

# Sundial: A Family of Highly Capable Time Series Foundation Models

Yong Liu<sup>\*1</sup> Guo Qin<sup>\*1</sup> Zhiyuan Shi<sup>1</sup> Zhi Chen<sup>1</sup> Caiyin Yang<sup>1</sup>  
Xiangdong Huang<sup>1</sup> Jianmin Wang<sup>1</sup> Mingsheng Long<sup>1</sup>

## Abstract

We introduce *Sundial*, a family of native, flexible, and scalable time series foundation models. To predict the next-patch’s distribution, we propose a *TimeFlow Loss* based on flow-matching, which facilitates native pre-training of Transformers on continuous-valued time series without discrete tokenization. Conditioned on arbitrary-length time series, our models are pre-trained without specifying any prior distribution and can generate multiple probable predictions, achieving more flexibility in representation learning than using parametric densities. Towards time series foundation models, we leverage minimal but crucial adaptations of Transformers and curate *TimeBench with one trillion time points*, comprising mostly real-world datasets and synthetic data. By mitigating mode collapse via *TimeFlow Loss*, we pre-train a family of Sundial models on *TimeBench*, which achieve unprecedented model capacity and generalization performance. In addition to excellent scalability, Sundial achieves state-of-the-art results on both point and probabilistic forecasting benchmarks with a just-in-time inference speed, i.e., making zero-shot predictions within a few milliseconds. We believe that Sundial’s pioneering generative forecasting capability can improve model reliability in real-world decision-making. Code is available at: <https://github.com/thuml/Sundial>.

## 1. Introduction

Time series forecasting has fascinated people for thousands of years. Although people have been able to determine the time using instruments like sundials in 3000 BC, time series forecasting is intrinsically *non-deterministic* (Box et al.,

<sup>\*</sup>Equal contribution <sup>1</sup>School of Software, BNRist, Tsinghua University. Yong Liu <liyong21@mails.tsinghua.edu.cn>. Guo Qin <qinguo24@mails.tsinghua.edu.cn>. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

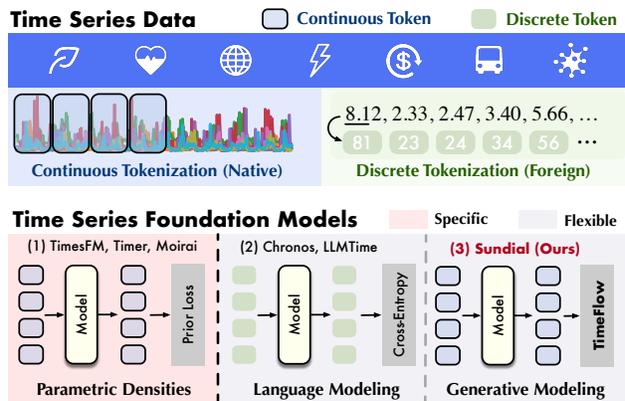


Figure 1. A *native* time series model operates on the original series of continuous values. A *flexible* foundation model is pre-trained without specifying prior distributions. Sundial is the first family of native and flexible time series foundation models.

2015). Therefore, generating a variety of probable predictions is crucial for decision-making. The growing demand has facilitated numerous statistical approaches over the past decades (Hyndman, 2018; Box, 2013), which provide high-profile theories and probabilistic tools for making reliable schedules. Recent advancements bring the boom of deftly designed models that automatically learn intricate dynamics and correlations from raw data (Oreshkin et al., 2019; Nie et al., 2022; Zhang & Yan, 2023; Liu et al., 2023a). Despite the impressive performance, deep models necessitate task-specific training on sufficient in-distribution data. Motivated by advances in large models (Bommasani et al., 2021), pre-trained time series foundation models have shown promising capabilities in out-of-distribution tasks (Das et al., 2023b; Liu et al., 2024b; Woo et al., 2024; Ansari et al., 2024).

Current research on time series foundation models has converged on building unified, scalable, and out-of-the-box forecasters, exhibiting zero-shot performance close to or sometimes surpassing supervised methods (Aksu et al., 2024). Notably, Transformers (Radford et al., 2018) are currently the *de facto* architecture of these models. While pre-trained Transformers with an inherent generative ability have facilitated great success in language, image, and video generation (Ramesh et al., 2021; OpenAI, 2023; Liu et al., 2024c), most time series foundation models are not “generative” or, more specifically, probabilistic forecasters, thereby limiting

reliability in decision-making. Although parametric densities specified with prior distributions (Wen et al., 2017; Woo et al., 2024) can be adopted to address uncertainty in time series forecasting, they can reduce the capacity of distributions learned by pre-trained models, especially on time series modality characterized by high heterogeneity. To learn arbitrarily intricate distributions without mode collapse, language modeling (Bengio et al., 2000) that learns the categorical distribution via cross-entropy loss inspires subsequent works (Gruber et al., 2023; Ansari et al., 2024), which treat time series as a *foreign* language using discrete tokenization. Still, discrepancies between continuous-valued time series and discrete language tokens can lead to out-of-vocabulary issues and coarse-grained prediction intervals.

As shown in Figure 1, Sundial is presented as the first family of generative models among time series foundation models. As foundation models intend to learn complicated distributions from extensive datasets and facilitate transferability across agnostic downstream datasets, we do not specify any prior parametric densities, such as unimodal and multimodal Gaussian mixtures. Instead, we delve into generative modeling to tame Transformers as native, flexible, and scalable time series foundation models. By comparing to denoising diffusion models (Li et al., 2024), we opt for a simple yet effective flow-matching framework (Lipman et al., 2022), which provides notable efficiency and sample quality (Tong et al., 2023). We propose *TimeFlow Loss*, a parameterized training objective (Zhang et al., 2018) for autoregressive models to learn and sample from each token’s predictive distribution. Optimizing models in the original continuous-valued domain, TimeFlow Loss facilitates patch-level generation and enables fast inference, which is naturally compatible with the time series modality.

In addition to TimeFlow, we enhance the Transformer with minimal but critical adaptations. We develop feasible patch tokenization for arbitrary-length input time series. We adopt RoPE (Su et al., 2024), Pre-LN (Xiong et al., 2020), FlashAttention (Dao et al., 2022), and KV Cache (Pope et al., 2023), which are crucial but generally neglected in the development of time series foundation models. Besides, we pre-train our models by multi-patch prediction to reduce autoregression steps. We realize a rapid generation of multiple samples by reusing a shared lookback representation. Beyond facilitating scalable pre-training, these adaptations help real-time long-context inference and long-term generation.

To validate the scaling law of time series foundation models, we collect and curate *TimeBench* with an unprecedented volume of a trillion time points. We present *Sundial* as a family of highly capable foundation models, which achieve state-of-the-art on three large-scale and best-recognized benchmarks, including Time-Series-Library (TSLib) (Wu et al., 2022), GIFT-Eval (Aksu et al., 2024), and FEV (Ansari et al., 2024).

Our contributions lie in these aspects:

- We propose TimeFlow Loss to predict next-patch’s distribution, allowing Transformers to be trained without discrete tokenization and make probable predictions.
- We present Sundial, a family of scalable and efficient time series foundation models built upon our enhanced Transformer and pre-trained on a trillion time points.
- Experimentally, Sundial achieves state-of-the-art zero-shot performance on point and probabilistic forecasting benchmarks, including TSLib, GIFT-Eval, and FEV, indicating a promising generative approach for the future improvement of time series foundation models.

## 2. Related Work

### 2.1. Time Series Forecasting

Forecasting is essential for decision-making. Advancements in deep learning for time series include theory-inspired deep modules (Wu et al., 2021; Liu et al., 2023b; Wu et al., 2022), architecture-oriented adaptations (Bai et al., 2018; Salinas et al., 2020; Lim et al., 2021), and time series preprocessing (Kim et al., 2021; Nie et al., 2022). Deep models learn the dataset-level distribution and benefit from strong generalization and model capacity. Statistical methods conduct case-by-case fitting on input series, achieving notable performance on small data (Ke et al., 2017; Hyndman, 2018).

One of the efforts towards more capable forecasters focuses on the foundation models (Bommasani et al., 2021), which address data-scarce scenarios by pre-training. More capable models support zero-shot forecasting, making inferences as fast as statistical methods and possessing large model capacity as deep models. Another aspect is to address uncertainty in time series forecasting. There is a growing research emphasis on probabilistic forecasting (Woo et al., 2024; Ansari et al., 2024). While parametric densities can be adopted as training objectives of probabilistic forecasting, they can be too specific to meet the heterogeneity of large-scale datasets, resulting in mode collapse in representation learning and over-smooth predictions (Figure 14-15). In this work, we introduce generative time series foundation models, which naturally address the uncertainty in forecasting.

### 2.2. Time Series Foundation Models

Recent research has concentrated on building versatile large time series models (Liang et al., 2024). With the advances made in large language models, Transformer has become the dominant architecture. Several works adapt Transformers to address the unique 2D-dimensionality and heterogeneity of time series (Woo et al., 2024; Liu et al., 2024a). Specifically, our work delves into tokenization and optimization.

Models such as TimesFM (Das et al., 2023b), Timer (Liu et al., 2024a;b), and Time-MoE (Shi et al., 2024b) embed continuous values and fit unimodal distributions via MSE or quantile loss (Wen et al., 2017). However, prior loss may result in mode collapse because predictive distributions are highly divergent across different domains. Besides, these models cannot provide the confidence level of predictions, limiting reliability for decision-making. Based on continuous tokenization, Moirai (Woo et al., 2024) presents a probabilistic model learning a mixture of distributions, but this prior can still fail to accommodate complex distributions. Inspired by language modeling, Chronos (Ansari et al., 2024) discretizes series via bucket quantization, learning more flexible categorical distributions by cross-entropy. Still, discrete tokenization is applied at each time point, which can lead to long contexts. Also, the final performance can be sensitive to quantization techniques. Unlike before, we tame Transformers as native time series foundation models, learning flexible distributions without discrete tokenization.

### 2.3. Generative Modeling for Time Series

By addressing complicated distributions during pre-training, generative modeling has become a focal point in the development of various foundation models (Zhao et al., 2023; Liu et al., 2024c). While this direction for time series mostly focused on time series generation (Tashiro et al., 2021) and task-specific forecasters (Rasul et al., 2021; Shen & Kwok, 2023; Kollovieh et al., 2024), generative modeling for time series foundation models is hardly explored. With the comparable flexibility in distribution learning as language modeling, diffusion denoising (Sohl-Dickstein et al., 2015) and flow-matching (Lipman et al., 2022) have gained increasing prevalence in continuous-valued modalities (Lipman et al., 2024). Compared with diffusion denoising models, flow-matching provides a simple yet efficient framework. With fewer steps involved in the forward and reverse processes, large models based on flow-matching have shown superior performance in image generation (Esser et al., 2024).

Despite the connection in value continuity, generating images and future time series are fundamentally different tasks due to the autoregressive property of forecasting. Our proposed TimeFlow Loss is designed for autoregressive models to conduct conditional generation, which is a parameterized loss function (Zhang et al., 2018) for arbitrary distributions and enhances representation learning of foundation models.

## 3. Preliminaries

### 3.1. Flow-Matching

The goal of generative modeling is to learn the underlying probability distribution that generates the data. The framework of flow-matching transforms a sample  $\mathbf{x}_0 \sim p_0$  drawn

from a source distribution into a sample  $\mathbf{x}_1 \sim p_1$  drawn from a target distribution. The transformation is continuous in time. For  $d$ -dimensional distributions, it is defined by a time-dependent velocity field  $u_t : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ , which is the solution of the ordinary differential equation (ODE):

$$\frac{d}{dt}\psi_t(\mathbf{x}) = u_t(\psi_t(\mathbf{x})) \text{ and } \psi_0(\mathbf{x}) = \mathbf{x}.$$

The velocity field  $u_t$  determines a flow  $\psi_t$ . For all  $t \in [0, 1]$ ,  $\psi_t$  generates the probability path  $p_t$  that interpolates  $p_0$  and  $p_1$ , i.e.,  $\mathbf{x}_t = \psi_t(\mathbf{x}_0) \sim p_t$  for  $\mathbf{x}_0 \sim p_0$ . The implementation of flow-matching is to train a network  $u_t^\theta$  parametrized by  $\theta$  to fit the velocity field  $u_t$ , which is a regression-based task formulated as the Flow-Matching objective:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t} \|u_t^\theta(\mathbf{x}_t) - u_t(\mathbf{x}_t)\|^2.$$

Furthermore, Lipman et al. (2022) proved the equivalence of optimizing the Conditional Flow-Matching objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t, \mathbf{x}_1} \|u_t^\theta(\mathbf{x}_t) - u_t(\mathbf{x}_t | \mathbf{x}_1)\|^2.$$

Leveraging the conditional optimal-transport (linear) path and a source Gaussian, the objective can be formulated as:

$$\mathcal{L}_{\text{CFM}}^{\text{Gauss}}(\theta) = \mathbb{E}_{t, \epsilon, \mathbf{x}_1} \|u_t^\theta(\mathbf{x}_t) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2. \quad (1)$$

where  $t \sim \mathcal{U}[0, 1]$ ,  $\mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{I})$  and  $\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\epsilon$ .

Consequently, we can train a generative network on given samples from the target distribution, and generate new samples by applying a push-forward process on samples drawn from a simple source Gaussian distribution:

$$\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = u_t^\theta(\mathbf{x}_t)\Delta t, \quad \mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad t \in [0, 1]. \quad (2)$$

### 3.2. Generative Models for Probabilistic Forecasting

Given a historical observation  $x_{1:t} = \{x_1, \dots, x_t\}$ , the target of time series forecasting is to predict future time series  $x_{t+1:t+f} = \{x_{t+1}, \dots, x_{t+f}\}$ . The task can be generally formulated as  $p(x_{t+1:t+f} | \mathbf{h}_t)$ , where  $\mathbf{h}_t = f_\phi(x_{1:t})$  is the learned representation from a deep model  $f_\phi$ . In probabilistic forecasting, explicit optimization objectives are utilized to predict the statistics of future series, e.g., MSE or quantile loss, which have specified  $p$  as a prior distribution. While using one parametric density generally fits well on a small amount of data, it can be the major bottleneck for scaling time series foundation models. Inspired by the success of large generative models (Rombach et al., 2022; OpenAI, 2023; Esser et al., 2024), we introduce generative modeling to realize probabilistic forecasting:

$$p(x_{t+1:t+f} | \mathbf{h}_t) = g_\theta(f_\phi(x_{1:t})). \quad (3)$$

$g_\theta$  is a small trainable generative network conditioned on the learned representations of  $f_\phi$ , which is jointly optimized

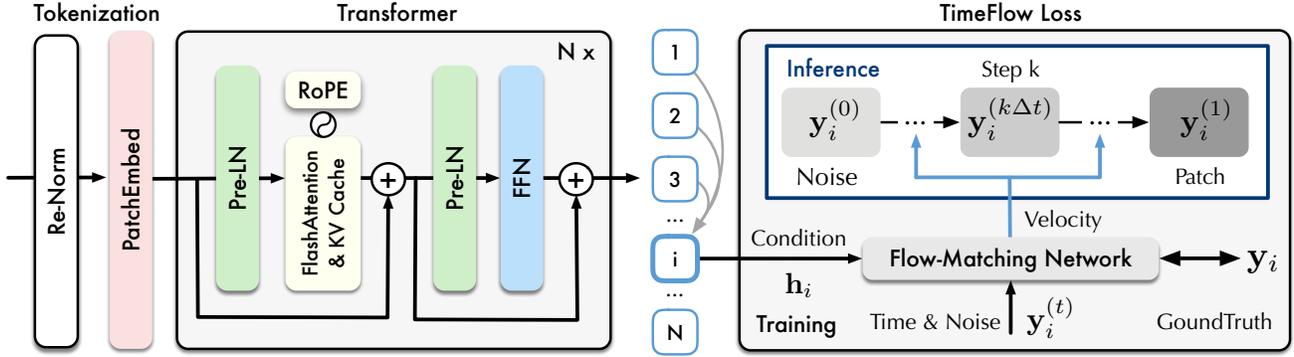


Figure 2. Overall architecture of Sundial. The input time series is divided into patch tokens, which are embedded from original continuous values. The patch embeddings are fed into a decoder-only Transformer, a stable and speedup version that learns token representations via causal self-attention. The model is optimized using our TimeFlow Loss, a parameterized loss function that models per-token probability distribution conditioned on the learned representations, and generates multiple plausible predictions under the flow-matching framework.

with  $f_\phi$ . While the generative model automatically fits the target distribution, it can sample raw predictions and calculating their statistics for probabilistic forecasting. The aim is conceptually related to conformal prediction (Vovk et al., 2005) but models uncertainty beyond prediction intervals.

## 4. Approach

In this work, we conduct a univariate pre-training paradigm, which adopts the S3 format proposed by Liu et al. (2024b) to address multivariate data. To mitigate value range discrepancy, we conduct normalization on time series individually per variable. Afterwards, we sample varying-length training samples with the maximum context length of 2880. As a foundation model, Sundial is required to predict on out-of-distribution series with varied lengths during inference.

### 4.1. Sundial

As shown in Figure 2, the Sundial models consist of three parts: (1) time series tokenization, including a context-level re-normalization and a patch embedding that addresses any-length time series, (2) a Transformer backbone that learns the per-token representation of time series, and (3) *TimeFlow Loss*, a parameterized loss function to model the per-token distribution and generate raw series during inference. Intuitively, Sundial can be regarded as an ARMA (Auto-Regression and Moving-Average) deep model, i.e., Transformer learns token representations autoregressively. Conditioned on the lookback representations, TimeFlow transforms random noises into non-deterministic predictions.

#### 4.1.1. TIME SERIES TOKENIZATION

**Re-Normalization** We adopt stationarization (Liu et al., 2022), a non-parametric two-stage instance normalization conducted within each sample, which is initially proposed to mitigate non-stationarity of time series. Here, it helps to ad-

dress temporal distribution shift and outlier ranges in input series, improving generalizability for zero-shot forecasting.

**Patch Embedding** Given a univariate time series  $\mathbf{X} = \{x_1, \dots, x_T\}$ , it is divided into patches  $\mathbf{x}_i = x_{1+(i-1)P:iP}$  with the length of  $P$ . To address non-divisible length, we pad the input at the beginning and use a binary mask  $\mathbf{m}_i \in \mathbb{R}^P$  for each patch to indicate the padded position. It will lead to  $N = \lceil T/P \rceil$  such input tokens. Subsequently, we use a shared MLP:  $\mathbb{R}^{2P} \mapsto \mathbb{R}^D$  to embed all patch tokens:

$$\mathbf{h}_i = \text{PatchEmbed}(\text{Concat}(\mathbf{x}_i, \mathbf{m}_i)), \quad (4)$$

where  $\mathbf{h}_i \in \mathbb{R}^D$  and  $D$  is the dimension of token embedding. Unlike point-level quantization (Ansari et al., 2024), we reserve original values without discrete quantization. It also reduces the context length (in token) of the Transformer.

#### 4.1.2. TRANSFORMER BACKBONE

Given  $N$  token embeddings  $\{\mathbf{h}_i\}$ , we adopt several crucial adaptations on a decoder-only Transformer to obtain per-token representations aggregated from all previous tokens. First, we adapt Pre-LN (Xiong et al., 2020) to improve pre-training stability. Second, we leverage a causal self-attention mechanism with RoPE (Su et al., 2024) that introduces the position information of patch tokens. It can be formulated as follows (the layer index is omitted for simplicity):

$$A_{ij} = \mathbf{h}_i^\top \mathbf{W}_q \mathbf{R}_{\Theta, i-j} \mathbf{W}_k^\top \mathbf{h}_j, \\ \text{Attention}(\mathbf{H}) = \text{Softmax}\left(\frac{\text{Mask}(\mathcal{A})}{\sqrt{d}}\right) \mathbf{H} \mathbf{W}_v, \quad (5)$$

where  $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{D \times d}$  project token embeddings  $\mathbf{H} = \{\mathbf{h}_i\}$  into  $d$ -dimensional queries, keys, and values.  $\mathbf{R}_{\Theta, t} \in \mathbb{R}^{d \times d}$  is the rotary matrix with rotation degree  $(t \cdot \Theta)$ . Lastly, we implement FlashAttention (Dao et al., 2022) and KV Cache (Pope et al., 2023), since these enhancements for deployment are increasingly emphasized in large foundation models (Shoeybi et al., 2019; Rasley et al., 2020).

### 4.1.3. TIMEFLOW LOSS

Given representations  $\{\mathbf{h}_i\}$  extracted by the last layer of the Transformer, we aim to generate length- $F$  predictions  $\hat{\mathbf{y}}_i = \hat{x}_{1+iP, F+iP}$  at each position  $i$  via our autoregressive model. Motivated by the empirical observation that a larger patch size improves the performance in decoder-only Transformers (Das et al., 2023b) while a small patch size can be more flexible to accommodate data of different frequencies, we adopt multi-patch predictions ( $F > P$ ) for pre-training, which also reduces the steps of autoregressive inference.

Based on Equations 1 and 3, we formulate a new generative forecasting conditioned on a sequential representation  $\mathbf{h}_i$ :

$$\mathcal{L}(\theta, \mathbf{h}_i) = \mathbb{E}_{t, \epsilon, \mathbf{y}_i} \left\| u_t^\theta(\mathbf{y}_i^{(t)} | \mathbf{h}_i) - (\mathbf{y}_i - \mathbf{y}_i^{(0)}) \right\|^2. \quad (6)$$

where  $\mathbf{y}_i \in \mathbb{R}^F$  is the groundtruth value and  $\mathbf{y}_i^{(0)}$  is a  $d$ -dimensional Gaussian noise,  $t$  is sampled from  $\mathcal{U}[0, 1]$ , and  $\mathbf{y}_i^{(t)} = t\mathbf{y}_i + (1-t)\mathbf{y}_i^{(0)}$  is constructed by the conditional optimal-transport path. It is important to note that the conditional representation  $\mathbf{h}_i$  differs from the conditional path and the conditional source distribution. Instead,  $\mathbf{h}_i$  is a condition of position  $i$ , also a time-invariant condition of the whole flow-matching process  $t \in [0, 1]$ . Technically, we implement the flow-matching network by a small MLP:

$$u_t^\theta(\mathbf{y}_i^{(t)} | \mathbf{h}_i) = \text{FM-Net}(\mathbf{y}_i^{(t)}, t, \mathbf{h}_i). \quad (7)$$

The training process involves sampling the noised  $\mathbf{y}_i^{(t)}$ , and jointly input it with  $t$ . The condition  $\mathbf{h}_i$  is integrated into the flow-matching network via AdaLN (Peebles & Xie, 2023). TimeFlow Loss for autoregressive models is formulated as:

$$\mathcal{L}_{\text{TimeFlow}} = \sum_{i=1}^N \left\| \text{FM-Net}(\mathbf{y}_i^{(t)}, t, \mathbf{h}_i) - (\mathbf{y}_i - \mathbf{y}_i^{(0)}) \right\|^2. \quad (8)$$

**Inference** Based on Equation 2, the push-forward process conditioned on a learned representation  $\mathbf{h}_i$  is formulated as

$$\mathbf{y}_i^{(t+\Delta t)} = \mathbf{y}_i^{(t)} + u_t^\theta(\mathbf{y}_i^{(t)} | \mathbf{h}_i) \Delta t. \quad (9)$$

Technically, we adopt a  $K$ -step uniform trajectory, and set  $\Delta t = 1/K$ . The sampling is done via starting from an initial Gaussian noise and advancing with the velocity generated by the trained FM-Net iteratively, as shown in Algorithm 1.

This procedure generates a predicted sample  $\hat{\mathbf{y}}_i$  at position  $i$ . To calibrate probabilistic forecasting results during inference, we repeat this procedure using different initial noises and estimate statistics such as the median and quantiles from a set of generated predictions. We implement an efficient repeated-sampling in the TimeFlow module. The condition (representation) of lookback series is shared and reused for different initial noises, thereby reducing the overhead of repeated forwarding in the Transformer backbone.

---

### Algorithm 1 TimeFlow Loss: Sampling

---

**Require:** condition  $\mathbf{h}_i \in \mathbb{R}^D$ , path steps  $K$ .

- 1: Sample initial noise  $\hat{\mathbf{y}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ .
  - 2:  $\Delta t = 1/K$
  - 3: **for**  $k$  **in**  $\{0, 1, \dots, K-1\}$  **do**
  - 4:      $\hat{\mathbf{y}}_i \leftarrow \hat{\mathbf{y}}_i + \text{FM-Net}(\hat{\mathbf{y}}_i, k\Delta t, \mathbf{h}_i) \Delta t$
  - 5: **end for**
  - 6: **Return:**  $\hat{\mathbf{y}}_i$
- 

## 4.2. TimeBench

We collected and curated *TimeBench*, which comprises over a trillion time points from various sources, as shown in Figure 3. Several datasets originate from research teams (Woo et al., 2024; Ansari et al., 2024; Liu et al., 2024a;b). While most datasets are collected from real-world records, a small portion (0.05%) is generated synthetically to enhance pattern diversity, following KernelSynth proposed by Ansari et al. (2024). We also leverage substantial meteorological data (Hersbach et al., 2020) because of the predictability of weather systems. Data of different frequencies encompasses common and comprehensive temporal dynamics.

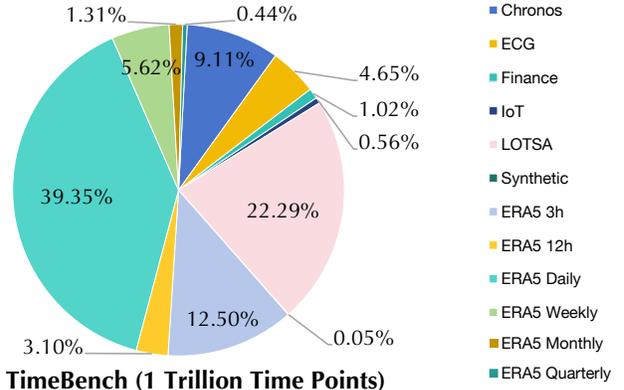


Figure 3. Ratios of data sources in TimeBench, the pre-training corpora of Sundial. Detailed statistics are provide in Table 4.

## 5. Experiments

We evaluate Sundial on best-recognized zero-shot forecasting benchmarks (Section 5.1) and investigate the scaling behavior of Sundial (Section 5.2). We compare TimeFlow with other training objectives (Section 5.3). We delve into test-time calibration of generative forecasters (Section 5.4). We conduct model adaptation of Sundial, i.e., instruction tuning (Section 5.5) and provide in-depth ablation studies to evaluate our modular enhancement (Section 5.6).

### 5.1. Time Series Forecasting

In this section, we focus on zero-shot forecasting, we compare Sundial with advanced time series foundation models on various benchmarks, including (1) point forecasting: we

## Sundial: A Family of Highly Capable Time Series Foundation Models

Table 1. Zero-shot forecasting results of time series foundation models on long-term forecasting datasets (Time-Series-Library) (Wu et al., 2022). Corresponding prediction lengths include {96, 192, 336, 720}. A lower MSE or MAE indicates a better prediction. Averaged results of four prediction lengths are reported here. 1<sup>st</sup> Count represents the number of wins achieved by a model under all prediction lengths and datasets. Results of baseline models are officially reported by Shi et al. (2024b). Datasets in pre-training are not evaluated on corresponding models, which are denoted by the dash (-). Full results under all prediction lengths are provided in Table 9.

Models	Sundial <sub>Small</sub> (Ours)		Sundial <sub>Base</sub> (Ours)		Sundial <sub>Large</sub> (Ours)		Time-MoE <sub>Base</sub> (2024b)		Time-MoE <sub>Large</sub> (2024b)		Time-MoE <sub>Ultra</sub> (2024b)		Timer-XL (2024a)		Moirai <sub>Base</sub> (2024)		Moirai <sub>Large</sub> (2024)		Chronos <sub>Base</sub> (2024)		Chronos <sub>Large</sub> (2024)		TimesFM (2023b)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.354	0.388	<u>0.336</u>	<u>0.377</u>	<b>0.331</b>	<b>0.369</b>	0.394	0.415	0.376	0.405	0.356	0.391	0.373	0.392	0.406	0.385	0.422	0.391	0.645	0.500	0.555	0.465	0.433	0.418
ETTm2	0.265	0.324	<u>0.258</u>	<u>0.320</u>	<b>0.254</b>	<b>0.315</b>	0.317	0.365	0.316	0.361	0.288	0.344	0.273	0.336	0.311	0.337	0.329	0.343	0.310	0.350	0.295	0.338	0.328	0.346
ETTh1	<b>0.390</b>	<u>0.418</u>	0.411	0.434	0.395	0.420	0.400	0.424	<u>0.394</u>	0.419	0.412	0.426	0.404	<b>0.417</b>	0.417	0.419	0.480	0.439	0.591	0.468	0.588	0.466	0.473	0.443
ETTh2	0.340	0.387	<b>0.333</b>	0.387	<u>0.334</u>	0.387	0.366	0.404	0.405	0.415	0.371	0.399	0.347	0.388	0.362	<u>0.382</u>	0.367	<b>0.377</b>	0.405	0.410	0.455	0.427	0.392	0.406
ECL	<u>0.169</u>	<u>0.265</u>	<u>0.169</u>	<u>0.265</u>	<b>0.166</b>	<b>0.262</b>	-	-	-	-	-	-	0.174	0.278	0.187	0.274	0.186	0.270	0.214	0.278	0.204	0.273	-	-
Weather	<b>0.233</b>	<u>0.271</u>	<u>0.234</u>	<b>0.270</b>	0.238	0.275	0.265	0.297	0.270	0.300	0.256	0.288	0.256	0.294	0.287	0.281	0.264	0.273	0.292	0.315	0.279	0.306	-	-
1 <sup>st</sup> Count	7	2	<u>8</u>	5	<b>16</b>	<b>16</b>	0	1	0	0	2	1	1	3	0	2	0	<u>6</u>	0	0	0	0	0	0

Table 2. GIFT-Eval comprises 23 datasets characterized by a variety of frequencies, variate numbers, and prediction lengths. We evaluate zero-shot performance using 100 generated series, being consistent with Woo et al. (2024). A lower MASE or CRPS indicates a better performance. Rank assigns a numerical ranking of all 97 configurations. Baseline results are officially reported by Aksu et al. (2024).

Type	Statistical Methods				Task-Specific Models (Supervised)					Time Series Foundation Models (Zero-Shot)				
Model	Naïve	Seasonal Naïve	Auto ARIMA	Auto Theta	DeepAR (2020)	TiDE (2023a)	N-BEATS (2019)	PTST (2022)	iTrans. (2023a)	TimesFM (2023b)	TabPFN (2025)	Chronos (2024)	Moirai (2024)	<b>Sundial (Ours)</b>
MASE	1.260	1.000	0.964	0.978	1.206	0.980	0.842	0.762	0.802	<u>0.680</u>	0.748	0.786	0.809	<b>0.673</b>
CRPS	1.383	1.000	0.770	1.051	0.721	0.652	0.689	0.496	0.524	<b>0.465</b>	0.480	0.551	0.515	<u>0.472</u>
Rank	28.072	26.175	21.515	24.031	18.938	18.557	21.381	10.052	11.320	<b>8.237</b>	<u>8.268</u>	14.309	10.175	9.062

adopt the long-term forecasting benchmark (Wu et al., 2022), which assesses the performance under different forecasting horizons using MSE and MAE; (2) probabilistic forecasting: we experiment on GIFT-Eval (Aksu et al., 2024) and FEV leaderboard (Ansari et al., 2024), following their official evaluation suite and assessing point (MASE) and probabilistic (CRPS and WQL) metrics. All evaluated datasets are excluded from the pre-training dataset. Model is available on HuggingFace<sup>1</sup> and configurations are detailed in Table 5.

### 5.1.1. POINT FORECASTING

As shown in Table 1, Sundial consistently outperforms other advanced time series foundation models. Compared with the previous state-of-the-art model Time-MoE (Shi et al., 2024b), the Sundial family using fewer parameters achieves the average MSE reduction of 7.57% and averaged MAE reduction of 4.71%. Notably, continuous tokenization allows our model to conduct patch-level forecasting with fewer autoregression steps, while Chronos using point-wise discrete tokenization may not be suitable in long-term forecasting.

### 5.1.2. PROBABILISTIC FORECASTING

Beyond point forecasting, Sundial possesses a unique generative capability for making probabilistic predictions. Fol-

lowing Ansari et al. (2024), we calculate the median and quantiles using a set of raw predictions of Sundial. While several baseline models have been pre-trained by the consistent objective function for probabilistic evaluation, e.g., quantile loss for WQL, Sundial calculates these statistics for evaluation without any prior knowledge.

**GIFT-Eval** Aggregated results are presented in Table 2. The benchmark evaluates performance from 23 datasets and 13 baseline models, encompassing statistical methods, task-specific models, and time series foundation models. Among supervised models and advanced foundation models, Sundial attains the first place in MASE and second place in CRPS on all unseen datasets. While the top PatchTST (Nie et al., 2022) is exhaustively trained and tweaked on each dataset, the zero-shot performance of Sundial highlights its simplicity and robustness on this comprehensive benchmark.

**FEV Leaderboard** We evaluate our Sundial on the open leaderboard established by AutoGluon (Ansari et al., 2024), which includes 27 datasets for probabilistic forecasting. As shown in Figure 4, the zero-shot forecasting performance of Sundial exceeds 70% statistical methods and deep models that are supervisedly trained in distribution. While Sundial is ranked as the second zero-shot pre-trained models after Chronos, Sundial realizes 35× inference speedup as shown in Figure 5. Based on patch-wise tokenization and multi-

<sup>1</sup><https://huggingface.co/thuml/sundial-base-128m>.

## Sundial: A Family of Highly Capable Time Series Foundation Models

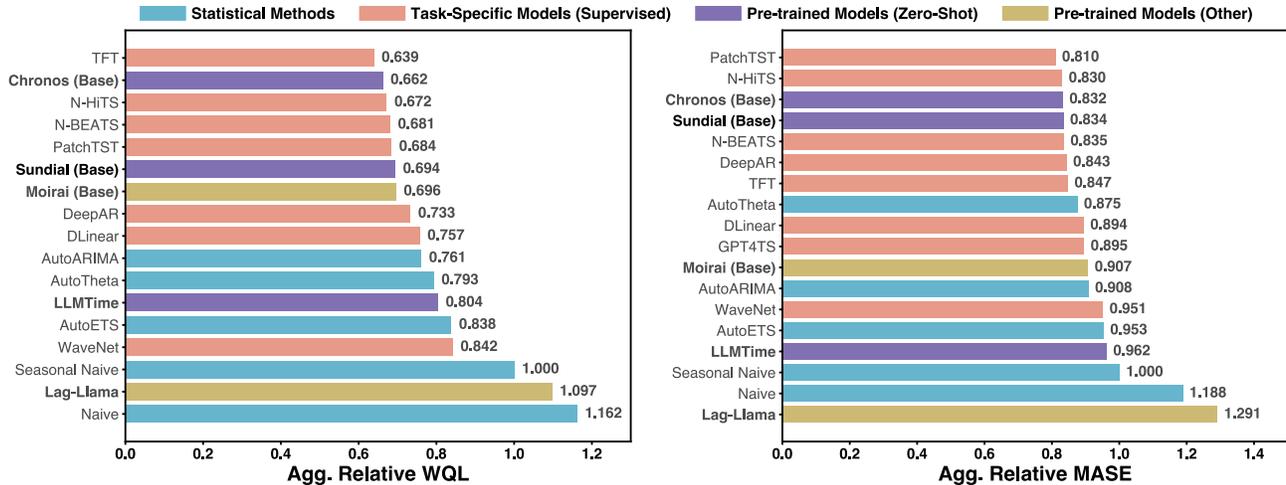


Figure 4. Model evaluation on the FEV leaderboard, which includes 27 datasets not seen by Sundial. Baseline models can be categorized into statistical methods fitting on each time series, task-specific deep models trained on each dataset, and pre-trained foundation models. Pre-trained Models that have seen several datasets during pre-training are denoted as Pre-trained Models (Other). A lower MASE/WQL indicates a better result. Sundial makes probabilistic predictions using 20 generated series, being consistent with Ansari et al. (2024).

patch prediction, our inference speed is near to N-BEATS.

Besides, we provide qualitative showcases in Appendix D. TimeFlow can generate highly eventful and coherent temporal patterns with input series. Beyond the mean or quantiles, our model enables the estimation of arbitrary statistics by sampling directly from the predictive distribution.

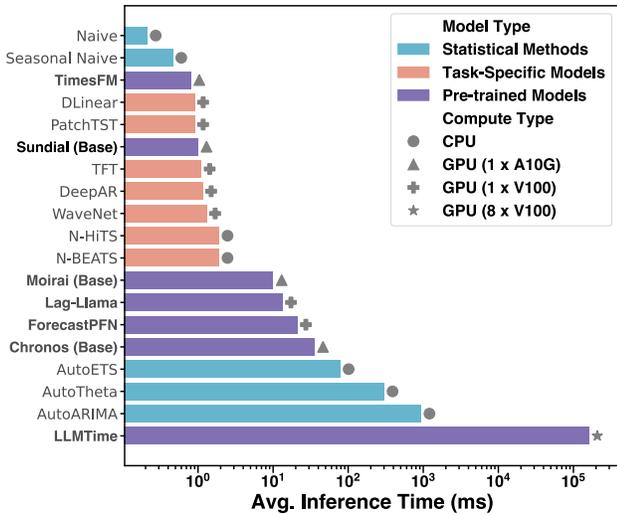


Figure 5. Inference time evaluation following Ansari et al. (2024), which is averaged from the FEV leaderboard. Computing resources of different models are marked. We plot the logarithmic x-axis.

### 5.2. Scalability

From Table 1, the larger Sundial model consistently achieves better performance with the scaling of parameters. Beyond downstream performance, we delve into the utilization of model capacity. Figure 6 shows training curves of different

sizes. Compared to Sundial (Small), the large version leads to 15.38% reduction in the converged training loss, exhibiting promising model capacity of generative forecasters.

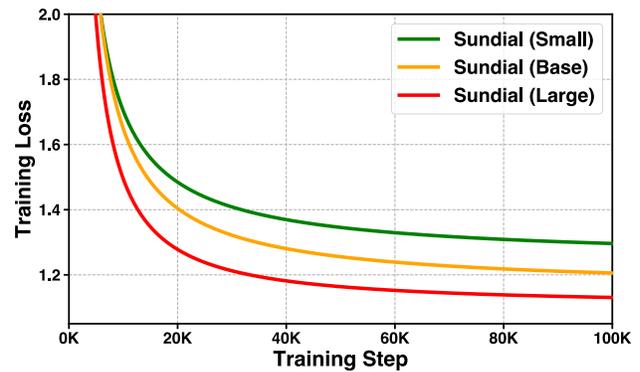


Figure 6. Training curves on TimeBench of different model sizes.

### 5.3. TimeFlow Loss

Based on the flow-matching framework, TimeFlow Loss allows autoregressive models to learn and generate flexible distributions while enhancing representation learning. To validate the effectiveness of this design, we implement two alternatives: (1) an MLP network and MSE Loss and (2) a parameterized training objective based on the denoising diffusion procedure (Li et al., 2024). We adopt the same parameterized network and Transformer backbone and pre-train them on TimeBench. Since the converged training loss is not comparable across different objective functions, we compare zero-shot performance in Table 3. Despite allowing for sampling predictions, performance using diffusion-based objective is notably inferior to TimeFlow Loss.

In addition to zero-shot performance, we provide showcases

Table 3. Zero-shot performance using different training objectives. We use the same model configuration and pre-training scale. Averaged MSE of four prediction lengths are reported here.

Objective	ETTh1	ETTh2	ETTm1	ETTm2	ECL	Weather	Avg.
<b>TimeFlow</b>	<b>0.336</b>	<b>0.258</b>	<b>0.411</b>	<b>0.333</b>	<b>0.169</b>	<b>0.234</b>	<b>0.290</b>
Diffusion	0.362	0.265	0.444	0.360	0.202	0.252	0.314
MSE	<b>0.360</b>	<b>0.264</b>	<b>0.404</b>	<b>0.341</b>	<b>0.175</b>	<b>0.231</b>	<b>0.296</b>

for quality evaluations in Appendix D.2. Pre-trained models optimized by the specific MSE Loss can only output a single prediction. And the prediction is sometimes over-smooth due to mode collapse (refer to Appendix C.1). Instead, generative modeling can accommodate significantly different future variations even if their lookback series are similar. We also provide a probabilistic metric CRPS to compare different objectives in Table 7, which validate that the predictive distribution modeled by TimeFlow is more coherent and diverse than counterpart training objectives. It benefits downstream tasks by generating multiple plausible predictions, conveys various future possibilities and enhances the reliability of decision-making.

#### 5.4. Test-Time Calibration

Generative modeling facilitates the flexibility to calibrate the final prediction during inference. Based on the median-based forecasting strategy, i.e., starting from multiple noise of a standard Gaussian and calculating the median of raw predictions, there are two configurations to calibrate final predictions: (1) the number of samples to calculate statistics and (2) sampling steps  $K$  used for flow-matching. Figure 7 shows the results using different configurations.

The top two figures conform to the central limit theorem. Generating more samples leads to more calibrated estimation of prediction and confidence interval. The bottom two figures indicate that using fine-grained steps during the push-forward process can leads to more precise predictions.

The trade-off between inference time and performance reveals the potential of test-time calibration, which does not require retraining models. The generative capability of Sundial provides flexibility for various use cases requiring different levels of uncertainty. In our experiments, sampling 20 predictions with each generated by 50 steps consumes nearly one second on a CPU, which is notably more efficient than tuning deep models or statistical methods. Advanced strategies of sampling and post-processing of raw prediction leave interesting directions for future exploration.

#### 5.5. Model Adaptation

Inspired by the prevalence of instruction tuning (Wei et al., 2021) that adapts foundation models on a collection of tasks.

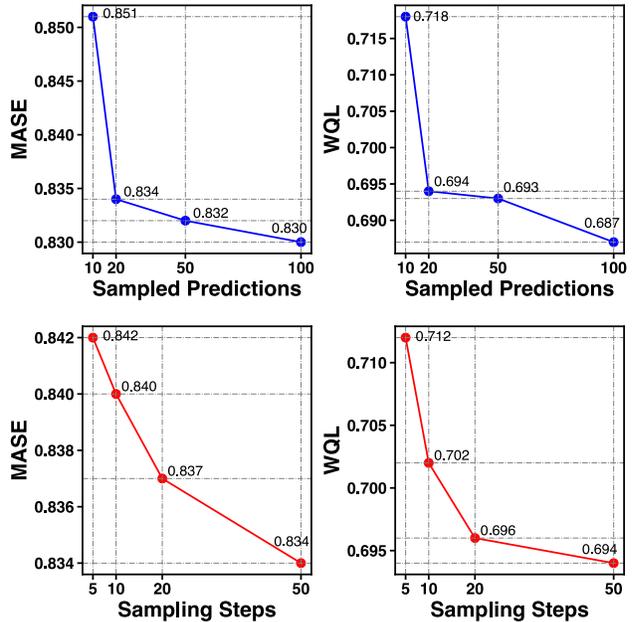


Figure 7. We show the MASE (left) and WQL (right) on FEV w.r.t. the number of generated raw predictions (top) and the steps to sample a prediction (down). More predictions or more sampling steps generally achieve better probabilistic metrics.

We fine-tune pre-trained Sundial (Base) on the FEV leaderboard, including short-term tasks with different prediction lengths. Our model is tuned once on all aggregated datasets. We evaluate the performance on unseen test splits (Figure 8). We observe that the performance can be further improved compared to zero-shot forecasting. Furthermore, training from scratch on aggregated datasets results in inferior performance, implying knowledge transfer in pre-trained models.

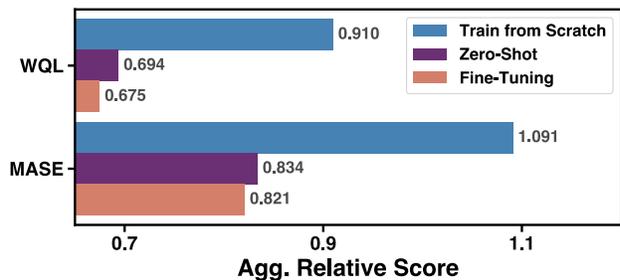


Figure 8. Performance on the FEV leaderboard, including (1) training Sundial from scratch on all datasets from the FEV leaderboard, (2) zero-shot forecasting using pre-trained Sundial, and (3) fine-tuning once on all datasets from the FEV leaderboard.

#### 5.6. Ablation Study

We conducted several ablation studies that provide insights into the enhancement made to Sundial’s architecture. We evaluate the overall zero-shot performance on TSLib, which covers six different datasets and four prediction lengths.

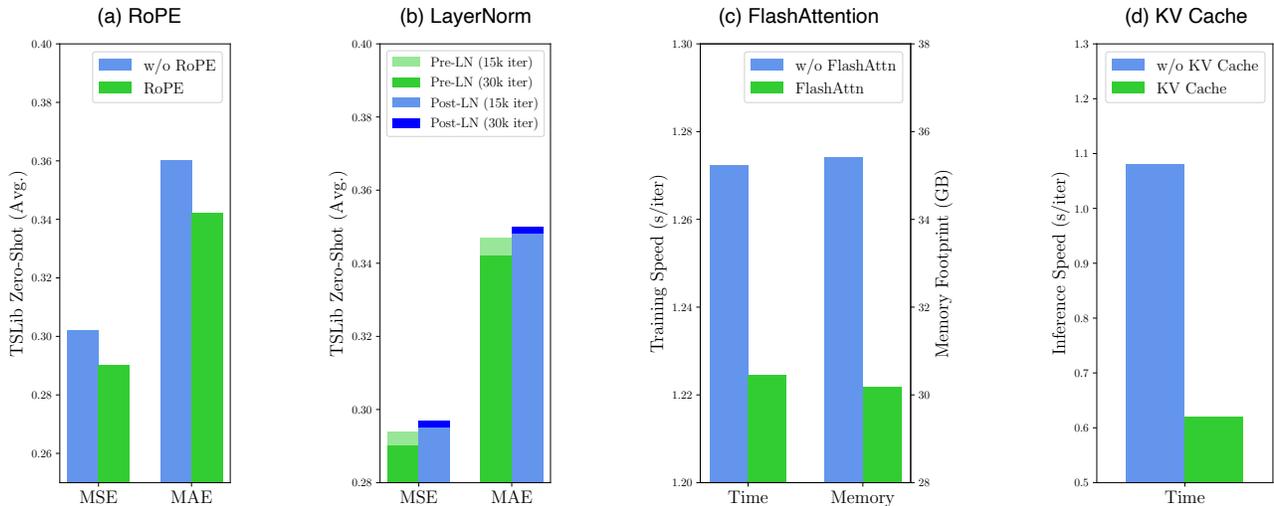


Figure 9. Ablation studies with respect to architectural enhancements. We report the averaged results of TSLib datasets (Wu et al., 2022) from four prediction lengths {96, 192, 336, 720} and all six datasets. The context length is set to 2880 and the patch length is 16.

**RoPE** Prior research (Liu et al., 2024a) observed that the introduction of RoPE (Su et al., 2024) yields better results in supervised forecasting tasks. As shown in Figure 9 (a), RoPE can also improve zero-shot forecasting, presenting a general enhancement for time series foundation models.

**Layer Normalization** Pre-LN (Baevski & Auli, 2018) is widely adopted in large language models (Touvron et al., 2023) due to the training stability. As depicted in Figure 9 (b), training with Pre-LN for more iterations yield better performance. In contrast, training with Post-LN, which is the predominant choice in supervised models, may adversely affect downstream results.

**FlashAttention and KV Cache** We make it to leverage FlashAttention (Dao et al., 2022) and KV Cache to reduce the computational costs. As shown in Figure 9 (c) and (d), they notably reduce 14.8% memory footprint and 43.6% inference time without affecting performance.

## 6. Conclusion

In this work, we collect and curate TimeBench, a trillion-scale time series dataset for building time series foundation models, which can benefit the research community. Towards time series foundation models, we delve into tokenization and optimization, presenting contributions in two aspects. First, we demonstrate that continuous tokenization, such as patch tokens, can be more effective and efficient for the time series modality, and generative modeling presents a native approach for learning on continuous-valued time series. Second, we propose a novel training objective to accommodate heterogeneous time series distribution. It endows autoregressive models with an inherent capability to sample

from non-categorical distribution. Our pre-trained Sundial models make substantial advances on best-recognized forecasting leaderboards. We hope this work can inspire future paradigms for pre-training time series foundation models and enhance their applicability to real-world applications.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (U2342217 and 62021002), the BNRist Innovation Fund (BNR2024RC01010), and the National Engineering Research Center for Big Data Software.

We extend our gratitude to Xingzhuo Guo for his expertise in flow-matching and meticulous proofreading of the method section. We further thank Jialong Wu, Yuezhou Ma and Yu Zhang for insightful discussions about generative models. Their collective support significantly enhanced this work.

## Impact Statement

This paper aims to advance the development of time series foundation models. We curated a pre-training dataset from publicly available resources. Our models employ an efficient tokenization and incorporate a generative training objective. The proposed TimeFlow Loss provides insights for training generative foundation models for time series forecasting. We released our pre-trained models that demonstrate notable zero-shot forecasting performance and versatility across different benchmarks. The generative forecasting paradigm enhances the model reliability for decision-making. The millisecond inference speed of our model can be beneficial to many practitioners. Our paper mainly focuses on scientific research and has no obvious negative social impact.

## References

- Aksu, T., Woo, G., Liu, J., Liu, X., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Gift-eval: A benchmark for general time series forecasting model evaluation. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.
- Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.
- Baevski, A. and Auli, M. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018.
- Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosse-lut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- Box, G. Box and jenkins: time series analysis, forecasting and control. In *A Very British Affair: Six Britons and the Development of Time Series Analysis During the 20th Century*, pp. 161–215. Springer, 2013.
- Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.
- Das, A., Kong, W., Leach, A., Sen, R., and Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023a.
- Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023b.
- Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.
- Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.
- Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. *arXiv preprint arXiv:2310.07820*, 2023.
- Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.
- Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730): 1999–2049, 2020.
- Hoo, S. B., Müller, S., Salinas, D., and Hutter, F. The tabular foundation model tabPFN outperforms specialized time series forecasting models based on simple features. *arXiv preprint arXiv:2501.02945*, 2025.
- Hyndman, R. *Forecasting: principles and practice*. OTexts, 2018.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
- Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Kollovich, M., Lienen, M., Lüdke, D., Schwinn, L., and Günemann, S. Flow matching with gaussian process priors for probabilistic time series forecasting. *arXiv preprint arXiv:2410.03024*, 2024.
- Li, T., Tian, Y., Li, H., Deng, M., and He, K. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

- Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S., and Wen, Q. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6555–6565, 2024.
- Lim, B., Arik, S. Ö., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.
- Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T., Lopez-Paz, D., Ben-Hamu, H., and Gat, I. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.
- Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.
- Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023a.
- Liu, Y., Li, C., Wang, J., and Long, M. Koopa: Learning non-stationary time series dynamics with koopman predictors. *arXiv preprint arXiv:2305.18803*, 2023b.
- Liu, Y., Qin, G., Huang, X., Wang, J., and Long, M. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024a.
- Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., and Long, M. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024b.
- Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024c.
- Muñoz-Sabater, J., Dutra, E., Agustí-Panareda, A., Albergel, C., Arduini, G., Balsamo, G., Boussetta, S., Choulga, M., Harrigan, S., Hersbach, H., et al. Era5-land: A state-of-the-art global reanalysis dataset for land applications. *Earth system science data*, 13(9):4349–4383, 2021.
- Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.
- OpenAI, R. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13, 2023.
- Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.
- PEMS. Traffic Dataset. <http://pems.dot.ca.gov/>.
- Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., and Dean, J. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5:606–624, 2023.
- Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.
- Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deep-speed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506, 2020.
- Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.
- Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Biloš, M., Ghonia, H., Hassen, N. V., Schneider, A., et al. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

- Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.
- Shen, L. and Kwok, J. Non-autoregressive conditional diffusion models for time series prediction. In *International Conference on Machine Learning*, pp. 31016–31029. PMLR, 2023.
- Shi, J., Ma, Q., Ma, H., and Li, L. Scaling law for time series forecasting. *arXiv preprint arXiv:2405.15124*, 2024a.
- Shi, X., Wang, S., Nie, Y., Li, D., Ye, Z., Wen, Q., and Jin, M. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024b.
- Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.
- Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- Tashiro, Y., Song, J., Song, Y., and Ermon, S. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.
- Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with mini-batch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- Wen, R., Torkkola, K., Narayanaswamy, B., and Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.
- Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.
- Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.
- Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.
- Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.
- Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

## A. Dataset Statistics

Large-scale datasets are of paramount importance for pre-training foundation models. Recent research has contributed significant time series datasets (Das et al., 2023b; Liu et al., 2024b; Shi et al., 2024b). While the scaling law of time series foundation models has been explored in the recent work (Shi et al., 2024a), the pre-training scale remains relatively limited. Given the heterogeneity of time series compared to other modalities, it raises the question of whether it is feasible to learn from enormous series. To address the question, we curated TimeBench with a trillion time points from various domains.

Unlike other modalities, most time series are unavailable on open websites or repositories. There are also limited domains that encompass typical and predictable time series, leading to slow progress on dataset construction. Therefore, we conducted tedious preprocessing, including missing values imputation, abnormalities exclusion, and normalization techniques. We conducted statistical analysis, examining time series through the lenses of intrinsic properties, e.g., non-stationarity, forecastability, and seasonality. This approach allows us to characterize the data quality inherent to time series, which affects the training stability of next-token prediction. We also adopt synthetic techniques to improve pattern diversity. Further, we adopt ERA5 (Muñoz-Sabater et al., 2021), including systematic real-world temporal observations.

The statistical details of TimeBench are summarized in Table 4. In addition to open-source datasets from research teams on time series foundation models (Woo et al., 2024; Ansari et al., 2024; Liu et al., 2024b;a), we collected substantial real-world time series from various domains such as finance, IoT, meteorology, and healthcare (Goldberger et al., 2000). These resources enable us to construct large-scale time-series corpora exceeding a trillion time points. The corpora include highly credible and predictable data with a wide range of frequencies, lengths, and numbers of variates, providing comprehensive temporal dynamics and variation patterns to facilitate downstream applications. To prevent data leakage, we exclude all datasets evaluated in Section 5.1 to make sure that Sundial conducts zero-shot forecasting.

Table 4. Key statistics of TimeBench, the pre-training dataset of Sundial.

Source	Chronos (2024)	ECG (2000)	Finance (Ours)	IoT (Ours)	LOSTA (2024)	Synthetic (2024)	ERA5 3h (2021)	ERA 12h (2021)	ERA5 Daily (2021)	ERA5 Weekly (2021)	ERA5 Monthly (2021)	ERA5 Quarterly (2021)	Total
# Pts.	94B	48B	10.5B	5.8B	230B	0.5B	129B	32B	406B	58B	13.5B	4.5B	1032B
%	9.11 %	4.65 %	1.02 %	0.56 %	22.29 %	0.05 %	12.50 %	3.10 %	39.35 %	5.62 %	1.31 %	0.44 %	100%

## B. Implementation Details

All experiments are implemented using PyTorch (Paszke et al., 2019) and executed with 32 NVIDIA A100 GPUs. We employ the AdamW optimizer (Kingma & Ba, 2014) for model optimization. We adopt S3 format (Liu et al., 2024b) for univariate pre-training. During training, data from different domains is sampled according to a predefined ratio to balance the domain weightings and ensure diversity in the training data. We implement a global shuffle strategy by loading time series into a standard parquet format. We use variable-wise normalization to unify the scope of values.

On the FEV leaderboard (Ansari et al., 2024), which consists of short-term forecasting datasets, we train Sundial models by TimeFlow Loss with the prediction length of  $F = 16$ . For the point forecasting (Wu et al., 2022) and GIFT-Eval (Aksu et al., 2024), which consist of forecasting datasets with a prediction length ranging from 6 to 900, we train Sundial models by TimeFlow Loss with the prediction length of  $F = 720$ . For the required prediction length less than the model prediction length, we truncate the output generated by Sundial. For the required length more than the prediction horizon, we conduct rolling forecasting. Following Chronos (Ansari et al., 2024), we sample 20 raw predictions to calculate MASE and WQL on FEV. Being consistent to Moirai (Woo et al., 2024), we sample 100 raw predictions to calculate MASE and CRPS for GIFT-Eval. The sampling step is fixed as  $K = 50$ . Configurations of Sundial in different sizes are provided in Table 5. We provide a model summary in Table 6, which summarizes several aspects of current time series foundation models.

## C. Supplementary Results

### C.1. Discussion of Mode Collapse

Mode collapse is a failure of representation learning, where a model generates a limited variety of outputs, ignoring the diversity in the training data. For time series foundation models, mode collapse stems from the heterogeneity of the time series distribution, e.g., a similar lookback time series goes into divergent trending. In other words, the semantics of time

**Sundial: A Family of Highly Capable Time Series Foundation Models**

Table 5. Model configurations of the Sundial family.

Model	Patch Size ( $P$ )	Context Length ( $T$ )	Prediction Length ( $F$ )	Layers ( $L$ )	Dimension ( $D, D_{ff}$ )	MHA Heads $H$	TimeFlow ( $D_{tf}, L_{tf}$ )	Total Parameters #Count
<b>Sundial<sub>Small</sub></b>	16	2880	{16, 720}	6	(512, 2048)	8	(512, 3)	32M
<b>Sundial<sub>Base</sub></b>	16	2880	{16, 720}	12	(768, 3072)	12	(768, 3)	128M
<b>Sundial<sub>Large</sub></b>	16	2880	{16, 720}	24	(1024, 4096)	16	(1024, 6)	444M

\*  $D$  is the embedding dimension of Transformer.  $D_{ff}$  is the hidden dimension of FFN.  $D_{tf}$  is the hidden dimension of the flow-matching network.  $L$  is the layer number of Transformer.  $L_{tf}$  is the layer number of the flow-matching network.

Table 6. Comparison of time series foundation models. *Architecture* denotes the Transformer category. *Model Size* presents parameter counts of different model sizes. *Pre-training Scale* measures pre-training datasets in time points. *Token Level* presents the graininess of time series tokens. *Tokenization* denotes what kind of values are embedded from time series. *Context Length* means the input length supported by the model. *Probabilistic* means generating multiple probable predictions, which is the opposite of deterministic forecasters.

Method	<b>Sundial (Ours)</b>	Time-MoE (2024b)	Timer-XL (2024a)	Moirai (2024)	MOMENT (2024)	LLMTime (2024)	Chronos (2024)	Lag-Llama (2023)	TimesFM (2023b)
Architecture	Decoder	Decoder	Decoder	Encoder	Encoder	Decoder	EncDec	Decoder	Decoder
Model Size	32M	113M	84M	14M	40M	-	46M	200M	17M
	128M	453M		91M	125M		200M		70M
	444M	2.4B		311M	385M		710M		200M
Pre-training Scale	1032B	300B	260B	231B	1.13B	-	84B	0.36B	100B
Token Level	Patch	Point	Patch	Patch	Patch	Point	Point	Point	Patch
Tokenization	Continuous	Continuous	Continuous	Continuous	Continuous	Discrete	Discrete	Continuous	Continuous
Context Length	≤2880	≤4096	≤2880	≤5000	= 512	-	≤512	≤1024	≤512
Probabilistic	True	False	False	True	False	True	True	True	False

series patterns are highly unstable. It sometimes leads to over-smooth predictions from models optimized by MSE because the results are global-optimal for this loss (See showcases on the right of Figure 14-15). Such a training objective pre-defines a unimodal predictive distribution of data, which struggles to accommodate large-scale datasets like TimeBench.

Our work addresses this phenomenon through generative modeling. Generative forecasters learn flexible distributions without relying on probabilistic priors. We evaluate the distributional metric Continuous Ranked Probability Score (CRPS) to assess the quality of generated predictions across different training objectives. The results indicate that the predictive distribution modeled by TimeFlow is more coherent and diverse compared to alternative training objectives, particularly on the highly diverse GIFT-Eval (Aksu et al., 2024). It validates the effectiveness of TimeFlow in mitigating mode collapse.

Table 7. Zero-shot probabilistic forecasting performance using different training objectives. Averaged CRPS is reported here.

Objective	ETTh1	ETTh2	ETTm1	ETTm2	ECL	Weather	GIFT-Eval
TimeFlow Loss	<b>0.0059</b>	<b>0.0037</b>	<b>0.0057</b>	<b>0.0029</b>	0.0082	<b>0.0021</b>	<b>0.5050</b>
Diffusion Loss	0.0082	0.0053	0.0070	0.0039	0.0095	0.0032	0.5340
MSE Loss	0.0063	0.0040	0.0058	0.0032	<b>0.0080</b>	0.0023	0.6420

### C.2. Scaling Behavior Using More Data

We compare Sundial with other time series foundation models that are pre-trained with smaller datasets: Chronos (Ansari et al., 2024) is pre-trained on 94 billion time points, and Moirai is pre-trained on 230 billion time points. As their pre-training datasets are part of the subset of TimeBench, we also conduct pre-training on Sundial using these subsets. As shown in Table 8, these results highlight the scaling behavior of Sundial using larger datasets. Additionally, Sundial still achieves better zero-shot forecasting than its counterpart models with the same pre-training dataset.

Table 8. Zero-shot forecasting performance of models trained on different scales of datasets (measured in time points, pts, and 1B means a billion). We report the averaged results from four prediction lengths {96, 192, 336, 720} on Time-Series-Library (Wu et al., 2022).

Model (pts.)	Chronos (94B)		Moirai (230B)		Sundial (94B)		Sundial (230B)		Sundial (1032B)	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.591	0.468	0.417	<b>0.419</b>	0.402	0.429	0.403	<b>0.419</b>	0.411	0.434
ETTh2	0.405	0.410	0.362	<b>0.382</b>	0.377	0.414	0.364	0.398	<b>0.333</b>	0.387
ETTh1m1	0.645	0.500	0.406	0.385	0.367	0.402	0.352	0.385	<b>0.336</b>	<b>0.377</b>
ETTh1m2	0.310	0.350	0.311	0.337	0.280	0.341	0.273	0.334	<b>0.258</b>	<b>0.320</b>
ECL	0.214	0.278	0.187	0.274	0.172	0.269	0.171	0.267	<b>0.169</b>	<b>0.265</b>
Weather	0.292	0.315	0.287	0.281	0.254	0.301	0.252	0.297	<b>0.234</b>	<b>0.270</b>

### C.3. Performance with Varying Lookback Lengths

Time series foundation models operate independently of training, functioning similarly to statistical methods. Given specific forecasting tasks, one of the most important hyperparameters is the lookback length. Unlike fixed-context models, Sundial offers flexibility for practitioners, allowing the context length to be dynamically adjusted during inference. In Figure 10, we present the performance of Sundial utilizing various lookback lengths. Based on our observations, we contend that performance is largely dependent on the forecasting task itself. Specifically, the size of the lookback window can be tuned to meet the forecasting horizon and data periodicity. Time series foundation models provide a training-free approach for rapid adjustments; still, they should enhance their fundamental long-context capabilities to handle high-frequency data.

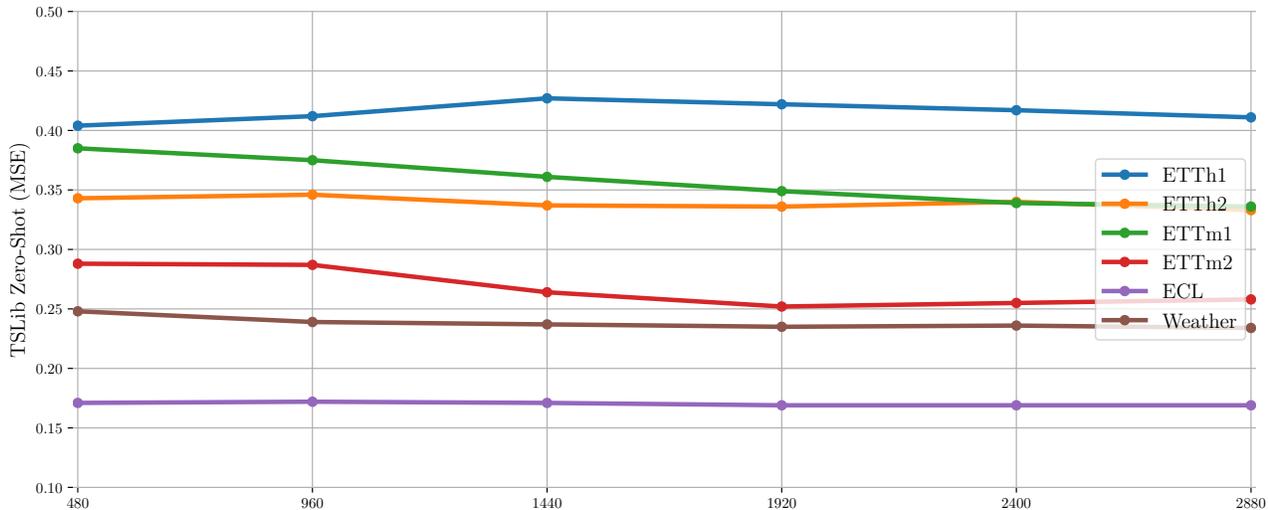


Figure 10. Zero-shot forecasting performance using different lookback lengths in {480, 960, 1440, 1920, 2400, 2880}. We report the averaged results from four prediction lengths {96, 192, 336, 720} on Time-Series-Library (Wu et al., 2022).

#### C.4. Zero-Shot Results of Point Forecasting

Table 9 provides full zero-shot results on Time-Series-Library forecasting benchmark (Wu et al., 2022), including prediction horizons in  $\{96, 192, 336, 720\}$ . We build Sundial with different model sizes with configurations in Table 5. The context length is fixed as 2880. We truncate the model’s predictions for tasks requiring a prediction length less than  $F = 720$ .

We compare the most advanced time series foundation models based on their official checkpoints, including Time-MoE (Shi et al., 2024b), Timer (Liu et al., 2024a;b), Moirai (Woo et al., 2024), TimesFM (Das et al., 2023b), and Chronos (Ansari et al., 2024). We conduct zero-shot evaluations on datasets that are not included during the pre-training of the corresponding models. For each of the evaluated model, we use their maximum input length during inference. Metrics (MSE/MAE) are calculated from all predicted windows in the test split of each dataset following Liu et al. (2024a).

#### C.5. Zero-Shot Results on GIFT-Eval and FEV Leaderboard

We evaluate our models on GIFT-Eval, a benchmark designed to comprehensively assess forecasting performance across diverse time series. GIFT-Eval includes 23 datasets covering 144,000 time series and 177 million data points, which constitute a total of 97 forecasting configurations. We use the official evaluation suite established by the research team of Salesforce and report aggregated results in Table 2. We evaluate the performance and inference time on the FEV leaderboard, which was originally proposed by Ansari et al. (2024) and established by AutoGluon, which comprises 27 datasets for zero-shot evaluation. We report aggregated metrics in Figure 4 and assess the inference time in Figure 5. We released the detailed results by submitting Sundial to their open benchmark<sup>2</sup>.

### D. Showcases

#### D.1. Showcases of Sundial

Figure 11-13 present zero-shot forecasting showcases on all the datasets from FEV (Ansari et al., 2024) and TSLib (Wu et al., 2022). By generating 20 predictions with different initial noise, we estimate the median and 80% prediction interval.

#### D.2. Showcases of Generative Forecasters and Deterministic Forecasters

As we introduce generative modeling in time series foundation models, we compare zero-shot forecasting showcases from two types of models, including (1) Sundial, a generative forecaster pre-trained by TimeFlow, which can predict multiple future possibilities based on a lookback series. (2) Using the same backbone and TimeBench, a Transformer pre-trained by MSE Loss. As a deterministic forecaster, the model can only output the mean prediction. As depicted in Figure 14-15, the unimodal Gaussian prior specified by MSE can be infeasible to handle large-scale pre-training, manifested as sometimes over-smooth predictions in downstream forecasting tasks. Therefore, we hope this work can inspire future paradigms for pre-training time series foundation models and enhance their applicability to real-world scenarios.

### E. Limitations

Our models represent an initial effort to incorporate generative modeling into time series foundation models, which enables pre-training on heterogeneous time series without specifying any prior distribution. This approach mitigates mode collapse in representation learning and generates a diverse range of probable predictions compared to previous deterministic forecasters. Despite significant progress in enlarging model capacity, the Sundial family may still face hallucinations. The performance on very high-frequency data is not guaranteed, since TimeBench contains many middle- and low-frequency time series. Therefore, an important future direction is to generalize Sundial to multi-scale time series. This situation may also indicate new opportunities during inference. As we only adopt a naïve sampling strategy that begins with random Gaussian noise, it leaves much room for future improvement in sampling strategy and post-processing, such as frequency normalization.

Another aspect of future development lies in model adaptation. Sundial is pre-trained in a univariate approach to address the discrepancy in variate numbers, which prevents it from explicitly utilizing variate correlations or covariate information. As an increasing number of studies address 2D dimensionality, multivariate pre-training is likely to be conducted for domain-specific time series foundation models. Lastly, while autoregressive models provide flexibility in the input context length, multiple steps of autoregression may still lead to over-smooth predictions and unreliable results.

<sup>2</sup><https://huggingface.co/spaces/Salesforce/GIFT-Eval>.

Sundial: A Family of Highly Capable Time Series Foundation Models

Table 9. Zero-shot forecasting results of time series foundation models on long-term forecasting datasets (Wu et al., 2022). A lower MSE or MAE indicates a better prediction. Averaged results of four prediction lengths are reported here. 1<sup>st</sup> Count represents the number of wins achieved by a model under all prediction lengths and datasets. Results of baseline models are officially reported by Shi et al. (2024b). Datasets for pre-training are not evaluated on corresponding models, which are denoted by the dash (–).

Models	Sundial <sub>Small</sub>		Sundial <sub>Base</sub>		Sundial <sub>Large</sub>		Time-MoE <sub>Base</sub>		Time-MoE <sub>Large</sub>		Time-MoE <sub>Ultra</sub>		Timer-XL		Moirai <sub>Base</sub>		Moirai <sub>Large</sub>		Chronos <sub>Base</sub>		Chronos <sub>Large</sub>		TimesFM		
	(Ours)		(Ours)		(Ours)		(2024b)		(2024b)		(2024b)		(2024a)		(2024)		(2024)		(2024)		(2024)		(2023b)		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETT <sub>m1</sub>	96	0.292	0.342	<u>0.280</u>	<u>0.334</u>	<b>0.273</b>	<b>0.329</b>	0.338	0.368	0.309	0.357	0.281	0.341	0.317	0.356	0.363	0.356	0.380	0.361	0.454	0.408	0.457	0.403	0.361	0.370
	192	0.337	0.376	0.321	0.366	<u>0.312</u>	<u>0.357</u>	0.353	0.388	0.346	0.381	<b>0.305</b>	<u>0.358</u>	0.358	0.381	0.388	0.375	0.412	0.383	0.567	0.477	0.530	0.450	0.414	0.405
	336	0.370	0.401	<u>0.350</u>	<u>0.389</u>	<b>0.343</b>	<b>0.378</b>	0.381	0.413	0.373	0.408	0.369	0.395	0.386	0.401	0.416	0.392	0.436	0.400	0.662	0.525	0.577	0.481	0.445	0.429
	720	0.418	0.433	<b>0.394</b>	<u>0.418</u>	<u>0.397</u>	<u>0.413</u>	0.504	0.493	0.475	0.477	0.469	0.472	0.430	0.431	0.460	0.418	0.462	0.420	0.900	0.591	0.660	0.526	0.512	0.471
	Avg	0.354	0.388	<u>0.336</u>	<u>0.377</u>	<b>0.331</b>	<b>0.369</b>	0.394	0.415	0.376	0.405	0.356	0.391	0.373	0.392	0.406	0.385	0.422	0.391	0.645	0.500	0.555	0.465	0.433	0.418
ETT <sub>m2</sub>	96	0.178	0.260	<b>0.170</b>	<u>0.256</u>	<u>0.172</u>	<b>0.255</b>	0.201	0.291	0.197	0.286	0.198	0.288	0.189	0.277	0.205	0.273	0.211	0.274	0.199	0.274	0.197	0.271	0.202	0.270
	192	0.235	0.304	<u>0.229</u>	<u>0.300</u>	<b>0.227</b>	<b>0.296</b>	0.258	0.334	0.250	0.322	0.235	0.312	0.241	0.315	0.275	0.316	0.281	0.318	0.261	0.322	0.254	0.314	0.289	0.321
	336	0.287	0.342	<u>0.281</u>	<u>0.337</u>	<b>0.275</b>	<b>0.331</b>	0.324	0.373	0.337	0.375	0.293	0.348	0.286	0.348	0.329	0.350	0.341	0.355	0.326	0.366	0.313	0.353	0.360	0.366
	720	0.360	0.390	<u>0.351</u>	<u>0.387</u>	<b>0.343</b>	<b>0.378</b>	0.488	0.464	0.480	0.461	0.427	0.428	0.375	0.402	0.437	0.411	0.485	0.428	0.455	0.439	0.416	0.415	0.462	0.430
	Avg	0.265	0.324	<u>0.258</u>	<u>0.320</u>	<b>0.254</b>	<b>0.315</b>	0.317	0.365	0.316	0.361	0.288	0.344	0.273	0.336	0.311	0.337	0.329	0.343	0.310	0.350	0.295	0.338	0.328	0.346
ETT <sub>h1</sub>	96	<b>0.341</b>	<u>0.381</u>	0.348	0.385	<u>0.346</u>	0.383	0.357	0.381	0.350	0.382	0.349	<b>0.379</b>	0.369	0.391	0.376	0.392	0.381	0.388	0.440	0.393	0.441	0.390	0.414	0.404
	192	<b>0.381</b>	<u>0.408</u>	0.393	0.418	0.386	0.410	<u>0.384</u>	<b>0.404</b>	0.388	0.412	0.395	0.413	0.405	0.413	0.412	0.413	0.434	0.415	0.492	0.426	0.502	0.524	0.465	0.434
	336	<b>0.405</b>	<u>0.424</u>	0.422	0.440	<u>0.410</u>	0.426	0.411	0.434	0.411	0.430	0.447	0.453	0.418	<b>0.423</b>	0.433	0.428	0.485	0.445	0.550	0.462	0.576	0.467	0.503	0.456
	720	0.433	0.458	0.481	0.493	0.438	0.459	0.449	0.477	<u>0.427</u>	0.455	0.457	0.462	<b>0.423</b>	<b>0.441</b>	0.447	<u>0.444</u>	0.611	0.510	0.882	0.591	0.835	0.583	0.511	0.481
	Avg	<b>0.390</b>	<u>0.418</u>	0.411	0.434	0.395	0.420	0.400	0.424	<u>0.394</u>	0.419	0.412	0.426	0.404	<b>0.417</b>	0.417	0.419	0.480	0.439	0.591	0.468	0.588	0.466	0.473	0.443
ETT <sub>h2</sub>	96	0.272	<u>0.332</u>	<u>0.271</u>	0.333	<b>0.269</b>	<b>0.330</b>	0.305	0.359	0.302	0.354	0.292	0.352	0.283	0.342	0.294	<b>0.330</b>	0.296	<b>0.330</b>	0.308	0.343	0.320	0.345	0.315	0.349
	192	0.329	0.374	<u>0.327</u>	0.376	<b>0.325</b>	<b>0.373</b>	0.351	0.386	0.364	0.385	0.347	0.379	0.340	0.379	0.365	0.375	0.361	<b>0.371</b>	0.384	0.392	0.406	0.399	0.388	0.395
	336	<u>0.357</u>	<u>0.399</u>	<b>0.354</b>	0.402	<b>0.354</b>	0.400	0.391	0.418	0.417	0.425	0.406	0.419	0.366	0.400	0.376	<b>0.390</b>	0.390	<b>0.390</b>	0.429	0.430	0.492	0.453	0.422	0.427
	720	0.401	0.442	<b>0.381</b>	0.435	<u>0.389</u>	0.443	0.419	0.454	0.537	0.496	0.439	0.447	0.397	<u>0.431</u>	0.416	0.433	0.423	<b>0.418</b>	0.501	0.477	0.603	0.511	0.443	0.454
	Avg	0.340	0.387	<b>0.333</b>	0.387	<u>0.334</u>	0.387	0.366	0.404	0.405	0.415	0.371	0.399	0.347	0.388	0.362	<u>0.382</u>	0.367	<b>0.377</b>	0.405	0.410	0.455	0.427	0.392	0.406
ECL	96	0.134	0.231	<u>0.132</u>	<u>0.229</u>	<b>0.130</b>	<b>0.227</b>	-	-	-	-	-	-	0.141	0.237	0.160	0.250	0.153	0.241	0.154	0.231	0.152	<u>0.229</u>	-	-
	192	0.154	0.251	<u>0.152</u>	<u>0.250</u>	<b>0.150</b>	<b>0.247</b>	-	-	-	-	-	-	0.159	0.254	0.175	0.263	0.169	0.255	0.179	0.254	0.172	<u>0.250</u>	-	-
	336	0.174	0.271	<u>0.173</u>	<u>0.271</u>	<b>0.170</b>	<b>0.268</b>	-	-	-	-	-	-	0.177	0.272	0.187	0.277	0.187	0.273	0.214	0.284	0.203	0.276	-	-
	720	<u>0.215</u>	<b>0.307</b>	0.218	0.311	<b>0.214</b>	<b>0.307</b>	-	-	-	-	-	-	0.219	<u>0.308</u>	0.228	0.309	0.237	0.313	0.311	0.346	0.289	0.337	-	-
	Avg	<u>0.169</u>	<u>0.265</u>	<u>0.169</u>	<u>0.265</u>	<b>0.166</b>	<b>0.262</b>	-	-	-	-	-	-	0.174	0.278	0.187	0.274	0.186	0.270	0.214	0.278	0.204	0.273	-	-
Weather	96	<u>0.158</u>	<u>0.206</u>	<b>0.157</b>	<b>0.205</b>	<b>0.157</b>	0.208	0.160	0.214	0.159	0.213	<b>0.157</b>	0.211	0.171	0.225	0.220	0.217	0.199	0.211	0.203	0.238	0.194	0.235	-	-
	192	<b>0.205</b>	<u>0.253</u>	<b>0.205</b>	<b>0.251</b>	<u>0.207</u>	0.256	0.210	0.260	0.215	0.266	0.208	0.256	0.221	0.271	0.271	0.259	0.246	<b>0.251</b>	0.256	0.290	0.249	0.285	-	-
	336	<u>0.254</u>	<u>0.290</u>	<b>0.253</b>	<b>0.289</b>	0.259	0.295	0.274	0.309	0.291	0.322	0.255	<u>0.290</u>	0.274	0.311	0.286	0.297	0.274	0.291	0.314	0.336	0.302	0.327	-	-
	720	<b>0.315</b>	<b>0.336</b>	<u>0.320</u>	<u>0.336</u>	0.327	<u>0.342</u>	0.418	0.405	0.415	0.400	0.405	0.397	0.356	0.370	0.373	0.354	0.337	0.340	0.397	0.396	0.372	0.378	-	-
	Avg	<b>0.233</b>	<u>0.271</u>	<u>0.234</u>	<u>0.270</u>	0.238	0.275	0.265	0.297	0.270	0.300	0.256	0.288	0.256	0.294	0.287	0.281	0.264	0.273	0.292	0.315	0.279	0.306	-	-
1 <sup>st</sup> Count	7	2	<u>8</u>	5	<b>16</b>	<b>16</b>	0	1	0	0	2	1	1	3	0	2	0	<u>6</u>	0	0	0	0	0	0	0

\* Traffic (PEMS) is not evaluated because it is included in the pre-training datasets of these time series foundation models.

## F. Societal Impacts

### F.1. Real-World Applications

In this work, we present Sundial, a family of time series foundation models to facilitate out-of-the-box forecasting. Our models employ native tokenization for continuous-valued time series and incorporate a flexible training objective, proposed as TimeFlow Loss, to enable probabilistic forecasting. With an unprecedented model capacity and a trillion-scale dataset, our models can be used directly or adapted for various forecasting scenarios, such as energy planning, weather forecasting, and financial risk prevention. With multiple predictions generation and a just-in-time inference speed, our model enhances the reliability of decision-making and streamlines the forecasting pipeline for practitioners. This paper primarily focuses on scientific research and does not present any evident negative social impact.

### F.2. Academic Research

We curate TimeBench, a trillion-level time series dataset for pre-training foundation models for time series analysis, which we believe will be beneficial to the research community. Technically, we propose a TimeFlow Loss to facilitate the learning of flexible next-patch distributions. Conditioned on the lookback representations acquired by autoregressive Transformers, our model is endowed with a novel generative capability for probabilistic forecasting, enhancing representation learning of Transformers without the need for discrete tokenization. Through pre-training on an unprecedented scale, we identify subtle scalability bottlenecks that are not solely attributable to architectural design but are predominantly influenced by the training objectives of foundation models. The proposed TimeFlow Loss applied to autoregressive and generative models may provide insights for the future development of time series foundation models.

# Sundial: A Family of Highly Capable Time Series Foundation Models

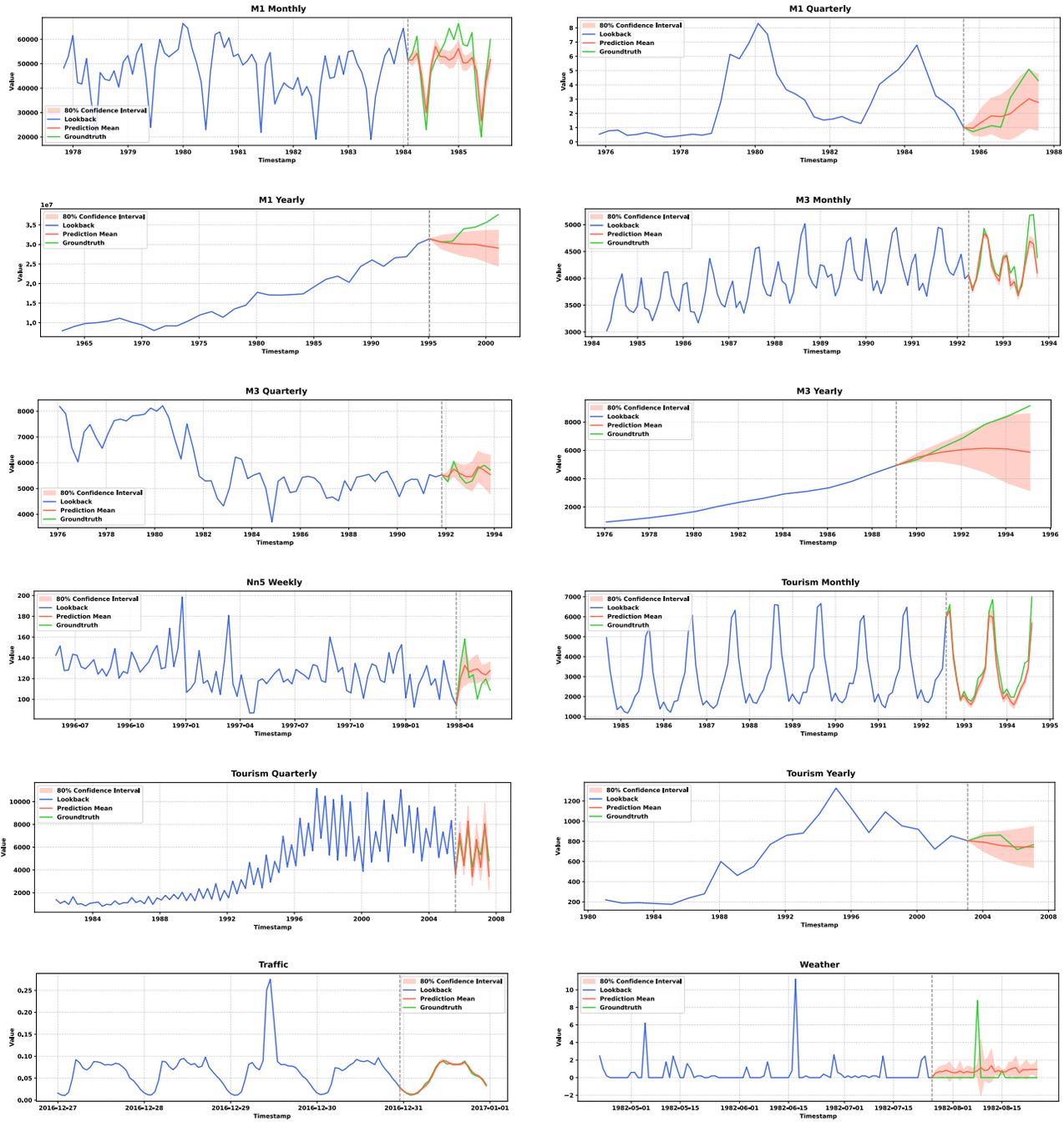


Figure 11. Showcases of zero-shot predictions from Sundial (Base) on the FEV leaderboard (Ansari et al., 2024).

# Sundial: A Family of Highly Capable Time Series Foundation Models

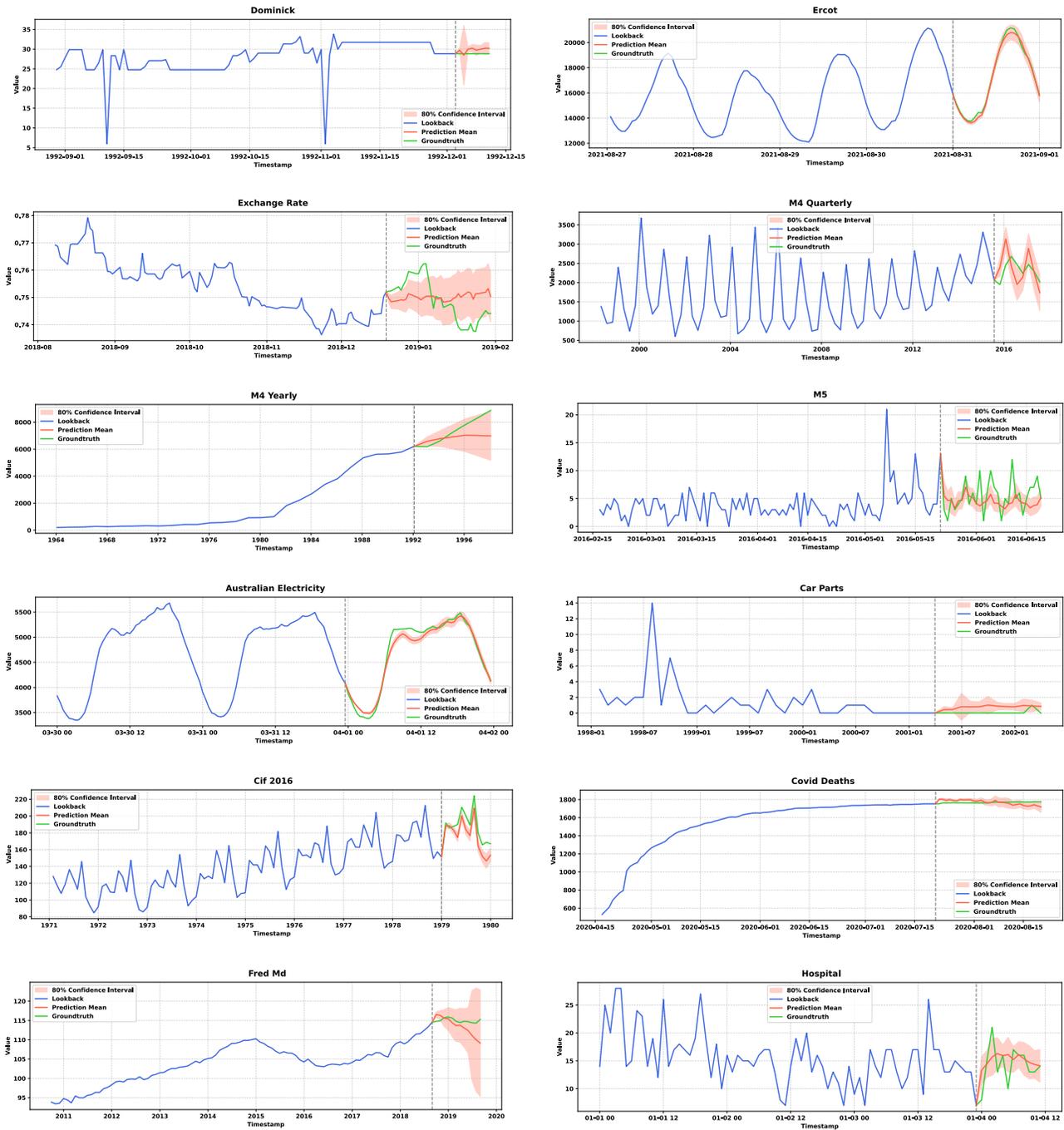


Figure 12. Showcases of zero-shot predictions from Sundial (Base) on the FEV leaderboard (Ansari et al., 2024).

## Sundial: A Family of Highly Capable Time Series Foundation Models

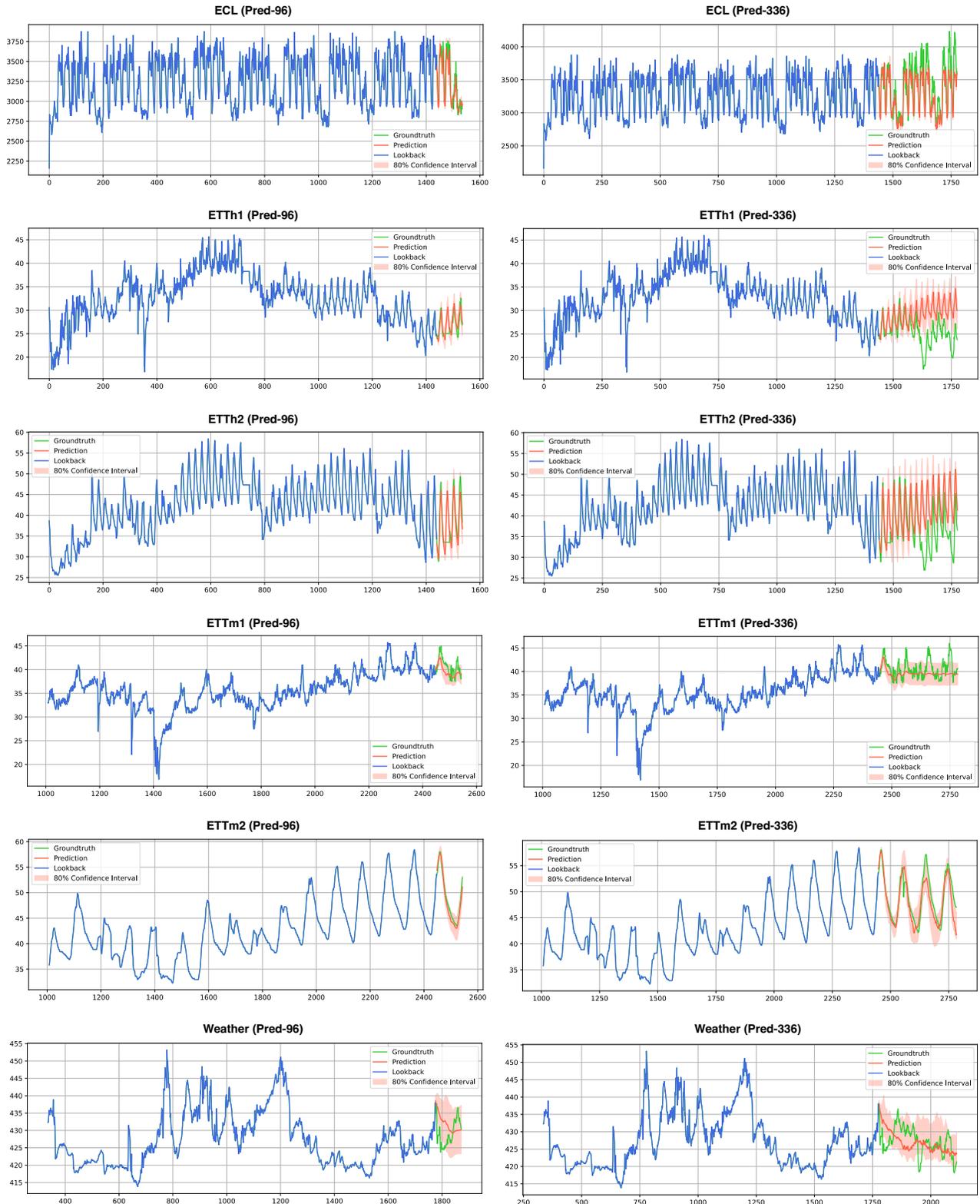


Figure 13. Showcases of zero-shot predictions from Sundial (Base) on long-term forecasting datasets (Wu et al., 2022).

## Sundial: A Family of Highly Capable Time Series Foundation Models

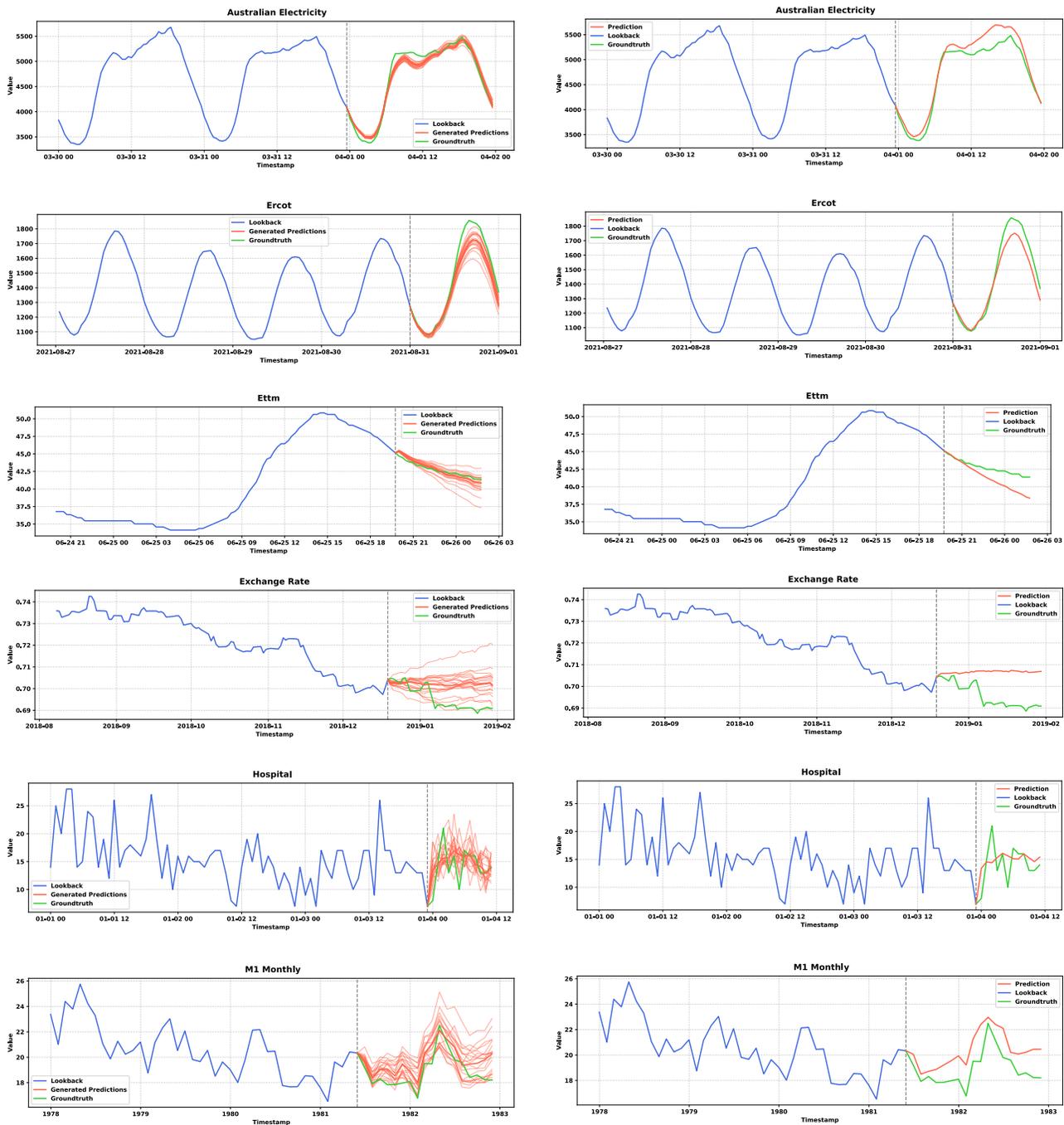


Figure 14. Showcases of Sundial (Left) and the same Transformer backbone pre-trained by MSE Loss (Right). MSE Loss optimizes a deterministic forecaster: given a lookback series, the model can only produce one prediction as the estimation of mean values. This objective may fail to accommodate divergent future variations during large-scale pre-training, leading to mode collapse and over-smooth results (as illustrated in the fourth row). TimeFlow optimizes a generative forecaster: it generates various possibilities observed in the pre-training dataset. Based on these raw predictions, we can estimate the underlying complicated distribution and different statistics. Besides, the greater concentration in generated predictions, the higher the model’s confidence in its predictions.

# Sundial: A Family of Highly Capable Time Series Foundation Models

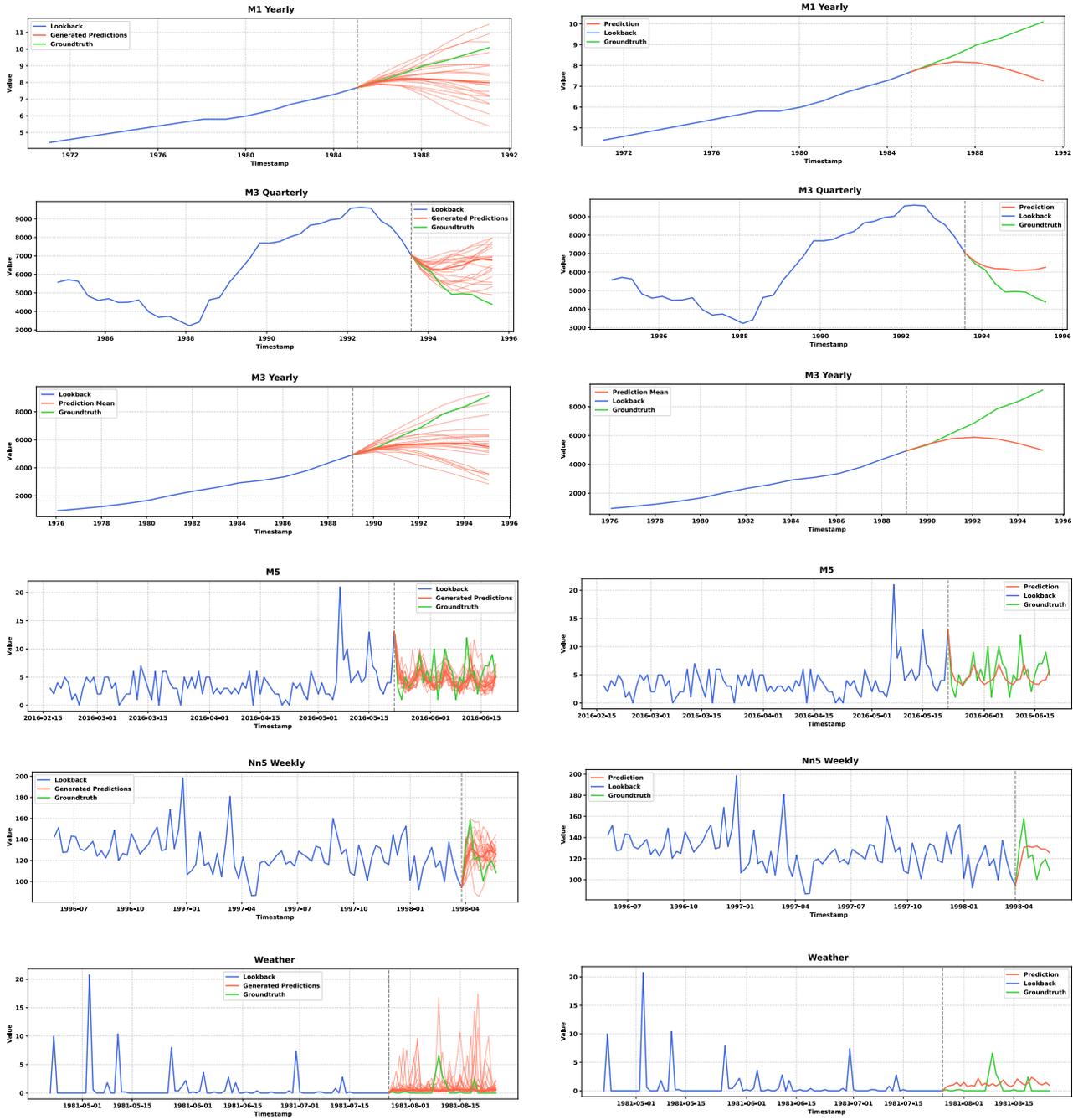


Figure 15. Supplementary showcases of Sundial (Left) and the same Transformer backbone pre-trained by MSE Loss (Right).