LEAF: Large Language Diffusion Model for Time Series Forecasting

Yuhang Pei[1] **Tao Ren**[1] **Yifan Wang**[2*] **Zhipeng Sun**[1] **Wei Ju**[3]
Chong Chen[4] **Xian-Sheng Hua**[4] **Xiao Luo**[5]
[1]Software College, Northeastern University
[2]University of International Business and Economics [3]Peking University
[4]Terminus Group [5]University of Wisconsin–Madison
{2210540,2410599}@stu.neu.edu.cn, chinarentao@163.com
yifanwang@uibe.edu.cn juwei@pku.edu.cn
{chenchong.cz,huaxiansheng}@gmail.com xiao.luo@wisc.edu

Abstract

This paper studies the problem of time series forecasting, which aims to generate future predictions given historical trajectories. Recent researchers have applied large language models (LLMs) into time series forecasting, which usually align the time series space with textual space and output future predictions with strong autoregressive reasoning abilities. Despite their remarkable progress, these approaches usually lack an understanding of holistic temporal patterns with potential error accumulation. Towards this end, this paper proposes a simple yet effective framework that marries **L**arg**e** Langu**a**ge Diffusion Model with time series **f**orecasting (LEAF). The core of our framework is to generate future predictions with a diffusion model from a holistic view. In particular, we first introduce a tokenization module to convert time series into tokens and then adopt the language diffusion models to capture the temporal dependencies. In this way, we can transform masked time series into all the predictions with the remasking strategy. Extensive experiments on various benchmark datasets validate the effectiveness of the proposed LEAF in comparison to various baselines.

1 Introduction

Time series forecasting (TSF) assumes a critical role in various domains, including finance (Deb et al., 2017), healthcare (Chimmula and Zhang, 2020), climate science (Pathak et al., 2022), and traffic prediction (Cirstea et al., 2022; Zhao et al., 2023). To achieve effective TSF, traditional methods like ARIMA (Box and Pierce, 1970) and exponential smoothing (ETS) (Gardner Jr, 1985) have been widely adopted for capturing temporal dependencies and trend patterns in time series data. Thanks to recent advances in deep learning, models such as recurrent neural networks (RNNs) (Salinas et al., 2020), convolutional neural networks (CNNs) (Wu et al., 2023), and Transformers (Wu et al., 2021) are highly effective in identifying intricate dynamics and long-range persistence in high-dimensional time series data. In practice, these TSF methods often rely on extensive domain expertise and task-specific designs. Meanwhile, real-world applications such as weather and financial forecasting require extrapolation from sparse observations, further complicating the task of making accurate predictions (Dooley et al., 2023).

Recently, large language models (LLMs), such as GPT (Brown et al., 2020) and Llama (Touvron et al., 2023a), have shown remarkable capabilities in capturing contextual dependencies in natural language. Since both language and time series data involve sequential structures and rely on learned token transitions, there is a growing interest in adapting off-the-shelf LLMs for time series forecasting, especially under few-shot and zero-shot settings. For example, LLMTime encodes time series data into sequences of numerical tokens and formulates TSF as a next-token prediction task (Gruver et al., 2023). AutoTimes projects time series data into the latent space of language tokens and generates future predictions in an autoregressive manner (Liu et al., 2024c). LSTPrompt further decomposes TSF into short- and long-term subtasks and formalizes the prediction into the Chain-of-Thought process (Liu et al., 2024b).

Despite the success of these LLM-based methods for TSF, we argue that the autoregressive generation framework is not inherently aligned with the nature of time series data, primarily due to two major challenges. ❶ *Limited understanding of the holistic temporal patterns.* Future intervals in time series often exhibit coherent global patterns, such as seasonality, trends, and periodic behaviors (Cao et al., 2024), which cannot be effectively captured through token-by-token prediction. ❷ *Error propagation and lack of internal consistency.* The autoregressive generation always suffers from error

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1076–1091
November 4-9, 2025 ©2025 Association for Computational Linguistics

accumulation as each prediction relies on previous ones, leading to increasingly distorted forecasts. This disrupts the internal consistency of the predicted sequence and further hinders the ability to capture global temporal patterns accurately. These limitations naturally prompt the question: *Can we design a TSF framework that captures the entire future trajectory, rather than predicting it step-by-step in an autoregressive manner?*

Diffusion models have emerged as an alternative to traditional autoregressive generative frameworks. Originally introduced for image generation (Ho et al., 2020; Song et al., 2021), diffusion models iteratively transform random noise into structured data via injecting noise in a forward process and learning to reverse it to recover the original distribution. In TSF tasks, diffusion models provide a key advantage over autoregressive methods by learning and predicting the entire trajectory simultaneously (Tashiro et al., 2021; Yuan and Qiao, 2024; Nie et al., 2025). LLaDA provides an initial attempt at the large language diffusion model, though it still remains limited to TSF (Nie et al., 2025). Therefore, the use of diffusion models for zero-shot TSF with LLMs presents a promising yet largely unexplored direction.

Towards this end, we propose a simple yet effective approach **L**arg**e** Langu**a**ge Diffusion Model for time series **f**orecasting (LEAF), which leverages a masked diffusion model incorporating a discrete random masking process along with a mask predictor to approximate the reverse process. Specifically, given the time series input, a dedicated tokenization module is first employed to transform the raw data into a sequence of discrete tokens, thereby enabling effective interaction with the LLMs. Then, a diffusion framework is employed to guide the LLMs' distribution, constructing a forward masking and a subsequent reverse process to capture temporal dependencies. For the inference phase, starting with the fully masked time series input, we discretize the reverse process and apply a confidence-based remasking strategy for the iterative TSF.

The contributions of this paper are as follows:
❶ *New Perspective.* We are the first to introduce the large language diffusion framework for TSF task, which captures the entire temporal trajectory and dependencies of time series data without relying on traditional autoregressive methods.
❷ *In-depth Analysis.* We formalize the model distribution through a forward masking process followed by a reverse process, and provide an in-depth analysis for the inference phase initialized from the fully masked input in TSF.
❸ *State-of-the-art Performance.* We conduct extensive experiments on several publicly available time series datasets and experimental results show that our LEAF significantly outperforms existing comparative methods.

2 Related Work

2.1 Time Series Forecasting

Time series forecasting has progressed from classical models like ARIMA (Box and Pierce, 1970) and ETS (Gardner Jr, 1985)—valued for interpretability but limited with complex, high-dimensional data—to deep learning approaches (Oord et al., 2016; Bai et al., 2018; Salinas et al., 2020; Wu et al., 2021, 2023) that capture hierarchical and non-linear temporal patterns. Transformer-based models such as Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and PatchTST (Nie et al., 2023) leverage self-attention to model long-range dependencies efficiently. However, the emergence of simple linear models such as LTSF-Linear (Zeng et al., 2023) highlights the underutilization of Transformers in TSF, thereby motivating researchers to explore enhancements across multiple dimensions, including cross-dimension attention (Zhang and Yan, 2023), patching techniques (Nie et al., 2023), integration of exogenous variables (Wang et al., 2024), and improvements in generalization and multi-scale modeling (Ilbert et al., 2024; Shabani et al., 2022). Meanwhile, diffusion models (Yuan and Qiao, 2024; Tashiro et al., 2021; Nie et al., 2025; Gao et al., 2025) have shown promise in capturing complex distributions for high-quality forecasts.

Despite their success, these models often require large labeled datasets and struggle with cross-domain generalization. Pre-trained models like Timer (Liu et al., 2024d) offer more generalizable representations, but enhancing data efficiency and domain adaptability remains an open challenge.

2.2 Large Models for Time Series Data

The success of foundation models in vision and language (Touvron et al., 2023b; Liu et al., 2024a; Li et al., 2023; Achiam et al., 2023; Radford et al., 2021) motivates extending their capabilities to time series. However, time series data present challenges due to domain variability and difficulties in large-scale collection arising from privacy and

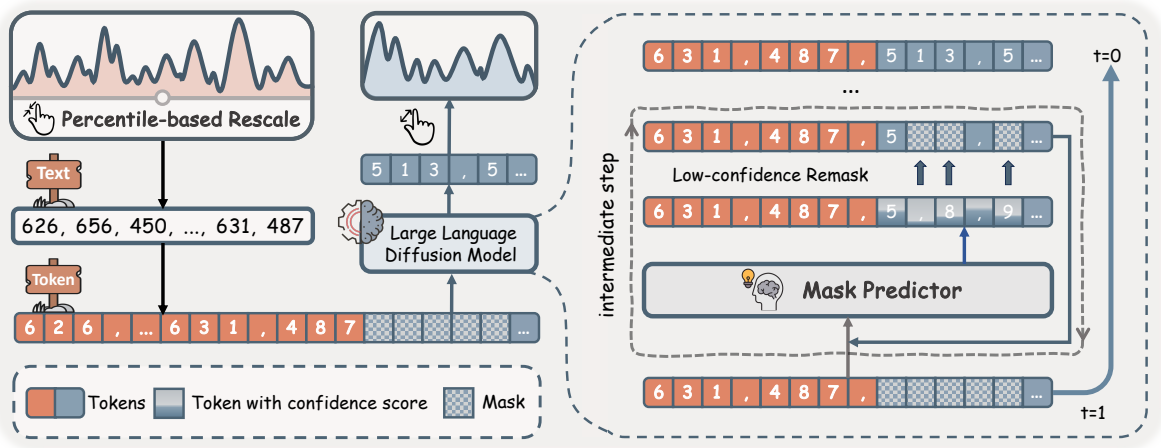


Figure 1: Overview of the proposed LEAF framework. Historical time series data (orange) is first rescaled and concatenated with a masked segment representing the prediction horizon (blue). Through iterative denoising, the masked tokens are progressively generated and rescaled back to the original scale to produce the final forecast.

cost constraints. These challenges have led to two primary research directions: 1) adapting LLMs to time series tasks, including fine-tuning (Zhou et al., 2023; Chang et al., 2025), reformulating numerical sequences as textual prompts (Xue and Salim, 2023), employing string-encoded sequences for zero-shot forecasting (Gruver et al., 2023), and using patch-based modality alignment (Jin et al., 2023); and 2) developing foundation models trained directly on time series data, such as constructing large-scale corpora for general-purpose forecasting (Rasul et al., 2023; Das et al., 2024) or using NLP-inspired tokenization to parse time series into semantic units (Ansari et al., 2024).

Nevertheless, current methods largely depend on autoregressive models. This approach is inherently constrained by the "reversal curse", limiting them to modeling forward temporal dependencies and struggling with bidirectional or global temporal patterns, thus restricting full exploitation of temporal structure. We introduce a novel Diffusion LLM framework for TSF. By leveraging a denoising diffusion process, our approach facilitates iterative prediction refinement and the capture of complex, bidirectional temporal dependencies.

3 The Proposed LEAF

In this paper, we first formalize the problem setting and then introduce the details of our proposed LEAF, which investigates the large language diffusion framework for zero-shot TSF. In particular, our framework consists of three components. Given the time series data, it is recognised as a string of numerical digits, and each digit is viewed as a discrete token. Then, we define a diffusion framework via a model distribution with forward masking and subsequent reverse process. Finally, for the inference phase, we employ a confidence-based remasking strategy for iteratively TSF. The overview of the framework is shown in Figure 1, with the details of each component presented as follows.

3.1 Problem Definition

Time series forecasting task aims to predict future values of a given time series based on its historical data. Formally, given a time series input $\boldsymbol{X}_{1:L} = [x_1, x_2, \ldots, x_L]$, where L denotes the length of the look-back window, the objective is to forecast the future values $\boldsymbol{X}_{L+1:L+H} = [x_{L+1}, x_{L+2}, \ldots, y_{L+H}]$ over a prediction horizon of length H. In the zero-shot setting, we leverage the generalization abilities of pretrained LLMs and build a foundation model $\mathcal{F}^*(\cdot)$ to directly map the look-back window with horizon:

$$\boldsymbol{X}_{L+1:L+H} = \mathcal{F}^*(\boldsymbol{X}_{1:L}). \tag{1}$$

3.2 Times Series Tokenization

To enable LLMs to process numerical time series data, we first transform the continuous input sequence into a discrete tokenized format. Standard tokenization methods, such as Byte Pair Encoding (BPE), often fragment numerical values into tokens inconsistent with the digit (Sennrich et al., 2016). Inspired by the tokenization strategy of LLaMA (Touvron et al., 2023a), LLMTime adopts a digit-level tokenization, where individual digit is separated by spaces and commas are utilized to

demarcate distinct time steps within the time series input (Gruver et al., 2023). For example:

$$0.345,\ 3.45,\ 34.5 \rightarrow \text{“3 4 , 3 4 5 , 3 4 5 0”}.$$

In our LEAF, we implement a percentile-based rescaling strategy (Gruver et al., 2023) to reduce the token consumption caused by large magnitudes. Specifically, each time series value is first offset by subtracting the β-percentile of the original series, and then scaled such that the α-percentile of the adjusted values is normalized to 1. So in this way, the majority of values fall within a manageable range while retaining statistical properties for accurate TSF. The details of the scaling strategy are in Appendix C.

3.3 Large Language Diffusion Model

In contrast to conventional autoregressive forecasting approaches, diffusion models (Ho et al., 2020; Tashiro et al., 2021; Ou et al., 2025) provide a probabilistic framework that models the data generation process via a forward noising process and its corresponding trained reverse denoising process, which could capture the complex nonlinear trajectory inherent in time series data. To integrate diffusion guidance into LLMs for TSF, discrete masking (Austin et al., 2021) is employed with the random masking ratio t sampled uniformly from $[0, 1]$ in a large language diffusion model (Nie et al., 2025). The iterative process for each token $x_i \in \boldsymbol{X}$ from $t = 0$ (original input) to $t = 1$ (fully masked) can be defined as:

$$q(x_i^t|x_i^0) = \text{Cat}(x_i^t; t\delta(x_i^0) + (1-t)\delta(\text{M})), \quad (2)$$

where $\text{Cat}(\cdot)$ is a categorical distribution with Dirac function $\delta(a) = \mathbb{I}(x_i^t = a)$, denoting the probability t of the token being masked, and M denote the special [MASK] token. Based on this, the reverse process inverts the noising process defined by q, where the denoising process can be:

$$p(x_i^0|x_i^t) = \begin{cases} 1, & x_i^t \neq \text{M}, \\ p(x_i^0|x_{\text{UM}}^t), & x_i^t = \text{M}. \end{cases} \quad (3)$$

where x_{UM}^t denotes the collection of unmasked tokens in the forward noising process.

3.4 Remarking Inference for Time Series Forecasting

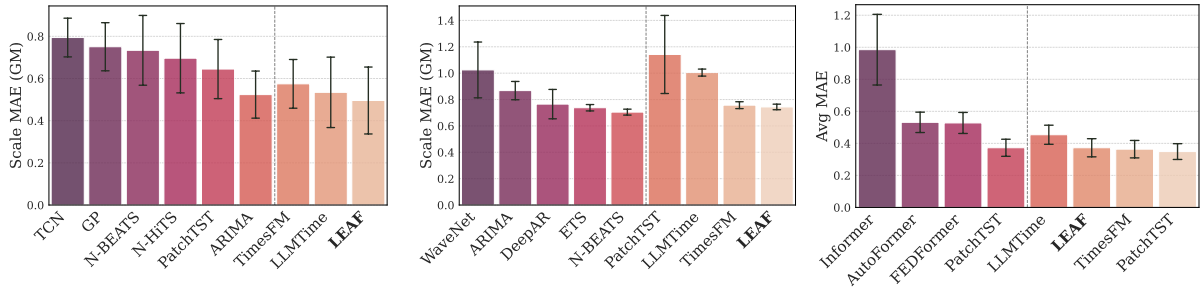
We evaluate the zero-shot ability of a large language diffusion model on TSF. Starting from the input $\boldsymbol{X}_{1:L}$ and fully masked $\boldsymbol{X}_{L+1:L+H}$, we predict the masked token and employ a remasking strategy for the reverse process. Specifically, we leverage a pretrained model (Nie et al., 2025) by feeding both $\boldsymbol{X}_{1:L}$ and fully masked $\boldsymbol{X}_{L+1:L+H}$ to predict all the masked token simultaneously. Then, we iteratively remask the $\boldsymbol{X}_{L+1:L+H}$ by transitioning from an intermediate step $t \in (0, 1]$ to a lower step $s \in [0, t)$, ensuring that the reverse inference process remains consistent with the formalized probabilistic model. Instead of random masking for step t, the predicted probability of each token is regarded as a confidence score and we remask out s/t of tokens with the lowest confidence in large language diffusion model (Nie et al., 2025). The masking ratio progressively decreases over iterations until all tokens are generated within T steps.

4 Experiments

4.1 Experimental Setup

Datasets and Baselines. To assess LEAF's performance, we evaluate it on three benchmark datasets: Darts, Monash, and Informer, which are widely used in time-series forecasting and span diverse domains and frequencies. A summary of these datasets is provided in Appendix A, with detailed descriptions presented below.

- **Darts Collection** (Herzen et al., 2022). For the Darts datasets, we evaluate LEAF against a comprehensive set of supervised and zero-shot baselines. The supervised methods include classical statistical models such as SM-GP (Wilson and Adams, 2013) and ARIMA (Box and Pierce, 1970), as well as modern deep learning approaches: Temporal Convolutional Networks (TCN) (Bai et al., 2018), N-BEATS (Oreshkin et al., 2019), N-HiTS (Challu et al., 2023), and PatchTST (Nie et al., 2023). For zero-shot forecasting, we include results from LLMTime (Gruver et al., 2023) and TimesFM (Das et al., 2024). This diverse set of baselines enables a thorough evaluation of LEAF's performance.
- **Monash Archive** (Godahewa et al., 2021). After filtering out datasets affected by missing values, 14 collections are retained for our evaluation. We benchmark our approach against classical statistical methods including ARIMA (Box and Pierce, 1970) and ETS (Gardner Jr, 1985), deep learning baselines including WaveNet (Oord et al., 2016), N-BEATS (Oreshkin et al., 2019), and DeepAR (Salinas et al., 2020). For zero-shot models, we use the same baselines as in the Darts



(a) Darts (b) Monash (c) Informer

Figure 2: Average performance of LEAF across the three dataset groups. The lower the scaled MAE the better. Error bars denote the standard error across all datasets.

Table 1: MAE on the Darts collection. We also report a naive baseline that repeatedly predicts the last observed value. The best zero-shot and supervised results are highlighted in **bold** and underlined, respectively.

		Supervised						Zero-Shot		
Dataset	NAIVE	GP	ARIMA	TCN	N-BEATS	N-HiTS	PatchTST	LLMTime	TimesFM	LEAF
AirPassengersDataset	81.45	34.67	24.03	54.96	97.89	59.16	44.65	34.37	62.51	**23.01**
AusBeerDataset	96.35	102.05	17.13	30.90	10.39	34.23	21.97	24.52	**11.94**	22.25
GasRateCO2Dataset	2.29	2.27	2.37	2.64	2.63	3.85	2.67	3.50	**2.50**	3.03
MonthlyMilkDataset	85.71	30.33	37.19	70.86	33.64	32.73	42.60	12.53	28.09	**8.00**
SunspotsDataset	48.24	53.74	43.56	51.82	73.15	49.93	62.33	47.34	**41.40**	44.83
WineDataset	4075.28	4552.06	2306.70	3287.14	4562.02	3909.51	2498.69	**1632.79**	2871.33	2505.76
WoolyDataset	1210.33	649.98	588.78	1158.79	903.01	382.09	542.28	812.07	**728.92**	880.63
HeartRateDataset	5.92	5.65	5.56	5.49	6.57	6.10	6.74	6.21	**5.85**	6.83
MSAE (AM)	1.00	0.82	0.60	0.84	0.92	0.81	0.74	0.68	0.68	**0.67**
MSAE (GM)	1.00	0.75	0.52	0.79	0.73	0.69	0.64	0.53	0.58	**0.50**

experiments, with PatchTST (Nie et al., 2023) additionally included as a zero-shot baseline.

- **Informer Collection** (Zhou et al., 2021). We employ a context window of 512 time steps and evaluate prediction horizons of 96 and 192 steps. We benchmark LEAF against a set of strong supervised and zero-shot baselines. The supervised methods include PatchTST (Nie et al., 2023), FEDFormer (Zhou et al., 2022), AutoFormer (Wu et al., 2021), and Informer (Zhou et al., 2021), which represent state-of-the-art deep learning approaches for long sequence time series forecasting. For zero-shot forecasting, the baselines from the Monash experiments are employed, including PatchTST, LLMTime (Gruver et al., 2023), and TimesFM (Das et al., 2024).

Metrics. Following standard evaluation practices, we report Mean Scaled Absolute Error (MSAE) metrics on Darts and Monash, normalized by a naive baseline that propagates the last observed value of each context window. Both Arithmetic Mean (AM) and Geometric Mean (GM) are used to aggregate MSAE across datasets, ensuring robust and equitable comparisons for time series with heterogeneous scales. For Informer, we report the Mean Absolute Error (MAE) averaged across all eight tasks (four datasets $\times$ two horizons). Further details are provided in Appendix D.1.

Implementation Details. We initialize LEAF with the publicly available pretrained LLaDA-base model weights (Nie et al., 2025), which correspond to an 8B-parameter diffusion LLM. For all datasets, we adopt a unified inference configuration: the low-confidence remask strategy is employed; the block length is set to 1.2 times the estimated token count of the target sequence to mitigate incomplete generation due to underestimated sequence length; the denoising steps are configured to be half of the block size, resulting in approximately 2 tokens being generated per step; and the classifier-free guidance (CFG) scale (Ho and Salimans, 2022) is set to 2.5 to strengthen the influence of the conditional context on the generated outputs. The numerical time-series data are directly tokenized and subsequently fed into the model without adding any extra prompts or instructions. The detail of the CFG scale is provided in the appendix D.4.

4.2 Zero-Shot Time Series Forecasting

We first report the performance of LEAF on the datasets without any task-specific training. Figure.

Table 2: MAE for Monash collection. The best results for zero-shot and supervised methods are indicated in **bold** and underlined, respectively.

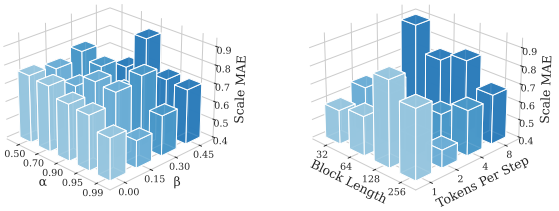
		Supervised					Zero-Shot			
Dataset	NAIVE	ETS	ARIMA	DeepAR	N-BEATS	WaveNet	PatchTST	LLMTime	TimesFM	LEAF
bitcoin	7.77e17	1.10e18	3.62e18	1.95e18	1.06e18	2.46e18	**1.11e18**	1.75e18	1.3e18	1.18e18
pedestrian counts	170.88	216.50	635.16	44.78	66.84	46.46	51.27	70.20	**40.71**	47.45
nn5 daily	8.26	3.72	4.41	3.94	4.92	3.97	3.77	9.39	**3.54**	5.56
nn5 weekly	16.71	15.70	15.38	14.69	14.19	19.34	17.00	15.91	**14.67**	17.80
tourism yearly	99456.05	94818.89	95033.24	71471.29	70951.80	69905.47	224411.89	140081.78	109977.29	**99293.90**
tourism quarterly	15845.10	8925.52	10475.47	9511.37	8640.56	9137.12	21276.98	14121.09	12102.04	**10424.06**
tourism monthly	5636.83	2004.51	2536.77	1871.69	2003.02	2095.13	4596.21	4724.94	3183.77	**2923.24**
cif 2016	386526.37	642421.42	469059.49	3200418.00	679034.80	5998224.62	8374813.14	715086.33	773980.44	**566806.36**
covid deaths	353.71	85.59	85.77	201.98	158.81	1049.48	348.60	304.68	209.80	**124.36**
traffic weekly	1.19	1.14	1.22	1.18	1.11	1.20	1.23	1.17	1.12	**0.96**
saugeenday	21.50	30.69	22.38	23.51	27.92	22.17	**22.33**	28.63	24.63	22.51
us births	1152.67	419.73	526.33	424.93	422.00	504.40	1193.28	459.43	437.27	**408.03**
hospital	24.07	17.97	19.60	18.25	20.18	19.35	20.87	24.62	**19.41**	28.09
solar weekly	1729.41	1131.01	839.88	721.59	1172.64	1996.89	**1093.46**	2049.09	1258.27	1480.23
MSAE (AM)	1.0000	0.86	1.23	1.30	0.80	2.12	2.49	1.11	0.87	**0.84**
MSAE (GM)	1.0000	0.74	0.87	0.77	0.70	1.02	1.14	1.00	0.76	**0.74**

2 shows the average performance of LEAF in three groups of datasets. The detailed results on the three groups of datasets are shown in Table 1, Table 2, and Table 3. From the results, we can obtain the following observations:

Observation 1. **LEAF** ***demonstrates highly competitive performance in TSF.*** Analysis of the experimental results shown in Figure 2 reveals that LEAF consistently achieves state-of-the-art or near state-of-the-art results among zero-shot approaches. Notably, its performance frequently surpasses that of established supervised methods, highlighting its strong generalization capability without requiring any task-specific fine-tuning.

Observation 2. Diffusion-based model enables effective global pattern modeling in seasonal time series. Examining performance on datasets known for distinct seasonality within the Darts collection, such as AirPassengers and MonthlyMilk, LEAF achieves excellent results. This suggests that the underlying diffusion model, which generates the entire forecast horizon potentially simultaneously rather than step-by-step, is adept at modeling and extrapolating the holistic, repeating patterns characteristic of seasonal data.

Observation 3. **LEAF** ***maintains robust performance and mitigates error accumulation in long-term forecasting.*** On the ETT datasets, it achieves results comparable to the leading methods across all prediction horizons. Although it is not always the top performer, its average MAE remains highly competitive. This indicates that the diffusion-based generation process is less prone to the compounding errors that typically affect autoregressive models over longer horizons, enabling LEAF to generate coherent and accurate long-term forecasts.



(a) α and β (b) Steps and Block Length

Figure 3: Parameter sensitivity analysis of LEAF on the Darts datasets. The left plot shows the sensitivity of α and β, while the right plot shows the sensitivity of denoising steps and block length.

4.3 Ablation Study

We further conduct an ablation study on Darts to analyze the impact of different components of LEAF. We consider the following components: (1) LEAF w/o rescaling strategy, which uses the minmax scaler to scale the time series data; (2) LEAF w/o offset, which does not use the offset strategy; (3) LEAF w/ semi-autoregressive (See Appendix D.3), which uses the semi-autoregressive strategy to predict the time series data; (4) LEAF w/o CFG scale, which uses the original logit as the final probability distribution. The results presented in Table 4 lead to the following observations.

Observation 4. Percentile-based rescaling with offset is essential for robust zero-shot generalization. Time series datasets exhibit significant heterogeneity in terms of scale, periodicity, and underlying patterns. Removing the specific percentile-based rescaling strategy and offset component leads to a worse overall performance. While the removal might coincidentally yield better results on a few

Table 3: MAE for Informer collection with prediction horizons 96 and 192. For consistency with LLMTime and TimesFM, we report results on the last window of the original test split. For each dataset, MAE is computed by aggregating the results from separate predictions for each column.

	Supervised				Zero-Shot			
Dataset	PatchTST	FEDFormer	AutoFormer	Informer	LLMTime	PatchTST	TimesFM	LEAF
ETTh1 (h=96)	0.41	0.58	0.55	0.76	0.42	**0.39**	0.45	0.41
ETTh1 (h=192)	0.49	0.64	0.64	0.78	**0.50**	**0.50**	0.53	0.61
ETTh2 (h=96)	0.28	0.67	0.65	1.94	0.33	0.37	0.35	**0.32**
ETTh2 (h=192)	0.68	0.82	0.82	2.02	0.70	**0.59**	0.62	**0.59**
ETTm1 (h=96)	0.33	0.41	0.54	0.71	0.37	0.24	0.19	**0.17**
ETTm1 (h=192)	0.31	0.49	0.46	0.68	0.71	**0.26**	**0.26**	0.38
ETTm2 (h=96)	0.23	0.36	0.29	0.48	0.29	**0.22**	0.24	0.23
ETTm2 (h=192)	0.25	0.25	0.30	0.51	0.31	**0.22**	0.27	0.27
Avg	0.37	0.53	0.53	0.99	0.45	**0.35**	0.36	0.37

Table 4: Ablation study on the Darts datasets. The red and blue numbers indicate the performance drop and improvement compared to LEAF, respectively. We bold the best results.

Dataset	NAIVE	LEAF	w/o rescaling strategy	w/o offset	w/ semi-autoregressive	w/o CFG scale
AirPassengersDataset	81.45	**23.01**	39.45$_{\uparrow 16.44}$	26.97$_{\uparrow 3.96}$	57.77$_{\uparrow 34.76}$	30.79$_{\uparrow 7.78}$
AusBeerDataset	96.35	22.25	32.88$_{\uparrow 10.63}$	23.92$_{\uparrow 1.67}$	**13.98**$_{\downarrow 8.27}$	14.27$_{\downarrow 7.98}$
GasRateCO2Dataset	2.29	**3.03**	5.38$_{\uparrow 2.35}$	4.86$_{\uparrow 1.83}$	3.37$_{\uparrow 0.34}$	5.25$_{\uparrow 2.22}$
MonthlyMilkDataset	85.71	**8.00**	13.93$_{\uparrow 5.93}$	13.10$_{\uparrow 5.10}$	68.08$_{\uparrow 60.08}$	58.50$_{\uparrow 50.50}$
SunspotsDataset	48.24	44.83	55.69$_{\uparrow 10.86}$	57.61$_{\uparrow 12.78}$	**37.11**$_{\downarrow 7.72}$	72.23$_{\uparrow 27.40}$
WineDataset	4075.28	2505.76	2475.92$_{\downarrow 29.84}$	3140.31$_{\uparrow 634.55}$	**2107.07**$_{\downarrow 398.69}$	2223.58$_{\downarrow 282.18}$
WoolyDataset	1210.33	880.63	**670.21**$_{\downarrow 210.42}$	749.19$_{\downarrow 131.44}$	910.36$_{\uparrow 29.73}$	859.35$_{\downarrow 21.28}$
HeartRateDataset	5.92	6.83	6.38$_{\downarrow 0.45}$	9.68$_{\uparrow 2.85}$	**5.33**$_{\downarrow 1.50}$	5.84$_{\downarrow 0.99}$
MSAE (AM)	1.00	**0.67**	0.84$_{\uparrow 0.17}$	0.88 $_{\uparrow 0.21}$	0.76$_{\uparrow 0.09}$	0.91$_{\uparrow 0.24}$
MSAE (GM)	1.00	**0.50**	0.63$_{\uparrow 0.13}$	0.63$_{\uparrow 0.13}$	0.65$_{\uparrow 0.15}$	0.69$_{\uparrow 0.19}$

specific datasets, the degradation in the overall geometric mean MAE clearly demonstrates the necessity of this combined strategy for robust generalization. It ensures the model receives consistently normalized inputs regardless of the original data's scale or distribution, which is crucial for a zero-shot setting where the model must perform reliably without dataset-specific tuning.

Observation 5. The synergy of low-confidence remasking and CFG forms a powerful denoising mechanism for TSF. First, semi-autoregressive generation, which predicts time series data step by step, leads to a notable performance drop. This indicates that the diffusion model's ability to generate and refine the entire future trajectory as a whole is more effective in capturing global temporal patterns and dependencies. Second, removing the CFG scale, which uses raw model logits instead of the weighted difference between conditional and unconditional predictions, also causes a clear decline in performance. This highlights CFG's crucial role in reinforcing the influence of the conditional input $\boldsymbol{X}_{1:L}$, helping the model generate future sequences $\boldsymbol{X}_{L+1:L+H}$ that are more coherent with the historical context.

4.4 Parameter Sensitivity

A systematic sensitivity study is carried out to quantify the influence of individual parameters on LEAF's predictive accuracy. We consider the following parameters: (1) α and β, which are the percentile values used in the rescaling strategy; (2) denoising steps and block length, which are the parameters used in the diffusion model to control the denoising process. Intuitively, it can be posited that the smaller the block length, the closer the denoising process is to the autoregressive process. As depicted in Figure 3, the results lead to the following observations.

Observation 6.* LEAF *exhibits relative robustness to the choice of rescaling percentiles α and β. Figure 3a shows that LEAF maintains stable forecasting results across a broad range of α and β values, indicating a degree of insensitivity to precise hyperparameter selection. Nevertheless, the original trend remains: higher β values generally pair best

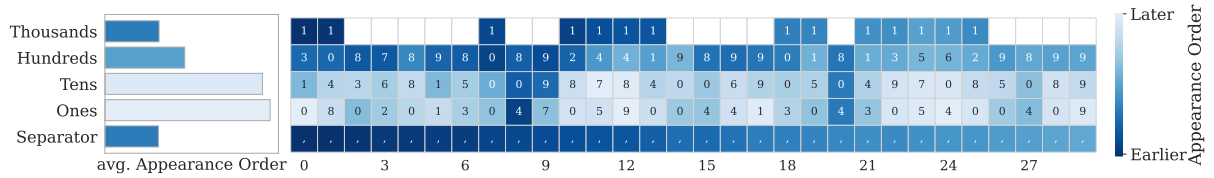


Figure 4: Visualization of the denoising steps performed by the LEAF model on the AirPassengersDataset. The digits are arranged in a grid, from top to bottom, with the first digit (thousands) at the top and the separators (",") at the bottom. Each digit (token) is colored according to the order in which it is denoised during the LEAF.

with lower α, while lower β values are more effective with higher α. This interaction likely arises because a larger offset (higher β) shifts the data distribution, requiring a less extreme upper percentile for scaling, whereas a smaller offset (lower β) preserves larger values and thus benefits from scaling based on a higher percentile to avoid excessively long token sequences.

Observation 7. **LEAF** ***exhibits sensitivity to the hyperparameters controlling the denoising process.*** Diffusion LLMs are sensitive to the inference hyperparameters (Nie et al., 2025), and results indicate that LEAF is the same. A larger block length allows the model to analyze and leverage correlations across a wider span of the predicted sequence, potentially enabling better inference from future context. Conversely, a smaller block length limits this contextual view. However, an exception might arise if the block length aligns well with the inherent periodicity of the time series data, which could lead to improved performance. Regarding the denoising steps, processing more steps (i.e., fewer tokens per step) generally leads to a more refined and detailed output. When dealing with a large block length, an increased number of steps might struggle to effectively denoise the entire block. It will invariably lead to higher computational costs due to the increased number of iterations.

4.5 Case Study

In the end, we conduct a case study to visually analyze the prediction process of the LEAF model. Specifically, we focus on visualizing the denoising steps performed by the model on the AirPassengersDataset. As illustrated in Figure 4, we represent the predicted time series where each token is colored according to the order in which it is denoised during the diffusion model's reverse process. We derive the following observation:

Observation 8. **LEAF** ***demonstrates an intrinsic understanding of numerical structure and range control.*** Denoising visualizations show that LEAF consistently prioritizes the reconstruction of separators (e.g., commas), which define numerical boundaries. This early identification of structural markers implicitly constrains the length and approximate range of the target value before detailed digits are generated. By first establishing separators, LEAF anchors the positional layout, enabling a coarse approximation of numerical scale (e.g., thousands vs. hundreds) ahead of precise refinement. This behavior reflects a human-like strategy: understanding numerical form and scope through structural cues before attending to fine-grained digits. Unlike traditional methods that rely on rule-based length constraints, LEAF achieves this implicitly through its diffusion dynamics.

Observation 9. **LEAF** ***implicitly performs coarse-to-fine hierarchical decomposition of numerical values.*** Further analysis reveals that LEAF denoises digit sequences in a structured order aligned with numerical significance. After resolving separators, it stabilizes higher-order digits (e.g., thousands) first, then progressively refines lower-order digits (e.g., tens, ones). This coarse-to-fine gradient mirrors the logic of hierarchical decomposition methods in time series modeling, such as first extracting the overall trend, then identifying seasonal patterns, and finally modeling the residuals. In contrast to autoregressive models—where early-stage errors in low-order digits can propagate—LEAF's parallel denoising establishes a stable scaffold with high-level semantics first. This reduces positional dependencies and enhances robustness in TSF.

5 Conclusion

In this paper, we present LEAF, a large language diffusion framework for zero-shot time series forecasting. By integrating a denoising diffusion process with digit-level tokenization, LEAF effectively captures global and holistic temporal trajectory and dependencies that are challenging for traditional autoregressive LLMs. Extensive experi-

ments across diverse benchmarks demonstrate that LEAF achieves highly competitive performance, frequently surpassing both zero-shot and supervised baselines, particularly on datasets with strong periodicity. Ablation and case studies further highlight the significance of percentile-based rescaling, the effectiveness of the diffusion mechanism, and the model's ability to prioritize structural and high-order information during generation.

Limitations

While our work presents a novel application of diffusion models to TSF, several constraints merit consideration. The digit-level tokenization may struggle with high-precision or large numerical values, potentially affecting fine-grained accuracy. The method shows a performance gap on certain datasets compared to state-of-the-art supervised approaches, and its iterative denoising steps result in slower inference. Our future work will focus on addressing these limitations, including enhancing the model's ability to handle high-precision data and improving computational efficiency.

Acknowledgements

Tao Ren is supported by the National Natural Science Foundation of China (62276058, 41774063), the Fundamental Research Funds for the Central Universities (N25GFZ011). Yifan Wang is supported by the Fundamental Research Funds for the Central Universities in UIBE (Grant No. 23QN02).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 17981–17993.

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

George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 1877–1901.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2024. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. In *Proceedings of the International Conference on Learning Representations*.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6989–6997.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2025. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–20.

Vinay Kumar Reddy Chimmula and Lei Zhang. 2020. Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, solitons & fractals*, 135:109864.

Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. 2022. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2900–2913. IEEE.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Proceedings of the International Conference on Machine Learning*.

Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. 2017. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. 2023. Forecastpfn: Synthetically-trained zero-shot forecasting. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 2403–2426.

Jiaxin Gao, Qinglong Cao, and Yuntian Chen. 2025. Auto-regressive moving diffusion models for time

series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16727–16735.

Everette S Gardner Jr. 1985. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2023. Large language models are zero-shot time series forecasters. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 19622–19635.

Julien Herzen, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, et al. 2022. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 6840–6851.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Romain Ilbert, Ambroise Odonnat, Vasilii Feofanov, Aladin Virmaux, Giuseppe Paolo, Themis Palpanas, and Ievgen Redko. 2024. Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. *arXiv preprint arXiv:2402.10198*.

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. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning*, pages 19730–19742. PMLR.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B Aditya Prakash. 2024b. Lst-prompt: Large language models as zero-shot time series forecasters by long-short-term prompting. In *Findings of the Annual Meeting of the Association for Computational Linguistics*, page 7832–7840.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024c. Autotimes: Autoregressive time series forecasters via large language models. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 122154–122184.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024d. Timer: Generative pre-trained transformers are large time series models. In *International Conference on Machine Learning*, pages 32369–32399. PMLR.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the International Conference on Learning Representations*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *Proceedings of the International Conference on Learning Representations*.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. 2022. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, et al. 2023. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Amin Shabani, Amir Abdi, Lili Meng, and Tristan Sylvain. 2022. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. *arXiv preprint arXiv:2206.04038*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations*.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 24804–24816.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. 2024. Timexer: Empowering transformers for time series forecasting with exogenous variables. *arXiv preprint arXiv:2402.19072*.

Andrew Wilson and Ryan Adams. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the International Conference on Machine Learning*, pages 1067–1075. PMLR.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of the International Conference on Learning Representations*.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 22419–22430.

Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864.

Xinyu Yuan and Yan Qiao. 2024. Diffusion-ts: Interpretable diffusion for general time series generation. In *Proceedings of the International Conference on Learning Representations*.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11121–11128.

Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *Proceedings of the International Conference on Learning Representations*.

Yusheng Zhao, Xiao Luo, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. 2023. Dynamic hypergraph structure learning for traffic flow forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2303–2316. IEEE.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One fits all: Power general time series analysis by pretrained lm. In *Proceedings of the Conference on Neural Information Processing Systems*.

A Datasets

Table 5 summarizes the datasets used in our experiments. The datasets are categorized into three collections:

Darts Collection. This collection comprises eight univariate time series datasets, characterized by diverse seasonal patterns coupled with both additive and multiplicative trends.

Monash Archive. The Monash Time Series Forecasting Archive initially includes 30 datasets spanning multiple domains (e.g., finance, traffic, weather) and various temporal granularities (ranging from minutes to years).

Informer Collection. For evaluating long-horizon forecasting capabilities, we utilize datasets from the Informer benchmark, specifically excluding those potentially used during the pretraining phase of compared models. Our focus centers on the Electricity Transformer Temperature (ETT) series (ETTm1, ETTm2, ETTh1, ETTh2), which contains two years of electricity transformer temperature recordings captured at 15-minute and 1-hour intervals, respectively.

Table 5: Summary of datasets used in our experiments.

Collection	Dataset	Horizon	Frequency
Darts	AirPassengersDataset	29	Seasonal
	AusBeerDataset	43	Seasonal
	GasRateCO2Dataset	60	Monthly
	MonthlyMilkDataset	34	Monthly
	SunspotsDataset	141	Monthly
	WineDataset	36	Monthly
	WoolyDataset	24	Seasonal
	HeartRateDataset	180	0.5s
Monash	bitcoin	30	1D
	pedestrian counts	48	1H
	nn5 daily	56	1D
	nn5 weekly	8	1W-MON
	tourism yearly	4	1Y
	tourism quarterly	8	1Q-JAN
	tourism monthly	24	1M
	cif 2016	12	1M
	covid deaths	30	1D
	traffic weekly	8	1W-WED
	saugeenday	30	1D
	us births	30	1D
	hospital	12	1M
	solar weekly	5	1W-SUN
Informer	ETTm1	96/192	15min
	ETTm2	96/192	15min
	ETTh1	96/192	1H
	ETTh2	96/192	1H

B Ethics and Data Usage

Artifact Licensing and Usage. We initialize LEAF with the publicly available pretrained LLaDA-base model weights (Nie et al., 2025), which are released under the MIT License, permitting unrestricted use for both research and commercial applications. Our use of the pretrained LLaDA model is consistent with its intended use. The model was specifically designed for NLP applications, and our LEAF framework extends its capabilities while maintaining the original architectural principles.

Data Privacy and Content. All datasets used in our experiments are well-established public benchmarks in the time series forecasting community. These datasets do not contain any personally identifiable information (PII) or offensive content. All benchmark datasets are used for their intended evaluation purposes in time series forecasting research, ensuring compliance with their original usage terms.

The model was trained on data without personal information, ensuring privacy compliance in our framework. No additional data collection or human annotation was required for this work, as we rely entirely on existing public datasets and pretrained models. This approach ensures that our research adheres to ethical standards while maintaining reproducibility and transparency.

C Details of the Rescaling Strategy

In classical MinMaxScale, each original data x_t is rescaled to a bounded range by computing

$$x'_t = \frac{x_t - m}{M - m} \tag{4}$$

where $m = \min(\boldsymbol{X})$ and $M = \max(\boldsymbol{X})$. This ensures that the smallest value maps to 0 and the largest to 1.

Starting from this affine-shift viewpoint, we introduce a two-step modification (Gruver et al., 2023). First, all observations are translated by an offset $\delta = \beta\,(M - m)$, with $\beta \in [0, 1]$ a hyperparameter, so that each x_t becomes $x_t + \delta$. After this shift, we subtract the original minimum m to re-anchor to zero, yielding a numerator $(x_t + \delta) - m$. Rather than dividing by the full range $M - m$, we use q_α, the α-percentile of the unshifted set $\boldsymbol{X}$, as a more robust scale factor. Consequently, the final scaled quantity is

$$x'_t = \frac{(x_t + \delta) - m}{q_\alpha}, \tag{5}$$

where q_α represents the threshold below which α of the original data lie. By shifting first and then normalizing with a percentile-based divisor instead of the absolute maximum, this formulation diminishes the influence of extreme outliers while preserving the relative ordering of values above the chosen percentile.

D Experimental Details

D.1 Experimental Setup

Following the experimental setup in LLMTime (Gruver et al., 2023), we only consider the univariate time series forecasting task. We use the same data splitting method as LLMTime, where the last 20% of the series in the Darts dataset is used as the prediction sequence. For the Monash dataset, we adopt the official splitting method. In the ETT dataset, the prediction sequences are set to lengths of 96 and 192 based on the last test data. We use a single NVIDIA A100 GPU with 80GB of memory for all experiments.

D.2 Metrics

We evaluate the performance of LEAF using the following metrics:

- **MAE (Mean Absolute Error)**: it calculates the average absolute difference between the predicted values $\hat{y}_t$ and the ground truth values y_t:

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t|. \tag{6}$$

- **MSAE (Mean Scaled Absolute Error)**: for each time series k we first compute

$$\mathrm{MSAE}_k = \frac{\mathrm{MAE}_k(\texttt{LEAF})}{\mathrm{MAE}_k(\text{Naive})}, \tag{7}$$

where the naive forecast is the last observed value. Two aggregation strategies over the K datasets are reported.

Arithmetic Mean:

$$\mathrm{MSAE}_{\text{arith}} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{MSAE}_k, \tag{8}$$

Geometric Mean:

$$\mathrm{MSAE}_{\text{geo}} = \Big(\prod_{k=1}^{K}\mathrm{MSAE}_k\Big)^{1/K}. \tag{9}$$

Table 6: Zero-shot imputation results (MAE) of LEAF on the Darts datasets.

Missing Rate	12.5%	25%	37.5%	50%
AirPassengersDataset	10.44	21.67	16.92	33.98
AusBeerDataset	7.55	13.71	17.11	19.59
MonthlyMilkDataset	9.25	8.74	13.97	102.29
WineDataset	1313.58	1863.02	2114.25	2179.67

D.3 Semi-autoregressive Strategy

The semi-autoregressive strategy divides the output sequence into multiple contiguous blocks and generates each block in a left-to-right order (Nie et al., 2025). For each block, the diffusion reverse process is applied to iteratively refine its predictions, while keeping the previously generated blocks fixed as context. This approach enables parallel generation within blocks while maintaining an overall autoregressive structure across blocks, balancing efficiency and temporal dependency modeling.

D.4 Classifier-free Guidance Scale

Classifier-Free Guidance (CFG) is a technique used in diffusion models to enhance conditional generation without the need for an external classifier. In our diffusion-based time series forecasting framework, CFG is employed to strengthen the model's adherence to the conditioning context (i.e., the historical time series $\boldsymbol{X}_{1:L}$).

At each denoising step, the model predicts the noise under both the conditional input (ϵ_{cond}) and an unconditional input (ϵ_{uncond}, typically using a special mask or empty prompt). The final guided prediction is computed as:

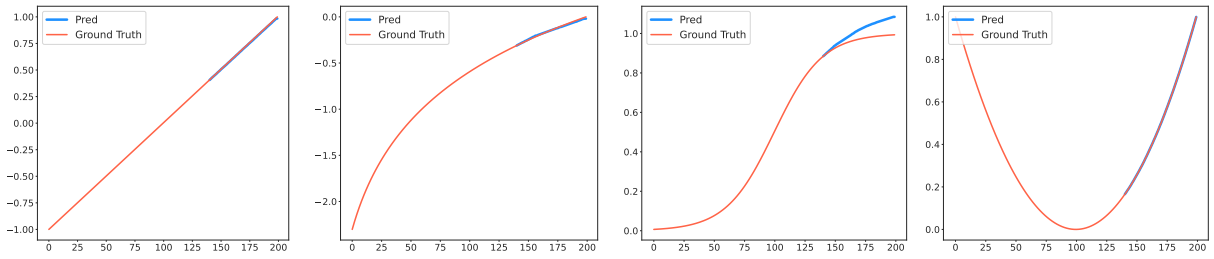
$$\epsilon_{\text{CFG}} = \epsilon_{\text{uncond}} + w \cdot (\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}) \tag{10}$$

where w is the guidance scale hyperparameter. Setting $w > 1$ amplifies the influence of the conditional context, encouraging the model to generate outputs more consistent with the provided history. In our experiments, we set $w = 2.5$ by default. This mechanism effectively improves the quality and relevance of zero-shot time series forecasts by leveraging both conditional and unconditional predictions in the diffusion process.

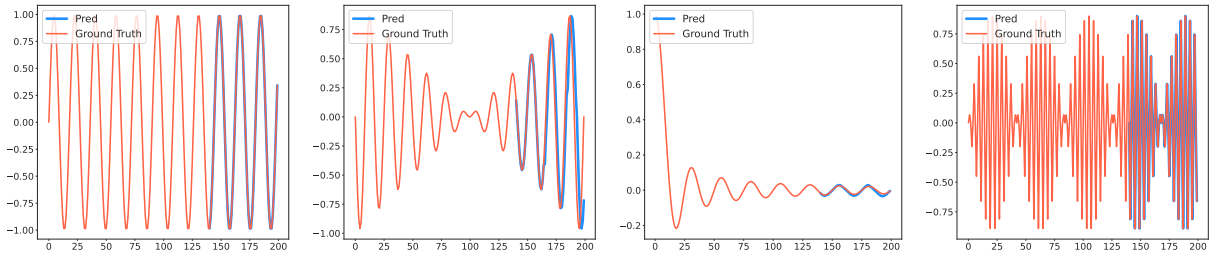
E Additional Experiment

E.1 Zero-Shot Time Series Imputation

Unlike autoregressive LLMs that operate via sequential token generation, LEAF leverages the principles of diffusion models. This allows it to iteratively refine sequences by progressively denoising



(a) Linear (b) Log (c) Sigmoid (d) Square



(e) Sine (f) $x \times sin(x)$ (g) Sinc (h) Beat

Figure 5: Synthetic data forecasting results of LEAF. The orange line represents the ground truth, while the blue line represents the prediction.

masked portions while potentially considering the entire available context (both preceding and succeeding information) within its prediction mechanism at each step. This inherent capability to condition on bidirectional context makes LEAF particularly well-suited for imputation tasks where filling missing values requires understanding the surrounding data points.

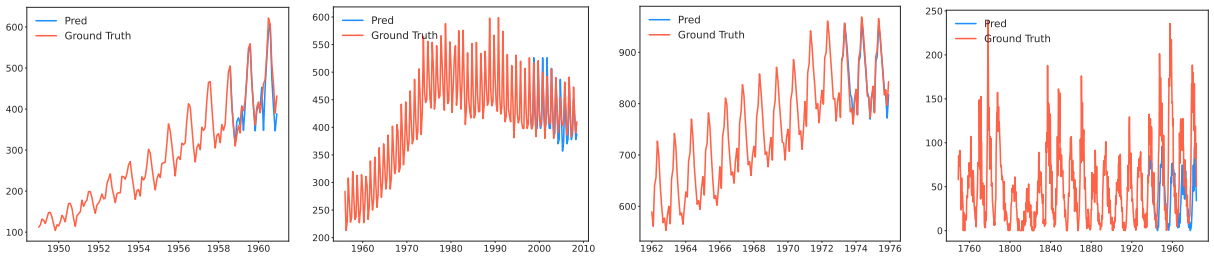
Consequently, we evaluated the zero-shot time series imputation capabilities of LEAF on the DARTS dataset. A challenge in applying the mask to numerical time series data is the variable number of tokens generated for different numerical values during tokenization. To ensure consistent and controllable mask lengths, we preprocessed the data by scaling time series using the transformation $\boldsymbol{X}_{scaled} = 0.1 + 0.9 MinMaxScale(\boldsymbol{X})$. This scaling normalizes the values to a specific range, thereby promoting a uniform token representation length for each numerical entry after tokenization, which is crucial for applying masks of a predefined size. The experimental results of this evaluation are presented in Table 6. Based on these results, we derive the following key observations:

Observation 10. **LEAF** ***can effectively implement the imputation task, but still struggles with a high rate of missing data.*** LEAF maintains relatively low MAE errors under 12.5–25% missing rates, indicating its ability to capture local and global temporal dependencies without fine-tuning. However, at 50% missing rate, MAE surges dramatically for certain datasets. This suggests that while LEAF excels in sparse observation extrapolation, excessive missing data disrupts contextual coherence, likely due to insufficient bidirectional context and tokenization limitations in preserving numerical precision.

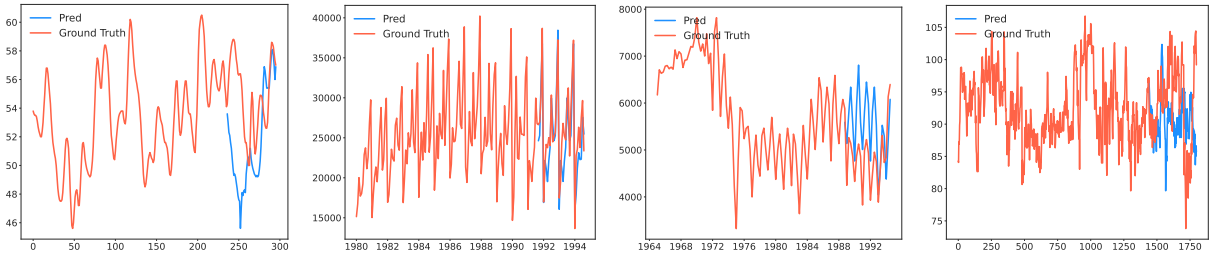
Observation 11. **LEAF** ***leverages bidirectional context effectively, demonstrating superior contextual understanding compared to autoregressive LLMs.*** The imputation results (Table 4) reveal that LEAF achieves comparable MAE in 50% missing-rate scenarios to its forecasting performance on 20% horizon tasks. This suggests that downstream information plays a critical role in temporal modeling. Unlike autoregressive LLMs constrained to unidirectional generation, LEAF's diffusion framework inherently captures dependencies from both past and future observations.

E.2 Synthetic Data Forecasting

We verify the effect of the method in common time series patterns by conducting experiments on synthetic data. We use some constructed data as in LLMTime (Gruver et al., 2023). Specifically, we test on the following synthetic patterns: **linear**, **log**, **sine**, **sinc**, **square**, **beat**, and $x \times \sin(x)$. These patterns cover a range of typical temporal behaviors, allowing us to comprehensively evaluate the forecasting ability of LEAF on diverse synthetic sig-



(a) AirPassengers (b) AusBeer (c) MonthlyMilk (d) Sunspots



(e) GasRateCO2 (f) Wine (g) Wooly (h) HeartRate

Figure 6: Visualization of time series forecasting results of LEAF on the Darts datasets. The orange line represents the ground truth, while the blue line represents the prediction.

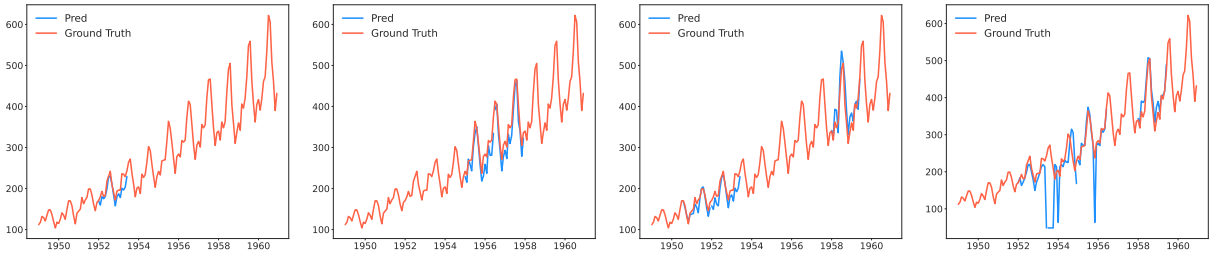
nals.

The visualization of the synthetic data is shown in Figure 5. From the figure, we can observe that LEAF is capable of capturing the underlying patterns of the synthetic data. The predictions align closely with the ground truth, demonstrating its effectiveness in modeling various temporal behaviors.

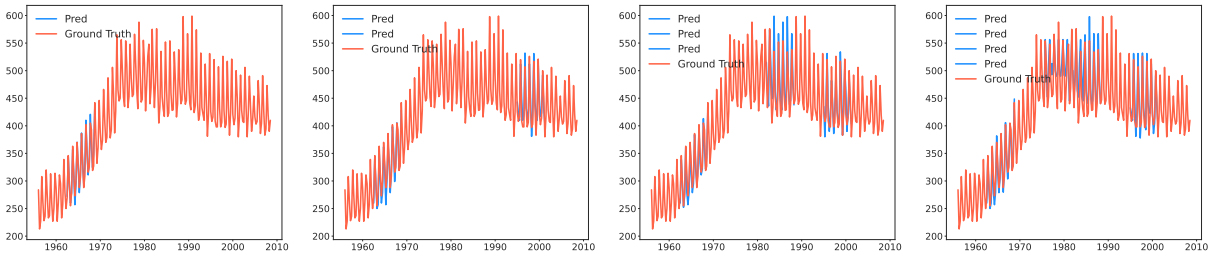
F Visualization

To provide a clearer understanding of the performance of LEAF in time series forecasting tasks, we present visualizations of the results on the Darts datasets. Figure 6 illustrates the forecasting results, where the orange line denotes the ground truth, while the blue line denotes the predictions made by LEAF.

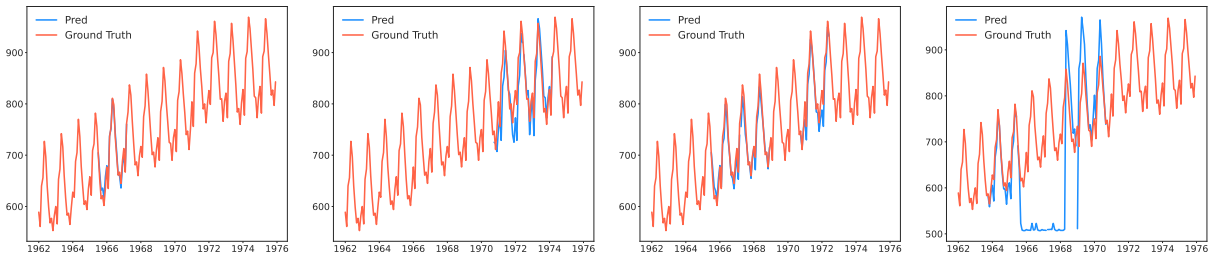
To further illustrate the imputation capabilities of LEAF, we visualize the imputation results on the Darts datasets with varying missing rates. The visualizations are shown in Figure 7.



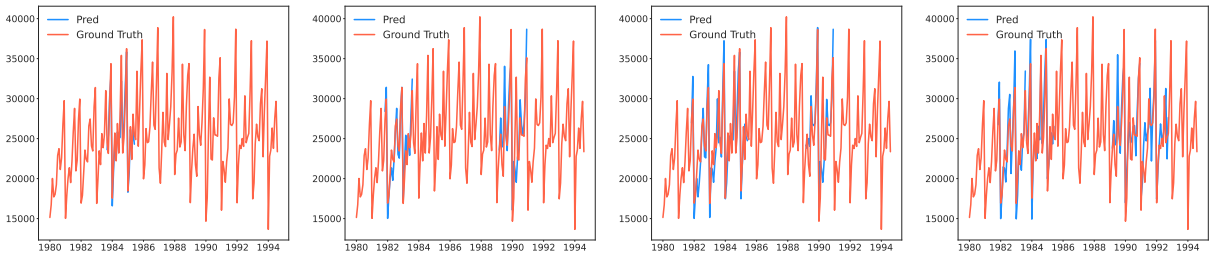
(a) AirPassengers 12.5% Missing (b) AirPassengers 25% Missing (c) AirPassengers 37.5% Missing (d) AirPassengers 50% Missing



(e) AusBeer 12.5% Missing (f) AusBeer 25% Missing (g) AusBeer 37.5% Missing (h) AusBeer 50% Missing



(i) MonthlyMilk 12.5% Missing (j) MonthlyMilk 25% Missing (k) MonthlyMilk 37.5% Missing (l) MonthlyMilk 50% Missing



(m) Wine 12.5% Missing (n) Wine 25% Missing (o) Wine 37.5% Missing (p) Wine 50% Missing

Figure 7: Visualization of time series imputation results of LEAF on the Darts datasets. The orange line represents the ground truth, while the blue line represents the prediction.