TimeFound: A Foundation Model for Time Series Forecasting

Congxi Xiao^{1,2,†}, Jingbo Zhou^{1,*}, Yixiong Xiao¹, Xinjiang Lu¹, Le Zhang¹, Hui Xiong^{3,*}

¹ Business Intelligence Lab, Baidu Research

² University of Science and Technology of China

³ The Hong Kong University of Science and Technology (Guangzhou)

xiaocongxi@mail.ustc.edu.cn,

{zhoujingbo, xiaoyixiong, luxinjiang, zhangle09}@baidu.com,

xionghui@ust.hk

Abstract

We present TimeFound, an encoder-decoder transformer-based time series foundation model for out-of-the-box zero-shot forecasting. To handle time series data from various domains, TimeFound employs a multi-resolution patching strategy to capture complex temporal patterns at multiple scales. We pre-train our model with two sizes (200M and 710M parameters) on a large time-series corpus comprising both real-world and synthetic datasets. Over a collection of unseen datasets across diverse domains and forecasting horizons, our empirical evaluations suggest that TimeFound can achieve superior or competitive zero-shot forecasting performance, compared to state-of-the-art time series foundation models.

1 Introduction

Time series forecasting [Hyndman and Athanasopoulos, 2018] plays a crucial role in industrial applications and scientific research in various domain [Chen et al., 2012, Zhou and Tung, 2015, Deb et al., 2017, Karmy and Maldonado, 2019, Kaushik et al., 2020, Zhu et al., 2023, Ji et al., 2023, Zhou et al., 2024], such as energy, retail, finance, manufacturing, and healthcare. In recent years, data-driven deep learning models have demonstrated remarkable success in time series forecasting [Salinas et al., 2020, Sen et al., 2019, Zhou et al., 2021, Zeng et al., 2023, Nie et al., 2023, Chen et al., 2021], surpassing traditional statistical models like ARIMA. Despite their impressive effectiveness, a major limitation of deep forecasters is the heavy reliance on substantial task-specific training data. This restricts their ability to generalize to diverse forecasting scenarios, especially on those where data is scarce and insufficient to support additional training, i.e., necessitating zero-shot forecasting.

Witnessing the recent advance of language foundation models (i.e. Large Language Models, LLMs), researchers have been inspired to develop time series foundation models that are generalizable to a broad range of forecasting scenarios [Liang et al., 2024]. Following the paradigm of building LLMs, recent studies (e.g., [Das et al., 2024, Ansari et al., 2024, Liu et al., 2024b, Shi et al., 2024, Garza et al., 2023, Woo et al., 2024]) collect a large scale of heterogeneous time series data from multiple domains to pre-train the time series foundation model in a self-supervised manner. After learning to capture common temporal patterns from extensive and diverse range of time series data, these models achieve superior forecasting performance and show promising generalization capabilities on unseen data without any training. Though still in its early stages, the development of time series foundation models marks a paradigm shift in forecasting, moving toward a more adaptable solution for building a universal forecaster across diverse data distributions.

^{*}Jingbo Zhou and Hui Xiong are corresponding authors. †This work was done when the first author was an intern at Baidu Research under the supervision of Jingbo Zhou

In this work, we continue exploring the development of effective foundation models for time series forecasting, and propose TimeFound, a transformer-based time series foundation model. In terms of the architecture, we employ an encoder-decoder design for time series modeling and forecasting, where the encoder enables contextual understanding of historical trends while the decoder maintains the causal future prediction. To tokenize time series data, we adopt a multi-resolution patching method that performs multiple divisions with different patch sizes, rather than fix-size patching. This design is driven by the need for a foundation model to deal with the time series across various domains with distinct dynamics and frequencies. In addition, even the same time series can exhibit diverse variations and fluctuations at different temporal scales [Ding et al., 2024, Chen et al., 2024]. Our approach facilitates the capture of temporal patterns at multiple scales, enhancing the model's ability to handle diverse forecasting scenarios.

We pre-train TimeFound in two size (*TimeFound-Base-*200M and *TimeFound-Large-*710M) using datasets opened by [Ansari et al., 2024]. The training objective is auto-regressive next patch prediction with teacher forcing, based on the historical contexts. We conduct empirical evaluations of TimeFound's zero-shot forecasting performance on 24 datasets. The experiment results demonstrate that our model achieves competitive or superior performance compared to state-of-the-art time series foundation models.

2 Related Work

Time Series Forecasting In the last decade, deep learning models have emerged as powerful tools in time series forecasting. Extensive studies have explored various architectures for building effective deep forecasting models, such as: Recurrent Neural Networks (RNNs) based models like DeepState [Rangapuram et al., 2018], DeepAR [Salinas et al., 2020], ESRNN [Smyl, 2020], and Convolutional Neural Networks (CNNs) based models like TCN [Bai et al., 2018], TimesNet [Wu et al., 2022]. As Transformers [Vaswani et al., 2017] exhibited powerful sequence modeling capability and promising scalability, it has become the most popular architecture to build time series forecasting models [Zhou et al., 2021, Chen et al., 2021, Zhou et al., 2022, Nie et al., 2023, Chen et al., 2024, Ding et al., 2024, Liu et al., 2024a, Zhang and Yan, 2023]. Some recent studies also developed linear forecasters achieving impressive performance, such as N-BEATS [Oreshkin et al., 2020], DLinear [Zeng et al., 2023] and TiDE [Das et al., 2023]. While these models achieve remarkable performance, they are trained independently for each application domain and fall short in generalizability to handle cross-domain data in a wide range forecasting scenarios.

Time Series Foundation Models There have been some research efforts focusing on building time series foundation models. As LLMs show strong generalizability, several works adopt LLMs for time-series forecasting. For instance, FPT [Zhou et al., 2023] fine-tunes the pre-trained GPT-2 model on different time-series related tasks. LLMTime [Gruver et al., 2023] proposes a tokenization method that encodes numerical time series as string. Time-LLM [Jin et al., 2023] aligns time series embedding to the text space via patch reprogramming and prompts LLM with aligned inputs to make future predictions. Another line of studies concentrate on pre-training the general foundation model from scratch on a large scale of time series data. For example, ForecastFPN [Dooley et al., 2023] is a pre-trained model purely on synthetic time series and used for zero-shot forecasting. There are also many works pre-training foundation models using real-world time series data, such as Timer [Liu et al., 2024b], MOIRAI [Woo et al., 2024], Moment [Goswami et al., 2024], and Lag-Llama [Rasul et al., 2023], or combining real-world and synthetic data together for general pre-training like TimesFM [Das et al., 2024], Chronos [Ansari et al., 2024] and TIME-MOE [Shi et al., 2024]. For example, TimesFM [Das et al., 2024] collects a massive amount of times series data from Google Trends and Wiki pageviews for pre-training. Another foundation model, TimeGPT-1 [Garza et al., 2023] is close-sourced and releases the commercial API for zero-shot forecasting.

3 Problem Definition

Our goal is to build a foundation model as zero-shot time series forecaster, which can use the historical time series to predict the future value across various domains. Formally, given the past points of a time series (also known the context) $\mathbf{x}_{1:C} = \{x_1, x_2, ..., x_C\}$, where C is the context length, such a foundation model f is expected to predict the future H time points: $f: (\mathbf{x}_{1:C}) \longrightarrow \mathbf{x}_{C+1:C+H}$.

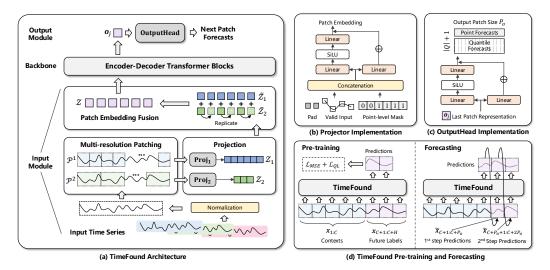


Figure 1: Illustration of TimeFound model. In (a), for simple illustration, we assume K=2 in multi-resolution patching method and it divides normalized time series using twp patch sizes P_i and P_i . (b) and (c) show the detailed implementation of the projector in Input Module and the prediction head in Output Module respectively. (d) presents the model's different behaviors during pre-training and forecasting.

In this work, we focus on *univariate* forecasting, where x_i is a scalar. For *multivariate* time series data, the univariate model can still be applied by performing channel-independent forecasting for each individual variate of the time series.

4 Method

In this section, we will introduce TimeFound, a transformer-based foundation model for time series modeling and forecasting. As illustrated in Figure 1, our model begins with an Input Module that pre-processes the raw time-series from different domains, where we propose a multi-resolution patching method capable in capturing the temporal dependencies at multiple scales. Next, we employ an encoder-decoder architecture, which enables both contextual understanding of historical trends and auto-regressive forecasting. Finally, the Output Module generates predictions of future patches. Below, we provide a detailed explanation of each deigns.

4.1 Input Module

Normalization Since our foundation model will be pre-trained on extensive time series data with varying amplitude, frequency and stationarity, the initial step is normalizing the input data to facilitate better optimization. We apply the commonly used standard scaling method that normalize based on the mean and standard deviation calculated over the entire input series. This mitigates the bias caused by different scales across multiple datasets, while reserving the patterns of the original series.

Multi-resolution Patching The next step is to divide the time series into patches, which is analogue to the tokenization in nature language processing. Patch-based modeling has been proved to be effective in capturing the semantic information of time series data Nie et al. [2023] and widely adopted in recent foundation models Das et al. [2024]. Different from the vanilla patching method with fixed patch size, we propose to perform multi-resolution patch division with different patch sizes. This design enables the model to handle time series data from different domains with distinct patterns and variations at different temporal scales, which is beneficial for building a generalizable foundation model.

Specifically, we define a collection of patch sizes $\{P_1, P_2, ..., P_K\}$, where each patch size corresponds to a division. In our framework, we restrict the value of P_i to be a power of 2 and assume that $P_1 < P_2 ... < P_K$. Given the input sequence $\boldsymbol{x}_{1:C}$, the k-th patch division with patch size P_k will break it

down into a series of N_k patches $\mathcal{P}^k = \{\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, ..., \boldsymbol{p}_{N_k}^k\}$, where patch $\boldsymbol{p}_j^k = \boldsymbol{x}_{(j-1)P_k+1:jP_k}$. Thus, with the multi-resolution patching method, we will obtain K groups of patches: $\{\mathcal{P}^1, \mathcal{P}^2, ..., \mathcal{P}^K\}$.

Projection Following previous works [Das et al., 2024, 2023], we encode the patches into the latent space using a two-layer Multi-layer Perceptron (MLP) projector with residual connection added to each layer. To accommodate the patches with different sizes resulted from multi-resolution patching, we employ K projectors $\{\operatorname{Proj}_1, \operatorname{Proj}_2, ..., \operatorname{Proj}_K\}$, where Proj_k is utilized to process the k-th group of patches \mathcal{P}^k .

Coupled with the patch, we further introduce a point-level binary mask as a part of inputs to the projector to mark special points, such as padding. To be specific, during batch training or inference, padding is commonly adopted to fill in missing values for aligning different samples. This binary mask takes a value of 1 at valid input parts and 0 at padding positions, enabling the model to differentiate between them. Formally, along with the input time series $x_{1:C}$, we define the point-level mask as $m_{1:C}$, which will be divided into patches with different sizes together. For the patch p_j^k , the corresponding mask segment is m_j^k , and they are both passed to the projector:

$$\boldsymbol{z}_{j}^{k} = \operatorname{Proj}_{k}(\boldsymbol{p}_{j}^{k} \oplus \boldsymbol{m}_{j}^{k}), \tag{1}$$

where \oplus denotes the concatenation operation and \boldsymbol{z}_{j}^{k} denotes the latent patch embedding. Thus, based on the multi-resolution patching, it yields K groups of patch embeddings $\{\mathcal{Z}^{1},\mathcal{Z}^{2},...,\mathcal{Z}^{K}\}$, where $\mathcal{Z}^{k}=\{\boldsymbol{z}_{1}^{k},\boldsymbol{z}_{2}^{k},...,\boldsymbol{z}_{N_{k}}^{k}\}$ under the k-th division.

Next, we fuse the patch embeddings from different groups to form the final input for the subsequent Transformer model. Since different groups have varying patch sizes, they also contain different numbers of patches. To align them, we upsample the coarser groups with larger patch sizes and smaller patch numbers (i.e., $\{\mathcal{Z}_2,...,\mathcal{Z}_K\}$ with $\{N_2,...,N_K\}$ patches) by replicating its patches, so that they match the highest-resolution group \mathcal{Z}_1 , which has the largest number of patches N_1 . Formally, within each group \mathcal{Z}_k , the every patch embedding z_j^k will be repeated N_1/N_k times to match the patch number of \mathcal{Z}_1 . The replicated patch sequence can be denoted as:

$$\tilde{\mathcal{Z}}^k = \{\tilde{z}_1^k, \tilde{z}_2^k, ..., \tilde{z}_{N_1}^k\}, \text{ where } \tilde{z}_j^k = z_{\lceil j \cdot N_k/N_1 \rceil}^k, \ j = 1, 2, ..., N_1.$$
 (2)

After that, all groups are aligned to have the same number of patches (i.e., N_1), and the final patch embedding sequence \mathcal{Z} is obtained by summing the corresponding patches across all groups:

$$\mathcal{Z} = \{ \boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{N_1} \}, \ \boldsymbol{z}_j = \sum_{k=1}^K \tilde{\boldsymbol{z}}_j^k,$$
 (3)

This fusion strategy ensures that information from multiple resolutions is effectively aggregated while maintaining a consistent sequence length for the subsequent Transformer processing.

4.2 Transformer Blocks

In our approach, we utilize the encoder-decoder architecture of T5 model [Raffel et al., 2020] as the backbone for time series modeling and forecasting. Briefly, each block consists of a multi-head attention for contextual understanding and a feed-forward network layer for feature transformation. The relative position embedding is introduced in the calculation of attention scores. Specifically, the encoder applies the bi-directional attention which enables to capture complex temporal dependencies among patches. While in the decoder, the causal attention is employed to ensure auto-regressive forecasting. Additionally, the decoder also integrates the cross-attention to leverage the encoded past trends and contextual information for generating predictions. Such an encoder-decoder architecture enables the model to learn rich temporal relationships from historical data while maintaining consistency in future value generation. Note that in the calculation of attention scores, we introduce a patch-level attention mask to filter out the padded segment, which is derived from the previously discussed point-level mask m_j^1 . If all values in a patch's point-level mask are 0, it means that this patch contains only padding values, in which case the patch-level attention mask is set to 0; otherwise, it is assigned 1.

4.3 Output Module

Finally, an output module will project the decoder outputs into future predictions. Similar to the input module, the output module is also implemented by a two-layer MLP block with residual path. It takes the representation vector of the last patch processed by the decoder as input, and produce the prediction of the next patch. Formally, given the input sequence $x_{1:C+h}$, where $x_{1:C}$ is the context fed into the encoder, and $x_{C+1:C+h}$ is the preceding points passed to the decoder (which can be either partial ground truth labels during training or previous predicted values during inference), both parts will be processed in a patch-wise manner. Assuming that the last patch representation derived from the decoder is denoted as o_i , the output module predicts the subsequent patch as follows:

$$\tilde{x}_{C+h+1:C+h+P_o} = {\tilde{x}_{C+h+1}, \tilde{x}_{C+h+2}, ..., \tilde{x}_{C+h+P_o}} = \text{OutputHead}(o_i)$$
 (4)

where P_o denotes the output patch size. Note that previous studies (e.g., [Das et al., 2024]) have indicated that a larger output patch has the advantages of improved performance and faster generation in long-term forecasting, so our approach also allows a larger output patch size than the input patch.

Though our model focuses on point forecasting, we also enable it to derive probabilistic forecasts. Following [Wen et al., 2017], we add another prediction head to generate quantile forecasts for each time point in the next patch:

$$\{\tilde{q}_{C+h+1}, \tilde{q}_{C+h+2}, ..., \tilde{q}_{C+h+P_o}\} = \text{OutputHead}(o_j), \text{ where } \tilde{q}_i = \{\tilde{x}_i^{q_1}, \tilde{x}_i^{q_2}, ..., \tilde{x}_i^{q_Q}\}$$
 (5)

where Q denotes the quantile set of interest (e.g. deciles) with $q(\cdot) \in Q$, and $\tilde{x}_i^{q(\cdot)}$ denotes the quantile forecast value. In the practical implementation, we use an MLP block with an output dimension of $P_o \times (|Q|+1)$ to jointly produce the point and quantiles forecasts of the next patch. And then the desired output (point or quantiles) can be obtained by slicing the results accordingly.

4.4 Training Objective

We pre-train TimeFound using two kinds of objectives. The first one is the commonly used Mean Squared Error (MSE) loss that minimizes the difference between point forecast values and the ground truth values. For an input sequence $x_{1:C+H}$, it consists of the visible historical context $x_{1:C}$ and the future labels $x_{C+1:C+H}$ to be predicted. We feed the context $x_{1:C}$ to the encoder, pass $x_{C+1:C+H}$ to the decoder for teacher forcing, and adopts the label shift-right method to compute the prediction error for each patch. The loss will be computed over the entire future sequence:

$$\mathcal{L}_{MSE} = \frac{1}{H} \sum_{i=C+1}^{C+H} ||x_i - \tilde{x}_i||$$
 (6)

Second, another training objective is to minimize the total Quantile Loss (QL), based on the quantile forecasts:

$$\mathcal{L}_{QL} = \frac{1}{H} \sum_{i=C+1}^{C+H} \sum_{q \in Q} q(x_i - \tilde{x}_i^q) + (1 - q)(\tilde{x}_i^q - x_i)$$
 (7)

The overall loss function is $\mathcal{L} = \mathcal{L}_{MSE} + \mathcal{L}_{QL}$, and the loss is averaged over a batch during training.

4.5 Forecasting

During the inference stage, our model will perform an auto-regressive forecasting in a patch-by-patch manner. Given the input time series $x_{1:C}$ with the goal to predict the future $x_{C+1:C+H}$, the forecasting process begins with the model generating an initial patch prediction $\tilde{x}_{C+1:C+P_o}$. Then, this predicted $\tilde{x}_{C+1:C+P_o}$ is fed into the decoder as part of the input to produce the next patch prediction $\tilde{x}_{C+P_o+1:C+2P_o}$. This process is iteratively repeated, where the model continuously integrates the predictions from the previous steps to generate subsequent patches, until the total forecasted length reaches or exceeds the target horizon length H. In the case when H is not an integer multiple of the output patch size P_o , the excess forecast points in the last patch will be discarded.

5 Experiments

5.1 Pre-training Details

Dataset To pre-train the TimeFound model, we utilized the pre-training dataset introduced by [Ansari et al., 2024]. This dataset encompasses a diverse set of publicly available time series datasets spanning multiple domains such as energy, finance, and weather, as well as varying frequencies from five minutes to yearly. Furthermore, they applied two data augmentation strategies to enhance the diversity of the training data, where one is TSMixup strategy that generates 10M training samples by interpolating between the collected real-world time series sequences, and another is generating synthetic time series data via Gaussian processes. This ensures that the pre-training dataset can cover a broad range of forecasting scenarios. For additional details on the dataset composition and augmentation techniques, please refer to their original paper [Ansari et al., 2024].

Configuration We pre-train TimeFound in two sizes, namely *TimeFound-Base* and *TimeFound-Large*, with key parameter details listed in Table 1. The models are trained for 200K steps, with a batch size of 1024. We use the AdamW optimizer with initial lr=1e-3, $\beta_1=0.9$, $\beta_2=0.999$ and weight_decay = 0.01. A linear learning rate decay strategy is applied over the training steps. For both models, we set the context length to 512 and the prediction length is set to 192.

Table 1: Configuration of TimeFound model.

	Model Size	# Encoder Layers	# Decoder Layers	Hidden Size	# Heads	Patch Size	Output Patch Size
Base	200M	12	12	768	12	$\{P_1 = 16, P_2 = 32\}$	$P_o = 32$
Large	710M	24	24	1024	16	$\{P_1 = 16, P_2 = 32\}$	$P_o = 32$

Table 2: Details of zero-shot evaluation datasets. This table is modified from Ansari et al. [2024].

Dataset	Domain	Frequency	Num. Series	Min. Length	Max. Length	Horizon Length
Australian Electricity	Energy	30min	5	230736	232272	60
CIF 2016	Banking	1 M	72	28	120	12
Car Parts	Retail	1 M	2674	51	51	12
Covid Deaths	Healthcare	1D	266	212	212	30
Dominick	Retail	1D	100014	201	399	8
ERCOT Load	Energy	1H	8	154854	154854	24
ETT (15 Min.)	Energy	15min	14	69680	69680	24
ETT (Hourly)	Energy	1H	14	17420	17420	24
Exchange Rate	Finance	1B	8	7588	7588	30
FRED-MD	Economics	1 M	107	728	728	12
Hospital	Healthcare	1 M	767	84	84	12
M1 (Monthly)	Various	1 M	617	48	150	18
M1 (Quarterly)	Various	3M	203	18	114	8
M1 (Yearly)	Various	1Y	181	15	58	6
M3 (Monthly)	Various	1 M	1428	66	144	18
M3 (Quarterly)	Various	3M	756	24	72	8
M3 (Yearly)	Various	1Y	645	20	47	6
M5	Retail	1D	30490	124	1969	28
NN5 (Daily)	Finance	1D	111	791	791	56
NN5 (Weekly)	Finance	1W	111	113	113	8
Tourism (Monthly)	Various	1 M	366	91	333	24
Tourism (Quarterly)	Various	1Q	427	30	130	8
Tourism (Yearly)	Various	1Y	518	11	47	4
Weather	Nature	1D	3010	1332	65981	30

5.2 Empirical Evaluation

We evaluate the zero-shot forecasting performance of our model in two settings. (1) **Standard Last Window**. We compare all methods on the last test window of each dataset, which follows the evaluation setting for time series foundation models established in recent studies [Ansari et al., 2024, Das et al., 2024, Gruver et al., 2024]. (2) **Rolling Validation**. This setting aims to have a more in depth understanding of our model's forecasting performance with a longer horizon. It is conducted on a subset of popular long sequence datasets, and compares the average error of the rolling validation task on the entire test set.

Table 3: Comparison of zero-shot forecasting performance under the *standard last window* setting. The best (second best) results are in red (blue). *Note*: for Chronos variants, we report the results on the median trajectory of 20 sampled trajectories.

	MASE↓					sMAPE↓				
	TimesFM	Chornos (Base)	Chornos (Large)	TimeFound (Base)	TimeFound (Large)	TimesFM	Chornos (Base)	Chornos (Large)	TimeFound (Base)	TimeFound (Large)
Australian Electricity	1.089	1.141	1.273	1.044	1.177	l 0.048	0.049	0.053	0.047	0.049
Car Parts	0.841	0.811	0.806	0.821	0.807	0.934	0.951	0.948	0.942	0.941
CIF 2016	1.036	0.992	0.979	1.013	0.971	0.071	0.073	0.071	0.070	0.074
Covid Deaths	7.803	6.409	6.518	5.486	6.020	0.228	0.201	0.205	0.190	0.197
Dominick	0.938	0.770	0.774	0.868	0.850	0.791	0.810	0.810	0.789	0.790
ERCOT Load	0.589	0.567	0.635	0.616	0.636	0.011	0.011	0.012	0.011	0.012
ETT (15 Min.)	0.602	0.648	0.761	0.682	0.679	0.093	0.101	0.116	0.103	0.104
ETT (Hourly)	0.890	0.779	0.758	0.789	0.777	0.098	0.090	0.091	0.096	0.097
Exchange Rate	1.698	2.103	1.954	1.605	1.494	0.005	0.006	0.006	0.005	0.005
FRED-MD	0.650	0.495	0.520	0.565	0.573	0.056	0.049	0.050	0.052	0.052
Hospital	0.783	0.815	0.809	0.798	0.792	0.089	0.094	0.093	0.090	0.090
M1 (Monthly)	1.068	1.131	1.100	1.139	1.103	0.075	0.079	0.077	0.079	0.077
M1 (Quarterly)	1.671	1.768	1.706	1.749	1.714	0.079	0.090	0.086	0.085	0.085
M1 (Yearly)	4,004	4.462	4.413	4.203	4.304	0.097	0.112	0.110	0.104	0.106
M3 (Monthly)	0.935	0.872	0.866	0.890	0.864	0.073	0.070	0.070	0.072	0.071
M3 (Quarterly)	1.151	1.214	1.198	1.232	1.190	0.049	0.051	0.049	0.051	0.050
M3 (Yearly)	2,697	3.178	3.070	2.999	2.931	0.080	0.092	0.089	0.089	0.087
M5	1,397	1.430	1.430	1.412	1.412	0.778	0.820	0.820	0.788	0.794
NN5 (Daily)	0.894	0.843	0.833	0.875	0.866	0.112	0.105	0.104	0.110	0.108
NN5 (Weekly)	0.949	0.932	0.949	0.935	0.937	0.058	0.058	0.059	0.058	0.058
Tourism (Monthly)	1.918	1.861	1.819	1.663	1.610	0.122	0.128	0.125	0.107	0.104
Tourism (Quarterly)	2.063	1.782	1.649	1.669	1.746	0.099	0.088	0.082	0.082	0.087
Tourism (Yearly)	3,233	3.895	3,686	3,961	3,808	0.181	0.232	0.213	0.239	0.235
Weather	0.627	0.561	0.565	0.559	0.546	0.320	0.331	0.330	0.305	0.311
Geometric Mean	0.869	0.857	0.859	0.846	0.842	1.105	1.133	1.130	1.109	1.109

5.2.1 Zero-shot Evaluation: Last Window Setting

Setups We evaluate the zero-shot forecasting ability of TimeFound on 24 datasets that were unseen during the pre-training stage. Table 2 lists the details of these datasets. These datasets are largely consistent with the benchmark II introduced in the [Ansari et al., 2024], which comprises 27 datasets, mostly from the Monash [Godahewa et al., 2021] and Informer [Zhou et al., 2021], except for the *ERCOT Load* and *Exchange Rate* datasets. The key difference is that we have removed three datasets that were included in the pre-training data of baseline models. These datasets covers multiple domains and granularities. For each dataset, we report the errors on the last test window. The forecast horizon is determined according to the sampling frequency.

We compare the performance of our model against two recent state-of-the-art foundation models for zero-shot time series forecasting, Chornos [Ansari et al., 2024] and TimesFM [Das et al., 2024]. The evaluation metrics include the Mean Absolute Scaled Error (MASE, Hyndman and Koehler [2006]) and symmetric Mean Absolute Percentage Error (sMAPE). Since the magnitude of metrics vary across datasets, we follow Ansari et al. [2024] to compute the relative score of each model relative to a baseline approach, Seasonal Naive, on each dataset. Then, we report the geometric mean of the relative scores across all datasets as the overall performance of the model.

Results The results are shown in Table 3. As we can see, the proposed TimeFound model demonstrates strong zero-shot forecasting performance. Notably, our model achieves the best average performance (geometric mean of MASE), compared with the state-of-the-art foundation models TimesFM and Chornos, which indicates its good generalization ability across diverse forecasting scenarios. Also, our model can rank the first or the second on most datasets, particularly in the MASE metric. While TimesFM attains the highest ranking on a slightly larger number of individual datasets, its accuracy on other datasets falls significantly behind other approaches, leading to a lower average performance. Given that this benchmark only evaluated the model on the last test window, the statistical significance of individual dataset results is limited, thus the geometric mean result is a more reliable measure for evaluating overall performance. Moreover, as a foundation model, it is crucial to achieve strong results across diverse forecasting scenarios rather than excelling on just a few datasets. In this regard, geometric mean serves as a better indicator of a model's generalization ability. Therefore, TimeFound with highest geometric mean results is considered to have superior overall effectiveness.

Table 4: Comparison of *long-horizon* zero-shot forecasting performance under the *rolling validation* setting. The best (second best) results are in red (blue). *Note*: for Chronos variants, we report the results on the median trajectory of 20 sampled trajectories.

		MAE↓					sMAPE↓				
Dataset	Horizon	Timer	TimesFM	Chornos	TimeFound	Timer	TimesFM	Chornos	TimeFound		
	96	0.388	0.405	0.404	0.384	0.721	0.725	0.719	0.707		
	192	0.411	0.432	0.451	0.417	0.743	0.758	0.772	0.737		
ETTh1	336	0.431	0.459	0.468	0.442	0.753	0.795	0.805	0.741		
	720	0.471	0.482	0.521	0.452	0.856	0.888	0.906	0.812		
	average	0.425	0.444	0.461	0.423	0.768	0.792	0.800	0.749		
	96	0.342	0.344	0.337	0.329	0.549	0.516	0.509	0.511		
	192	0.399	0.393	0.383	0.394	0.617	0.581	0.587	0.586		
ETTh2	336	0.405	0.403	0.398	0.400	0.630	0.617	0.631	0.604		
	720	0.430	0.484	0.485	0.474	0.693	0.712	0.637	0.666		
	average	0.394	0.406	0.401	0.399	0.622	0.606	0.591	0.592		
	96	0.369	0.351	0.378	0.320	0.689	0.675	0.708	0.619		
	192	0.400	0.390	0.434	0.358	0.729	0.726	0.794	0.659		
ETTm1	336	0.426	0.420	0.469	0.384	0.780	0.778	0.875	0.702		
	720	0.490	0.472	0.524	0.443	0.883	0.850	0.956	0.784		
	average	0.421	0.408	0.451	0.376	0.770	0.757	0.833	0.691		
	96	0.274	0.257	0.262	0.246	0.466	0.435	0.447	0.415		
	192	0.313	0.314	0.296	0.288	0.505	0.486	0.477	0.450		
ETTm2	336	0.365	0.397	0.378	0.360	0.549	0.544	0.549	0.511		
	720	0.412	0.445	0.446	0.408	0.596	0.604	0.608	0.565		
	average	0.341	0.353	0.345	0.326	0.529	0.517	0.520	0.486		

5.2.2 Long-horizon Zero-shot Evaluation: Rolling Validation Setting

Setups We further benchmark our models' long-horizon forecasting ability on four electricity transformer temperature datasets (ETTh₁, ETTh₂, ETTm₁, ETTm₂) collected by Zhou et al. [2021]. In this setting, each model performs rolling forecasting across the entire test set. We compare their performance on horizon lengths of {96, 192, 336 and 720}, while the context length is fixed at 512.

In this experiment, another state-of-the-art foundation model Timer [Liu et al., 2024b] is included for comparison. This model is not compared in the *Last Window* setting because most of the test datasets have been seen during its pre-training stage. We select the Mean Absolute Error (MAE) and symmetric Mean Absolute Percentage Error (sMAPE) as metrics, and report the results calculated on the standard normalized data.

Results Table 4 presents the results, where we report the results of *Large* model for Chronos and our TimeFound. It shows that our model achieves the best overall performance in the long-horizon zero-shot forecasting task on these datasets, and it can consistently deliver strong results across different horizon lengths. We also observe that the Chronos baseline shows relatively poor performance. This is likely because it adopts point-based modeling for time series data (while other approaches are patch-based), and auto-regressively generates future predictions in a point-by-point manner, which leads to significant error accumulation in long-horizon forecasting. This provides valuable insights that patch-based modeling and prediction are crucial for building strong time series foundation models, particularly when the model is applied in long-horizon forecasting tasks.

6 Conclusion

In this paper, we introduced TimeFound, a foundation model designed for zero-shot time series forecasting across various domains. Our model employs a multi-resolution patching strategy within an encoder-decoder transformer architecture to capture complex temporal dynamics across different scales from a diverse set of time series data. Our experiments show that TimeFound achieves promising zero-shot results, superior to or competitive with the state-of-the-art time series foundation models, on multiple unseen datasets.

References

- Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.
- 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.
- Cathy WS Chen, Richard Gerlach, Edward MH Lin, and WCW Lee. Bayesian forecasting for financial risk management, pre and post the global financial crisis. *Journal of Forecasting*, 31(8): 661–687, 2012.
- Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12270–12280, 2021.
- Peng Chen, Yingying ZHANG, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.
- Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023.
- Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *ICLR*, 2024.
- Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924, 2017.
- Ruixin Ding, Yuqi Chen, Yu-Ting Lan, and Wei Zhang. Drformer: Multi-scale transformer utilizing diverse receptive fields for long time-series forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 446–456, 2024.
- Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. Advances in Neural Information Processing Systems, 36:2403–2426, 2023.
- Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. arXiv preprint arXiv:2310.03589, 2023.
- Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.
- Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.
- Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023.
- Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.
- Rob J Hyndman and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.
- Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.
- Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4356–4364, 2023.

- Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.
- Juan Pablo Karmy and Sebastián Maldonado. Hierarchical time series forecasting via support vector regression in the european travel retail industry. *Expert Systems with Applications*, 137:59–73, 2019.
- Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.
- Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6555–6565, 2024.
- Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a.
- Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024b.
- Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.
- Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.
- Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.
- David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191, 2020.
- Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.
- Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Timemoe: Billion-scale time series foundation models with mixture of experts. *arXiv* preprint *arXiv*:2409.16040, 2024.
- Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International journal of forecasting*, 36(1):75–85, 2020.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

- Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.
- Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.
- Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186, 2022.
- Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.
- Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.
- Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.
- Jingbo Zhou and Anthony KH Tung. Smiler: A semi-lazy time series prediction system for sensors. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1871–1886, 2015.
- Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jian Tang, Jiantao Su, Yu Li, Ji Liu, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: a dataset for spatial dynamic wind power forecasting over a large turbine array. *Scientific Data*, 11(1):649, 2024.
- Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.
- Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.
- Zhaoyang Zhu, Weiqi Chen, Rui Xia, Tian Zhou, Peisong Niu, Bingqing Peng, Wenwei Wang, Hengbo Liu, Ziqing Ma, Xinyue Gu, et al. Energy forecasting with robust, flexible, and explainable machine learning algorithms. *AI Magazine*, 44(4):377–393, 2023.