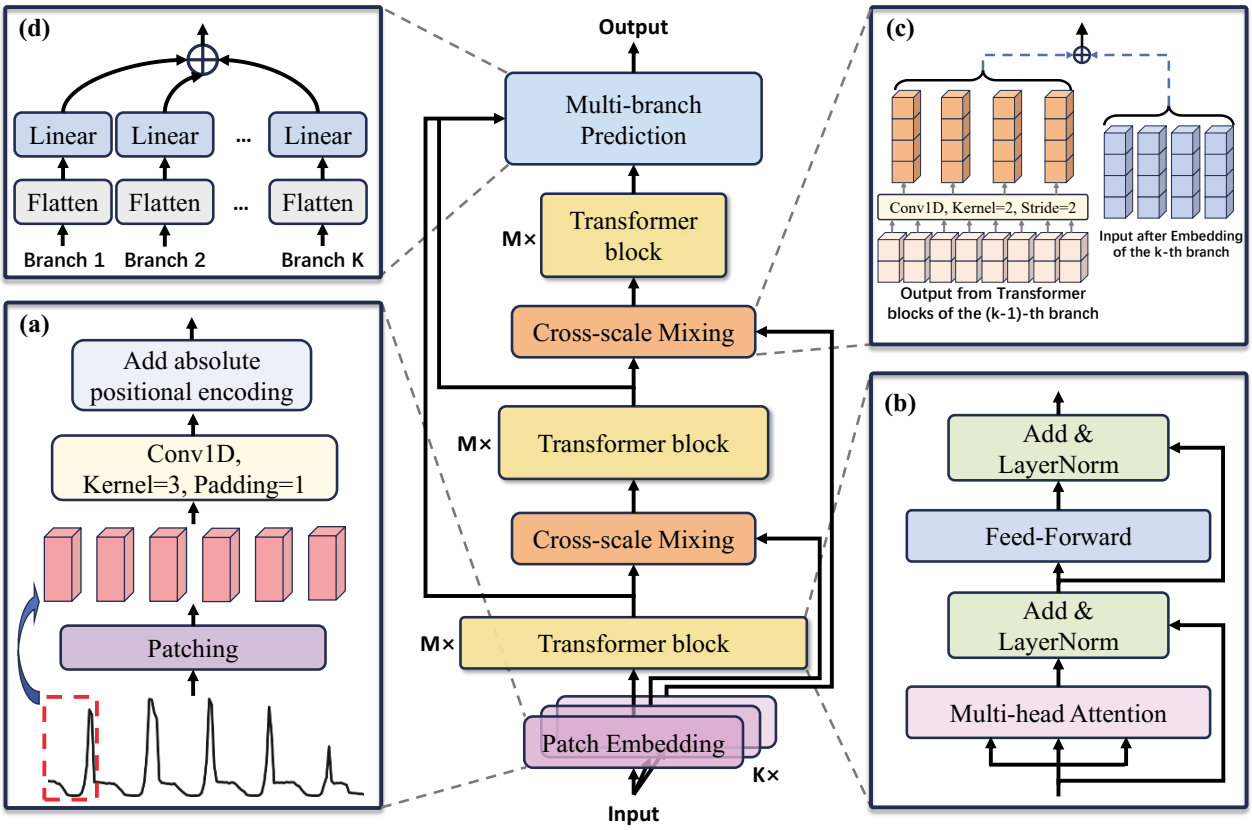

Figure 2: Overall framework of the proposed HMformer with hierarchical temporal branches across different scales. (a) The input sequence is embedded into multiple temporal branches at varying scales. (b) The block structure follows the conventional Transformer architecture. (c) The output from blocks of the (k-1)-th branch undergoes cross-scale mixing with the embedded input of the k-th branch. (d) Final predictions are obtained by summing the multi-branch forecasts from all K branches.

figured within each branch. Specifically, each Transformer block consists of a multi-head self-attention mechanism with residual connections and Layer Normalization, alongside a feed-forward neural network with residual connections and Layer Normalization. The computation process of the adopted scaled dot-product attention (Vaswani et al. 2017) is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2}$$

It's noteworthy that to further enhance HMformer's focus on temporal positional information, rotary position embedding (Su et al. 2024) is incorporated during the attention mechanism computation. The k-th branch's M Transformer blocks are computed as:

$$\text{for } m: 1 \to M \text{ do: } \mathbf{z}_k^m = \text{TransformerBlock}_{m-1}(\mathbf{z}_k^{m-1}), \tag{3}$$

where $\mathbf{z}_k^i \in \mathbb{R}^{n_k \times d_k}$ denotes the input of the i-th block in the k-th branch, and $\text{Transformerblock}_i(\cdot)$ represents the computational operation of the i-th block.

Cross-scale Mixing. HMformer incorporates a hierarchical cross-scale mixing strategy to integrate multi-scale temporal patterns in a fine-to-coarse manner. This facilitates the modeling of complex, dynamic, and multi-period overlapping real-world time series. From a technical perspective, the output $\mathbf{z}_k^M$ of the final block in the k-th branch (where $k \in \{1, \ldots, K-1\}$) is downsampled using a 1D convolutional layer with a stride of 2, a kernel size of 2, and an output dimension that doubles. The resulting downsampled sequence is then fused with the time series $\mathbf{z}_{k+1}$ corresponding to the next scale, serving as the input $\mathbf{z}_{k+1}^0$ for the subsequent temporal branch's blocks. Combining the computational aspects of Transformer blocks within each branch, the process of calculating the latent representations of multi-scale time series $\{\mathbf{z}_1^M, \mathbf{z}_2^M, ..., \mathbf{z}_K^M\}$ after cross-scale mixing analysis can be summarized by the following formula:

$$\begin{aligned} &\mathbf{z}_1^0 = \mathbf{z}_1, \\ &\text{for } k: 1 \to K-1 \text{ do: } \left\{ \begin{array}{l} \mathbf{z}_k^M = \text{Blocks}_k(\mathbf{z}_k^0), \\ \mathbf{z}_{k+1}^0 = \mathbf{z}_{k+1} + \text{Conv1d}_k(\mathbf{z}_k^M) \end{array} \right\}, \\ &\mathbf{z}_K^M = \text{TransformerBlocks}_K(\mathbf{z}_K^0), \end{aligned} \tag{4}$$

where $\text{Blocks}_k(\cdot)$ denotes the computational process of all Transformer blocks in the k-th branch, $\text{Conv1d}_k(\cdot)$ represents the 1D convolutional layer between the k-th and

Models	**HMformer (Ours)**		AMD (2025)		TimeMixer (2024a)		SAMformer (2024)		MTST (2024)		TimeXer (2024c)		iTransformer (2024)		RLinear (2023)		MICN (2023)		PatchTST (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.374**	**0.391**	0.380	0.395	0.381	0.395	0.404	0.401	0.377	0.395	0.375	**0.391**	0.407	0.410	0.414	0.407	0.423	0.422	0.387	0.400
ETTm2	0.277	**0.322**	0.282	0.326	**0.275**	0.323	0.293	0.335	0.282	0.325	0.278	0.323	0.288	0.332	0.286	0.327	0.353	0.402	0.281	0.326
ETTh1	**0.422**	**0.430**	0.444	0.431	0.447	0.440	0.452	0.438	0.438	0.439	0.431	0.432	0.454	0.447	0.446	0.434	0.475	0.480	0.469	0.454
ETTh2	**0.363**	**0.395**	0.378	0.400	0.364	**0.395**	0.383	0.404	0.369	0.397	0.378	0.403	0.383	0.407	0.374	0.398	0.574	0.531	0.387	0.407
ECL	0.183	**0.269**	0.192	0.279	0.182	0.272	0.187	0.273	0.188	0.274	0.180	0.274	**0.178**	0.270	0.219	0.298	0.196	0.309	0.205	0.290
Traffic	**0.420**	**0.266**	0.474	0.290	0.484	0.297	0.462	0.289	0.451	0.281	0.447	0.295	0.428	0.282	0.626	0.378	0.593	0.356	0.481	0.304
Weather	0.252	0.276	0.256	0.280	**0.240**	**0.271**	0.261	0.281	0.255	0.277	0.254	0.276	0.258	0.278	0.272	0.291	0.268	0.321	0.259	0.281
Solar	0.229	**0.260**	0.248	0.289	**0.216**	0.280	0.254	0.281	0.232	0.262	0.238	0.269	0.233	0.262	0.369	0.356	0.283	0.358	0.270	0.307
Avg	**0.315**	**0.326**	0.332	0.336	0.324	0.334	0.337	0.338	0.324	0.331	0.323	0.333	0.329	0.336	0.376	0.361	0.396	0.397	0.342	0.346

Table 1: Experimental results on eight long-term forecasting datasets. All input lengths are fixed to 96, and all results represent the average prediction errors across four different prediction lengths {96, 192, 336, 720}. **Bold**: best, Underlined: second best.

$(k+1)$-th branches used for downsampling.

Multi-Branch Complementary Prediction. During the prediction phase, we provide K independent predictors for the outputs $\{\mathbf{z}_1^M, \mathbf{z}_2^M, ..., \mathbf{z}_K^M\}$ of the K branches after cross-scale mixing analysis. These predictors leverage hierarchical multi-scale historical information to forecast future sequences. Each predictor consists of a flattening layer followed by a linear layer with weights of dimension $F \times ND$. The prediction result from the k-th branch is denoted as $\hat{\mathbf{y}}_k \in \mathbb{R}^{F\times 1}$. Finally, the results from the complementary K branches are summed up to obtain the final model output $\hat{\mathbf{y}} \in \mathbb{R}^{F\times 1}$. This prediction process can be summarized by the following formula:

$$\hat{\mathbf{y}}_k = \text{Predictor}_k(\mathbf{z}_k^M), \quad k \in \{1, \ldots, K\}, \quad \hat{\mathbf{y}} = \sum_{k=1}^{K} \hat{\mathbf{y}}_k, \tag{5}$$

where $\text{Predictor}_k(\cdot)$ denotes the predictior for the k-th branch.

Experiments

To comprehensively and rigorously evaluate the effectiveness and full potential of the proposed HMformer for time series forecasting tasks, we conducted extensive experiments on nine publicly available real-world datasets, benchmarking against 21 well-established baseline models.

Datasets. The experimental evaluation utilizes eight benchmark datasets for long-term forecasting: Traffic, Weather, Solar-Energy, Electricity, and ETT (including its four subsets ETTh1, ETTh2, ETTm1, and ETTm2) (Lai et al. 2018; Wu et al. 2021; Zhou et al. 2021), along with the M4 dataset commonly employed for short-term forecasting. These datasets cover multiple domains closely related to time series, such as transportation, power, and weather, and have been extensively utilized in the evaluation of numerous well-known neural network-based forecasting models.

Baselines. We select 21 state-of-the-art models from the time series forecasting domain in recent years as baselines for comparison with HMformer, including Linear-based models: LightTS (2022), DLinear (2023), RLinear (2023), TiDE (2023), N-HiTS (2023), TimeMixer (2024a), and AMD (2025); CNN-based models: SCINet (2022a), TimesNet (2023a), and MICN (2023); and Transformer-based models: Autoformer (2021), Informer (2021), FEDformer (2022), Stationary (2022b), Crossformer (2023), PatchTST (2023), TIME-LLM (2024), iTransformer (2024), TimeXer (2024c), SAMformer (2024), and MTST (2024).

Implementation Details. We follow the experimental settings of Timesnet (Wu et al. 2023a) to conduct a thorough evaluation of HMformer's performance and refer to the experimental results of the baseline models presented in the article. The final prediction results are obtained by averaging the outcomes of three repeated experiments to ensure the reliability of the evaluation data. All experimental networks are implemented in PyTorch.

Main Results

Long-Term Forecasting. The results of the multivariate long-term time series forecasting task are presented in the table. Lower values of MAE and MSE indicate better predictive performance of the model. As can be observed from the Table 1, our proposed HMformer achieves state-of-the-art performance on nearly all benchmarks compared to other models, spanning high- and low-dimensional datasets such as Traffic and ETT. Even recently leading models such as iTransformer and TimeXer can only match HMformer's performance on a subset of datasets, demonstrating that Transformer-based models trained using channel-independent methods can still achieve top-tier results in long-term forecasting tasks. More importantly, when compared to MICN, TimeMixer, AMD, and MTST, all of which utilize similar multi-scale analysis methods, HMformer still demonstrates a competitive advantage. These experimental results convincingly highlight the immense potential of

Models	**HMformer (Ours)**	iTransformer (2024)	TIME-LLM (2024)	TimesNet (2023a)	N-HiTS (2023)	SCINet (2022a)	PatchTST (2023)	MICN (2023)	LightTS (2022)	DLinear (2023)	FEDformer (2022)	Stationary (2022b)	Autoformer (2021)	Informer (2021)
SMAPE	**11.775**	12.684	11.983	<u>11.829</u>	11.927	15.542	13.152	19.638	13.525	13.639	12.840	12.780	12.909	14.086
MASE	**1.581**	1.764	1.595	<u>1.585</u>	1.613	2.816	1.945	5.947	2.111	2.095	1.701	1.756	1.771	2.718
OWA	**0.848**	0.929	0.859	<u>0.851</u>	0.861	1.309	0.998	2.279	1.051	1.051	0.918	0.930	0.939	1.230

Table 2: Experimental results on short-term forecasting using the M4 dataset. The prediction lengths range from 6 to 48, and all results are weighted averages of prediction errors across multiple sub-datasets with different sampling frequencies. **Bold**: best, <u>Underlined</u>: second best.

Datasets	Traffic							ETTh1			ETTh2		
Branch1	✓	✓	✓	×	✓	×	×	✓	✓	×	✓	✓	×
Branch2	✓	✓	×	✓	×	✓	×	✓	×	✓	✓	×	✓
Branch3	✓	×	✓	✓	×	×	✓	\	\	\	\	\	\
MSE	**0.420**	0.426	0.423	0.423	0.442	0.425	0.423	**0.422**	0.429	0.424	**0.363**	0.373	0.369
MAE	**0.266**	0.274	0.269	0.267	0.281	0.272	0.268	**0.430**	0.435	0.432	**0.395**	0.400	0.399

Table 3: Ablation results of the multi-branch complementary prediction framework. All results represent the average prediction errors across four forecast lengths {96, 192, 336, 720}. Symbols ✓and × indicate whether each branch was retained or ablated in the final prediction, while '\' denotes the absence of that branch option for the current dataset. **Bold**: best.

Transformer-based models in long-term forecasting and suggest that this potential can be further unlocked by integrating hierarchical multi-scale analysis methods.

Short-Term Forecasting. As demonstrated in Table 2, our proposed HMformer maintains superior performance over all baseline models including TimesNet on the univariate short-term forecasting benchmark M4, further validating its effectiveness and generalizability.

Model Analysis

Ablation Studies. To further substantiate that the adoption of a hierarchical multi-scale analysis method is the key factor behind HMformer's outstanding performance in forecasting tasks and to validate the rationality of our proposed multi-scale temporal branching architecture, comprehensive and rigorous ablations were conducted from three distinct perspectives: whether each branch's output contributes to future predictions, whether cross-scale information fusion is necessary, and whether the SAFE design is effective. It should be noted that HMformer's predictive performance varies across datasets when the number of branches K changes. Therefore, to ensure the rigor and reliability of the experimental findings, we selected three representative datasets for the ablation studies: Traffic (which achieves better performance at $K = 3$) and ETTh1/ETTh2 (which exhibit superior results at $K = 2$). Furthermore, compared to other models, the fundamental architecture of the Attention mechanism in Transformers possesses an inherent advantage in effectively capturing long-range dependencies. To systematically investigate the necessity of the Attention mechanism in HMformer architecture, we performed component ablation through replacement with alternative operators (MLP/ convolutional layers).

Multi-Branch Complementary Prediction. We systematically evaluated the contribution of each temporal branch by configuring the HMformer's final output to retain or discard predictions from individual branches, resulting in $2^K - 1$ possible model variants. Each configuration was trained and evaluated under identical hyperparameters, and the results are summarized in Table 3. It is demonstrated that the model incorporating predictions from all branches achieves optimal performance, indicating that each temporal branch provides indispensable information for the final forecast. Furthermore, variants that retain predictions from deeper branches generally achieve superior accuracy. This observation aligns with the design philosophy of HMformer, in which deeper branches not only possess a larger local receptive field, but also inherently integrate cross-scale temporal information propagated from earlier shallow branches, thereby more effectively capturing complex and intertwined multi-scale temporal patterns.

Cross-Scale Mixing. Given the inherently complex nature of real-world time series, where multiple periodic patterns often overlap and interact, HMformer incorporates the cross-scale mixing module to effectively model these composite temporal patterns across multiple scales. To validate this design, we conducted ablation studies by training an HMformer variant without the mixing module and evaluating its prediction error. As demonstrated in the Table 4, the removal of cross-scale mixing consistently leads to significant deterioration in prediction accuracy across all experimental settings. These results empirically confirm that our proposed mixing module is both theoretically justified and practically effective in enhancing model performance. Fundamentally, the variant model without the cross-scale mixing module shares a structural similarity with MTST (Zhang et al. 2024), as both adopt a parallel, scale-wise independent modeling strategy for time series, neglecting the inter-scale dependencies and intertwined temporal patterns across different scales. This architectural limitation partially explains why HMformer outperforms MTST on the majority of datasets.

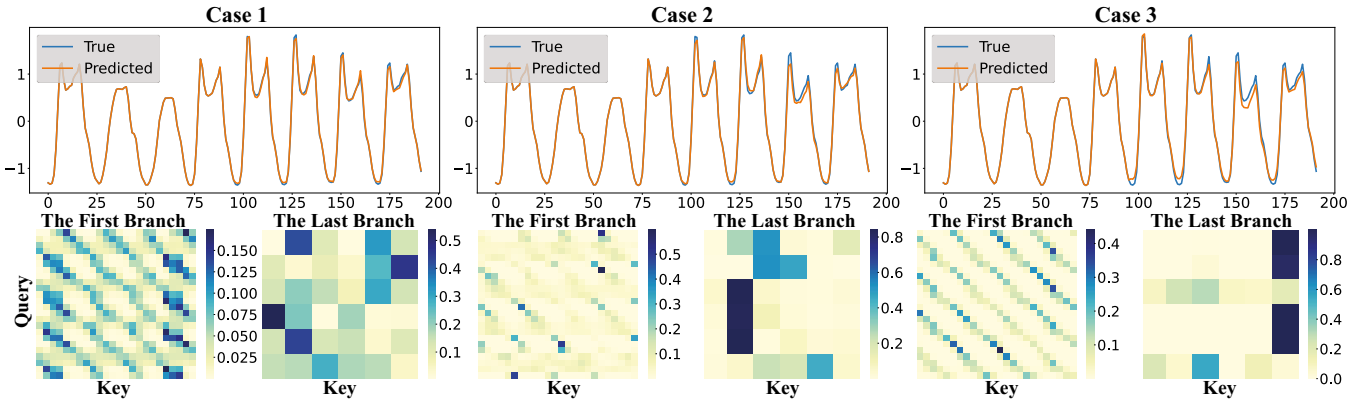

Figure 3: Visualization of prediction results and attention maps from the first and last branches under three different configurations. Case 1: Standard HMformer. Case 2: Without SAFE (with fixed d_k). Case 3: Without the cross-scale mixing module.

Dataset	Traffic		ETTh1		ETTh2	
Cross-scale Mixing	✓	×	✓	×	✓	×
MSE	**0.420**	0.429	**0.422**	0.428	**0.363**	0.370
MAE	**0.266**	0.271	**0.430**	0.433	**0.395**	0.400

Table 4: Ablation results for cross-scale mixing. All reported values represent mean prediction errors averaged across four forecast lengths {96, 192, 336, 720}. **Bold**: best.

Scale-Adaptive Feature Expansion. In HMformer, individual tokens in deeper temporal branches inherently encompass more extensive multi-scale time-series information, which motivated our proposed SAFE design. This architectural principle aligns conceptually with the increasing channel dimensionality in deeper convolutional layers of CNNs, which enhances high-level feature extraction. To validate the rationale behind SAFE, we conducted controlled experiments by enforcing uniform feature dimensions across all transformer blocks in HMformer. As Table 5 demonstrates, ablation studies consistently show that maintaining identical feature dimensions—regardless of their absolute size—fails to achieve the superior predictive performance of HMformer with varying feature dimensions.

Analysis of Attention Patterns Across Model Variants. To provide an intuitive illustration of the hierarchical multi-scale modeling in HMformer and enhance model interpretability, we present visualizations of prediction results and attention maps from the first and last branches for the same test sample under three configurations: (1) standard HMformer, (2) without SAFE (with fixed d_k), (3) without the cross-scale mixing module, as shown in Figure 3. It can be observed that the standard HMformer achieves the most accurate prediction, while the other two variants exhibit noticeable prediction errors. From the attention maps, in Cases 1 and 3, the first branch—characterized by smaller feature dimensions and local receptive fields (corresponding to fine-scale input)—effectively captures basic, single-scale periodic patterns in the time series. In contrast, in Case 2, the excessively large feature dimension introduces more parameters, which hinders the model's ability to learn these simple periodicities, suggesting overfitting to local time series. Moreover, In the last branch of Case 1, the attention scores exhibit higher dispersion, whereas those in Case 3 remain highly concentrated. Given their prediction results, the enhanced token-to-token interaction and stronger dependency modeling in Case 1's last branch promote the learning of advanced, deeply overlapping multi-scale temporal patterns, leading to more accurate future predictions. In contrast, although the last branch of Case 2 also incorporates cross-scale temporal feature integration from shallow layers, its inadequate feature dimensions for capturing high-level representations result in inferior predictive performance.

Datasets	Traffic				ETTh1				ETTh2			
Metric	SAFE	d_k=64	d_k=128	d_k=256	SAFE	d_k=32	d_k=64		SAFE	d_k=32	d_k=64	
MSE	**0.420**	0.428	0.425	0.431	**0.422**	0.427	0.433		**0.363**	0.371	0.375	
MAE	**0.266**	0.273	0.269	0.272	**0.430**	0.434	0.441		**0.395**	0.400	0.402	

Table 5: Ablation results of the SAFE design. All results represent the average prediction errors across four forecast lengths {96, 192, 336, 720}. In the baseline configuration without SAFE, all blocks across branches maintain uniform feature dimensions. **Bold**: best.

Conclusion

In the field of time series forecasting, the muti-scale method is not inherently well-suited to the classical Transformer architecture. Through a hierarchical multi-scale structural design, the proposed HMformer not only preserves the Transformer's inherent strength in capturing long-range dependencies but also models complex multi-scale temporal patterns, thereby achieving multi-branch complementary predictions for future sequences. Extensive experimental results demonstrate that HMformer achieves advanced performance across various time series forecasting tasks, further unlocking the immense potential of the Transformer in this domain.

Acknowledgments

This work was partly supported by the National Key R&D Program of China (Grant No. 2023YFC3806004), the National Natural Science Foundation of China (Grant No. 62373104), the Hainan Province Science and Technology Special Fund (Grant No. ZDYF2023GXJS150), the Fundamental Research Funds for the Central Universities (Grant No. 2242024k30035), and Zhongguancun Academy, Beijing (Grant No. C20250403).

References

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Ramirez, F. G.; Canseco, M. M.; and Dubrawski, A. 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 6989–6997.

Chang, C.; Wang, W.-Y.; Peng, W.-C.; and Chen, T.-F. 2025. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3): 1–20.

Dai, T.; Wu, B.; Liu, P.; Li, N.; Bao, J.; Jiang, Y.; and Xia, S.-T. 2024. Periodicity decoupling framework for long-term series forecasting. In *The Twelfth International Conference on Learning Representations*.

Das, A.; Kong, W.; Leach, A.; Mathur, S.; Sen, R.; and Yu, R. 2023. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words. *arXiv preprint arXiv:2010.11929*, 7.

Hu, Y.; Liu, P.; Zhu, P.; Cheng, D.; and Dai, T. 2025. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17359–17367.

Ilbert, R.; Odonnat, A.; Feofanov, V.; Virmaux, A.; Paolo, G.; Palpanas, T.; and Redko, I. 2024. Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. *arXiv preprint arXiv:2402.10198*.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; et al. 2024. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Li, Z.; Qi, S.; Li, Y.; and Xu, Z. 2023. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*.

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. *arXiv preprint arXiv:2310.06625*.

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.

Ma, C.; Dai, G.; and Zhou, J. 2021. Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 5615–5624.

Martín, L.; Zarzalejo, L. F.; Polo, J.; Navarro, A.; Marchante, R.; and Cony, M. 2010. Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. *Solar Energy*, 84(10): 1772–1781.

Muttaqi, K. M.; Sutanto, D.; et al. 2021. Adaptive and predictive energy management strategy for real-time optimal power dispatch from VPPs integrated with renewable energy and energy storage. *IEEE Transactions on Industry Applications*, 57(3): 1958–1972.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

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.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and Zhou, J. 2024a. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Long, M.; and Wang, J. 2024b. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*.

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

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023a. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

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.

Wu, H.; Zhou, H.; Long, M.; and Wang, J. 2023b. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 5(6): 602–611.

Xue, W.; Zhou, T.; Wen, Q.; Gao, J.; Ding, B.; and Jin, R. 2023. Card: Channel aligned robust blend transformer for time series forecasting. *arXiv preprint arXiv:2305.12095*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; and Li, J. 2022. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*.

Zhang, Y.; Ma, L.; Pal, S.; Zhang, Y.; and Coates, M. 2024. Multi-resolution time-series transformer for long-term forecasting. In *International conference on artificial intelligence and statistics*, 4222–4230. PMLR.

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 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*, 27268–27286. PMLR.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.