FLAME: Flow Enhanced Legendre Memory Models for General Time Series Forecasting

Xingjian Wu*, **Zhengyu Li***, **Hanyin Cheng***, **Xiangfei Qiu***,
Jilin Hu, **Chenjuan Guo**, **Bin Yang**✉
East China Normal University
{xjwu,lizhengyu,hycheng,xfqiu}@stu.ecnu.edu.cn,
{jlhu,cjguo,byang}@dase.ecnu.edu.cn

Abstract

In this work, we introduce **FLAME**, a family of extremely lightweight and capable Time Series Foundation Models, which support versatile forecasting tasks via generative probabilistic modeling, while ensuring both efficiency and robustness. FLAME utilizes the Legendre Memory for strong generalization capabilities. By adapting variants of Legendre Memory, i.e., translated Legendre (LegT) and scaled Legendre (LegS), in the Encoding and Decoding phases, FLAME can effectively capture the inherent inductive bias within data and make efficient long-range inferences. To enhance the accuracy of probabilistic forecasting while remaining efficient, FLAME adopts a normalizing-flow-based forecasting head, which can model complex distributions over the forecasting horizon in a generative manner. Comprehensive experiments on three well-recognized benchmarks, including TSFM-Bench, ProbTS, and TFB, demonstrate that FLAME is a strong out-of-the-box tool for decision intelligence.

https://github.com/decisionintelligence/FLAME.
https://huggingface.co/DecisionIntelligence/FLAME.

1 Introduction

Time Series Forecasting is a critical and widely studied task due to the attractive charm of peeking into the future, which creates value across fields by supporting planning. Time series can fluctuate like a *flame*, making their underlying dynamics difficult to capture. Therefore, to control risks and make reasonable decisions, quantifying uncertainties through Probabilistic Time Series Forecasting (PTSF) is a more reliable approach, which has been explored for decades from statistical approaches [1, 2] with solid theories and interpretability to data-driven approaches [3–5] with strong fitting capability. Among them, deep learning models [3–6] show impressive performance on in-distribution tasks through effectively capturing the inherent dynamics within raw data. Inspired by the success of pretrained models in other domains such as CV [7, 8] and NLP [9, 10] with strong general-

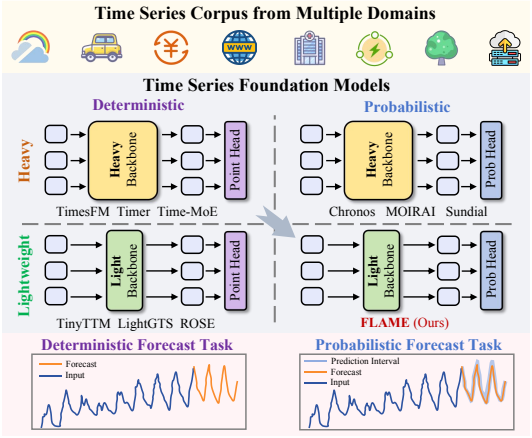


Figure 1: FLAME is pretrained on Time Series Corpus from Multiple Domains, possessing a *lightweight* backbone while supporting generative *probabilistic* forecasting.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

ization capabilities, a surge of pretrained time series foundation models [11–16] have been developed for out-of-distribution tasks, demonstrating promising trends.

Recent studies on time series foundation models [17, 14, 15, 18] have achieved some progress in exploiting the generalization capabilities and exhibit excellent forecasting performance in zero-shot scenarios. Their generalization capabilities mainly come from two perspectives–see Figure 1: 1) some works make efforts in devising appropriate Transformer-based architectures, and expanding the parameters to hundreds of millions or billions of parameters [14, 19, 17, 12], or with the help of the massive parameters from LLMs [20, 21] and ViTs [22, 23]. Though these works achieve excellent performance in zero-shot scenarios, they incur substantial inference costs under limited resources, and some of them are even slower than training end-to-end supervised models from scratch, thus *hindering the efficiency and practicality*; 2) another line of work explores the effective capture of inductive bias within data, with meticulously-designed lightweight structures [24, 13, 15] but strong generalization capabilities. However, although they demonstrate strong performance in deterministic forecasting tasks, they *lack the support of probabilistic forecasting*. Moreover, predicting accurate distributions over the forecasting horizon is difficult because assuming prior distributions like Isotropic Gaussian Distributions constrains the modeling capabilities of real data distributions. On the other hand, modeling arbitrarily complex distributions relies on generative mechanisms [25, 26], which may increase the model complexity and hinder the efficiency.

As shown in Figure 1, our proposed **FLAME** pioneers the exploration of *lightweight* time series foundation models capable of *both deterministic and probabilistic forecasting*. To enhance the generalization capability while remaining lightweight, we utilize the *Legendre Memory* [27, 28] in both Encoding and Decoding phases to capture the inherent inductive bias within data, which contributes to a lightweight but strong framework. Originating from ordinary differential equations (ODEs) and characterized by orthogonal Legendre polynomials, the Legendre Memory has been utilized in control theory to model continuous time-delay systems [29] for several decades, which has the potential to compress time series of varying lengths into vectors of the same size. This helps keep the model lightweight, and we employ Legendre Memory in both the encoder and decoder.

First, in the pretraining process of time series foundation models, the inherent periods and trends of data from distinct domains can lead to complex local characteristics but the patching technique cannot handle this due to the fixed receptive field. This inspires us to devise a *Local-Perception* module to unlock the power of Legendre Memory in integrating intricate and varying local characteristics within data as supplementary environmental information for patch tokens. Second, we also adopt the Structured State-Space Duality (SSD [30]) layers in the decoding phase, which utilize Legendre Memory to initialize the state transition matrices [28], excelling at modeling long time series based on its selective states and adaptive memory utilization.

Besides the adaptations of Legendre Memory, we also devise a *Flow-based Head* to support generative probabilistic forecasting. Specifically, we focus on token-wise modeling, which constructs the distributions over dozens of time series points at once, and using the Flow-based Head driven by normalizing flow [31, 32] can preserve both considerable efficiency and interpretability compared with prevailing diffusion models [26, 33]. Additionally, we also aim to capture the within-token autoregressive dependencies in the Flow-based Head, which is important but neglected in current decoding methods [18, 11, 14], and this can help preserve temporal dependencies within tokens to boost the fine-grained modeling. Through these adaptations, the Flow-based Head can preserve both the accuracy and efficiency through constructing lightweight autoregressive affine layers to simulate the reversible conditional transfer process from fixed prior Gaussian distributions to flexible predictive distributions. With the Legendre Memory and Flow-based Head facilitating the strong generative paradigm, FLAME can make extremely fast and accurate probabilistic forecasts. Our contributions are summarized as:

- We present **FLAME**, a family of **Fl**ow enh**a**nced Legendre **Me**mory Foundation Models, including three parameter versions, i.e., 2M, 6M, 10M. Even the largest version of FLAME (10M) is far more lightweight than other time series foundation models which support probabilistic forecasting.
- We adapt the Legendre Memory to enhance the generalization capability. Both the Local-Perception Module and SSD-Decoder are driven by Legendre Memory and able to adaptively utilize the inductive bias within data.
- We devise a Flow-based Head to model the intricate token-wise distributions over the forecasting horizon, considering the causality within tokens, with strong efficiency and accuracy.

- Experimentally, FLAME achieves state-of-the-art zero-shot performance on both deterministic and probabilistic forecasting tasks, including TSFM-Bench [34], ProbTS [35], and TFB [36], demonstrating a strong out-of-the-box tool for decision intelligence.

2 Legendre Memory

When modeling sequence data with recurrent networks, "memory" is a common concept denoting the online compression capability of historical observations. Legendre Memory [27, 28] is a method for continuous online function approximation, which utilizes orthogonal Legendre polynomials to compress the continuous-time history f into fixed-size state vectors m. From the perspective of *Real Analysis*, given a time-varying *measure* $\mu^{(t)}$ and a *high-order polynomial operator*, recent history f can be compressed into state vectors m with bounded approximation error [28]. When the operator is the Legendre polynomial, there exist two common time-varying measures: *translated Legendre (LegT)* and *scaled Legendre measure (LegS)*.

LegT assigns uniform weights to the most recent history $[t-\theta, t]$, where θ denotes the length of the sliding window, or the length of history that is being summarized as LegT : $\mu^{(t)}(x) = \frac{1}{\theta}\mathbb{I}_{[t-\theta,t]}(x)$. Under the LegT measure, the process of compressing the historical information f into state vectors m can be represented as solving d coupled linear time-invariant ODEs:

$$m'(t) = -Am(t) + Bf(t),\ B_n = \frac{1}{\theta}(2n+1)(-1)^n, \tag{1}$$

$$A_{nk} = \frac{1}{\theta}\begin{cases}(-1)^{n-k}(2n+1) & \text{if } n > k,\\ 2n+1 & \text{if } n \le k\end{cases}, \tag{2}$$

where $n, k \in \{0, \ldots, d-1\}$, and $A \in \mathbb{R}^{d\times d}, B \in \mathbb{R}^{d\times 1}$ are parameters derived using a Padé approximant [37, 38]. In the forward process, the history over $[t-\theta, t]$ can be compressed into the memory cell $m(t) \in \mathbb{R}^d$ recurrently using the discretized form of Equation (1):

$$m_t = \bar{A}m_{t-1} + \bar{B}f_t,\ \bar{A} = I - \Delta t A, \quad \bar{B} = \Delta t B, \tag{3}$$

where $\bar{A}$ and $\bar{B}$ are obtained through forward Euler discretization; the factor $1/\theta$ is already included in A and B. The "memory" lies in the capability of using $m(t)$ to approximate the history f over $[t-\theta, t]$, thereby approximating a θ time-delay system [29]:

$$\begin{aligned} f(t-\theta') &\approx \sum_{i=0}^{d-1} \mathcal{P}_i\left(\frac{\theta'}{\theta}\right) m_i(t), \qquad 0 \le \theta' \le \theta,\\ \mathcal{P}_i(r) &= (-1)^i \sum_{j=0}^{i} \binom{i}{j}\binom{i+j}{j}(-r)^j, \end{aligned} \tag{4}$$

where $\mathcal{P}_i(r)$ is the i-th shifted Legendre polynomial [27]. Here $r = \theta'/\theta$ is the normalized lag: $r = 0$ corresponds to the newest point in the window and $r = 1$ to the oldest. The same d memory coefficients therefore provide approximate reconstructions at different lags by evaluating the corresponding basis functions. Theorem 2.1 [28, 39] bounds the error of the exact polynomial projection as the number d of Legendre basis functions increases. The finite-state recurrence and its discretization introduce additional approximation errors. We provide a detailed proof of the projection bound in Appendix H.1.

Theorem 2.1. *Let $F : \mathbb{R}_+ \to \mathbb{R}$ be a differentiable function, $\theta > 0$, $t \ge \theta$, and $d \ge 1$. Let $f_{[t-\theta,t]}$ denote the **HiPPO-LegT projection** of F, defined as its optimal polynomial approximation of degree at most $d-1$ under the uniform probability measure $\mu^{(t)}$ on $[t-\theta, t]$. If F is L-Lipschitz on this window, then $\|F - f_{[t-\theta,t]}\|_{\mu^{(t)}} = \mathcal{O}(\theta L/\sqrt{d})$. Moreover, for a fixed integer $k \ge 1$, if $F \in C^k([t-\theta, t])$, $M_k = \sup_{x\in[t-\theta,t]} |F^{(k)}(x)| < \infty$, and $d \ge k$, then $\|F - f_{[t-\theta,t]}\|_{\mu^{(t)}} = \mathcal{O}(M_k \theta^k d^{-k+1/2})$, with the implied constant depending only on k.*

Compared with LegT, LegS assigns uniform weights to all history $[0, t]$ to support the compression of the entire history: LegS : $\mu^{(t)}(x) = \frac{1}{t}\mathbb{I}_{[0,t]}(x)$.

Under the LegS measure, the compression process is:

$$m'(t) = -\frac{1}{t}Am(t) + \frac{1}{t}Bf(t),\ B_n = (2n+1)^{\frac{1}{2}}, \tag{5}$$

$$A_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k, \\ n+1 & \text{if } n = k, \\ 0 & \text{if } n < k \end{cases} \tag{6}$$

where $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times 1}$, with $n, k \in \{0, \ldots, d-1\}$. Unlike LegT, LegS is time-varying: a forward Euler step from time $\tau > 0$ uses $\bar{A}_\tau = I - (\Delta t/\tau)A$ and $\bar{B}_\tau = (\Delta t/\tau)B$.

3 FLAME

FLAME adopts the Channel-Independent [3] pretraining paradigm, and each variable is preprocessed through Instance Normalization [40] to mitigate the value discrepancy. As shown in Figure 2, FLAME utilizes the Re-Norm [41] to further mitigate the statistical differences between inputs and forecasts, and its backbone mainly consists of three modules: 1) Encoding, including Time Series Tokenization, Local-Perception, and MSA-Encoder, which tokenize the time series and enhance them through fusing the local environmental information with LegT; 2) Decoding, including LegS based SSD-Decoder and MCA-Decoder, which utilize the SSD layers and MCA layers to make long-term inference; 3) Flow-based Head, which leverages the Normalizing Flow to support generative probabilistic forecasting.

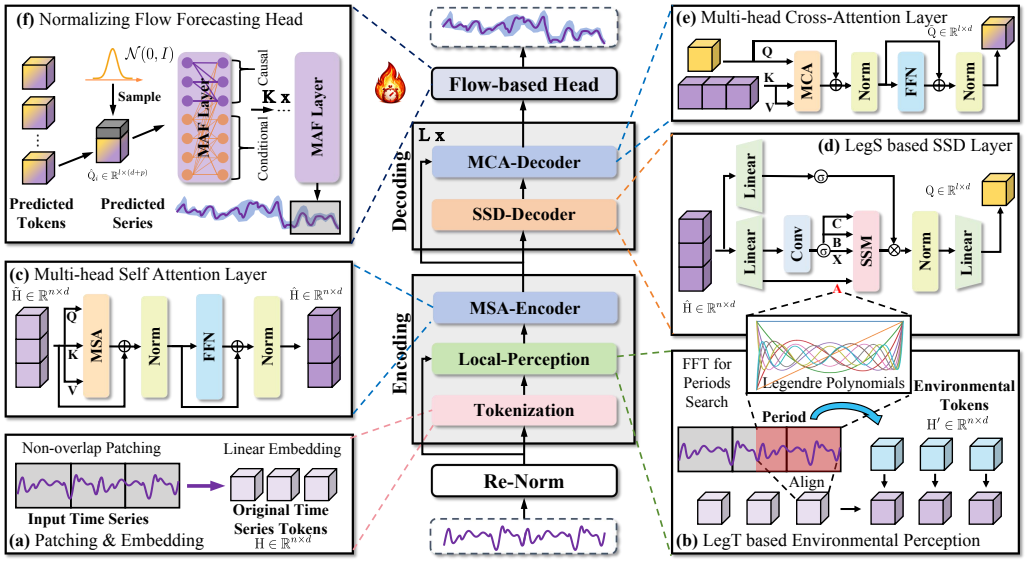


Figure 2: The overall framework of FLAME. The data flow follows the sequence of (a) - (f).

3.1 Encoding

3.1.1 Time Series Tokenization

In the pretraining Time Series Corpus, all data are preprocessed into univariate time series and received by FLAME in a Channel-Independent paradigm. Following the commonly used patching technique, which was introduced by Triformer [42], we first divide each univariate time series $\mathrm{X} \in \mathbb{R}^T$ into n non-overlapping time series patches $[x_1, x_2, \cdots, x_n]$, where $x_i \in \mathbb{R}^p$ is a patch of size p, and $n = \lceil T/p \rceil$ is the number of patches. If T is not divisible by p, the boundary patch is padded to length p. Subsequently, we use a linear layer: $\mathbb{R}^p \to \mathbb{R}^d$ to transform the patches to tokens:

$$\mathrm{H} = [\texttt{Linear}(x_1), \texttt{Linear}(x_2), \cdots, \texttt{Linear}(x_n)], \tag{7}$$

where $\mathrm{H} \in \mathbb{R}^{n\times d}$ are the embedded time series tokens. Patch-wise tokens preserve semantic information and reduce the number compared with point-based tokens, thus enhancing efficiency.

3.1.2 Local-Perception

Although the patching technique [42, 3] has achieved some progress in time series foundation models, its fixed receptive field may limit models' generalization capabilities, thus requiring heavy backbones. The core reason is that the inherent periods and trends within data vary across datasets from distinct domains, causing the diversity of frequencies, but the fixed patch size in pretrained models only captures the same span without considering the changing surrounding environmental information.

Specifically, in pretraining scenarios, a patch with the fixed length may cover the whole period in one pretraining dataset but only 70% of a period in another–see Figure 2(b), which makes this semantic variation difficult to model. To tackle this limitation, we manage to recognize the varying periods, then align their information with patch-wise tokens to supplement the environmental semantics. First, we adopt the FFT [43] to find the period of the input series:

$$\mathcal{A} = \texttt{Amp}(\texttt{FFT}(\mathrm{X})), \quad \mathcal{F} = \underset{1 \le f \le \lfloor T/2 \rfloor}{\arg\max} \mathcal{A}[f], \quad \mathrm{P} = \left\lceil \frac{T}{\mathcal{F}} \right\rceil, \tag{8}$$

where $\texttt{FFT}(\cdot)$ and $\texttt{Amp}(\cdot)$ denote the FFT and amplitude calculation. The search excludes the zero-frequency (DC) component, and P is the estimated period length. Then we consider the relative positions of each patch and its surrounding period to align them:

$$s = \lfloor (p + \mathrm{P} - 1)/\mathrm{P} \rfloor \cdot \mathrm{P} - p, \mathrm{X}' = \texttt{LeftPadding}(\mathrm{X}, s), \tag{9}$$

$$\tilde{\mathrm{X}} = \texttt{Patching}(\mathrm{X}') = [w_1, w_2, \cdots, w_n], \tag{10}$$

where `Patching` uses a window of length $p + s$ and stride p, producing local environmental patches $\tilde{\mathrm{X}} \in \mathbb{R}^{n \times (p+s)}$. The padding length s is used to make each local environmental patch $w_i \in \mathbb{R}^{p+s}$ cover the corresponding patch $x_i \in \mathbb{R}^p$, and $p + s$ is the integer multiple of the period P. These operations can help align the intricate and varying local information w_i to the patch x_i.

For each $w_i \in \mathbb{R}^{p+s}$, its length varies with the sample during pretraining due to the multiple frequencies of the time series corpus, which is difficult to encode using networks with fixed input dimensions, such as MLPs. To overcome this, we propose to utilize the LegT Legendre Memory–see Section 2, to encode the varying local environmental information, i.e., the varying periods, into vectors of the fixed size. For each $w_i \in \mathbb{R}^{p+s}$, we use the discretized form of LegT to compress it like a state space model in an autoregressive manner:

$$\bar{A} = I - \Delta t A, \quad \bar{B} = \Delta t B, \tag{11}$$

$$f_t = w_{i,t}, m_0 = 0, m_t = \bar{A} m_{t-1} + \bar{B} f_t, \tag{12}$$

where $\theta = p + s$ is determined by the length of w_i, $A \in \mathbb{R}^{d \times d}$ and $B \in \mathbb{R}^{d \times 1}$ are fixed and initialized using Equation (2), and Δt is the predefined step size for forward Euler discretization. This mechanism compresses each local environmental patch $w_i \in \mathbb{R}^{p+s}$ with *varying lengths* into a fixed-size memory cell $m_t \in \mathbb{R}^d$. The generated environmental tokens are $\mathrm{H}' = [w'_1, w'_2, \cdots, w'_n]$, where each token $w'_i \in \mathbb{R}^d$ corresponds to the last state of the memory cell. We then fuse the original tokens H and the environmental tokens H' into $\tilde{\mathrm{H}} \in \mathbb{R}^{n \times d}$ via $\tilde{\mathrm{H}} = \mathrm{H} + \mathrm{H}'$. Theorem 2.1 motivates increasing d to reduce the polynomial projection error; it does not by itself guarantee lossless reconstruction when $d > \theta$. We evaluate reconstruction empirically in Appendix B and provide the projection proof in Appendix H.

3.1.3 MSA-Encoder

After obtaining the fused tokens, we use the Multi-head Self-Attention Layer [44] as the encoder to generate the hidden representations of them. Through fully-connected interaction, the MSA-Encoder aims at learning the similarities and differences among patterns and environmental information within the tokens $\hat{\mathrm{H}} = \texttt{MSA-Encoder}(\tilde{\mathrm{H}})$, where the representations $\hat{\mathrm{H}} \in \mathbb{R}^{n \times d}$ are the outputs of the Encoding phase, and provide rich semantics within data.

3.2 Decoding

3.2.1 SSD-Decoder

We utilize LegT in Local-Perception of the Encoding phase to enrich the local environmental information. In the Decoding phase, the long-range modeling capability gains more attention, which

calls for adaptive memory utilization when making long-term forecasts. The Structured State-Space Duality (SSD) layer in Mamba-2 [30], uses the selective state space modeling mechanism to adaptively process historical information, significantly improving efficiency while achieving similar accuracy to Transformers. Its mechanism takes the form as follows:

$$m_t = A_t m_{t-1} + B_t x_t, y_t = C_t^T m_t, \tag{13}$$

where the form shares similarities with LegS–see Formula (5). Previous works [45, 30] demonstrate that adapting A, B, C to selective A_t, B_t, C_t can enhance the non-linearity capability. Among them, A_t is obtained from A through the learnable granularity Δt of discretization, so that the initialization of A still matters. To preserve the capability of adaptive memory utilization and ensure the linear complexity of SSD, we use the diagonal of A in LegS to initialize the continuous-time transition parameters before discretization. We then use the SSD layer with initialized A_t as the Decoder to forecast the future tokens:

$$\mathrm{Q} = \texttt{SSD-Decoder}(\hat{\mathrm{H}}), \tag{14}$$

where $\mathrm{Q} \in \mathbb{R}^{l \times d}$. Supposing the length of forecasting horizon is L, $l = \lceil L/p \rceil$. Based on the LegS, SSD-Decoder can efficiently summarize historical information and adaptively use it for forecasting.

3.2.2 MCA-Decoder

We then use the Multi-head Cross-Attention [44] Layer to further enhance the outputs through recalling the similar representations in the inputs. Specifically, we set Q as Query, $\hat{\mathrm{H}}$ as Key and Value. Through the skip connections in MCA-Decoder, the feature sequence Q is refined as $\tilde{\mathrm{Q}} = \texttt{MCA-Decoder}(\mathrm{Q}, \hat{\mathrm{H}})$, where $\tilde{\mathrm{Q}} \in \mathbb{R}^{l \times d}$ denotes the output of the Decoding phase and captures the high-order semantic information learned from the token transition process in the hidden space.

3.3 Flow-based Head

Through pretraining on Time Series Corpus from multiple domains, the Encoding and Decoding phases of FLAME possess strong capability of modeling the token transition principles, which means $\tilde{\mathrm{Q}} = [q_1, q_2, \cdots, q_l] \in \mathbb{R}^{l \times d}$ are high-quality representations for forecasting. In time series foundation models, making token-wise forecasts through relatively lightweight heads can preserve efficiency and ensure the heavy backbones do learn the "knowledge" [11, 18].

However, due to the inherent temporal causality within time series data, patch-based forecasting models only model the token-wise causal relationships in decoding. When transforming these high-dimensional tokens into patch sizes for forecasts, the fine-grained causal relationships within each patch are often neglected. To support generative probabilistic forecasting, remain lightweight, and capture such fine-grained causal relationships, we adopt a Masked Autoregressive Flow (MAF) [46] to model the intricate probabilistic distributions of real data. The whole generative process is formulated as:

$$\tilde{\mathrm{Q}} = [q_1, q_2, \cdots, q_l] \in \mathbb{R}^{l \times d},\ \epsilon_i \sim \mathcal{N}(0, I), \tag{15}$$

$$\hat{\mathrm{Y}}_i = T_1^{-1} \circ \cdots \circ T_K^{-1}(\epsilon_i \mid q_i), \tag{16}$$

where $q_i \in \mathbb{R}^d$ is the conditioning context for the i-th predicted patch $\hat{\mathrm{Y}}_i \in \mathbb{R}^p$, and $\epsilon_i \in \mathbb{R}^p$ is sampled from the base distribution. Each T_j maps data toward latent space; generation applies the inverse layers in reverse order. Define $\mathrm{Z}_0^i = \mathrm{Y}_i$ and $\mathrm{Z}_K^i = \epsilon_i$ in the data-to-latent direction. The j-th layer is an autoregressive affine transformation parameterized by a masked MLP:

$$\begin{aligned} (\alpha_{i,k}^{(j)},\ \beta_{i,k}^{(j)}) &= \texttt{MLP}_{\mathcal{M},k}^{j}(\mathrm{Z}_{j-1}^i[1{:}k{-}1],\ q_i), \\ \mathrm{Z}_j^i[k] &= \exp(\alpha_{i,k}^{(j)})\, \mathrm{Z}_{j-1}^i[k] + \beta_{i,k}^{(j)}, \end{aligned} \tag{17}$$

where $\texttt{MLP}_{\mathcal{M},k}^{j}$ denotes the pair of outputs for dimension k of the shared masked MLP. It generates element-wise affine parameters $(\alpha_{i,k}^{(j)}, \beta_{i,k}^{(j)})$. The effective input-to-parameter dependency mask is:

$$\mathcal{M}_{r,k} = \begin{cases} 1 & \text{if } k < r, \\ 0 & \text{if } k \geq r, \end{cases} \qquad r, k \in \{1, \ldots, p\}, \tag{18}$$

where $\mathcal{M}_{r,k} = 0$ means that the affine parameters for dimension r cannot depend on input dimension k. The transformed value still directly depends on its own input through the affine term. Thus, $(\alpha_{i,k}^{(j)}, \beta_{i,k}^{(j)})$ depend only on preceding input dimensions and the full conditioning token q_i. This preserves the autoregressive ordering within each patch (Figure 2(f)) and ensures **exact invertibility**: $\mathrm{Z}_{j-1}^{i}[k] = (\mathrm{Z}_{j}^{i}[k] - \beta_{i,k}^{(j)}) / \exp(\alpha_{i,k}^{(j)})$, evaluated sequentially for $k = 1, \ldots, p$ within each inverse layer. Applying all K inverse layers gives the predicted patches, which are concatenated as:

$$\hat{\mathrm{Y}} = [\hat{\mathrm{Y}}_1, \hat{\mathrm{Y}}_2, \cdots, \hat{\mathrm{Y}}_l], \tag{19}$$

where $\hat{\mathrm{Y}} \in \mathbb{R}^{l \times p}$ can be further flattened and truncated to the first L points to obtain forecasts.

With the affine autoregressive layers, the Flow-based Head achieves intricate distributional transformation while remaining lightweight. Due to the probabilistic generative mechanism, multiple samples are recommended to evaluate the forecasted distributions. Let $\mathrm{Y}_i \in \mathbb{R}^p$ denote the i-th ground-truth target patch. For training, we apply the encoding direction $T = T_K \circ \cdots \circ T_1$ to map data to latent space. Since the input Y_i is known, the masked MLP computes all affine parameters in parallel. The log-likelihood is:

$$\begin{aligned} \epsilon_i = T_K \circ \cdots \circ T_1(\mathrm{Y}_i \mid q_i), \log p_\theta(\mathrm{Y}_i \mid q_i) = \log p_\epsilon(\epsilon_i) + \sum_{j=1}^{K} \sum_{k=1}^{p} \alpha_{i,k}^{(j)}, \\ \mathcal{L}_{\text{NLL}} = -\frac{1}{l} \sum_{i=1}^{l} \log p_\theta(\mathrm{Y}_i \mid q_i). \end{aligned} \tag{20}$$

Each autoregressive layer has a triangular Jacobian with diagonal entries $\exp(\alpha_{i,k}^{(j)})$, making the double sum the log absolute determinant of the data-to-latent mapping. Appendix H.2 proves global invertibility, this exact likelihood formula, and normalization of the conditional density.

4 Experiments

Detailed experimental settings are provided in Appendix A. We evaluate deterministic forecasting on TSFM-Bench and TFB in Section 4.1, and probabilistic forecasting on ProbTS in Section 4.2. We also conduct a detailed analysis of key components of FLAME to reveal the source of its generalization capability in Section 4.3. To analyze the scalability of FLAME, we make studies on training and inference in Section 4.4. As an important property for deployment, we also analyze the efficiency of FLAME in Section 4.5. In summary–see Figure 3, FLAME achieves the most first-place results with a small parameter count. *We provide detailed results in Appendix C, including full results on TSFM-Bench and ProbTS, as well as results under full-shot and few-shot settings.*

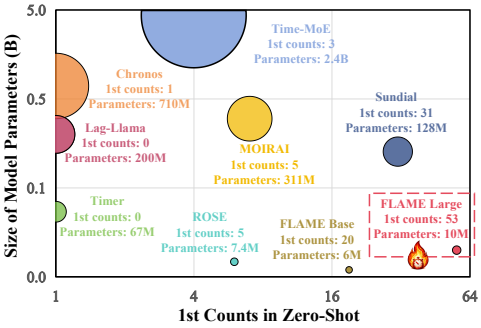


Figure 3: The parameter scale and zero-shot performance of Time Series Foundation models. FLAME Large achieves the most first-place results with approximately 10M parameters.

4.1 Deterministic Forecasting

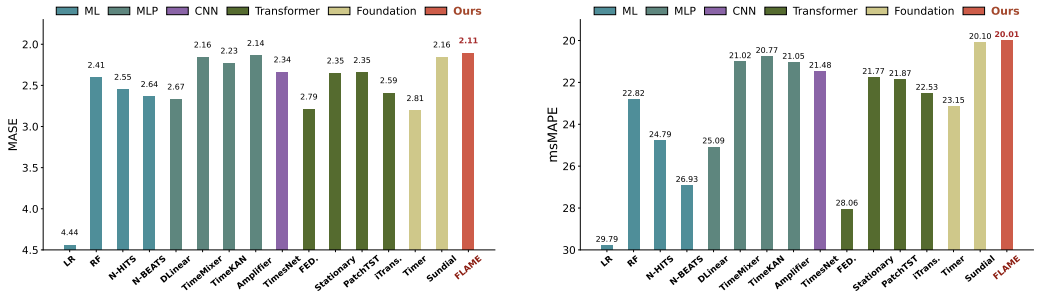
As shown in Table 1, the FLAME family outperforms other advanced time series foundation models on datasets from TSFM-Bench. Compared with the recent state-of-the-art model Sundial, FLAME Large outperforms it in most settings, achieving the most first-place results in zero-shot deterministic forecasting tasks. Note that even the FLAME Small (2M) can achieve competitive performance with Sundial Base (128M) while only possessing the *1/64* parameter scale. Compared with the heaviest model Time-MoE Ultra (2.4B), FLAME Small possesses the *1/1,200* parameter scale but achieves average MSE reduction of *15.0%*. Considering lightweight baselines like UniTS and ROSE, FLAME Small also shows excellent advantages in generalization capabilities, achieving significant average MSE reductions of *26.3%* and *17.4%*.

Table 1: Average results of zero-shot deterministic forecasting. ("-") denotes datasets included in the model's pretraining and therefore excluded from testing. **Red**: the best, Blue: the 2nd best. Full results are listed in Table 16. We also provide strict comparisons in Table 18, excluding datasets used in some baselines' pretraining.

Models	FLAME Small		FLAME Base		FLAME Large		Sundial (2025)		LightGTS (2025)		ROSE (2025)		Timer (2024)		Chronos (2024)		Time-MoE (2024)		TinyTTM (2024)		MOIRAI (2024)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT (Avg)	0.339	0.382	0.339	0.381	0.344	0.385	**0.335**	0.379	0.339	**0.379**	0.393	0.411	0.551	0.478	0.442	0.408	0.357	0.390	0.441	0.430	0.382	0.388
Weather	**0.215**	**0.261**	0.222	0.269	0.227	0.273	0.234	0.270	0.219	0.267	0.265	0.305	0.292	0.313	0.288	0.309	0.256	0.289	0.265	0.307	0.260	0.275
Electricity	0.287	0.371	0.222	0.319	0.217	0.313	**0.169**	**0.265**	0.233	0.319	0.234	0.320	0.297	0.375	-	-	-	-	0.222	0.317	0.188	0.273
Traffic	0.673	0.461	0.576	0.396	**0.546**	**0.384**	-	-	0.610	0.399	0.588	0.412	0.613	0.407	0.615	0.421	-	-	0.564	0.386	-	-
Solar	0.203	0.278	0.184	0.277	**0.168**	0.272	0.221	**0.252**	0.219	0.305	0.505	0.549	0.771	0.604	0.393	0.319	0.411	0.428	0.815	0.710	0.714	0.704
PEMS08	0.324	0.384	0.312	0.374	**0.293**	**0.351**	-	-	0.812	0.692	1.369	0.979	0.866	0.695	1.707	1.024	-	-	1.730	1.066	-	-
Wind	1.144	0.765	1.120	0.758	**1.042**	**0.733**	1.186	0.772	1.292	0.836	1.251	0.820	1.201	0.783	1.478	0.834	-	-	1.337	0.829	1.299	0.795
NYSE	0.527	0.553	**0.442**	**0.430**	0.529	0.507	0.880	0.642	0.620	0.626	-	-	0.988	0.704	1.129	0.720	-	-	-	-	0.492	0.441
1^{st} **Count**	8	8	6	4	**15**	**9**	9	9	0	3	3	2	0	0	0	0	1	2	2	3	0	4
2^{nd} **Count**	3	1	**15**	9	2	**10**	2	7	9	6	2	3	0	0	0	0	2	1	5	3	4	4

Short-Term Forecasting. We further evaluate FLAME Large on short-term forecasting using 8,068 univariate time series from TFB [36]. Figure 4 compares FLAME with machine learning methods, end-to-end supervised models, and time series foundation models. FLAME Large achieves the lowest average MASE of **2.11** and msMAPE of **20.01** among the compared methods. For comparison, Sundial obtains 2.16 MASE and 20.10 msMAPE.

The baseline rankings differ across the two metrics. For example, Amplifier achieves a lower MASE than TimeKAN (2.14 versus 2.23), whereas TimeKAN obtains a lower msMAPE (20.77 versus 21.05). FLAME ranks first under both criteria, showing that its aggregate advantage holds across the two error measures. These results complement the long-horizon multivariate evaluations on TSFM-Bench, demonstrating that FLAME's forecasting capability also extends to short-term univariate settings.



(a) MASE. (b) msMAPE.

Figure 4: Short-term forecasting results averaged over 8,068 univariate time series from TFB. Lower values indicate better performance.

4.2 Probabilistic Forecasting

Table 2 summarizes the results of probabilistic forecasting on ProbTS. FLAME shows strong performance against the other probabilistic foundation models. FLAME Large achieves the most first-place results in zero-shot probabilistic forecasting tasks, excelling in both periodic scenarios like ETT and Weather, and non-stationary scenarios like Exchange and ILI. Compared with the well-established baselines MOIRAI Large and Chronos Large, FLAME Large achieves average CRPS reductions of *34.3%* and *18.1%*, respectively. On Weather, FLAME Large reduces CRPS from Sundial's 0.087 to 0.074 and NMAE from 0.102 to 0.090. These paired reductions show improvements in both distributional and point forecast accuracy.

FLAME's zero-shot performance also compares favorably with several full-shot baselines. Compared with diffusion-based CSDI and Flow-based GRU NVP, which need dozens of hours to train from scratch on a large dataset, FLAME Large achieves average *23.5%* and *35.1%* CRPS reduction. This demonstrates the effectiveness of Flow-based Head in FLAME, which provides both efficiency and accuracy.

Table 2: Average results of zero-shot probabilistic forecasting. ("-") denotes datasets included in the model's pretraining and therefore excluded from testing. ("/") denotes the excessive time consumption. **Red**: the best, Blue: the 2nd best. Full results are in Table 17. We also provide strict comparisons in Table 19, excluding datasets used in some baselines' pretraining.

Models	**FLAME Small**		**FLAME Base**		**FLAME Large**		**Sundial** (2025)		**Chronos** (2024)		**MOIRAI** (2024)		**Lag-Llama** (2023)		**CSDI** (2022)		**TimeGrad** (2022)		**GRU NVP** (2021)	
Metrics	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE
ETT (Avg)	0.235	0.293	0.250	0.317	0.236	0.295	**0.231**	**0.273**	0.290	0.316	0.366	0.377	0.273	0.310	0.304	0.389	0.493	0.619	0.476	0.605
Weather	0.089	0.114	0.087	0.105	**0.074**	**0.090**	0.087	0.102	0.142	0.158	0.179	0.143	0.096	0.106	0.077	0.093	0.125	0.155	0.119	0.147
Electricity	0.093	0.118	0.086	0.110	**0.081**	**0.097**	0.081	0.098	-	-	0.247	0.290	-	-	/	/	0.102	0.126	0.101	0.127
Traffic	0.268	0.337	0.248	**0.306**	**0.247**	0.308	-	-	0.269	0.295	-	-	0.330	0.385	/	/	0.211	0.246	0.198	0.245
Exchange	0.043	0.048	**0.040**	**0.047**	0.042	0.047	0.045	0.049	0.044	0.047	0.045	0.050	0.057	0.069	0.069	0.086	0.082	0.095	0.073	0.093
ILI	0.162	0.198	**0.130**	**0.166**	0.154	0.188	0.148	0.166	0.170	0.197	0.159	0.197	0.156	0.211	0.276	0.290	0.284	0.310	0.283	0.309
1^{st} **Count**	3	3	7	3	**15**	**14**	5	8	0	1	0	1	0	0	2	2	0	2	4	2
2^{nd} **Count**	7	5	5	5	**7**	6	5	**8**	0	2	1	1	1	0	6	5	4	2	0	2

4.3 Model Analysis

To identify the source of FLAME's generalization capability, we analyze the key components of FLAME including Local-Perception, SSD-Decoder, and Flow-based Head in Table 3. We choose ETT (Avg), Weather, Electricity and Traffic (shared in TSFM-Bench and ProbTS), and report average MSE and CRPS to evaluate deterministic and probabilistic forecasting. The results support the effectiveness of the proposed components–see Appendix B for full results, analyses, and implementation of variants.

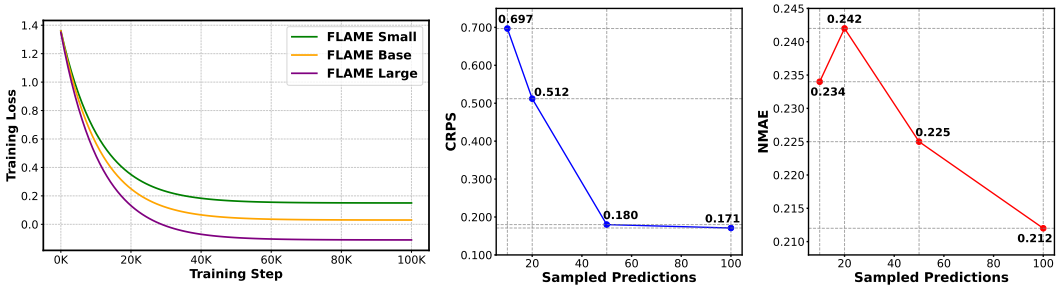
Table 3: Ablation studies on deterministic and probabilistic forecasting.

Variants	ETT (Avg)		Weather		Electricity		Traffic	
Metrics	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS
w/o Local-Perception	0.374	0.266	0.278	0.104	0.293	0.109	0.690	0.283
w/o Period Alignment	0.368	0.251	0.248	0.083	0.275	0.096	0.627	0.264
Learnable LegT	0.348	0.245	0.237	0.074	0.222	0.084	0.566	0.253
Self-Attention Decoder	0.391	0.300	0.253	0.106	0.300	0.131	0.654	0.341
Causal-Attention Decoder	0.369	0.260	0.255	0.103	0.230	0.095	0.560	0.266
Mamba v1 Decoder	0.362	0.254	0.256	0.098	0.218	0.087	0.557	0.277
VAE Head	0.366	0.262	0.264	0.093	0.291	0.138	0.745	0.506
Diffusion Head	0.360	0.252	0.230	0.077	0.235	0.088	0.568	0.266
Flow Match Head	0.358	0.251	0.233	0.080	0.232	0.088	0.595	0.297
FLAME Large	**0.344**	**0.236**	**0.227**	**0.074**	**0.217**	**0.081**	**0.546**	**0.247**

4.4 Scalability Analysis

Though the FLAME family possesses extremely lightweight backbones while achieving state-of-the-art performance, we reveal that it also possesses the potential of scalability. As shown in Figure 5a, the training curves of the FLAME family indicate that the model capacity increases with the parameter scale. According to the results in Table 1–2, the zero-shot performance also gets better and the FLAME Large achieves the most first-place results in both deterministic and probabilistic forecasting tasks.

Since FLAME models distributions over the forecasting horizon, we also study the scalability of generative forecasting by exploring the correlations between the sampling number and performance–see Figure 5b. The results show that CRPS and NMAE generally improve as the number of samples increases, with 100 samples providing the best overall performance among the tested settings.



(a) The training curves of the FLAME family. (b) Analysis on sampling number. Full results are in Table 10.

Figure 5: Scalability analysis of the FLAME family.

4.5 Efficiency Analysis

As a characteristic property of FLAME, the extreme efficiency owes to the lightweight backbone and Flow-based Head. For detailed quantification, we report the Parameter scale, Multiply-Accumulate Operations (MACs), Max GPU Memory Occupation, and Inference Time in Table 4. We compare the FLAME Large with time series foundation models and end-to-end supervised probabilistic models.

Table 4: Efficiency analysis of FLAME and other baselines on ETTm1 dataset, evaluated with the horizon of 720 and batch size of 1. We report the Parameter scale, MACs, Max GPU Memory, and Inference Time.

Models	Parameters	MACs	GPU (MB)	Inference (s)
Timer	67.4 M	52.6 G	1,435	0.08
TimesFM	200 M	624.9 G	1,395	0.16
Chronos Large	700 M	92327.9 G	10,269	34.33
MOIRAI Large	300 M	97.36 G	2,009	0.1
Time-MoE Ultra	453 M	5252.9 G	14,131	2.13
Sundial Base	128 M	432.3 G	603	0.1
TimeGrad	12.3 M	19.3 G	522	28.61
CSDI	15.9 M	37.5 G	817	86.18
FLAME Large	**10.6 M**	**0.238 G**	**129**	**0.037**

Results show FLAME Large achieves consistent efficiency advantages over other time series foundation models. Compared with end-to-end supervised models like TimeGrad and CSDI, FLAME Large also has fewer parameters, faster inference speed, and fewer MACs and lower GPU memory usage.

5 Related Work

5.1 Time Series Forecasting

Deterministic Time Series Forecasting is central to decision intelligence and has driven many deep learning methods, such as PatchTST [3], DUET [4], and SEER [47]. LMU [27] and FiLM [39] introduced Legendre Memory (LegT) into time series forecasting, but they compress the entire input series into state vectors, which may cause information loss. In contrast, FLAME leverages LegT to compress varying token-wise environmental information and align it with the original tokens, enabling adaptation to multiple frequencies in the pretraining corpus and yielding strong generalization. Another important direction is Probabilistic Time Series Forecasting, which quantifies uncertainty and is more practical in real-world applications. Recently, generative mechanisms such as VAE, Diffusion, and Flow have been widely studied. Among them, D^3VAE [48] and K^2VAE [5] are the most advanced VAE models in probabilistic time series forecasting. Considering multi-step generative mechanisms like Diffusion and Normalizing Flow, recent models include GRU NVP [31], CSDI [49], TimeGrad [33], and TSDiff [26], showing strong performance.

5.2 Time Series Foundation Models

Recent studies on Time Series Foundation Models have explored the source of generalization capability. Many works design specialized Transformer-based architectures and scale them to massive time series corpora to capture the inherent inductive bias in data. Representative deterministic forecasting models, such as TimesFM [12], Timer [11], Time-MoE [14], and VisionTS [22, 23] achieve strong zero-shot performance but rely on heavy backbones. In contrast, UniTS [13], TinyTTM [24], and ROSE [16] show that lightweight backbones can also achieve state-of-the-art deterministic forecasting performance when inductive bias is captured more efficiently. Some foundation models also support probabilistic forecasting, including Lag-Llama [50], Chronos [19], Moirai [17], and Sundial [18], which outperform several end-to-end supervised probabilistic models. However, they still rely on heavy backbones, making probabilistic forecasting on large downstream datasets inefficient. In this paper, we propose **FLAME** to address this bottleneck.

6 Conclusion

In this work, we propose FLAME, a family of lightweight yet capable time series foundation models. FLAME adapts Legendre Memory in the backbone to encode local environmental information and support adaptive memory utilization for long-term forecasting, yielding strong generalization capabilities. It also supports probabilistic time series forecasting through a normalizing-flow-based head. Comprehensive experiments show that FLAME is an accurate and efficient out-of-the-box tool for decision intelligence.

Acknowledgments and Disclosure of Funding

This work was partially supported by National Natural Science Foundation of China (62372179).

Xingjian Wu, Zhengyu Li, Hanyin Cheng, and Xiangfei Qiu contributed equally to this work. Bin Yang is the corresponding author.

References

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

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

[3] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

[4] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *SIGKDD*, pages 1185–1196, 2025.

[5] Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. K^2VAE: A koopman-kalman enhanced variational autoencoder for probabilistic time series forecasting. In *ICML*, 2025.

[6] Wei Shao, Ziquan Fang, Lu Chen, and Yunjun Gao. Towards trajectory anomaly detection: a fine-grained and noise-resilient framework. In *SIGKDD*, pages 2490–2501, 2025.

[7] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54 (10s):1–41, 2022.

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[9] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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

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

[12] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[13] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: Building a unified time series model. *arXiv*, 2024. URL `https://arxiv.org/pdf/2403.00131.pdf`.

[14] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv e-prints*, pages arXiv–2409, 2024.

[15] Yihang Wang, Yuying Qiu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Guo Chenjuan. Lightgts: A lightweight general time series forecasting model. In *ICML*, 2025.

[16] Yihang Wang, Yuying Qiu, kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Guo Chenjuan. Towards a general time series forecasting model with unified representation and adaptive transfer. In *ICML*, 2025.

[17] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

[18] Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

[19] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

[20] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One Fits All: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

[21] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[22] Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*, 2024.

[23] Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL `https://proceedings.mlr.press/v267/chen25be.html`.

[24] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181, 2024.

[25] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[26] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *NeurIPS*, 2023.

[27] Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. *Advances in neural information processing systems*, 32, 2019.

[28] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

[29] Jean-Pierre Richard. Time-delay systems: an overview of some recent advances and open problems. *automatica*, 39(10):1667–1694, 2003.

[30] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[31] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *ICLR*, 2021.

[32] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

[33] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *ICML*, volume 139, pages 8857–8868, 2021.

[34] Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, and Bin Yang. TSFM-Bench: Comprehensive and unified benchmarking of foundation models for time series forecasting. In *SIGKDD*, 2025.

[35] Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. In *NeurIPS*, 2024.

[36] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. In *Proc. VLDB Endow.*, pages 2363–2377, 2024.

[37] Henri Padé. Sur la représentation approchée d'une fonction par des fractions rationnelles. In *Annales scientifiques de l'Ecole normale supérieure*, volume 9, pages 3–93, 1892.

[38] Aaron Russell Voelker. Dynamical systems in spiking neuromorphic hardware. 2019.

[39] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems*, 35:12677–12690, 2022.

[40] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[41] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2021.

[42] Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting. In *IJCAI*, pages 1994–2001, 2022.

[43] E Oran Brigham and RE Morrow. The fast fourier transform. *IEEE spectrum*, 4(12):63–70, 1967.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[45] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[46] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[47] Anonymous Authors. Seer: Transformer-based robust time series forecasting via automated patch enhancement and replacement. Concurrent Submission to ICML, 2026. Filename: SEER_ICML_2026.pdf.

[48] Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35:23009–23022, 2022.

[49] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, pages 24804–24816, 2021.

[50] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[51] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[52] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[53] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.

[54] Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. Contrast everything: A hierarchical contrastive framework for medical time-series. *NeurIPS*, 36, 2024.

[55] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *NeurIPS*, 35:5816–5828, 2022.

[56] Michael W McCracken and Serena Ng. Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

[57] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083, 2012.

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[59] Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. TAB: Unified benchmarking of time series anomaly detection methods. In *Proc. VLDB Endow.*, pages 2775–2789, 2025.

[60] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

[61] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL `https://arxiv.org/abs/2006.11239`.

[62] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

[63] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL `https://doi.org/10.1109/ICCV51070.2023.00387`.

[64] Xingjian Wu, Jianxin Jin, Wanghui Qiu, Peng Chen, Yang Shu, Bin Yang, and Chenjuan Guo. Aurora: Towards universal generative multimodal time series forecasting. In *ICLR*, 2026.

[65] Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. In *NeurIPS*, 2025.

[66] Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *NeurIPS*, 2025.

[67] Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *AAAI*, 2026.

[68] Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. In *ICML*, 2026.

[69] Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. In *ICLR*, 2026.

[70] Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, Zhulin An, Qi Wang, and Yongjun Xu. Ginar+: A robust end-to-end framework for multivariate time series forecasting with missing values. *IEEE Transactions on Knowledge and Data Engineering*, 37 (8):4635–4648, 2025.

[71] Chengqing Yu, Fei Wang, Chuanguang Yang, Zezhi Shao, Tao Sun, Tangwen Qian, Wei Wei, Zhulin An, and Yongjun Xu. Merlin: Multi-view representation learning for robust multivariate time series forecasting with unfixed missing rates. In *SIGKDD*, pages 3633–3644, 2025.

[72] Xiangfei Qiu, Yuhan Zhu, Zhengyu Li, Xingjian Wu, Bin Yang, and Jilin Hu. DAG: A dual correlation network for time series forecasting with exogenous variables. In *ICML*, 2026.

[73] Zhengyu Li, Xiangfei Qiu, Yuhan Zhu, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. GCGNet: Graph-consistent generative network for time series forecasting with exogenous variables. In *ICLR*, 2026.

[74] Hanyin Cheng, Jingrong Zhou, Yang Shu, and Chenjuan Guo. KITE: Knowledge-guided probabilistic modeling for time series forecasting with exogenous variables. In *ICML*, 2026.

[75] Fang Cheng, Hui Liu, and Xinwei Lv. Metagnsdformer: Meta-learning enhanced gated non-stationary informer with frequency-aware attention for point-interval remaining useful life prediction of lithium-ion batteries. *Advanced Engineering Informatics*, 69:103798, 2026.

[76] Hanyin Cheng, Xingjian Wu, Xiangfei Qiu, Yang Shu, Bin Yang, and Chenjuan Guo. CCD: Capturing cross-correlations with deformable convolutional networks for multivariate time series forecasting. In *KDD*, 2026.

[77] Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *ICLR*, 2025.

[78] Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. CoRA: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. In *ICLR*, 2026.

[79] Hanyin Cheng, Ruitong Zhang, Yuning Lu, Peng Chen, Meng Wang, Yang Shu, Bin Yang, and Chenjuan Guo. STAR: Boosting time series foundation models for anomaly detection through state-aware adapter. In *NeurIPS*, 2026.

[80] Xingjian Wu, Junkai Lu, Zhengyu Li, Xiangfei Qiu, Jilin Hu, Chenjuan Guo, Christian S Jensen, and Bin Yang. TimeART: Towards agentic time series reasoning via tool-augmentation. *arXiv preprint arXiv:2601.13653*, 2026.

[81] Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Xuanhua He, Chenyang Zhu, Hongyu Liu, Yingqing He, et al. Controllable video generation: A survey. *arXiv preprint arXiv:2507.16869*, 2025.

[82] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Siran Chen, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. In *AAAI*, volume 38, pages 4117–4125, 2024.

[83] Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, et al. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. In *SIGGRAPH Asia*, pages 1–12, 2024.

[84] Yue Ma, Jiangming Wang, Yucheng Wang, Xilai Wang, Zhiyuan Li, Xinyu Wang, Hongyu Liu, Ruofan Liang, Songchun Zhang, Yuxuan Xue, et al. Livelight: Real-time streaming video relighting with interactive control. *arXiv preprint arXiv:2608.01771*, 2026.

A Implementation Details

A.1 Pretraining Corpus

To pre-train our model, we utilize a broad collection of time series datasets drawn from multiple sources, including selected subsets from the Monash [1], UEA [51], and UCR [52] repositories, along with several widely-used benchmarks [53–57]. A full inventory of these datasets is provided in Table 5. We ensure that none of the pre-training datasets overlap with those used in downstream evaluations. The pretraining Weather dataset is univariate, whereas the downstream Weather dataset contains multivariate sequences. We categorize the pre-training datasets into six domain-specific groups: Energy, Economic, Health, Transport, Web, and Nature. These datasets vary significantly in sampling rates, ranging from milliseconds to monthly intervals, capturing the diversity and richness of real-world time series. All datasets are decomposed into univariate sequences and FLAME is trained using a Channel Independent (CI) strategy.

Table 5: List of pretraining datasets.

Domain	Dataset	Frequency	Point Number	Source
Energy	Aus. Electricity Demand	Half Hourly	1155264	Monash [1]
	Wind	4 Seconds	7397147	Monash [1]
	Wind Farms	Minutely	172178060	Monash [1]
	Solar Power	4 Seconds	7397222	Monash [1]
	London Smart Meters	Half Hourly	166527216	Monash [1]
Economic	FRED_MD	Monthly	77896	[56]
	Bitcoin	Daily	75364	Monash [1]
	NN5	Daily	87801	[57]
Health	MotorImagery	0.001 Seconds	72576000	UEA [51]
	SelfRegulationSCP1	0.004 Seconds	3015936	UEA [51]
	SelfRegulationSCP2	0.004 Seconds	3064320	UEA [51]
	AtrialFibrillation	0.008 Seconds	38400	UEA [51]
	PigArtPressure	-	624000	UCR [52]
	PIGCVP	-	624000	UCR [52]
	TDbrain	0.002 Seconds	79232703	[54]
Transport	Pems03	5 Minute	9382464	[55]
	Pems07	5 Minute	24921792	[55]
	Pems-bay	5 Minute	16937700	[55]
	Pedestrian_Counts	Hourly	3132346	Monash [1]
Web	Web Traffic	Daily	116485589	Monash [1]
Nature	Phoneme	-	2160640	UCR [52]
	EigenWorms	-	27947136	UEA [51]
	PRSA	Hourly	4628448	[53]
	Temperature Rain	Daily	23252200	Monash [1]
	StarLightCurves	-	9457664	UCR [52]
	Worms	0.033 Seconds	232200	UCR [52]
	Saugeen River Flow	Daily	23741	Monash [1]
	Sunspot	Daily	73924	Monash [1]
	Weather	Daily	43032000	Monash [1]
	KDD Cup 2018	Daily	2942364	Monash [1]
	US Births	Daily	7305	Monash [1]

A.2 Baselines

For zero-shot deterministic forecasting, we compare the FLAME family against 8 advanced foundation models: Sundial [18], ROSE [16], Timer [11], MOIRAI [17], Chronos [19], LightGTS [15], Time-MoE [14], and TinyTTM [24]. For zero-shot probabilistic forecasting, we compare the FLAME family with 4 foundation models: Sundial [18], MOIRAI [17], Chronos [19], and Lag-Llama [50],

and 4 advanced end-to-end supervised models including TSDiff [26], TimeGrad [33], CSDI [49], and GRU NVP [31]. The corresponding codebases and implementation details are summarized in Table 6.

Table 6: Code repositories for baselines.

Model Types	Models	Code Repositories
End2End	TSDiff	https://github.com/amazon-science/unconditional-time-series-diffusion
	CSDI	https://github.com/ermongroup/CSDI
	TimeGrad	https://github.com/Zjh152/TimeGrad
	GRU NVP	https://github.com/microsoft/ProbTS
Foundation	Sundial	https://github.com/thuml/Sundial
	ROSE	https://github.com/decisionintelligence/TSFM-Bench
	Timer	https://github.com/thuml/Large-Time-Series-Model
	MOIRAI	https://github.com/redoules/moirai
	Chronos	https://github.com/amazon-science/chronos-forecasting
	LightGTS	https://github.com/decisionintelligence/LightGTS
	Time-MoE	https://github.com/Time-MoE/Time-MoE
	Lag-Llama	https://github.com/time-series-foundation-models/lag-llama
	TinyTTM	https://huggingface.co/ibm-granite/granite-timeseries-ttm-r1

A.3 Benchmarks

To thoroughly assess the effectiveness of FLAME, we conduct experiments on TSFM-Bench [34] and ProbTS [35]. Our evaluation spans 11 deterministic and 9 probabilistic forecasting datasets. In FLAME's flow-based head, samples are drawn i.i.d. from the base distribution and processed through layers to reach the target distribution. For probabilistic forecasting, we draw 100 random samples; for deterministic predictions, we average these 100 samples.

For deterministic forecasting, we use ETTm1, ETTm2, ETTh1, ETTh2, Weather, Electricity, Traffic, Solar, PEMS08, Wind, and NYSE from TSFM-Bench. For most datasets, the prediction length is set to $L \in \{96, 192, 336, 720\}$, while NYSE uses $L \in \{24, 36, 48, 60\}$.

For probabilistic forecasting, we adopt ETTm1, ETTm2, ETTh1, ETTh2, Weather, Electricity, Traffic, Exchange, and ILI from ProbTS. The prediction length is set to $L \in \{96, 192, 336, 720\}$ for most datasets, while ILI uses $L \in \{24, 36, 48, 60\}$.

All models are configured with the contextual length of the best performance recommended in their papers. It is important to clarify that datasets shared in both the TSFM-Bench and ProbTS (e.g., ETTh1, Traffic) are different in evaluation settings like strides. A summary of dataset statistics can be found in Table 7.

Table 7: Statistics of benchmark datasets.

Dataset	Domain	Frequency	Lengths	Dim	Split	Stride	Benchmark	Description
ETTm1	Electricity	15 mins	57,600	7	6:2:2	1	TSFM-Bench	Power transformer 1, comprising seven indicators such as oil temperature and useful load
ETTm2	Electricity	15 mins	57,600	7	6:2:2	1	TSFM-Bench	Power transformer 2, comprising seven indicators such as oil temperature and useful load
ETTh1	Electricity	1 hour	14,400	7	6:2:2	1	TSFM-Bench	Power transformer 1, comprising seven indicators such as oil temperature and useful load
ETTh2	Electricity	1 hour	14,400	7	6:2:2	1	TSFM-Bench	Power transformer 2, comprising seven indicators such as oil temperature and useful load
Weather	Environment	10 mins	52,696	21	7:1:2	1	TSFM-Bench	Recorded every 10 minutes throughout 2020, with 21 meteorological indicators
Electricity	Electricity	1 hour	26,304	321	7:1:2	1	TSFM-Bench	Electricity records the electricity consumption in kWh every 1 hour from 2012 to 2014
Traffic	Traffic	1 hour	17,544	862	7:1:2	1	TSFM-Bench	Road occupancy rates measured by 862 sensors on San Francisco Bay area freeways
Solar	Energy	10 mins	52,560	137	6:2:2	1	TSFM-Bench	Solar production records collected from 137 PV plants in Alabama
PEMS08	Traffic	5 mins	17,856	170	6:2:2	1	TSFM-Bench	Traffic flow time series collected from the CalTrans PeMS
Wind	Energy	15 mins	48,673	7	7:1:2	1	TSFM-Bench	Wind power records from 2020-2021 at 15-minute intervals
NYSE	Stock	1 day	1,243	5	7:1:2	1	TSFM-Bench	Records opening price, closing price, trading volume, lowest price, and highest price
ETTm1	Electricity	15 mins	57,600	7	6:2:2	96	ProbTS	Power transformer 1, comprising seven indicators such as oil temperature and useful load
ETTm2	Electricity	15 mins	57,600	7	6:2:2	96	ProbTS	Power transformer 2, comprising seven indicators such as oil temperature and useful load
ETTh1	Electricity	1 hour	14,400	7	6:2:2	96	ProbTS	Power transformer 1, comprising seven indicators such as oil temperature and useful load
ETTh2	Electricity	1 hour	14,400	7	6:2:2	96	ProbTS	Power transformer 2, comprising seven indicators such as oil temperature and useful load
Weather	Environment	10 mins	52,696	21	7:1:2	96	ProbTS	Recorded every 10 minutes throughout 2020, with 21 meteorological indicators
Electricity	Electricity	1 hour	26,304	321	7:1:2	96	ProbTS	Electricity records the electricity consumption in kWh every 1 hour from 2012 to 2014
Traffic	Traffic	1 hour	17,544	862	7:1:2	96	ProbTS	Road occupancy rates measured by 862 sensors on San Francisco Bay area freeways
Exchange	Economic	1 day	7,588	8	7:1:2	96	ProbTS	ExchangeRate collects the daily exchange rates of eight countries
ILI	Health	1 week	966	7	7:1:2	96	ProbTS	Recorded indicators of patient data from Centers for Disease Control and Prevention

A.4 Experimental Settings

Pretraining We implement the FLAME family using Distributed Data Parallel in PyTorch [58], and conduct all experiments on 8 NVIDIA A800 GPUs with 80GB GPU memory. The model is

optimized using the AdamW optimizer with an initial learning rate of 5×10^{-5}. To progressively reduce the learning rate during training, we employ a step decay schedule via the StepLR scheduler. During pre-training, we use 12 historical tokens and 4 prediction tokens, with a reference patch size $p = 48$. The batch size is set to 8,192. Full details on model configurations and parameter sizes are summarized in Table 8.

Table 8: Detailed model configurations of the FLAME family and corresponding parameter counts.

Models	Encoder Layers	Decoder Layers	Flow Layers	Model Dim d	FFN Dim d'	Parameters
FLAME Small	1	1	3	256	512	2M
FLAME Base	1	3	5	256	512	6M
FLAME Large	1	6	7	256	512	10M

Downstream Forecasting For downstream forecasting tasks, we apply periodic patching strategies adapted to each dataset's temporal characteristics. The number of past tokens remains fixed at 11, while the prediction lengths are evaluated at 96, 192, 336, and 720 steps.

We also address the *"Drop Last"* issue highlighted in recent studies [36, 59, 34], where setting *drop_last*=True during test evaluation may lead to misleading results due to incomplete batches. To ensure consistency and fairness, we set *drop_last*=False for all baseline models in our experiments.

A.5 Evaluation Metrics

Regarding evaluation metrics, following the experimental setup in TSFM-Bench, we adopt Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics for deterministic forecasting. For probabilistic forecasting, we use Continuous Ranked Probability Score (CRPS) and Normalized Mean Absolute Error (NMAE) in ProbTS. Consider a forecast with K variables and a prediction horizon of T steps.

Mean Squared Error (MSE) The Mean Squared Error quantifies the average of the squared differences between predicted values and their corresponding ground truths. It penalizes larger errors more heavily due to the squaring operation, making it sensitive to outliers. Formally, it is defined as:

$$\text{MSE} = \frac{1}{K \times T} \sum_{k=1}^{K} \sum_{t=1}^{T} (x_t^k - \hat{x}_t^k)^2, \tag{21}$$

where K denotes the number of variables, T the prediction horizon, x_t^k the true value, and $\hat{x}_t^k$ the predicted value.

Mean Absolute Error (MAE) The Mean Absolute Error calculates the average magnitude of prediction errors without considering their direction. By focusing on the absolute differences, MAE provides a robust and interpretable measure of accuracy:

$$\text{MAE} = \frac{1}{K \times T} \sum_{k=1}^{K} \sum_{t=1}^{T} |x_t^k - \hat{x}_t^k|, \tag{22}$$

where all terms follow the same definition as above. Unlike MSE, MAE treats all errors equally and is less sensitive to large deviations.

Continuous Ranked Probability Score (CRPS) The CRPS assesses the quality of probabilistic forecasts by comparing the predicted cumulative distribution function (CDF) F with the observed outcome x. It is computed as:

$$\text{CRPS} = \int_{\mathbb{R}} (F(z) - \mathbb{I}\{x \leq z\})^2 dz, \tag{23}$$

where $\mathbb{I}\{x \leq z\}$ is the indicator function. CRPS evaluates both calibration and sharpness; its expected value is minimized when the predictive distribution matches the true conditional distribution. In practice, we approximate CRPS using the empirical CDF $\hat{F}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{X_i \leq z\}$ based on $n = 100$ scalar samples X_i drawn from the corresponding predictive marginal.

Normalized Mean Absolute Error (NMAE) The NMAE extends MAE by normalizing it against the total magnitude of ground-truth values, thereby enabling fair comparison across datasets with varying scales. Its formula is:

$$\text{NMAE} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T}|x_t^k - \hat{x}_t^k|}{\sum_{k=1}^{K}\sum_{t=1}^{T}|x_t^k|} \tag{24}$$

B Full Model Analysis

B.1 The Convergence of LegT based Local Perception

In FLAME, θ is a varying parameter related to the input's periodicity. Empirically, common dataset periods (e.g., ETT, Electricity, Traffic) are typically 24, 48, 96, or 168, which are smaller than $d = 256$. Theorem 2.1 gives a projection error bound that decreases with d for fixed window length and smoothness; the condition $d > \theta$ alone does not guarantee a small error for the discretized memory. To validate this, we randomly select 100 time-series segments for each length $\theta \in \{24, 48, 96, 168\}$, compress them using LegT with model sizes $d \in \{64, 128, 256, 512\}$, then reconstruct them and report the average Mean Squared Error. The experiments and visual showcases are provided in Table 9 and Figure 6 . We observe that $d = 256$ sufficiently maintains a low approximation error.

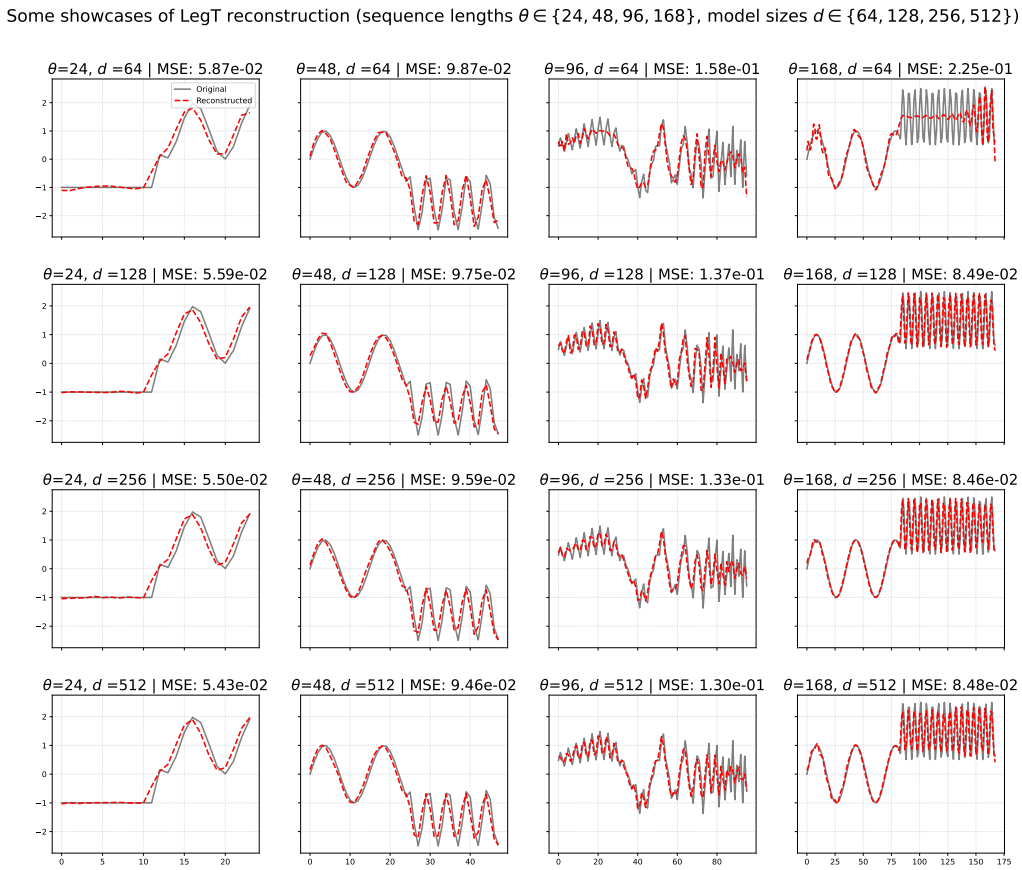


Figure 6: The reconstruction results of LegT based Local Perception Module.

Table 9: MSE values of LegT reconstruction results. Each value is calculated over 100 time series samples. We observe that $d = 256$ is sufficient to maintain low approximation error under common periods $\theta \in \{24, 48, 96, 168\}$. Although $d = 512$ further improves accuracy, the improvement is marginal. Therefore, we choose $d = 256$ to balance approximation accuracy and efficiency.

Seq. Length θ	$d = 64$	$d = 128$	$d = 256$	$d = 512$
24	0.088	0.084	**0.083**	**0.083**
48	0.124	0.121	0.118	**0.117**
96	0.183	0.146	0.141	**0.139**
168	0.233	0.094	0.093	**0.092**

B.2 Analysis of Sampling Number

Given that FLAME models distributions across the forecasting horizon, we conduct an in-depth study on the scalability of generative forecasting by examining the correlations between the sampling number and performance, as presented in Table 10. Specifically, we choose four sampling numbers: 10, 20, 50, and 100. The findings indicate that CRPS and NMAE generally improve as the number of samples increases, although individual settings do not improve monotonically. Moreover, a sampling number of 100 leads to excellent performance, and this is unlikely to impose significant efficiency burdens in practical applications.

Table 10: Full results of probabilistic forecasting experiments with different sampling numbers.

Models		FLAME-10		FLAME-20		FLAME-50		FLAME-100	
Metric		CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE
ETTm1	96	1.255	0.412	0.388	0.477	0.323	0.418	**0.315**	**0.407**
	192	0.905	0.444	0.372	0.433	0.358	0.432	**0.337**	**0.426**
	336	0.369	0.468	0.448	0.682	0.352	0.448	**0.351**	**0.447**
	720	0.388	0.486	0.394	0.491	0.394	0.488	**0.383**	**0.476**
	avg	0.729	0.453	0.401	0.521	0.357	0.447	**0.347**	**0.439**
ETTm2	96	0.134	0.173	0.129	0.167	0.125	0.163	**0.124**	**0.155**
	192	0.208	0.196	0.187	0.190	0.144	0.186	**0.143**	**0.175**
	336	0.477	0.212	0.427	0.208	0.159	0.204	**0.158**	**0.193**
	720	0.662	0.236	0.631	0.232	0.187	0.229	**0.184**	**0.219**
	avg	0.370	0.204	0.344	0.199	0.154	0.196	**0.152**	**0.186**
ETTh1	96	0.299	0.346	0.269	0.331	0.262	0.324	**0.259**	**0.310**
	192	0.294	0.365	0.285	0.354	0.279	0.347	**0.278**	**0.342**
	336	0.309	0.383	0.298	0.372	0.293	0.365	**0.290**	**0.362**
	720	0.328	0.406	0.317	0.396	0.311	0.390	**0.303**	**0.367**
	avg	0.308	0.375	0.292	0.363	0.286	0.357	**0.283**	**0.345**
ETTh2	96	0.158	0.200	0.150	0.191	0.146	0.188	**0.144**	**0.186**
	192	0.167	0.211	0.162	0.207	0.160	0.206	**0.159**	**0.204**
	336	0.181	0.226	0.176	0.222	0.174	0.221	**0.172**	**0.217**
	720	0.191	0.234	0.187	0.231	0.183	0.229	**0.182**	**0.228**
	avg	0.174	0.218	0.169	0.213	0.166	0.211	**0.164**	**0.209**
Weather	96	0.080	0.098	0.178	0.093	0.073	0.091	**0.067**	**0.083**
	192	0.103	0.112	0.092	0.105	0.084	0.103	**0.068**	**0.082**
	336	1.382	0.121	0.233	0.117	0.092	0.114	**0.079**	**0.094**
	720	2.471	0.133	1.384	0.129	0.101	0.127	**0.083**	**0.100**
	avg	1.009	0.116	0.472	0.111	0.088	0.109	**0.074**	**0.090**
Electricity	96	1.734	0.097	0.072	0.092	0.070	0.090	**0.069**	**0.089**
	192	0.093	0.119	1.662	0.116	0.086	0.112	**0.072**	**0.088**
	336	0.662	0.147	0.375	0.143	0.107	0.140	**0.081**	**0.096**
	720	2.406	0.173	1.354	0.168	0.125	0.166	**0.100**	**0.116**
	avg	1.224	0.134	0.866	0.130	0.097	0.127	**0.081**	**0.097**
Traffic	96	0.562	0.301	0.491	0.309	0.224	0.273	**0.220**	**0.268**
	192	0.811	0.324	0.681	0.333	0.240	0.298	**0.238**	**0.294**
	336	2.911	0.348	3.284	0.364	0.257	0.323	**0.254**	**0.319**
	720	3.491	0.378	2.824	0.484	0.278	0.354	**0.276**	**0.351**
	avg	1.944	0.338	1.820	0.373	0.250	0.312	**0.247**	**0.308**
Exchange	96	0.023	0.027	0.022	0.027	0.022	0.027	**0.020**	**0.026**
	192	0.032	0.037	0.032	0.037	0.031	0.037	**0.031**	**0.035**
	336	0.045	0.049	0.044	0.049	0.044	0.049	**0.044**	**0.046**
	720	1.234	0.098	0.145	0.082	0.074	0.079	**0.073**	**0.079**
	avg	0.334	0.053	0.061	0.049	0.043	0.048	**0.042**	**0.047**
ILI	24	0.156	0.188	0.157	0.197	0.157	0.194	**0.136**	**0.168**
	36	0.178	0.216	0.181	0.225	0.177	0.214	**0.145**	**0.179**
	48	0.197	0.234	0.201	0.244	0.196	0.235	**0.164**	**0.200**
	60	0.196	0.234	0.190	0.231	0.192	0.233	**0.169**	**0.204**
	avg	0.182	0.218	0.182	0.224	0.181	0.219	**0.154**	**0.188**

B.3 Analysis of Local-Perception

As shown in Table 11, we conduct ablation studies to assess the effectiveness of the proposed Local-Perception module. Specifically, we consider three variants: removing the entire Local-Perception module, denoted as "w/o Local-Perception"; disabling Period Alignment, denoted as "w/o Period Alignment"; and making the LegT matrices A and B learnable, denoted as "Learnable LegT".

The results demonstrate that Local-Perception plays a crucial role in zero-shot forecasting. Removing the entire module leads to the most severe degradation, indicating that Period Alignment and LegT jointly contribute to constructing informative local environments for patch-wise tokens. We further observe that Period Alignment is especially important for datasets with complex periodic structures, such as Electricity and Traffic. Without Period Alignment, the model fixes the local window length θ to the patch size and thus fails to anchor each token to its period-aware local environment, resulting in substantially worse zero-shot performance.

In addition, the Learnable LegT variant also underperforms the default design with fixed matrices. This suggests that the analytically initialized LegT matrices preserve a stable Legendre basis for online function approximation, while making A and B learnable may shift the underlying orthogonal space during large-scale pretraining and weaken the approximation capability. Overall, these results verify that the effectiveness of Local-Perception comes from the complementary design of Period Alignment and fixed LegT: Period Alignment determines an appropriate period-aware environment boundary, and LegT compresses the aligned environment into a fixed-size representation without losing its online approximation property.

Table 11: Experimental results on Local Perception analysis, comparing configurations without Local Perception (LP), without Period Alignment (P.A.), and with Learnable LegT (L.L.).

Models		w/o LP		w/o P.A.		with L.L.		FLAME Large	
Metric		MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS
ETTm1	96	0.365	0.352	0.362	0.332	**0.297**	0.317	0.301	**0.315**
	192	0.389	0.385	0.387	0.364	**0.327**	**0.335**	0.328	0.337
	336	0.417	0.407	0.408	0.398	0.353	0.359	**0.351**	**0.351**
	720	0.470	0.462	0.483	0.423	0.397	0.390	**0.392**	**0.383**
	avg	0.410	0.402	0.410	0.379	0.344	0.350	**0.343**	**0.347**
ETTm2	96	0.229	0.144	0.216	0.133	**0.178**	**0.121**	0.186	0.124
	192	0.283	0.154	0.258	0.149	0.253	0.146	**0.247**	**0.143**
	336	0.311	0.186	0.304	0.171	**0.297**	0.164	0.300	**0.158**
	720	0.362	0.200	0.363	0.195	0.365	0.191	**0.357**	**0.184**
	avg	0.296	0.171	0.285	0.162	0.273	0.156	**0.273**	**0.152**
ETTh1	96	0.372	0.283	0.362	0.279	**0.352**	**0.255**	0.358	0.259
	192	0.418	0.304	0.402	0.291	0.408	0.291	**0.392**	**0.278**
	336	0.433	0.322	0.428	0.300	0.413	0.304	**0.411**	**0.290**
	720	0.450	0.376	0.459	0.318	**0.438**	0.328	0.444	**0.303**
	avg	0.418	0.321	0.413	0.297	0.403	0.295	**0.401**	**0.283**
ETTh2	96	0.312	0.148	0.303	0.145	0.311	0.161	**0.300**	**0.144**
	192	0.368	0.167	0.358	0.159	0.364	0.178	**0.351**	**0.159**
	336	0.384	0.182	0.375	0.174	0.391	0.199	**0.374**	**0.172**
	720	0.425	0.188	0.422	0.186	0.422	0.185	**0.416**	**0.182**
	avg	0.372	0.171	0.365	0.166	0.372	0.181	**0.360**	**0.164**
Weather	96	0.233	0.076	0.195	0.071	0.168	0.067	**0.152**	**0.067**
	192	0.270	0.099	0.225	0.074	0.214	0.069	**0.199**	**0.068**
	336	0.295	0.112	0.261	0.088	0.255	**0.078**	**0.250**	0.079
	720	0.312	0.127	0.311	0.098	0.312	**0.081**	**0.307**	0.083
	avg	0.278	0.104	0.248	0.083	0.237	**0.074**	**0.227**	0.074
Electricity	96	0.214	0.088	0.202	0.077	0.166	0.072	**0.161**	**0.069**
	192	0.253	0.098	0.237	0.091	0.198	0.073	**0.190**	**0.072**
	336	0.309	0.112	0.288	0.104	0.233	0.085	**0.227**	**0.081**
	720	0.397	0.138	0.374	0.112	0.292	0.105	**0.289**	**0.100**
	avg	0.293	0.109	0.275	0.096	0.222	0.084	**0.217**	**0.081**
Traffic	96	0.599	0.239	0.557	0.229	0.503	0.222	**0.480**	**0.220**
	192	0.631	0.275	0.592	0.251	0.526	0.246	**0.511**	**0.238**
	336	0.722	0.301	0.647	0.282	0.589	0.265	**0.553**	**0.254**
	720	0.808	0.318	0.712	0.294	0.647	0.279	**0.639**	**0.276**
	avg	0.690	0.283	0.627	0.264	0.566	0.253	**0.546**	**0.247**

B.4 Analysis of SSD-Decoder

Subsequently, we compare the SSD-Decoder with several representative decoder variants in Table 12, including a Self-Attention Decoder, a Causal-Attention Decoder, and a Mamba v1 Decoder. The main objective of these variants is to validate whether LegS provides a more suitable mechanism for the decoding stage, which is expected to support long-term forecasting while maintaining computational efficiency. For fair comparison, we adopt the standard implementations of these commonly used structures and keep the hidden dimension consistent with that of the SSD-Decoder.

The results show that the SSD-Decoder achieves the best overall performance. In forecasting tasks, the decoding process should respect temporal causality and effectively utilize historical memory for future prediction. Therefore, the Self-Attention Decoder, which does not explicitly impose causal constraints, yields the weakest performance. Compared with the Causal-Attention Decoder and the Mamba v1 Decoder, the SSD-Decoder further improves forecasting accuracy, demonstrating its stronger capability in adaptive memory utilization and long-range temporal modeling. These results suggest that the LegS-based SSD mechanism is better suited for the decoder design in FLAME.

For transparency, we report the key hyperparameters of each decoder variant in Table 13. All variants use the same hidden dimension and number of decoder layers to ensure a fair comparison.

Table 12: Forecasting performance comparison of different decoders: Self-Attention (SA), Causal-Attention (CA), Mamba v1, and our SSD-Decoder used in FLAME.

Models		SA		CA		Mamba v1		FLAME	
Metric		MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS
ETTm1	96	0.389	0.448	0.377	0.358	0.337	0.332	**0.301**	**0.315**
	192	0.421	0.467	0.395	0.382	0.359	0.341	**0.328**	**0.337**
	336	0.449	0.485	0.413	0.401	0.390	0.367	**0.351**	**0.351**
	720	0.491	0.508	0.461	0.442	0.444	0.390	**0.392**	**0.383**
	avg	0.438	0.477	0.412	0.396	0.383	0.358	**0.343**	**0.347**
ETTm2	96	0.235	0.167	0.209	0.141	0.212	0.139	**0.186**	**0.124**
	192	0.272	0.198	0.251	0.167	0.256	0.162	**0.247**	**0.143**
	336	0.352	0.223	0.313	0.182	**0.292**	0.179	0.300	**0.158**
	720	0.398	0.257	0.368	0.224	**0.356**	0.198	0.357	**0.184**
	avg	0.314	0.211	0.285	0.179	0.279	0.170	**0.273**	**0.152**
ETTh1	96	0.358	0.269	**0.354**	0.263	0.363	0.271	0.358	**0.259**
	192	0.404	0.302	0.395	0.286	0.404	0.303	**0.392**	**0.278**
	336	0.420	0.331	0.423	0.297	0.428	0.325	**0.411**	**0.290**
	720	0.455	0.352	0.475	0.325	0.468	0.344	**0.444**	**0.303**
	avg	0.409	0.314	0.412	0.293	0.416	0.311	**0.401**	**0.283**
ETTh2	96	0.336	0.175	**0.288**	**0.139**	0.303	0.148	0.300	0.144
	192	0.382	0.203	0.362	0.170	0.370	0.172	**0.351**	**0.159**
	336	0.414	0.212	0.389	0.177	0.401	0.186	**0.374**	**0.172**
	720	0.472	0.209	0.432	0.198	**0.409**	0.202	0.416	**0.182**
	avg	0.401	0.200	0.368	0.171	0.371	0.177	**0.360**	**0.164**
Weather	96	0.188	0.088	0.209	0.093	0.207	0.087	**0.152**	**0.067**
	192	0.221	0.094	0.237	0.097	0.237	0.091	**0.199**	**0.068**
	336	0.275	0.113	0.261	0.105	0.269	0.102	**0.250**	**0.079**
	720	0.328	0.127	0.311	0.117	0.311	0.112	**0.307**	**0.083**
	avg	0.253	0.106	0.255	0.103	0.256	0.098	**0.227**	**0.074**
Electricity	96	0.210	0.115	0.172	0.075	0.163	0.072	**0.161**	**0.069**
	192	0.255	0.127	0.204	0.084	0.191	0.077	**0.190**	**0.072**
	336	0.318	0.132	0.243	0.098	0.234	0.090	**0.227**	**0.081**
	720	0.415	0.149	0.302	0.122	**0.285**	0.108	0.289	**0.100**
	avg	0.300	0.131	0.230	0.095	0.218	0.087	**0.217**	**0.081**
Traffic	96	0.587	0.311	0.492	0.238	0.501	0.244	**0.480**	**0.220**
	192	0.617	0.338	0.521	0.256	0.522	0.270	**0.511**	**0.238**
	336	0.662	0.343	0.572	0.275	0.562	0.294	**0.553**	**0.254**
	720	0.749	0.372	0.655	0.294	0.642	0.298	**0.639**	**0.276**
	avg	0.654	0.341	0.560	0.266	0.557	0.277	**0.546**	**0.247**

Table 13: Key hyperparameters of different decoder variants. All variants use the same hidden dimension and number of decoder layers for fair comparison.

Self-Attention	
Hyperparameter	Value
Decoder Type	Self-Attention
Layers	6
Hidden Dim.	256
FFN Dim.	512
Num Heads	8
Head Dim.	32
Dropout	0.1
Causal Mask	No
Pos. Encoding	Learnable
Norm.	LayerNorm

Causal-Attention	
Hyperparameter	Value
Decoder Type	Causal-Attention
Layers	6
Hidden Dim.	256
FFN Dim.	512
Num Heads	8
Head Dim.	32
Dropout	0.1
Causal Mask	Yes
Pos. Encoding	Learnable
Norm.	LayerNorm

Mamba v1	
Hyperparameter	Value
Decoder Type	Mamba v1
Layers	6
Hidden Dim.	256
State Dim.	16
Conv. Kernel	4
Expansion	2
Num Heads	–
Head Dim.	–
Dropout	0.1
Causal Modeling	Yes
Norm.	RMSNorm

SSD-Decoder	
Hyperparameter	Value
Decoder Type	Mamba v2 / SSD
Layers	6
Hidden Dim.	256
State Dim.	64
Conv. Kernel	4
Expansion	2
SSM Heads	8
Head Dim.	32
LegS Order	256
Dropout	0.1
Causal Modeling	Yes
Memory	LegS-based SSD
Norm.	RMSNorm

B.5 Analysis of Flow-based Head

To assess the efficacy of the proposed Flow-based Head, we compare it with several representative generative forecasting heads in Table 15, including Variational Autoencoder (VAE) [60], Diffusion [61], and Flow Matching (FM) [62]. The main objective of this ablation is to examine whether the proposed normalizing-flow-based design is more suitable for probabilistic forecasting than other commonly used generative mechanisms, under comparable conditioning and hidden-size settings.

For a fair and informative comparison, we implement all alternative heads with concise but standard designs. Specifically, the VAE baseline is implemented as a conditional VAE (CVAE) decoder [60], where the decoder receives the forecasting condition token from the backbone and generates the target patch from the latent variable, and it adopts the MLPs as network structures. For Diffusion [61] and Flow Matching [62], we adopt a DiT-style conditional architecture [63], where the condition is injected through AdaLN-Zero, while the Transformer blocks are replaced by lightweight MLP blocks to keep the parameter scale comparable to our forecasting head. For the Diffusion baseline, we use a DDPM-style probability path. For the Flow Matching baseline, we adopt the optimal-transport path used in Sundial [18] and Aurora [64], which is a commonly used probability path for learning continuous transformations between the base and data distributions. These settings ensure that the baselines follow standard generative modeling principles while remaining lightweight enough to serve as forecasting heads. Note that the VAE, Flow Matching, and Diffusion all adopt the same hidden dimension $d = 256$ as the FLAME's Flow-based Head, all other parameters follow the standard design and maintain an appropriate multiple relationship with the hidden dimension.

In the proposed Flow-based Head, samples are independently drawn from the base Gaussian distribution and then transformed through a sequence of autoregressive affine layers to obtain samples from the target forecasting distribution. For probabilistic forecasting, we randomly sample 100 times from the base distribution. The deterministic prediction is then obtained by averaging the 100 generated samples, which provides an empirical estimate of the predictive mean.

The results show that the proposed Flow-based Head achieves the best overall performance. The VAE baseline follows a one-step generation paradigm, which makes it less effective on datasets with complex real-world temporal dynamics, such as Electricity and Traffic. Although Diffusion and Flow Matching provide more expressive multi-step generation mechanisms, they introduce additional sampling dynamics and rely heavily on the choice of probability paths, solvers, and velocity or denoising networks. In contrast, our Flow-based Head directly learns an invertible transformation from the base distribution to the target forecasting distribution through lightweight autoregressive affine layers. This design supports efficient probabilistic sampling while preserving fine-grained dependencies within each predicted patch. These results suggest that the normalizing-flow-based head provides a more suitable balance between generative expressiveness, inference efficiency, and forecasting accuracy for FLAME.

To further clarify the implementation details of the generative forecasting heads, we report their key hyperparameters in Table 14. For fair comparison, all variants adopt the same hidden dimension $d = 256$, the same forecasting condition token, and the same number of network layers. The main differences lie in their generative mechanisms, conditioning strategies, and sampling procedures.

Table 14: Key hyperparameters of different generative forecasting heads. All variants use the same hidden dimension $d = 256$, the same forecasting condition token, and the same number of layers for fair comparison.

VAE Head	
Hyperparameter	Value
Generative Type	CVAE
Layers	7
Hidden Dim.	256
Latent Dim.	128
Encoder Network	MLP
Decoder Network	MLP
Condition Injection	Concatenation
Prior Distribution	Gaussian
Sampling Steps	1
Training Objective	ELBO
Output Samples	100

Diffusion Head	
Hyperparameter	Value
Generative Type	DDPM
Layers	7
Hidden Dim.	256
Network Type	MLP Blocks
Condition Injection	AdaLN-Zero
Probability Path	DDPM-style
Noise Schedule	Linear
Prediction Target	Noise
Sampling Steps	100
Training Objective	Denoising Loss
Output Samples	100

Flow Matching Head	
Hyperparameter	Value
Generative Type	Flow Matching
Layers	7
Hidden Dim.	256
Network Type	MLP Blocks
Condition Injection	AdaLN-Zero
Probability Path	OT Path
Solver	Euler
Prediction Target	Velocity
Sampling Steps	100
Training Objective	FM Loss
Output Samples	100

Flow-based Head	
Hyperparameter	Value
Generative Type	Normalizing Flow
Layers	7
Hidden Dim.	256
Base Distribution	Gaussian
Flow Transform	Autoregressive Affine
Condition Injection	Masked MLP
Jacobian Type	Triangular
Sampling Steps	p per flow layer
Training Objective	NLL
Output Samples	100

Table 15: Forecasting performance comparison of various generative heads: VAE, Diffusion, Flow Matching (FM), and our Flow-based Head used in FLAME. We also compare FLAME with vanilla Gaussian Head.

Models		Gaussian		VAE		Diffusion		FM		FLAME	
Metric		MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS
ETTm1	96	0.342	0.357	0.334	0.348	0.305	0.318	0.311	0.323	**0.301**	**0.315**
	192	0.362	0.379	0.364	0.382	0.340	0.345	0.331	0.340	**0.328**	**0.337**
	336	0.395	0.413	0.388	0.397	0.359	0.362	0.361	0.360	**0.351**	**0.351**
	720	0.427	0.463	0.421	0.448	0.401	0.390	0.398	0.386	**0.392**	**0.383**
	avg	0.382	0.403	0.377	0.394	0.351	0.354	0.350	0.352	**0.343**	**0.347**
ETTm2	96	0.231	0.166	0.223	0.152	0.194	0.128	0.215	0.139	**0.186**	**0.124**
	192	0.312	0.198	0.292	0.179	0.261	0.145	0.257	0.144	**0.247**	**0.143**
	336	0.372	0.225	0.345	0.197	0.322	0.163	0.309	0.168	**0.300**	**0.158**
	720	0.413	0.231	0.392	0.215	0.370	0.200	0.381	0.205	**0.357**	**0.184**
	avg	0.332	0.205	0.313	0.186	0.287	0.159	0.291	0.164	**0.273**	**0.152**
ETTh1	96	0.374	0.291	0.366	0.279	0.359	0.261	0.363	0.267	**0.358**	**0.259**
	192	0.408	0.304	0.398	0.285	0.394	0.288	0.404	0.295	**0.392**	**0.278**
	336	0.441	0.344	0.423	0.301	0.428	0.318	0.427	0.330	**0.411**	**0.290**
	720	0.488	0.372	0.457	0.320	0.479	0.348	0.462	0.327	**0.444**	**0.303**
	avg	0.420	0.328	0.411	0.296	0.415	0.304	0.414	0.305	**0.401**	**0.283**
ETTh2	96	0.354	0.193	**0.296**	**0.142**	0.332	0.169	0.328	0.164	0.300	0.144
	192	0.395	0.206	0.358	0.167	0.378	0.180	0.364	0.168	**0.351**	**0.159**
	336	0.405	0.217	0.379	0.182	0.394	0.198	0.384	0.191	**0.374**	**0.172**
	720	0.472	0.246	0.425	0.197	0.448	0.215	0.428	0.202	**0.416**	**0.182**
	avg	0.407	0.216	0.365	0.172	0.388	0.191	0.376	0.181	**0.360**	**0.164**
Weather	96	0.186	0.097	0.176	0.083	**0.149**	0.068	0.155	0.072	0.152	**0.067**
	192	0.237	0.098	0.229	0.086	0.199	0.069	0.206	0.074	**0.199**	**0.068**
	336	0.317	0.114	0.292	0.098	0.256	0.083	0.257	0.084	**0.250**	**0.079**
	720	0.369	0.118	0.357	0.105	0.314	0.089	0.312	0.088	**0.307**	**0.083**
	avg	0.277	0.107	0.264	0.093	0.230	0.077	0.233	0.080	**0.227**	**0.074**
Electricity	96	0.255	0.141	0.241	0.129	0.172	0.078	0.167	0.075	**0.161**	**0.069**
	192	0.263	0.144	0.260	0.134	0.199	0.079	0.195	0.079	**0.190**	**0.072**
	336	0.312	0.148	0.304	0.142	0.251	0.088	0.246	0.086	**0.227**	**0.081**
	720	0.367	0.155	0.357	0.147	0.318	0.105	0.318	0.110	**0.289**	**0.100**
	avg	0.299	0.147	0.291	0.138	0.235	0.088	0.232	0.088	**0.217**	**0.081**
Traffic	96	0.785	0.533	0.721	0.482	0.488	0.236	0.507	0.248	**0.480**	**0.220**
	192	0.796	0.542	0.742	0.498	0.528	0.264	0.557	0.296	**0.511**	**0.238**
	336	0.817	0.579	0.747	0.512	0.572	0.270	0.603	0.305	**0.553**	**0.254**
	720	0.838	0.601	0.769	0.532	0.682	0.293	0.712	0.337	**0.639**	**0.276**
	avg	0.809	0.564	0.745	0.506	0.568	0.266	0.595	0.297	**0.546**	**0.247**

C Full Results

C.1 Full zero-shot results

We provide all the results of the zero-shot deterministic forecasting in Table 16. As shown in Table 16, we include 11 representative real-world datasets from TSFM-Bench, demonstrating that FLAME achieves state-of-the-art forecasting performance.

We provide all the results of the zero-shot probabilistic forecasting in Table 17. As shown in Table 17, we include 9 representative real-world datasets from ProbTS, demonstrating that FLAME achieves state-of-the-art forecasting performance.

C.2 Full-shot and few-shot results

We further compare FLAME Small with end-to-end supervised models. We provide all the results of deterministic forecasting in Table 21, comparing FLAME Small of different training budgets with most advanced end-to-end deterministic forecasting models.

Table 16: Full results of zero-shot deterministic forecasting experiments on datasets from TSFM-Bench. Lower MSE or MAE values indicate better predictions. ("-") denotes datasets included in the model's pretraining and therefore excluded from testing. **Red**: the best, Blue: the 2nd best.

Models		**FLAME Small**		**FLAME Base**		**FLAME Large**		Sundial (2025)		LightGTS (2025)		ROSE (2025)		Timer (2024)		Chronos (2024)		Time-MoE (2024)		TinyTTM (2024)		MOIRAI (2024)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.305	0.353	0.298	0.351	0.301	0.353	**0.280**	**0.334**	0.307	0.359	0.512	0.460	0.817	0.611	0.402	0.373	0.281	0.341	0.738	0.541	0.353	0.363
	192	0.328	0.370	0.324	0.369	0.328	0.373	0.321	0.366	0.332	0.374	0.512	0.462	0.927	0.659	0.510	0.435	**0.305**	**0.358**	0.698	0.547	0.376	0.380
	336	0.349	**0.384**	**0.346**	0.386	0.351	0.389	0.350	0.389	0.353	0.393	0.523	0.470	1.043	0.704	0.590	0.477	0.369	0.395	0.670	0.533	0.399	0.395
	720	0.392	0.416	**0.385**	0.414	0.392	0.419	0.394	0.418	0.388	**0.386**	0.552	0.490	1.044	0.722	0.703	0.525	0.469	0.472	0.660	0.550	0.432	0.417
	Avg	0.344	0.381	0.338	0.380	0.343	0.384	**0.336**	**0.377**	0.345	0.378	0.525	0.471	0.958	0.674	0.551	0.453	0.356	0.392	0.692	0.543	0.390	0.389
ETTm2	96	**0.165**	**0.255**	0.171	0.262	0.186	0.272	0.170	0.256	0.166	0.260	0.224	0.309	0.225	0.300	0.192	0.263	0.198	0.288	0.226	0.309	0.189	0.260
	192	**0.217**	**0.297**	0.227	0.302	0.247	0.315	0.229	0.300	0.222	0.300	0.266	0.333	0.286	0.339	0.256	0.308	0.235	0.312	0.311	0.360	0.247	0.300
	336	**0.262**	**0.330**	0.280	0.340	0.300	0.352	0.281	0.337	0.267	0.333	0.310	0.358	0.335	0.369	0.315	0.346	0.293	0.348	0.350	0.383	0.295	0.334
	720	**0.333**	0.379	0.348	0.387	0.357	0.394	0.351	0.387	0.340	**0.378**	0.395	0.407	0.414	0.416	0.409	0.405	0.427	0.428	0.446	0.435	0.372	0.386
	Avg	**0.244**	**0.315**	0.257	0.323	0.273	0.333	0.258	0.320	0.249	0.318	0.299	0.352	0.315	0.356	0.293	0.331	0.288	0.344	0.333	0.372	0.276	0.320
ETTh1	96	0.362	0.395	0.361	0.392	0.358	0.389	**0.348**	0.385	0.359	0.390	0.382	0.408	0.454	0.434	0.444	0.409	0.349	**0.379**	0.364	0.389	0.380	0.398
	192	0.406	0.427	0.400	0.422	0.392	0.416	0.393	0.418	0.390	0.409	0.400	0.420	0.522	0.465	0.502	0.443	0.395	0.413	**0.386**	**0.407**	0.440	0.434
	336	0.436	0.449	0.417	0.436	0.411	0.432	0.422	0.440	0.420	0.436	0.404	0.426	0.559	0.484	0.580	0.460	0.447	0.453	**0.404**	**0.422**	0.514	0.474
	720	0.471	0.482	0.433	0.459	0.444	0.463	0.481	0.493	0.437	0.460	**0.420**	**0.447**	0.714	0.549	0.605	0.495	0.457	0.462	0.424	0.448	0.705	0.568
	Avg	0.419	0.438	0.403	0.427	0.401	0.425	0.411	0.434	0.402	0.424	0.402	0.425	0.562	0.483	0.533	0.452	0.412	0.427	**0.395**	**0.417**	0.510	0.469
ETTh2	96	0.278	0.339	0.295	0.345	0.300	0.350	**0.271**	0.333	0.278	0.342	0.298	0.362	0.316	0.359	0.306	0.338	0.292	0.352	0.277	0.335	0.287	**0.325**
	192	0.336	0.381	0.353	0.388	0.351	0.390	**0.327**	0.376	0.345	0.383	0.336	0.385	0.374	0.398	0.396	0.394	0.347	0.379	0.334	0.373	0.347	**0.367**
	336	0.373	0.410	0.373	0.409	0.374	0.411	0.354	0.402	0.394	0.416	**0.353**	0.399	0.381	0.410	0.423	0.417	0.406	0.419	0.362	0.402	0.377	**0.393**
	720	0.411	0.441	0.413	0.440	0.416	0.444	**0.381**	0.435	0.430	0.445	0.395	0.432	0.408	0.434	0.442	0.439	0.439	0.447	0.408	0.444	0.404	**0.421**
	Avg	0.350	0.393	0.359	0.396	0.360	0.399	**0.333**	0.387	0.362	0.397	0.346	0.395	0.370	0.400	0.392	0.397	0.371	0.399	0.345	0.389	0.354	**0.377**
Weather	96	**0.148**	**0.204**	0.151	0.209	0.152	0.211	0.157	0.205	0.151	0.211	0.200	0.260	0.190	0.236	0.186	0.208	0.157	0.211	0.183	0.242	0.177	0.208
	192	**0.189**	**0.243**	0.194	0.250	0.199	0.254	0.205	0.251	0.191	0.248	0.239	0.288	0.261	0.293	0.238	0.258	0.208	0.256	0.229	0.285	0.219	0.249
	336	**0.233**	**0.278**	0.241	0.286	0.250	0.293	0.253	0.289	0.236	0.284	0.279	0.315	0.332	0.340	0.313	0.353	0.255	0.290	0.289	0.330	0.277	0.292
	720	**0.289**	**0.320**	0.301	0.330	0.307	0.333	0.320	0.336	0.299	0.323	0.340	0.357	0.385	0.381	0.416	0.415	0.405	0.397	0.359	0.370	0.365	0.350
	Avg	**0.215**	**0.261**	0.222	0.269	0.227	0.273	0.234	0.270	0.219	0.267	0.265	0.305	0.292	0.313	0.288	0.309	0.256	0.289	0.265	0.307	0.260	0.275
Electricity	96	0.206	0.294	0.170	0.269	0.161	0.261	**0.132**	**0.229**	0.181	0.272	0.209	0.307	0.210	0.312	-	-	-	-	0.166	0.263	0.152	0.242
	192	0.247	0.338	0.198	0.298	0.190	0.292	**0.152**	**0.250**	0.195	0.284	0.219	0.315	0.239	0.337	-	-	-	-	0.191	0.286	0.171	0.259
	336	0.310	0.398	0.230	0.331	0.227	0.326	**0.173**	**0.271**	0.253	0.338	0.236	0.330	0.284	0.372	-	-	-	-	0.237	0.336	0.192	0.278
	720	0.383	0.455	0.288	0.379	0.289	0.372	**0.218**	**0.311**	0.304	0.382	0.273	0.328	0.456	0.479	-	-	-	-	0.292	0.384	0.236	0.313
	Avg	0.287	0.371	0.222	0.319	0.217	0.313	**0.169**	**0.265**	0.233	0.319	0.234	0.320	0.297	0.375	-	-	-	-	0.222	0.317	0.188	0.273
Traffic	96	0.591	0.401	0.522	0.360	**0.480**	**0.344**	-	-	0.525	0.356	0.572	0.407	0.526	0.368	0.562	0.378	-	-	0.514	0.347	-	-
	192	0.623	0.431	0.538	0.374	**0.511**	0.369	-	-	0.547	**0.365**	0.575	0.406	0.561	0.385	0.579	0.412	-	-	0.543	0.373	-	-
	336	0.697	0.485	0.574	0.400	**0.553**	0.394	-	-	0.645	0.419	0.588	0.411	0.614	0.412	0.594	0.420	-	-	0.575	**0.389**	-	-
	720	0.782	0.528	0.669	0.449	0.639	0.429	-	-	0.722	0.457	**0.618**	**0.422**	0.749	0.464	0.723	0.472	-	-	0.622	0.433	-	-
	Avg	0.673	0.461	0.576	0.396	**0.546**	**0.384**	-	-	0.610	0.399	0.588	0.412	0.613	0.407	0.615	0.421	-	-	0.564	0.386	-	-
Solar	96	0.185	0.263	0.167	0.259	**0.153**	0.255	0.204	**0.230**	0.209	0.309	0.524	0.557	0.591	0.504	0.373	0.304	0.304	0.345	0.863	0.664	0.682	0.688
	192	0.201	0.276	0.180	0.273	**0.165**	0.269	0.221	**0.248**	0.220	0.315	0.507	0.550	0.689	0.567	0.363	0.303	0.309	0.342	0.823	0.695	0.694	0.695
	336	0.212	0.283	0.190	0.282	**0.174**	0.278	0.225	**0.260**	0.228	0.314	0.508	0.553	0.831	0.636	0.391	0.319	0.433	0.450	0.835	0.741	0.719	0.706
	720	0.213	0.291	0.198	0.295	**0.179**	0.287	0.233	**0.272**	0.218	0.283	0.479	0.534	0.972	0.710	0.444	0.349	0.599	0.576	0.738	0.738	0.759	0.725
	Avg	0.203	0.278	0.184	0.277	**0.168**	0.272	0.221	**0.252**	0.219	0.305	0.505	0.549	0.771	0.604	0.393	0.319	0.411	0.428	0.815	0.710	0.714	0.704
PEMS08	96	0.249	0.357	0.241	0.353	**0.224**	**0.331**	-	-	0.608	0.562	1.373	0.995	0.625	0.580	1.538	0.983	-	-	1.284	0.888	-	-
	192	0.299	0.375	0.288	0.368	**0.273**	**0.342**	-	-	0.774	0.648	1.365	0.979	0.798	0.661	1.719	0.983	-	-	1.638	1.039	-	-
	336	0.330	0.376	0.324	0.367	**0.305**	**0.348**	-	-	0.892	0.693	1.338	0.960	0.910	0.716	1.768	0.998	-	-	1.979	1.160	-	-
	720	0.418	0.428	0.395	0.406	**0.368**	**0.384**	-	-	0.975	0.864	1.401	0.980	1.131	0.824	1.802	1.133	-	-	2.020	1.175	-	-
	Avg	0.324	0.384	0.312	0.374	**0.293**	**0.351**	-	-	0.812	0.692	1.369	0.979	0.866	0.695	1.707	1.024	-	-	1.730	1.066	-	-
Wind	96	0.948	0.663	0.906	0.650	**0.867**	**0.634**	0.931	0.646	0.988	0.678	1.072	0.747	0.946	0.659	1.273	0.738	-	-	1.077	0.701	0.992	0.656
	192	1.107	0.746	1.062	0.735	**1.005**	**0.714**	1.136	0.751	1.275	0.862	1.209	0.804	1.142	0.758	1.439	0.817	-	-	1.350	0.825	1.221	0.765
	336	1.215	0.801	1.208	0.801	**1.097**	**0.765**	1.285	0.820	1.332	0.897	1.318	0.848	1.300	0.830	1.550	0.869	-	-	1.473	0.891	1.403	0.844
	720	1.307	0.849	1.303	0.847	**1.197**	**0.817**	1.390	0.870	1.572	0.906	1.404	0.881	1.417	0.884	1.649	0.914	-	-	1.447	0.899	1.581	0.915
	Avg	1.144	0.765	1.120	0.758	**1.042**	**0.733**	1.186	0.772	1.292	0.836	1.251	0.820	1.201	0.783	1.478	0.834	-	-	1.337	0.829	1.299	0.795
NYSE	24	0.323	0.429	**0.240**	**0.327**	0.301	0.385	0.370	0.393	0.362	0.403	-	-	0.388	0.409	0.480	0.449	-	-	-	-	-	-
	36	0.451	0.520	**0.361**	**0.391**	0.437	0.461	0.681	0.566	0.574	0.592	-	-	0.778	0.613	0.912	0.655	-	-	-	-	-	-
	48	0.571	0.593	**0.487**	**0.446**	0.589	0.538	1.043	0.729	0.688	0.708	-	-	1.200	0.792	1.371	0.818	-	-	-	-	-	-
	60	0.762	0.671	**0.681**	**0.557**	0.789	0.644	1.425	0.880	0.855	0.802	-	-	1.584	1.000	1.754	0.958	-	-	-	-	-	-
	Avg	0.527	0.553	**0.442**	**0.430**	0.529	0.507	0.880	0.642	0.620	0.626	-	-	0.988	0.704	1.129	0.720	-	-	-	-	-	-
1st Count		8	8	6	4	**15**	**9**	9	9	0	3	3	2	0	0	0	0	1	2	2	3	0	4
2nd Count		3	1	**15**	9	2	**10**	2	7	9	6	2	3	0	0	0	0	2	1	5	3	4	4

Table 17: Full results of zero-shot probabilistic forecasting experiments on datasets from ProbTS. Lower CRPS or NMAE values indicate better predictions. ("-") denotes datasets included in the model's pretraining and therefore excluded from testing. ("/") denotes the excessive time consumption. **Red**: the best, Blue: the 2nd best.

Models		**FLAME Small**		**FLAME Base**		**FLAME Large**		**Sundial** (2025)		**Chronos** (2024)		**MOIRAI** (2024)		**Lag-Llama** (2023)		**TSDiff** (2023)		**CSDI** (2022)		**TimeGrad** (2022)		**GRU NVP** (2021)	
Metrics		CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE
ETTm1	96	0.294	0.368	0.326	0.417	0.315	0.407	0.253	0.308	0.360	0.422	0.464	0.522	0.354	0.402	0.344	0.441	**0.236**	**0.308**	0.522	0.645	0.383	0.488
	192	0.332	0.420	0.363	0.469	0.337	0.426	**0.279**	**0.337**	0.404	0.450	0.467	0.531	0.368	0.415	0.345	0.441	0.291	0.377	0.603	0.748	0.396	0.514
	336	0.341	0.427	0.393	0.511	0.351	0.447	**0.291**	**0.350**	0.425	0.456	0.524	0.558	0.387	0.436	0.462	0.571	0.322	0.419	0.601	0.759	0.486	0.630
	720	0.364	0.454	0.477	0.622	0.383	0.476	**0.318**	**0.380**	0.461	0.478	0.514	0.535	0.403	0.466	0.478	0.622	0.448	0.578	0.621	0.793	0.546	0.707
	avg	0.333	0.417	0.390	0.505	0.347	0.439	**0.285**	**0.344**	0.413	0.451	0.492	0.536	0.378	0.430	0.407	0.519	0.324	0.421	0.587	0.736	0.453	0.585
ETTm2	96	0.120	0.153	0.121	0.153	0.124	0.155	0.128	0.153	0.134	0.158	0.176	0.186	0.163	0.192	0.175	0.224	**0.115**	**0.146**	0.427	0.525	0.319	0.413
	192	0.146	0.182	0.147	0.184	**0.143**	**0.175**	0.150	0.177	0.163	0.183	0.197	0.207	0.181	0.207	0.255	0.316	0.147	0.189	0.424	0.530	0.326	0.427
	336	0.163	0.202	0.164	0.205	**0.158**	**0.193**	0.167	0.195	0.190	0.204	0.229	0.227	0.206	0.229	0.328	0.397	0.190	0.248	0.469	0.566	0.449	0.580
	720	0.185	0.223	0.186	0.231	**0.184**	0.219	0.189	**0.217**	0.223	0.230	0.321	0.258	0.227	0.249	0.344	0.416	0.239	0.306	0.470	0.561	0.561	0.749
	avg	0.154	0.190	0.155	0.193	**0.152**	**0.186**	0.158	0.186	0.178	0.194	0.231	0.220	0.194	0.219	0.276	0.338	0.173	0.222	0.448	0.546	0.414	0.542
ETTh1	96	0.265	0.322	0.261	0.319	**0.259**	0.310	0.260	**0.307**	0.293	0.341	0.469	0.488	0.297	0.340	0.395	0.510	0.437	0.557	0.455	0.585	0.379	0.481
	192	0.288	0.346	0.282	0.351	**0.278**	**0.342**	0.297	0.347	0.348	0.384	0.496	0.492	0.312	0.356	0.467	0.596	0.496	0.625	0.516	0.680	0.425	0.531
	336	0.300	0.371	0.297	0.368	**0.290**	**0.362**	0.314	0.365	0.384	0.409	0.503	0.489	0.326	0.370	0.450	0.581	0.454	0.574	0.512	0.666	0.458	0.580
	720	0.320	0.402	0.312	0.386	**0.303**	0.367	0.312	**0.366**	0.413	0.425	0.526	0.508	0.340	0.405	0.516	0.657	0.528	0.657	0.523	0.672	0.502	0.643
	avg	0.293	0.360	0.288	0.356	**0.283**	**0.345**	0.296	0.346	0.360	0.390	0.498	0.494	0.319	0.368	0.457	0.586	0.479	0.603	0.502	0.651	0.441	0.559
ETTh2	96	**0.139**	**0.177**	0.145	0.185	0.144	0.186	0.157	0.186	0.163	0.191	0.212	0.238	0.178	0.204	0.336	0.421	0.164	0.214	0.358	0.448	0.432	0.548
	192	**0.152**	**0.195**	0.161	0.207	0.159	0.204	0.183	0.213	0.195	0.218	0.229	0.250	0.193	0.217	0.265	0.339	0.226	0.294	0.457	0.575	0.625	0.766
	336	**0.170**	**0.216**	0.174	0.224	0.172	0.217	0.203	0.232	0.226	0.241	0.253	0.263	0.211	0.230	0.350	0.427	0.274	0.353	0.481	0.606	0.793	0.942
	720	0.183	0.231	0.191	0.241	**0.182**	**0.228**	0.203	0.233	0.255	0.263	0.279	0.273	0.222	0.238	0.406	0.482	0.302	0.382	0.445	0.550	0.539	0.688
	avg	**0.161**	**0.205**	0.168	0.214	0.164	0.209	0.186	0.216	0.210	0.228	0.243	0.256	0.201	0.222	0.339	0.417	0.242	0.311	0.435	0.545	0.597	0.736
Weather	96	0.071	0.088	0.071	0.085	**0.067**	**0.083**	0.078	0.091	0.136	0.155	0.161	0.141	0.085	0.094	0.104	0.113	0.068	0.087	0.130	0.164	0.116	0.145
	192	0.089	0.113	0.081	0.100	**0.068**	**0.082**	0.083	0.097	0.145	0.165	0.173	0.139	0.094	0.102	0.134	0.144	0.068	0.086	0.127	0.158	0.122	0.147
	336	0.094	0.121	0.090	0.110	**0.079**	**0.094**	0.090	0.106	0.136	0.151	0.149	0.134	0.098	0.109	0.137	0.138	0.083	0.098	0.130	0.162	0.128	0.160
	720	0.103	0.132	0.106	0.125	**0.083**	**0.100**	0.096	0.113	0.151	0.161	0.233	0.155	0.107	0.120	0.152	0.141	0.087	0.102	0.113	0.136	0.110	0.135
	avg	0.089	0.114	0.087	0.105	**0.074**	**0.090**	0.087	0.102	0.142	0.158	0.179	0.143	0.096	0.106	0.132	0.134	0.077	0.093	0.125	0.155	0.119	0.147
Electricity	96	0.071	0.090	**0.066**	0.084	0.069	0.089	0.068	**0.083**	-	-	0.231	0.275	-	-	0.344	0.441	0.153	0.203	0.096	0.119	0.094	0.118
	192	0.086	0.103	0.076	0.098	**0.072**	**0.088**	0.074	0.090	-	-	0.236	0.279	-	-	0.345	0.441	0.200	0.264	0.100	0.124	0.097	0.121
	336	0.096	0.125	0.091	0.117	**0.081**	**0.096**	0.083	0.100	-	-	0.245	0.289	-	-	0.462	0.571	/	/	0.102	0.126	0.099	0.123
	720	0.117	0.154	0.110	0.141	0.100	**0.116**	**0.098**	0.118	-	-	0.274	0.318	-	-	0.478	0.622	/	/	0.108	0.134	0.114	0.144
	avg	0.093	0.118	0.086	0.110	**0.081**	**0.097**	0.081	0.098	-	-	0.247	0.290	-	-	0.407	0.519	/	/	0.102	0.126	0.101	0.127
Traffic	96	0.240	0.293	0.229	0.277	0.220	0.268	-	-	0.250	0.289	-	-	0.255	0.297	0.294	0.342	/	/	0.202	0.234	**0.187**	**0.231**
	192	0.257	0.319	0.239	0.291	0.238	0.294	-	-	0.249	0.278	-	-	0.295	0.343	0.306	0.354	/	/	0.208	0.239	**0.192**	**0.236**
	336	0.271	0.341	0.250	0.311	0.254	0.319	-	-	0.269	0.290	-	-	0.335	0.384	0.317	0.392	/	/	0.213	**0.246**	**0.201**	0.248
	720	0.304	0.395	0.274	0.346	0.276	0.351	-	-	0.310	0.321	-	-	0.434	0.517	0.391	0.478	/	/	0.220	**0.263**	**0.211**	0.264
	avg	0.268	0.337	0.248	0.306	0.247	0.308	-	-	0.269	0.295	-	-	0.330	0.385	0.327	0.392	/	/	0.211	0.246	**0.198**	**0.245**
Exchange	96	0.023	0.028	0.021	0.027	**0.020**	0.026	0.024	0.027	0.021	**0.025**	0.025	0.030	0.042	0.051	0.079	0.090	0.028	0.036	0.068	0.079	0.071	0.091
	192	0.032	0.037	**0.030**	0.036	0.031	**0.035**	0.034	0.038	0.032	0.036	0.034	0.039	0.047	0.058	0.093	0.106	0.045	0.058	0.087	0.100	0.068	0.087
	336	0.044	0.049	**0.041**	0.047	0.044	**0.046**	0.046	0.050	0.045	0.048	0.047	0.052	0.061	0.073	0.081	0.106	0.060	0.076	0.074	0.086	0.072	0.091
	720	0.074	0.079	**0.069**	**0.077**	0.073	0.079	0.076	0.079	0.078	0.080	0.073	0.081	0.078	0.094	0.082	0.142	0.143	0.173	0.099	0.113	0.079	0.103
	avg	0.043	0.048	**0.040**	0.047	0.042	**0.047**	0.045	0.049	0.044	0.047	0.045	0.050	0.057	0.069	0.084	0.111	0.069	0.086	0.082	0.095	0.073	0.093
ILI	24	0.135	0.166	0.111	0.143	0.136	0.168	**0.108**	**0.123**	0.120	0.139	0.150	0.196	0.135	0.173	0.228	0.242	0.250	0.263	0.275	0.296	0.257	0.283
	36	0.164	0.200	**0.119**	**0.153**	0.145	0.179	0.152	0.169	0.179	0.205	0.171	0.222	0.163	0.227	0.235	0.246	0.285	0.298	0.272	0.298	0.281	0.307
	48	0.176	0.216	**0.139**	**0.178**	0.164	0.200	0.164	0.184	0.186	0.215	0.151	0.184	0.171	0.233	0.265	0.275	0.285	0.301	0.295	0.320	0.288	0.314
	60	0.171	0.211	**0.149**	0.189	0.169	0.204	0.168	0.190	0.196	0.228	0.163	**0.188**	0.156	0.211	0.263	0.272	0.283	0.299	0.295	0.325	0.307	0.333
	avg	0.162	0.198	**0.130**	**0.166**	0.154	0.188	0.148	0.166	0.170	0.197	0.159	0.197	0.156	0.211	0.248	0.259	0.276	0.290	0.284	0.310	0.283	0.309
1st Count		3	3	7	3	**15**	**14**	5	8	0	1	0	1	0	0	0	0	2	2	0	2	4	2
2nd Count		7	5	5	5	**7**	6	5	**8**	0	2	1	1	1	0	0	0	6	5	4	2	0	2

We also provide all the results of probabilistic forecasting in Table 20, comparing FLAME Small of different training budgets with most advanced end-to-end probabilistic forecasting models.

Table 18: Full results of zero-shot deterministic forecasting experiments on datasets from TSFM-Bench. Lower MSE or MAE values indicate better predictions. Compared with Table 16, we exclude the datasets used for pre-training several baseline models. **Red**: the best, Blue: the 2nd best.

Models		**FLAME Small**		**FLAME Base**		**FLAME Large**		Sundial (2025)		LightGTS (2025)		ROSE (2025)		Timer (2024)		Chronos (2024)		Time-MoE (2024)		TinyTTM (2024)		MOIRAI (2024)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.305	0.353	0.298	0.351	0.301	0.353	**0.280**	**0.334**	0.307	0.359	0.512	0.460	0.817	0.611	0.402	0.373	0.281	0.341	0.738	0.541	0.353	0.363
	192	0.328	0.370	0.324	0.369	0.328	0.373	0.321	0.366	0.332	0.374	0.512	0.462	0.927	0.659	0.510	0.435	**0.305**	**0.358**	0.698	0.547	0.376	0.380
	336	0.349	**0.384**	**0.346**	0.386	0.351	0.389	0.350	0.389	0.353	0.393	0.523	0.470	1.043	0.704	0.590	0.477	0.369	0.395	0.670	0.533	0.399	0.395
	720	0.392	0.416	**0.385**	0.414	0.392	0.419	0.394	0.418	0.388	**0.386**	0.552	0.490	1.044	0.722	0.703	0.525	0.469	0.472	0.660	0.550	0.432	0.417
	Avg	0.344	0.381	0.338	0.380	0.343	0.384	**0.336**	**0.377**	0.345	0.378	0.525	0.471	0.958	0.674	0.551	0.453	0.356	0.392	0.692	0.543	0.390	0.389
ETTm2	96	**0.165**	**0.255**	0.171	0.262	0.186	0.272	0.170	0.256	0.166	0.260	0.224	0.309	0.225	0.300	0.192	0.263	0.198	0.288	0.226	0.309	0.189	0.260
	192	**0.217**	**0.297**	0.227	0.302	0.247	0.315	0.229	0.300	0.222	0.300	0.266	0.333	0.286	0.339	0.256	0.308	0.235	0.312	0.311	0.360	0.247	0.300
	336	**0.262**	**0.330**	0.280	0.340	0.300	0.352	0.281	0.337	0.267	0.333	0.310	0.358	0.335	0.369	0.315	0.346	0.293	0.348	0.350	0.383	0.295	0.334
	720	**0.333**	0.379	0.348	0.387	0.357	0.394	0.351	0.387	0.340	**0.378**	0.395	0.407	0.414	0.416	0.409	0.405	0.427	0.428	0.446	0.435	0.372	0.386
	Avg	**0.244**	**0.315**	0.257	0.323	0.273	0.333	0.258	0.320	0.249	0.318	0.299	0.352	0.315	0.356	0.293	0.331	0.288	0.344	0.333	0.372	0.276	0.320
ETTh1	96	0.362	0.395	0.361	0.392	0.358	0.389	**0.348**	0.385	0.359	0.390	0.382	0.408	0.454	0.434	0.444	0.409	0.349	**0.379**	0.364	0.389	0.380	0.398
	192	0.406	0.427	0.400	0.422	0.392	0.416	0.393	0.418	0.390	0.409	0.400	0.420	0.522	0.465	0.502	0.443	0.395	0.413	**0.386**	**0.407**	0.440	0.434
	336	0.436	0.449	0.417	0.436	0.411	0.432	0.422	0.440	0.420	0.436	0.404	0.426	0.559	0.484	0.580	0.460	0.447	0.453	**0.404**	**0.422**	0.514	0.474
	720	0.471	0.482	0.433	0.459	0.444	0.463	0.481	0.493	0.437	0.460	**0.420**	**0.447**	0.714	0.549	0.605	0.495	0.457	0.462	0.424	0.448	0.705	0.568
	Avg	0.419	0.438	0.403	0.427	0.401	0.425	0.411	0.434	0.402	0.424	0.402	0.425	0.562	0.483	0.533	0.452	0.412	0.427	**0.395**	**0.417**	0.510	0.469
ETTh2	96	0.278	0.339	0.295	0.345	0.300	0.350	**0.271**	0.333	0.278	0.342	0.298	0.362	0.316	0.359	0.306	0.338	0.292	0.352	0.277	0.335	0.287	**0.325**
	192	0.336	0.381	0.353	0.388	0.351	0.390	**0.327**	0.376	0.345	0.383	0.336	0.385	0.374	0.398	0.396	0.394	0.347	0.379	0.334	0.373	0.347	**0.367**
	336	0.373	0.410	0.373	0.409	0.374	0.411	0.354	0.402	0.394	0.416	**0.353**	0.399	0.381	0.410	0.423	0.417	0.406	0.419	0.362	0.402	0.377	**0.393**
	720	0.411	0.441	0.413	0.440	0.416	0.444	**0.381**	0.435	0.430	0.445	0.395	0.432	0.408	0.434	0.442	0.439	0.439	0.447	0.408	0.444	0.404	**0.421**
	Avg	0.350	0.393	0.359	0.396	0.360	0.399	**0.333**	0.387	0.362	0.397	0.346	0.395	0.370	0.400	0.392	0.397	0.371	0.399	0.345	0.389	0.354	**0.377**
Weather	96	**0.148**	**0.204**	0.151	0.209	0.152	0.211	0.157	0.205	0.151	0.211	0.200	0.260	0.190	0.236	0.186	0.208	0.157	0.211	0.183	0.242	0.177	0.208
	192	**0.189**	**0.243**	0.194	0.250	0.199	0.254	0.205	0.251	0.191	0.248	0.239	0.288	0.261	0.293	0.238	0.258	0.208	0.256	0.229	0.285	0.219	0.249
	336	**0.233**	**0.278**	0.241	0.286	0.250	0.293	0.253	0.289	0.236	0.284	0.279	0.315	0.332	0.340	0.313	0.353	0.255	0.290	0.289	0.330	0.277	0.292
	720	**0.289**	**0.320**	0.301	0.330	0.307	0.333	0.320	0.336	0.299	0.323	0.340	0.357	0.385	0.381	0.416	0.415	0.405	0.397	0.359	0.370	0.365	0.350
	Avg	**0.215**	**0.261**	0.222	0.269	0.227	0.273	0.234	0.270	0.219	0.267	0.265	0.305	0.292	0.313	0.288	0.309	0.256	0.289	0.265	0.307	0.260	0.275
Solar	96	0.185	0.263	0.167	0.259	**0.153**	0.255	0.204	**0.230**	0.209	0.309	0.524	0.557	0.591	0.504	0.373	0.304	0.304	0.345	0.863	0.664	0.682	0.688
	192	0.201	0.276	0.180	0.273	**0.165**	0.269	0.221	**0.248**	0.220	0.315	0.507	0.550	0.689	0.567	0.363	0.303	0.309	0.342	0.823	0.695	0.694	0.695
	336	0.212	0.283	0.190	0.282	**0.174**	0.278	0.225	**0.260**	0.228	0.314	0.508	0.553	0.831	0.636	0.391	0.319	0.433	0.450	0.835	0.741	0.719	0.706
	720	0.213	0.291	0.198	0.295	**0.179**	0.287	0.233	**0.272**	0.218	0.283	0.479	0.534	0.972	0.710	0.444	0.349	0.599	0.576	0.738	0.738	0.759	0.725
	Avg	0.203	0.278	0.184	0.277	**0.168**	0.272	0.221	**0.252**	0.219	0.305	0.505	0.549	0.771	0.604	0.393	0.319	0.411	0.428	0.815	0.710	0.714	0.704
1st Count		**8**	**8**	2	0	4	0	5	5	0	2	2	1	0	0	0	0	1	2	2	2	0	4
2nd Count		1	1	5	2	0	3	2	**6**	**9**	6	2	3	0	0	0	0	2	1	4	2	0	0

Table 19: Full results of zero-shot probabilistic forecasting experiments on datasets from ProbTS. Lower CRPS or NMAE values indicate better predictions. Compared with Table 17, we exclude the datasets used for pre-training several baseline models. **Red**: the best, Blue: the 2nd best.

Models		**FLAME Small**		**FLAME Base**		**FLAME Large**		**Sundial** (2025)		**Chronos** (2024)		**MOIRAI** (2024)		**Lag-Llama** (2023)		**TSDiff** (2023)		**CSDI** (2022)		**TimeGrad** (2022)		**GRU NVP** (2021)	
Metrics		CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE
ETTm1	96	0.294	0.368	0.326	0.417	0.315	0.407	0.253	0.308	0.360	0.422	0.464	0.522	0.354	0.402	0.344	0.441	**0.236**	**0.308**	0.522	0.645	0.383	0.488
	192	0.332	0.420	0.363	0.469	0.337	0.426	**0.279**	**0.337**	0.404	0.450	0.467	0.531	0.368	0.415	0.345	0.441	0.291	0.377	0.603	0.748	0.396	0.514
	336	0.341	0.427	0.393	0.511	0.351	0.447	**0.291**	**0.350**	0.425	0.456	0.524	0.558	0.387	0.436	0.462	0.571	0.322	0.419	0.601	0.759	0.486	0.630
	720	0.364	0.454	0.477	0.622	0.383	0.476	**0.318**	**0.380**	0.461	0.478	0.514	0.535	0.403	0.466	0.478	0.622	0.448	0.578	0.621	0.793	0.546	0.707
	avg	0.333	0.417	0.390	0.505	0.347	0.439	**0.285**	**0.344**	0.413	0.451	0.492	0.536	0.378	0.430	0.407	0.519	0.324	0.421	0.587	0.736	0.453	0.585
ETTm2	96	0.120	0.153	0.121	0.153	0.124	0.155	0.128	0.153	0.134	0.158	0.176	0.186	0.163	0.192	0.175	0.224	**0.115**	**0.146**	0.427	0.525	0.319	0.413
	192	0.146	0.182	0.147	0.184	**0.143**	**0.175**	0.150	0.177	0.163	0.183	0.197	0.207	0.181	0.207	0.255	0.316	0.147	0.189	0.424	0.530	0.326	0.427
	336	0.163	0.202	0.164	0.205	**0.158**	**0.193**	0.167	0.195	0.190	0.204	0.229	0.227	0.206	0.229	0.328	0.397	0.190	0.248	0.469	0.566	0.449	0.580
	720	0.185	0.223	0.186	0.231	**0.184**	0.219	0.189	**0.217**	0.223	0.230	0.321	0.258	0.227	0.249	0.344	0.416	0.239	0.306	0.470	0.561	0.561	0.749
	avg	0.154	0.190	0.155	0.193	**0.152**	**0.186**	0.158	0.186	0.178	0.194	0.231	0.220	0.194	0.219	0.276	0.338	0.173	0.222	0.448	0.546	0.414	0.542
ETTh1	96	0.265	0.322	0.261	0.319	**0.259**	0.310	0.260	**0.307**	0.293	0.341	0.469	0.488	0.297	0.340	0.395	0.510	0.437	0.557	0.455	0.585	0.379	0.481
	192	0.288	0.346	0.282	0.351	**0.278**	**0.342**	0.297	0.347	0.348	0.384	0.496	0.492	0.312	0.356	0.467	0.596	0.496	0.625	0.516	0.680	0.425	0.531
	336	0.300	0.371	0.297	0.368	**0.290**	**0.362**	0.314	0.365	0.384	0.409	0.503	0.489	0.326	0.370	0.450	0.581	0.454	0.574	0.512	0.666	0.458	0.580
	720	0.320	0.402	0.312	0.386	**0.303**	0.367	0.312	**0.366**	0.413	0.425	0.526	0.508	0.340	0.405	0.516	0.657	0.528	0.657	0.523	0.672	0.502	0.643
	avg	0.293	0.360	0.288	0.356	**0.283**	**0.345**	0.296	0.346	0.360	0.390	0.498	0.494	0.319	0.368	0.457	0.586	0.479	0.603	0.502	0.651	0.441	0.559
ETTh2	96	**0.139**	**0.177**	0.145	0.185	0.144	0.186	0.157	0.186	0.163	0.191	0.212	0.238	0.178	0.204	0.336	0.421	0.164	0.214	0.358	0.448	0.432	0.548
	192	**0.152**	**0.195**	0.161	0.207	0.159	0.204	0.183	0.213	0.195	0.218	0.229	0.250	0.193	0.217	0.265	0.339	0.226	0.294	0.457	0.575	0.625	0.766
	336	**0.170**	**0.216**	0.174	0.224	0.172	0.217	0.203	0.232	0.226	0.241	0.253	0.263	0.211	0.230	0.350	0.427	0.274	0.353	0.481	0.606	0.793	0.942
	720	0.183	0.231	0.191	0.241	**0.182**	**0.228**	0.203	0.233	0.255	0.263	0.279	0.273	0.222	0.238	0.406	0.482	0.302	0.382	0.445	0.550	0.539	0.688
	avg	**0.161**	**0.205**	0.168	0.214	0.164	0.209	0.186	0.216	0.210	0.228	0.243	0.256	0.201	0.222	0.339	0.417	0.242	0.311	0.435	0.545	0.597	0.736
Weather	96	0.071	0.088	0.071	0.085	**0.067**	**0.083**	0.078	0.091	0.136	0.155	0.161	0.141	0.085	0.094	0.104	0.113	0.068	0.087	0.130	0.164	0.116	0.145
	192	0.089	0.113	0.081	0.100	**0.068**	**0.082**	0.083	0.097	0.145	0.165	0.173	0.139	0.094	0.102	0.134	0.144	0.068	0.086	0.127	0.158	0.122	0.147
	336	0.094	0.121	0.090	0.110	**0.079**	**0.094**	0.090	0.106	0.136	0.151	0.149	0.134	0.098	0.109	0.137	0.138	0.083	0.098	0.130	0.162	0.128	0.160
	720	0.103	0.132	0.106	0.125	**0.083**	**0.100**	0.096	0.113	0.151	0.161	0.233	0.155	0.107	0.120	0.152	0.141	0.087	0.102	0.113	0.136	0.110	0.135
	avg	0.089	0.114	0.087	0.105	**0.074**	**0.090**	0.087	0.102	0.142	0.158	0.179	0.143	0.096	0.106	0.132	0.134	0.077	0.093	0.125	0.155	0.119	0.147
Exchange	96	0.023	0.028	0.021	0.027	**0.020**	0.026	0.024	0.027	0.021	**0.025**	0.025	0.030	0.042	0.051	0.079	0.090	0.028	0.036	0.068	0.079	0.071	0.091
	192	0.032	0.037	**0.030**	0.036	0.031	**0.035**	0.034	0.038	0.032	0.036	0.034	0.039	0.047	0.058	0.093	0.106	0.045	0.058	0.087	0.100	0.068	0.087
	336	0.044	0.049	**0.041**	0.047	0.044	**0.046**	0.046	0.050	0.045	0.048	0.047	0.052	0.061	0.073	0.081	0.106	0.060	0.076	0.074	0.086	0.072	0.091
	720	0.074	0.079	**0.069**	**0.077**	0.073	0.079	0.076	0.079	0.078	0.080	0.073	0.081	0.078	0.094	0.082	0.142	0.143	0.173	0.099	0.113	0.079	0.103
	avg	0.043	0.048	**0.040**	0.047	0.042	**0.047**	0.045	0.049	0.044	0.047	0.045	0.050	0.057	0.069	0.084	0.111	0.069	0.086	0.082	0.095	0.073	0.093
IL1	24	0.135	0.166	0.111	0.143	0.136	0.168	**0.108**	**0.123**	0.120	0.139	0.150	0.196	0.135	0.173	0.228	0.242	0.250	0.263	0.275	0.296	0.257	0.283
	36	0.164	0.200	**0.119**	**0.153**	0.145	0.179	0.152	0.169	0.179	0.205	0.171	0.222	0.163	0.227	0.235	0.246	0.285	0.298	0.272	0.298	0.281	0.307
	48	0.176	0.216	**0.139**	**0.178**	0.164	0.200	0.164	0.184	0.186	0.215	0.151	0.184	0.171	0.233	0.265	0.275	0.285	0.301	0.295	0.320	0.288	0.314
	60	0.171	0.211	**0.149**	0.189	0.169	0.204	0.168	0.190	0.196	0.228	0.163	**0.188**	0.156	0.211	0.263	0.272	0.283	0.299	0.295	0.325	0.307	0.333
	avg	0.162	0.198	**0.130**	**0.166**	0.154	0.188	0.148	0.166	0.170	0.197	0.159	0.197	0.156	0.211	0.248	0.259	0.276	0.290	0.284	0.310	0.283	0.309
1st Count		3	3	6	3	**13**	**11**	4	7	0	1	0	1	0	0	0	0	2	2	0	0	0	0
2nd Count		**7**	5	5	4	6	**6**	2	5	0	2	1	1	1	0	0	0	6	5	0	0	0	0

Table 20: Probabilistic forecasting results. Full-shot, 10% few-shot and zero-shot results of FLAME Small vs. the most advanced end-to-end supervised models on commonly-used datasets in ProbTS. Lower CRPS or NMAE values indicate better predictions. **Red**: the best, Blue: the second best.

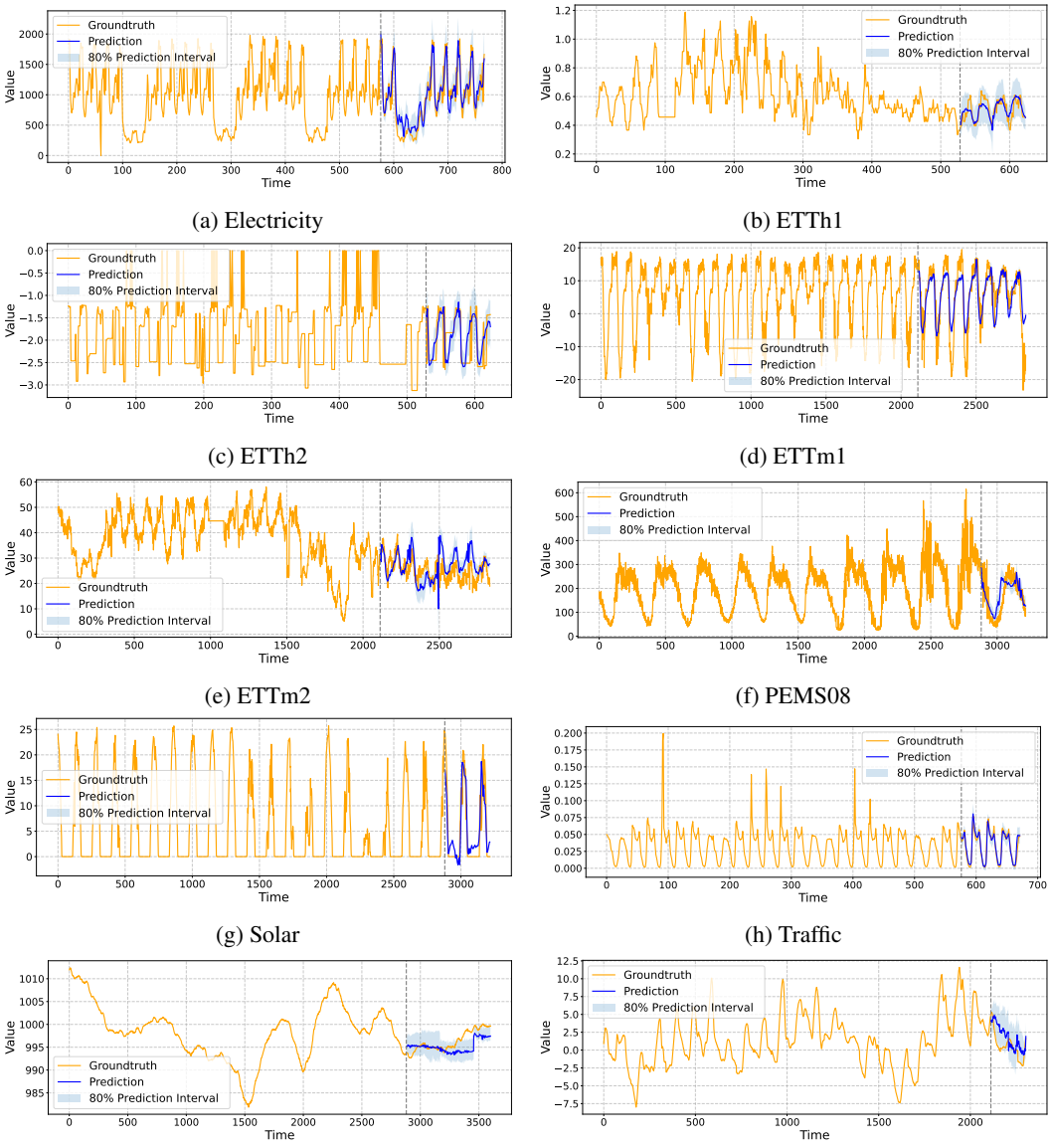
Models		FLAME Small (Zero-shot)		FLAME Small (10% few-shot)		FLAME Small (Full-shot)		NSDIFF (2025)		K^2VAE (2025)		TSDiff (2023)		CSDI (2022)		TimeGrad (2022)		GRU NVP (2021)	
Metric		CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE	CRPS	NMAE
ETTm1	96	0.294	0.368	0.233	0.287	**0.211**	**0.275**	0.295	0.364	0.235	0.294	0.344	0.441	0.236	0.308	0.522	0.645	0.383	0.488
	192	0.332	0.420	0.259	0.317	**0.232**	**0.298**	0.285	0.350	0.261	0.330	0.345	0.441	0.291	0.377	0.603	0.748	0.396	0.514
	336	0.341	0.427	0.292	**0.351**	**0.279**	0.355	0.289	0.357	0.304	0.386	0.462	0.571	0.322	0.419	0.601	0.759	0.486	0.630
	720	0.364	0.454	0.304	0.392	**0.282**	**0.358**	0.411	0.500	0.302	0.387	0.478	0.622	0.448	0.578	0.621	0.793	0.546	0.707
	Avg	0.333	0.417	0.272	0.337	**0.251**	**0.322**	0.320	0.393	0.276	0.349	0.407	0.519	0.324	0.421	0.587	0.736	0.453	0.585
ETTm2	96	0.120	0.153	0.117	0.144	**0.114**	**0.141**	0.145	0.176	0.128	0.148	0.175	0.224	0.115	0.146	0.427	0.525	0.319	0.413
	192	0.146	0.182	0.145	0.178	**0.138**	0.166	0.162	0.196	0.161	**0.163**	0.255	0.316	0.147	0.189	0.424	0.530	0.326	0.427
	336	0.163	0.202	0.160	0.199	**0.151**	**0.182**	0.172	0.209	0.180	0.183	0.328	0.397	0.190	0.248	0.469	0.566	0.449	0.580
	720	0.185	0.223	0.181	0.231	**0.170**	**0.211**	0.364	0.428	0.224	0.224	0.344	0.416	0.239	0.306	0.470	0.561	0.561	0.749
	Avg	0.154	0.190	0.151	0.188	**0.143**	**0.175**	0.211	0.252	0.173	0.180	0.276	0.338	0.173	0.222	0.448	0.546	0.414	0.542
ETTh1	96	0.265	0.322	0.254	0.311	**0.247**	**0.301**	0.301	0.380	0.280	0.358	0.395	0.510	0.437	0.557	0.455	0.585	0.379	0.481
	192	0.288	0.346	0.272	0.336	**0.268**	**0.322**	0.309	0.386	0.304	0.392	0.467	0.596	0.496	0.625	0.516	0.680	0.425	0.531
	336	0.300	0.371	0.297	0.369	**0.288**	**0.363**	0.357	0.448	0.326	0.413	0.450	0.581	0.454	0.574	0.512	0.666	0.458	0.580
	720	0.320	0.402	0.308	0.395	**0.300**	**0.392**	0.384	0.513	0.331	0.428	0.516	0.657	0.528	0.657	0.523	0.672	0.502	0.643
	Avg	0.293	0.360	0.283	0.353	**0.276**	**0.345**	0.337	0.432	0.310	0.398	0.457	0.586	0.479	0.603	0.502	0.651	0.441	0.559
ETTh2	96	0.139	0.177	0.133	0.175	**0.122**	**0.162**	0.175	0.220	0.182	0.206	0.336	0.421	0.164	0.214	0.358	0.448	0.432	0.548
	192	0.152	0.195	0.148	0.193	**0.137**	**0.188**	0.200	0.245	0.185	0.220	0.265	0.339	0.226	0.294	0.457	0.575	0.625	0.766
	336	0.170	0.216	0.167	0.209	**0.155**	**0.194**	0.207	0.257	0.245	0.277	0.350	0.427	0.274	0.353	0.481	0.606	0.793	0.942
	720	0.183	0.231	0.177	0.228	**0.171**	**0.222**	0.209	0.260	0.237	0.272	0.406	0.482	0.302	0.382	0.445	0.550	0.539	0.688
	Avg	0.161	0.205	0.156	0.201	**0.146**	**0.192**	0.198	0.246	0.212	0.244	0.339	0.417	0.242	0.311	0.435	0.545	0.597	0.736
Weather	96	0.071	0.088	0.069	0.085	**0.065**	**0.083**	0.123	0.129	0.083	0.087	0.104	0.113	0.068	0.087	0.130	0.164	0.116	0.145
	192	0.089	0.113	0.078	0.101	0.073	0.088	0.132	0.132	0.085	0.090	0.134	0.144	**0.068**	**0.086**	0.127	0.158	0.122	0.147
	336	0.094	0.121	0.082	0.114	**0.075**	**0.089**	0.130	0.128	0.087	0.091	0.137	0.138	0.083	0.098	0.130	0.162	0.128	0.160
	720	0.103	0.132	0.098	0.119	0.091	0.096	0.142	0.147	0.090	**0.094**	0.152	0.141	**0.087**	0.102	0.113	0.136	0.110	0.135
	Avg	0.089	0.114	0.082	0.105	**0.076**	**0.089**	0.132	0.134	0.086	0.091	0.132	0.134	0.077	0.093	0.125	0.155	0.119	0.147
Electricity	96	0.071	0.090	0.069	0.088	**0.069**	**0.088**	0.160	0.202	0.089	0.093	0.344	0.441	0.153	0.203	0.096	0.119	0.094	0.118
	192	0.086	0.103	**0.083**	0.097	0.084	**0.095**	0.248	0.358	0.097	0.102	0.345	0.441	0.200	0.264	0.100	0.124	0.097	0.121
	336	0.096	0.125	0.097	0.116	**0.094**	0.114	0.252	0.362	0.101	**0.105**	0.462	0.571	/	/	0.102	0.126	0.099	0.123
	720	0.117	0.154	0.108	0.123	**0.104**	0.117	0.259	0.368	0.105	**0.110**	0.478	0.622	/	/	0.108	0.134	0.114	0.144
	Avg	0.093	0.118	0.089	0.106	**0.088**	0.104	0.230	0.322	0.098	**0.103**	0.407	0.519	/	/	0.102	0.126	0.101	0.127
Traffic	96	0.240	0.293	0.227	0.262	0.203	**0.217**	0.393	0.500	0.216	0.222	0.294	0.342	/	/	0.202	0.234	**0.187**	0.231
	192	0.257	0.319	0.243	0.289	0.212	**0.223**	0.448	0.585	0.220	0.227	0.306	0.354	/	/	0.208	0.239	**0.192**	0.236
	336	0.271	0.341	0.266	0.317	0.220	**0.228**	0.455	0.592	0.228	0.235	0.317	0.392	/	/	0.213	0.246	**0.201**	0.248
	720	0.304	0.395	0.288	0.322	0.231	**0.237**	0.452	0.588	0.235	0.242	0.391	0.478	/	/	0.220	0.263	**0.211**	0.264
	Avg	0.268	0.337	0.256	0.298	0.217	**0.226**	0.437	0.566	0.225	0.232	0.327	0.392	/	/	0.211	0.246	**0.198**	0.245
Exchange	96	0.023	0.028	0.024	0.031	**0.022**	**0.027**	0.023	0.029	0.027	0.028	0.079	0.090	0.028	0.036	0.068	0.079	0.071	0.091
	192	0.032	0.037	0.028	0.036	**0.025**	**0.031**	0.029	0.037	0.035	0.036	0.093	0.106	0.045	0.058	0.087	0.100	0.068	0.087
	336	0.044	0.049	0.035	0.037	0.031	0.033	0.043	0.053	**0.027**	**0.028**	0.081	0.106	0.060	0.076	0.074	0.086	0.072	0.091
	720	0.074	0.079	0.067	0.074	**0.053**	**0.070**	0.057	0.071	0.081	0.083	0.082	0.142	0.143	0.173	0.099	0.113	0.079	0.103
	Avg	0.043	0.048	0.039	0.045	**0.033**	**0.040**	0.038	0.047	0.043	0.044	0.084	0.111	0.069	0.086	0.082	0.095	0.073	0.093
ILI	24	0.135	0.166	0.112	0.133	**0.088**	**0.103**	0.092	0.108	0.110	0.113	0.228	0.242	0.250	0.263	0.275	0.296	0.257	0.283
	36	0.164	0.200	0.148	0.185	**0.137**	0.168	0.151	0.178	0.140	**0.144**	0.235	0.246	0.285	0.298	0.272	0.298	0.281	0.307
	48	0.176	0.216	0.169	0.205	**0.142**	0.171	0.151	0.178	0.153	**0.157**	0.265	0.275	0.285	0.301	0.295	0.320	0.288	0.314
	60	0.171	0.211	0.165	0.198	**0.152**	0.184	0.168	0.198	0.164	**0.167**	0.263	0.272	0.283	0.299	0.295	0.325	0.307	0.333
	Avg	0.162	0.198	0.149	0.180	**0.130**	0.157	0.140	0.165	0.142	**0.145**	0.248	0.259	0.276	0.290	0.284	0.310	0.283	0.309
1st Count		0	0	1	1	**28**	**26**	0	0	1	8	0	0	2	1	0	0	4	0

Table 21: Deterministic forecasting results. Full-shot, 10% few-shot and zero-shot results of FLAME Small vs. the most advanced end-to-end supervised models on commonly-used datasets in TSFM-Bench. Lower MSE or MAE values indicate better predictions. **Red**: the best, Blue: the second best.

Models		FLAME Small (Zero-shot)		FLAME Small (10% few-shot))		FLAME Small (Full-shot)		TimeKAN (2025)		AMD (2025)		TimePro (2025)		TimeXer (2024)		Fredformer (2024)		iTransformer (2024)		PatchTST (2023)		TimesNet (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.305	0.353	0.278	0.328	**0.264**	**0.318**	0.327	0.365	0.327	0.361	0.326	0.364	0.309	0.352	0.326	0.361	0.334	0.368	0.329	0.367	0.338	0.375
	192	0.328	0.370	0.320	0.359	**0.311**	**0.352**	0.363	0.387	0.366	0.383	0.367	0.383	0.355	0.378	0.363	0.384	0.377	0.391	0.367	0.385	0.374	0.387
	336	0.349	0.384	0.338	0.371	**0.332**	**0.368**	0.389	0.407	0.398	0.404	0.402	0.409	0.387	0.399	0.395	0.406	0.426	0.420	0.399	0.410	0.410	0.411
	720	0.392	0.416	0.383	0.404	**0.372**	**0.399**	0.457	0.445	0.464	0.437	0.469	0.446	0.448	0.435	0.456	0.441	0.491	0.459	0.454	0.439	0.478	0.450
	Avg	0.344	0.381	0.330	0.366	**0.320**	**0.359**	0.384	0.401	0.389	0.396	0.391	0.401	0.375	0.391	0.385	0.398	0.407	0.410	0.387	0.400	0.400	0.406
ETTm2	96	0.165	0.255	0.155	0.250	**0.150**	**0.243**	0.178	0.262	0.176	0.259	0.178	0.260	0.171	0.255	0.177	0.258	0.180	0.264	0.175	0.259	0.187	0.267
	192	0.217	0.297	0.215	0.294	**0.214**	**0.292**	0.244	0.308	0.242	0.302	0.242	0.303	0.238	0.300	0.243	0.301	0.250	0.309	0.241	0.302	0.249	0.309
	336	0.262	0.330	0.258	0.318	**0.253**	**0.312**	0.305	0.346	0.298	0.337	0.303	0.342	0.301	0.340	0.302	0.340	0.311	0.348	0.305	0.343	0.321	0.351
	720	0.333	0.379	0.329	0.368	**0.328**	**0.364**	0.402	0.404	0.396	0.394	0.400	0.399	0.401	0.397	0.404	0.398	0.412	0.407	0.402	0.400	0.408	0.403
	Avg	0.244	0.315	0.239	0.308	**0.236**	**0.303**	0.282	0.330	0.278	0.323	0.281	0.326	0.278	0.323	0.281	0.324	0.288	0.332	0.281	0.326	0.291	0.333
ETTh1	96	0.362	0.395	0.357	0.386	**0.342**	**0.375**	0.374	0.391	0.375	0.392	0.375	0.398	0.377	0.397	0.378	0.395	0.386	0.405	0.414	0.419	0.384	0.402
	192	0.406	0.427	0.392	0.413	**0.384**	**0.404**	0.421	0.421	0.430	0.422	0.427	0.429	0.425	0.425	0.435	0.424	0.441	0.436	0.460	0.445	0.436	0.429
	336	0.436	0.449	0.420	0.435	**0.407**	**0.428**	0.464	0.440	0.471	0.443	0.472	0.450	0.457	0.441	0.485	0.447	0.487	0.458	0.501	0.466	0.491	0.469
	720	0.471	0.482	0.436	0.450	**0.418**	**0.436**	0.466	0.462	0.478	0.464	0.476	0.474	0.464	0.463	0.496	0.472	0.503	0.491	0.500	0.488	0.521	0.500
	Avg	0.419	0.438	0.401	0.421	**0.388**	**0.411**	0.431	0.429	0.438	0.430	0.438	0.438	0.431	0.432	0.448	0.435	0.454	0.448	0.469	0.455	0.458	0.450
ETTh2	96	0.278	0.339	0.271	0.327	**0.264**	**0.320**	0.293	0.343	0.287	0.338	0.293	0.345	0.289	0.340	0.291	0.342	0.297	0.349	0.302	0.348	0.340	0.374
	192	0.336	0.381	0.335	0.376	**0.332**	**0.364**	0.375	0.396	0.367	0.388	0.367	0.394	0.370	0.391	0.372	0.390	0.380	0.400	0.388	0.400	0.402	0.414
	336	0.373	0.410	0.364	0.402	**0.360**	**0.399**	0.429	0.441	0.410	0.424	0.419	0.431	0.422	0.434	0.419	0.431	0.428	0.432	0.426	0.433	0.452	0.452
	720	0.411	0.441	0.391	0.428	**0.382**	**0.423**	0.466	0.468	0.421	0.440	0.427	0.445	0.429	0.445	0.431	0.450	0.427	0.445	0.431	0.446	0.462	0.468
	Avg	0.350	0.393	0.340	0.383	**0.335**	**0.377**	0.391	0.412	0.371	0.397	0.377	0.404	0.378	0.403	0.378	0.403	0.383	0.407	0.387	0.407	0.414	0.427
Weather	96	0.148	0.204	0.144	0.197	**0.141**	**0.184**	0.164	0.210	0.174	0.221	0.166	0.207	0.168	0.209	0.163	0.207	0.174	0.214	0.177	0.218	0.172	0.220
	192	0.189	0.243	0.184	0.241	**0.182**	**0.240**	0.209	0.250	0.219	0.259	0.216	0.254	0.220	0.254	0.224	0.258	0.221	0.254	0.255	0.259	0.219	0.261
	336	0.233	0.278	0.230	0.276	**0.224**	**0.272**	0.264	0.290	0.273	0.296	0.273	0.296	0.276	0.296	0.278	0.298	0.278	0.296	0.278	0.297	0.280	0.306
	720	0.289	0.320	0.287	0.318	**0.280**	**0.306**	0.343	0.342	0.349	0.345	0.351	0.346	0.353	0.347	0.357	0.350	0.358	0.349	0.354	0.348	0.365	0.359
	Avg	0.215	0.261	0.211	0.258	**0.207**	**0.251**	0.245	0.273	0.254	0.280	0.252	0.276	0.254	0.277	0.256	0.278	0.258	0.278	0.266	0.281	0.259	0.287
Electricity	96	0.206	0.294	0.142	0.235	**0.124**	**0.218**	0.174	0.266	0.147	0.251	0.139	0.234	0.151	0.247	0.148	0.242	0.148	0.240	0.195	0.285	0.168	0.272
	192	0.247	0.338	0.168	0.255	**0.149**	**0.236**	0.182	0.273	0.176	0.262	0.156	0.249	0.165	0.261	0.165	0.257	0.162	0.253	0.199	0.289	0.184	0.289
	336	0.310	0.398	0.188	0.271	**0.164**	**0.252**	0.197	0.286	0.193	0.281	0.172	0.267	0.183	0.280	0.180	0.274	0.178	0.269	0.215	0.305	0.198	0.300
	720	0.383	0.455	**0.192**	**0.277**	0.205	0.288	0.236	0.320	0.232	0.329	0.209	0.299	0.220	0.309	0.218	0.305	0.225	0.317	0.256	0.337	0.220	0.320
	Avg	0.287	0.371	0.173	0.260	**0.161**	**0.249**	0.197	0.286	0.187	0.281	0.169	0.262	0.180	0.274	0.178	0.270	0.178	0.270	0.216	0.304	0.193	0.295
Traffic	96	0.591	0.401	0.375	0.262	**0.358**	**0.246**	0.423	0.286	0.443	0.298	0.426	0.292	0.416	0.280	0.403	0.274	0.395	0.268	0.544	0.359	0.593	0.321
	192	0.623	0.431	0.388	0.265	**0.379**	**0.248**	0.442	0.295	0.496	0.323	0.439	0.298	0.435	0.288	0.429	0.289	0.417	0.276	0.540	0.354	0.617	0.336
	336	0.697	0.485	0.397	0.285	**0.386**	**0.262**	0.473	0.335	0.520	0.330	0.449	0.307	0.451	0.296	0.441	0.295	0.433	0.283	0.551	0.358	0.629	0.336
	720	0.782	0.528	0.451	0.332	**0.427**	**0.282**	0.481	0.357	0.540	0.344	0.475	0.309	0.484	0.314	0.463	0.300	0.467	0.302	0.586	0.375	0.640	0.350
	Avg	0.673	0.461	0.403	0.286	**0.388**	**0.260**	0.455	0.318	0.500	0.324	0.447	0.302	0.447	0.295	0.434	0.289	0.428	0.282	0.555	0.362	0.620	0.336
1st Count		0	0	1	1	**34**	**34**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D Showcases

We visualize the forecasting results of FLAME Large on TSFM-Bench and TFB in Figures 7 and 8. The figures show that FLAME can accurately model the future patterns by providing prediction intervals.



(a) Electricity (b) ETTh1

(c) ETTh2 (d) ETTm1

(e) ETTm2 (f) PEMS08

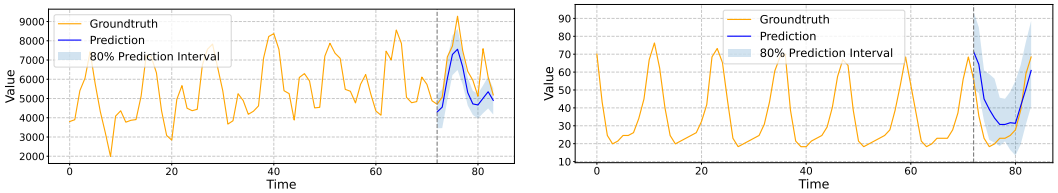
(g) Solar (h) Traffic

(i) Weather (j) Wind

Figure 7: Visualization of TSFM-Bench

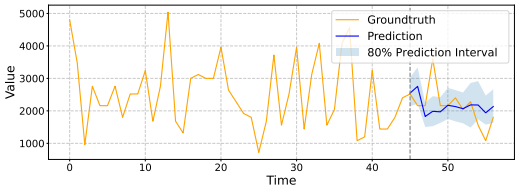
E Limitations

Despite the strong performance of FLAME, this work still has some limitations. FLAME's local perception mechanism is particularly designed for time series with strong periodicity. For non-stationary scenarios, the mechanism may degrade, but it does not introduce adverse effects. Experiments on datasets like ILI, Exchange, NYSE, and multiple very short time series in TFB, demonstrate that FLAME also achieves state-of-the-art performance. On the other hand, FLAME is proposed for regular time series, and does not adapt to irregular time series.

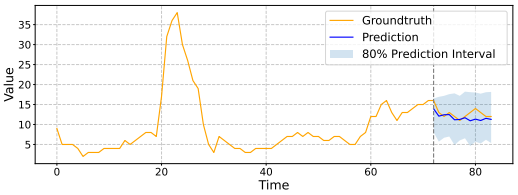


(a) economics_36

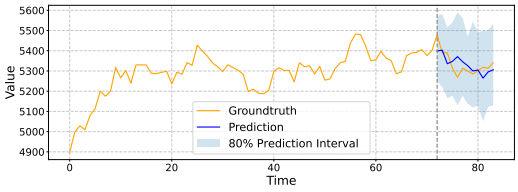
(b) human_58



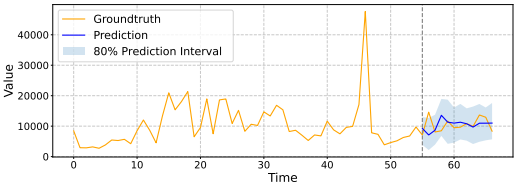
(c) m3_monthly_dataset_223



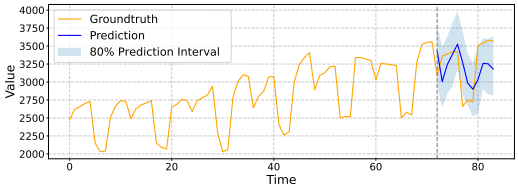
(d) KDD Cup 2018, series 60



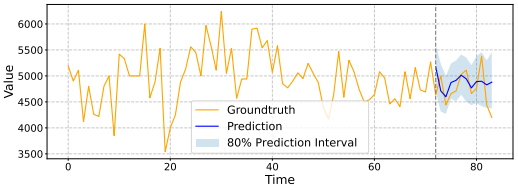
(e) m3_other_dataset_7



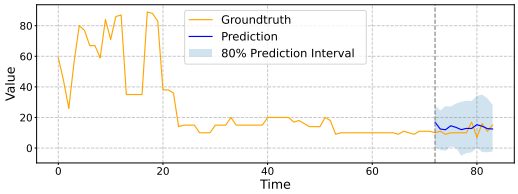
(f) m4_quarterly_dataset_18009



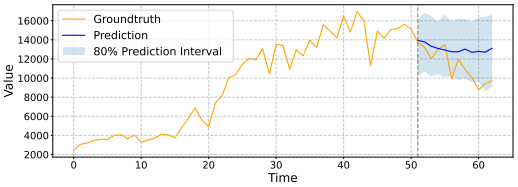
(g) m4_monthly_dataset_2019



(h) m4_monthly_dataset_84



(i) Vehicle trips, series 82



(j) cif_2016_dataset_65

Figure 8: Visualization of Univariate Datasets in TFB

F Impact Statement

This paper presents a family of time series foundation models for both deterministic and probabilistic forecasting. The release of code and experimental protocols also promotes reproducible research and further exploration of time series models.

G The Use of Large Language Models

LLMs were used for language editing and assistance with drafting and checking the mathematical exposition and proofs. Their use did not involve designing the model or generating experimental results.

H Theoretical Proof

H.1 Legendre Approximation Bound

Theorem 2.1 (restated). Let $\theta > 0$, $t \geq \theta$, and $d \geq 1$. For a differentiable function $F : \mathbb{R}_+ \to \mathbb{R}$, let $f_{[t-\theta,t]}$ be its optimal polynomial approximation of degree at most $d-1$ under the uniform probability measure $\mu^{(t)}$ on $[t-\theta, t]$. If F is L-Lipschitz on this window, then

$$\|F - f_{[t-\theta,t]}\|_{\mu^{(t)}} = \mathcal{O}(\theta L/\sqrt{d}).$$

For a fixed integer $k \geq 1$, if $F \in C^k([t-\theta, t])$, $M_k = \sup_{x\in[t-\theta,t]} |F^{(k)}(x)| < \infty$, and $d \geq k$, then

$$\|F - f_{[t-\theta,t]}\|_{\mu^{(t)}} = \mathcal{O}(M_k \theta^k d^{-k+1/2}),$$

where the implied constant depends only on k. These bounds concern the exact projection; they do not include the finite-state recurrence or discretization errors.

Proof. Fix a time t. The HiPPO-LegT framework utilizes the uniform measure $\mu^{(t)}$ supported on the sliding window $[t-\theta, t]$, with the density function $\omega(t, x) = \frac{1}{\theta}\mathbb{I}_{[t-\theta,t]}(x)$. The corresponding orthonormal polynomial basis is given by:

$$p_n^{(t)}(x) = \sqrt{2n+1} P_n\left(\frac{2(x-t)}{\theta} + 1\right)$$

where P_n denotes the canonical Legendre polynomial of degree n.

Let $c_n(t) = \langle F_{[t-\theta,t]}, p_n^{(t)}\rangle_{\mu^{(t)}}$ be the projection coefficients for $n = 0, 1, \ldots$. The projection $f_{[t-\theta,t]}$ is constructed by a linear combination of the basis functions up to degree $d-1$:

$$f_{[t-\theta,t]}(x) = \sum_{n=0}^{d-1} c_n(t) p_n^{(t)}(x)$$

Since $p_n^{(t)}(x)$ forms an orthonormal basis in the Hilbert space defined by the inner product $\langle\cdot,\cdot\rangle_{\mu^{(t)}}$, we apply Parseval's identity to express the approximation error as the sum of squared high-order coefficients:

$$\|F_{[t-\theta,t]} - f_{[t-\theta,t]}\|^2_{\mu^{(t)}} = \sum_{n=d}^{\infty} c_n^2(t)$$

To bound the error, it suffices to bound $c_n(t)$ for $n \geq d$. By definition:

$$c_n(t) = \frac{\sqrt{2n+1}}{\theta} \int_{t-\theta}^{t} F(x) P_n\left(\frac{2(x-t)}{\theta} + 1\right) dx$$

Applying the change of variable $y = \frac{2(x-t)}{\theta} + 1$, we have $dx = \frac{\theta}{2} dy$. The integral domain transforms to $[-1, 1]$:

$$c_n(t) = \frac{\sqrt{2n+1}}{2} \int_{-1}^{1} F\left(t + \frac{\theta}{2}(y-1)\right) P_n(y) dy$$

Recall the property of Legendre polynomials: $P_n(y) = \frac{1}{2n+1}\frac{d}{dy}(P_{n+1}(y) - P_{n-1}(y))$. Applying integration by parts yields:

$$c_n(t) = \frac{\sqrt{2n+1}}{2(2n+1)}\left[F\left(t+\frac{\theta}{2}(y-1)\right)(P_{n+1}(y)-P_{n-1}(y))\right]_{-1}^{1} - \frac{\sqrt{2n+1}}{2(2n+1)}\int_{-1}^{1}\frac{\theta}{2}F'\left(t+\frac{\theta}{2}(y-1)\right)(P_{n+1}(y)-P_{n-1}(y))dy$$

Since $P_m(1) = 1$ and $P_m(-1) = (-1)^m$, the boundary terms evaluate to zero. The expression simplifies to:

$$c_n(t) = -\frac{\theta}{4\sqrt{2n+1}}\int_{-1}^{1}F'\left(t+\frac{\theta}{2}(y-1)\right)(P_{n+1}(y)-P_{n-1}(y))dy$$

Case 1: $F(x)$ is L-Lipschitz
If F is L-Lipschitz, its first derivative is bounded by L (i.e., $|F'| \le L$). By applying the Cauchy-Schwarz inequality, we bound the squared coefficient:

$$\begin{aligned} c_n^2(t) &\le \frac{\theta^2 L^2}{16(2n+1)}\left[\int_{-1}^{1}|P_{n+1}(y)-P_{n-1}(y)|dy\right]^2 \\ &\le \frac{\theta^2 L^2}{16(2n+1)}\cdot 2\int_{-1}^{1}(P_{n+1}(y)-P_{n-1}(y))^2 dy \end{aligned}$$

Due to the orthogonality of Legendre polynomials:

$$\int_{-1}^{1}(P_{n+1}(y)-P_{n-1}(y))^2dy = \int_{-1}^{1}P_{n+1}^2(y)dy + \int_{-1}^{1}P_{n-1}^2(y)dy = \frac{2}{2n+3}+\frac{2}{2n-1}$$

Thus, $c_n^2(t) \le \mathcal{O}(1)\theta^2L^2\frac{1}{n^2}$. Summing over all $n \ge d$ yields:

$$\|F_{[t-\theta,t]} - f_{[t-\theta,t]}\|_{\mu^{(t)}}^2 = \sum_{n=d}^{\infty}c_n^2(t) = \mathcal{O}(1)\theta^2L^2\sum_{n=d}^{\infty}\frac{1}{n^2} = \mathcal{O}\left(\frac{\theta^2L^2}{d}\right)$$

Taking the square root gives $\|F_{[t-\theta,t]} - f_{[t-\theta,t]}\|_{\mu^{(t)}} \le \mathcal{O}(\theta L/\sqrt{d})$, which concludes the first part of the theorem.

Case 2: F has a bounded k-th derivative
Let $F \in C^k([t-\theta,t])$ and $\|F^{(k)}\|_\infty \le M_k$. For $n \ge d \ge k$, define the k-fold antiderivative

$$q_{n,k}(y) = \frac{1}{(k-1)!}\int_{-1}^{y}(y-u)^{k-1}P_n(u)\,du.$$

Then $q_{n,k}^{(k)} = P_n$ and $q_{n,k}^{(j)}(-1) = 0$ for $0 \le j < k$. At $y = 1$, these derivatives are integrals of P_n against polynomials of degree at most $k-1 < n$, so orthogonality also gives $q_{n,k}^{(j)}(1) = 0$. Consequently, all boundary terms vanish when integrating by parts k times:

$$c_n(t) = \frac{(-1)^k\sqrt{2n+1}}{2}\left(\frac{\theta}{2}\right)^k\int_{-1}^{1}F^{(k)}\left(t+\frac{\theta}{2}(y-1)\right)q_{n,k}(y)\,dy.$$

Repeatedly applying $\int_{-1}^{y}P_m(u)\,du = (P_{m+1}(y)-P_{m-1}(y))/(2m+1)$ for $m \ge 1$ expresses $q_{n,k}$ as a linear combination of at most $k+1$ Legendre polynomials, with indices between $n-k$ and $n+k$. For fixed k, their coefficients are $\mathcal{O}_k(n^{-k})$. Using $\|P_m\|_{L^2([-1,1])}^2 = 2/(2m+1)$ and absorbing the finitely many small values of n into the constant gives

$$\int_{-1}^{1}q_{n,k}^2(y)\,dy = \mathcal{O}_k(n^{-2k-1}).$$

The Cauchy–Schwarz inequality therefore yields

$$c_n^2(t) \le \frac{(2n+1)M_k^2\theta^{2k}}{2^{2k+1}} \int_{-1}^{1} q_{n,k}^2(y)\,dy = \mathcal{O}_k(M_k^2\theta^{2k}n^{-2k}).$$

Summing the squared coefficients and taking the square root gives

$$\|F - f_{[t-\theta,t]}\|_{\mu^{(t)}}^2 = \sum_{n=d}^{\infty} c_n^2(t) = \mathcal{O}_k\left(M_k^2\theta^{2k}d^{-2k+1}\right),$$
$$\|F - f_{[t-\theta,t]}\|_{\mu^{(t)}} = \mathcal{O}_k\left(M_k\theta^k d^{-k+1/2}\right).$$

This completes the proof. ■

H.2 Properties of the Conditional MAF

We establish the properties of the conditional affine MAF used in FLAME, following the normalizing-flow construction of Papamakarios et al. [46] and the data-to-latent convention in Eq. (17). The proofs below concern the transformation within one target patch; they do not require the conditioning network itself to be invertible.

Notation and assumptions. Fix a patch index i and a conditioning token $q = q_i$. Let $p, K \ge 1$ be integers. To simplify notation, write $y = \mathrm{Y}_i$, $z^{(0)} = y$, and $z^{(j)} = T_j(z^{(j-1)}; q)$. For $x \in \mathbb{R}^p$, the j-th layer is

$$[T_j(x;q)]_k = e^{a_{j,k}(x_{<k},q)}x_k + b_{j,k}(x_{<k},q), \qquad k = 1,\dots,p, \tag{25}$$

where $x_{<k} = (x_1,\dots,x_{k-1})$. The functions $a_{j,k}$ and $b_{j,k}$ are the masked MLP outputs, whose evaluations at $z_{<k}^{(j-1)}$ are $\alpha_{i,k}^{(j)}$ and $\beta_{i,k}^{(j)}$ in the main text. For $k = 1$, the prefix is empty and both functions depend only on q.

We assume that these functions are finite and real-valued for every finite prefix, and that the effective mask enforces dependence only on $x_{<k}$ and q. All layers use the same coordinate order as in Eq. (17). For the classical Jacobian and change-of-variables arguments, we additionally assume that the functions are continuously differentiable in the prefix. The token q is held fixed throughout all derivatives and inversions; it is not recomputed from the target patch being transformed. Bijectivity itself only requires the finite-valued, strictly autoregressive parameterization.

Proposition H.1 (Global invertibility)**.** *For each fixed q, every layer $T_j(\cdot;q)$ in Eq.* (25) *is a bijection from $\mathbb{R}^p$ to $\mathbb{R}^p$. Its inverse is uniquely determined by the recursion*

$$x_k = e^{-a_{j,k}(x_{<k},q)}\big(z_k - b_{j,k}(x_{<k},q)\big), \qquad k = 1,\dots,p, \tag{26}$$

where $z = T_j(x;q)$. Consequently, the full map $T = T_K \circ \cdots \circ T_1$ is bijective, with

$$T^{-1} = T_1^{-1} \circ \cdots \circ T_K^{-1}. \tag{27}$$

If the parameter functions are continuous, both T and T^{-1} are continuous. Under the continuous differentiability assumption, they are continuously differentiable as well.

Proof. Fix a layer j and an arbitrary desired output $z \in \mathbb{R}^p$. For the first coordinate, $a_{j,1}(q)$ and $b_{j,1}(q)$ are known constants. Since $e^{a_{j,1}(q)} > 0$, the equation $z_1 = e^{a_{j,1}(q)}x_1 + b_{j,1}(q)$ has exactly one real solution, namely Eq. (26) for $k = 1$.

Suppose that $x_1,\dots,x_{k-1}$ have already been recovered uniquely. The strict autoregressive dependence makes $a_{j,k}(x_{<k},q)$ and $b_{j,k}(x_{<k},q)$ known at this step. They are finite, so the coefficient $e^{a_{j,k}(x_{<k},q)}$ is nonzero. The equation for z_k therefore has the unique finite solution in Eq. (26). Induction recovers all p coordinates.

This construction provides a preimage for every z, proving surjectivity. Any two preimages must agree in the first coordinate, then the second, and so on by the same recursion, proving injectivity. Substitution into Eq. (25) gives $T_j(T_j^{-1}(z;q);q) = z$; uniqueness also gives $T_j^{-1}(T_j(x;q);q) = x$.

A composition of these bijections is bijective. Cancelling adjacent inverse pairs yields Eq. (27): generation starts from $z^{(K)} = \epsilon$ and applies $T_K^{-1}, T_{K-1}^{-1},\dots,T_1^{-1}$ in that execution order. Finally,

continuity, or continuous differentiability, of the inverse follows inductively from Eq. (26), since exponentiation and the finite algebraic operations preserve these properties. Composition gives the corresponding regularity of T and T^{-1}. □

Proposition H.2 (Triangular Jacobian and exact conditional likelihood)**.** *Under the continuous differentiability assumption, each $T_j(\cdot; q)$ has a lower triangular Jacobian with a strictly positive diagonal. At $z^{(0)} = y$ and $\epsilon = T(y; q)$, the full transformation satisfies*

$$\begin{aligned} \log |\det D_y T(y; q)| &= \sum_{j=1}^{K} \sum_{k=1}^{p} a_{j,k}(z_{<k}^{(j-1)}, q), \\ p_\theta(y \mid q) &= p_\epsilon(T(y; q)) \, |\det D_y T(y; q)| \, . \end{aligned} \tag{28}$$

For any normalized base density p_ϵ, this defines a normalized conditional density. For the standard Gaussian base used in FLAME,

$$\log p_\theta(y \mid q) = -\frac{p}{2} \log(2\pi) - \frac{1}{2} \|\epsilon\|_2^2 + \sum_{j=1}^{K} \sum_{k=1}^{p} a_{j,k}(z_{<k}^{(j-1)}, q). \tag{29}$$

Proof. Suppress the layer index and write $a_k = a_{j,k}(x_{<k}, q)$ and $b_k = b_{j,k}(x_{<k}, q)$. Differentiating Eq. (25) while holding q fixed gives

$$\frac{\partial [T_j(x; q)]_k}{\partial x_r} = \begin{cases} e^{a_k} x_k \, \dfrac{\partial a_k}{\partial x_r} + \dfrac{\partial b_k}{\partial x_r}, & r < k, \\ e^{a_k}, & r = k, \\ 0, & r > k. \end{cases} \tag{30}$$

The diagonal formula uses the strict mask: neither a_k nor b_k depends on x_k. Thus the parameter dependency mask is strictly lower triangular, whereas the transformation Jacobian has a nonzero diagonal. Its determinant is

$$\det D_x T_j(x; q) = \prod_{k=1}^{p} e^{a_{j,k}(x_{<k}, q)} = \exp\left(\sum_{k=1}^{p} a_{j,k}(x_{<k}, q) \right) > 0. \tag{31}$$

By the chain rule, $D_y T$ is the product of the layer Jacobians, evaluated at $z^{(0)}, \ldots, z^{(K-1)}$. Determinants multiply, and taking logarithms proves the first identity in Eq. (28). This also shows that the composition remains nonsingular and orientation-preserving.

Proposition H.1 establishes a global continuously differentiable inverse. The change-of-variables formula therefore gives the second identity in Eq. (28). In particular,

$$\begin{aligned} \int_{\mathbb{R}^p} p_\theta(y \mid q) \, dy &= \int_{\mathbb{R}^p} p_\epsilon(T(y; q)) \, |\det D_y T(y; q)| \, dy \\ &= \int_{\mathbb{R}^p} p_\epsilon(\epsilon) \, d\epsilon = 1. \end{aligned} \tag{32}$$

Substituting the standard Gaussian log density yields Eq. (29). The Jacobian term has a plus sign because T maps data to the base variable. Restoring the patch index i and averaging the negative log densities over $i = 1, \ldots, l$ gives exactly the objective in Eq. (20). □

Corollary H.3 (Autoregressive order and exact sampling)**.** *For fixed q, the k-th coordinate of $T(y; q)$ depends only on $y_1, \ldots, y_k$, and the k-th coordinate of $T^{-1}(\epsilon; q)$ depends only on $\epsilon_1, \ldots, \epsilon_k$. Each coordinate map is strictly increasing in its own coordinate when the preceding coordinates are fixed. With a standard Gaussian base, the conditional density admits the autoregressive factorization*

$$\begin{aligned} p_\theta(y \mid q) &= \prod_{k=1}^{p} p_\theta(y_k \mid y_{<k}, q), \\ p_\theta(y_k \mid y_{<k}, q) &= \varphi([T(y; q)]_k) \frac{\partial [T(y; q)]_k}{\partial y_k}, \end{aligned} \tag{33}$$

where φ is the standard univariate Gaussian density. Drawing $\epsilon \sim \mathcal{N}(0, I)$ and evaluating $T^{-1}(\epsilon; q)$ produces a sample from this conditional density.

Proof. The prefix property holds for each layer by Eq. (25). If the first k coordinates of $z^{(j-1)}$ depend only on $y_{\le k}$, the same holds for the first k coordinates of $z^{(j)}$. Induction over layers proves the claim for T. Equation (26) gives the corresponding property for each inverse layer, and hence for their composition. Products of lower triangular matrices are lower triangular, and their diagonal entries multiply. Therefore,

$$\frac{\partial [T(y;q)]_k}{\partial y_k} = \exp\left(\sum_{j=1}^{K} a_{j,k}(z_{<k}^{(j-1)}, q)\right) > 0. \tag{34}$$

For a fixed $y_{<k}$, all intermediate prefixes $z_{<k}^{(j)}$ are fixed. The map from y_k to $[T(y;q)]_k$ is then a composition of affine maps with positive slopes, so it is strictly increasing and onto $\mathbb{R}$. The inverse is also coordinatewise increasing: its diagonal Jacobian entries are the reciprocals of those in Eq. (34), evaluated at $y = T^{-1}(\epsilon; q)$.

The standard Gaussian base factorizes over coordinates, and the triangular determinant is the product of the diagonal derivatives. Substitution into Eq. (28) yields the product in Eq. (33). Each factor depends only on $y_{\le k}$ and q, and a one-dimensional change of variable gives

$$\int_{\mathbb{R}} \varphi([T(y;q)]_k) \frac{\partial [T(y;q)]_k}{\partial y_k}\, dy_k = \int_{\mathbb{R}} \varphi(u)\, du = 1 \tag{35}$$

for every fixed prefix. Integrating the later coordinates in reverse order confirms that these factors are the conditional densities of the joint model. Finally, for any measurable set $A \subseteq \mathbb{R}^p$, the change of variables $\epsilon = T(y;q)$ gives

$$\Pr\{T^{-1}(\epsilon;q) \in A\} = \int_A p_\epsilon(T(y;q)) |\det D_y T(y;q)|\, dy = \int_A p_\theta(y \mid q)\, dy,$$

which proves the sampling claim. □

Computational consequences. When evaluating a known target patch, every input coordinate to a layer is available. A masked MLP therefore produces all p pairs of affine parameters in one forward pass, and density evaluation uses K such passes plus the sum of Kp log-scales. The latter sum does not require constructing a dense Jacobian; its cost is separate from the MLP evaluation cost. During generation, Eq. (26) requires the recovered prefix before the next coordinate can be computed. A direct implementation has p sequential coordinate updates per inverse layer, applied through the K layers in reverse order. Independent draws with fixed conditioning can still be batched. These are the density-evaluation and sampling trade-offs of MAF [46].

Scope and regularity. The ordering established above is conditional on q and concerns coordinates within a patch; it is not a claim of causal identification. The normalized density and exact likelihood concern complete p-dimensional patches. Truncating an output patch gives a marginal distribution on the retained coordinates and does not make the truncation invertible. If the parameter networks are locally Lipschitz rather than continuously differentiable, both the forward and recursive inverse maps are locally Lipschitz; the Jacobian and density identities then hold almost everywhere by the change-of-variables formula for such bijections. Finally, mathematical invertibility does not by itself guarantee good numerical conditioning: extremely small scales or large off-diagonal derivatives can amplify roundoff errors. The propositions establish structural properties of the specified flow, not universal approximation of arbitrary conditional densities.

I Additional Related Work

Learning compact temporal representations. FLAME's Legendre memory centers on retaining predictive information in a compact state. The broader forecasting literature examines representation design through selective patch spaces [65] and supervision through decomposition-based losses [66]. The search for economical designs also extends to irregular observations, where a simple forecasting baseline has been studied [67]. Joint time–frequency representations [68] and adaptive spatio-temporal graph interactions [69] offer further perspectives on encoding such data. These approaches place temporal memory within a wider set of choices about what information a forecaster represents and how that representation is learned.

Predictive uncertainty and available information. The distribution of future values depends on the information available to the predictor. GinAR+ [70] considers forecasting from incomplete multivariate series, and Merlin [71] develops multi-view representations for varying missing rates. Additional information enters through exogenous variables in DAG's dual correlation network [72], GCGNet's graph-consistent generative network [73], and KITE's knowledge-guided probabilistic model [74]. At the application level, point–interval battery lifetime prediction [75] illustrates how uncertainty estimates accompany a specific prognostic target. These settings motivate studying efficient distributional prediction under different information constraints, alongside FLAME's generative forecasting head.

Temporal structure across objectives. Dependencies among variables are relevant to both prediction and detection. CCD uses deformable convolutions to capture forecasting cross-correlations [76], whereas CATCH investigates channel-aware frequency patches for identifying anomalies [77]. Foundation-model adaptation likewise spans multivariate forecasting in CoRA [78] and anomaly detection in STAR [79]. TAB supplies a unified benchmark for the latter task [59]; TimeART broadens the task landscape through agentic reasoning augmented with tools [80]. This literature raises complementary questions about which temporal regularities can be reused when the learning objective changes.

Generation with explicit controls. FLAME models numerical futures, while research on visual sequences studies other forms of conditional generation. The controllable video generation survey [81] provides an overview of this field. Individual examples include Follow Your Pose for pose-guided synthesis [82] and Follow-Your-Emoji for expressive portrait animation [83]. LiveLight addresses interactive relighting of streaming video in real time [84]. Their output modalities and control mechanisms give a broader perspective on generative modeling, in which computational cost and the specification of desired outputs are also central design considerations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Appendix H.1 proves Theorem 2.1. Appendix H.2 states the assumptions and proves the conditional MAF's invertibility, Jacobian formula, normalized likelihood, and autoregressive sampling properties.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We set a fixed random seed to ensure deterministic results across all runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide an anonymous repository link for releasing the model code, and the complete reproducible model code is already available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide full experimental settings in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We use a fixed random seed but do not report error bars or statistical significance tests across independent runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the corresponding information in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We ensure that our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our research does not involve these concerns.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We follow the licences of open-source code and datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets introduced in the paper are well documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our research does not involve these concerns.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research does not involve these concerns.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Appendix G discloses the use of LLMs for language editing and assistance with mathematical exposition and proofs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.