Dirichlet-Guided Group Forecasting for Alleviating Over-smoothing in Time Series Forecasting

Xingyu Zhang[1,2] **Jingyao Wang**[1,2] **Xin Yu**[1,2] **Zeen Song**[3]
Jianqi Zhang[1,2] **Changwen Zheng**[1] **Wenwen Qiang**[1*]
[1] National Key Laboratory of Space Integrated Information System, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Guizhou Power Grid Co., Ltd. Digital-Intelligent Operations Center, Guizhou, China
{zhangxingyu23,wangjingyao22,yuxin261}@mails.ucas.ac.cn, zeensong@qq.com
jluzhangjianqi@163.com, {changwen,qiangwenwen}@iscas.ac.cn

Abstract

Time series forecasting often suffers from over-smoothing, especially when future dynamics are multi-modal. Forecasts may follow the coarse trend of the observed future, but fail to preserve sharp changes, oscillations, turning points, and regime transitions that define plausible dynamic evolution. In this work, we revisit over-smoothing from the perspective of latent dynamical mode compression: under partial observation and single-realization supervision, multiple plausible future modes can be weakened, merged, or averaged during forecasting. Based on this view, we propose Dirichlet-Guided Group Forecasting (DGF), a mode-preserving forecasting framework that explicitly models multiple mode-conditioned predictive distributions and uncertainty over their selection probabilities. DGF uses a Dirichlet-guided hierarchical sampling mechanism and reward-based optimization to encourage forecasts that are accurate, dynamically consistent, and mode-distinct. Extensive experiments on real-world forecasting benchmarks show that DGF reduces over-smoothing while improving forecasting accuracy, diversity, and dynamical consistency. Our code is publicly available at `https://github.com/Zh-XY22/DGF`.

1 Introduction

Time series forecasting (TSF) is a fundamental task in modern data-driven decision-making, with broad applications [Bilal et al., 2022, Duchon and Hale, 2012, Li et al., 2015]. However, many real-world forecasting models still suffer from over-smoothing, especially in multi-step prediction: forecasts may capture the coarse trend but fail to reproduce sharp peaks, abrupt changes, local oscillations, frequency variations, turning points, and regime shifts [Sun et al., 2025]. As illustrated in Figure 1, such forecasts may achieve acceptable average error, yet lose the dynamic structures that are critical for downstream decisions [Guen and Thome, 2019].

We rethink over-smoothing as a problem of latent dynamical mode compression. In many real systems, the observed history is only a partial view of the underlying state, while future evolution may also depend on unobserved disturbances, external conditions, hidden states, or stochastic mechanisms [Rangapuram et al., 2018, Salinas et al., 2020]. Consequently, similar observed histories can lead to qualitatively different future patterns, such as rising, declining, oscillatory, abrupt-transition, or regime-switching trajectories [Klushyn et al., 2026, Rasul et al., 2021a]. These patterns are not merely small perturbations around a common future; they may correspond to incompatible dynamic modes whose valid trajectory sets are not closed under linear averaging, a notion we formalize in Section 4.

*Corresponding author.

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

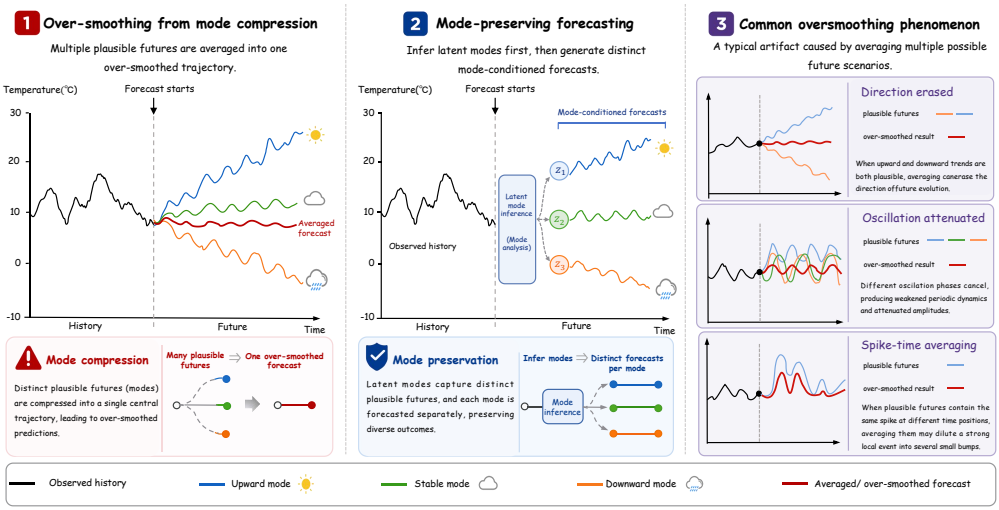


Figure 1: Illustration of over-smoothing and mode-preserving forecasting. Under the same observed history, multiple latent dynamical modes may lead to distinct plausible futures. A centralized forecast can compress these incompatible modes into an averaged trajectory with weakened dynamics, whereas mode-preserving forecasting maintains separate mode-conditioned trajectories.

However, each training input is usually paired with only one realized future trajectory, which reveals only one possible mode and leaves other dynamically plausible modes unobserved. Under such single-realization supervision, a forecasting model may suppress less frequently realized modes, merge nearby modes, or average incompatible modes into a trajectory that no longer preserves the structure of any individual future. From this perspective, over-smoothing is not only a loss-induced or frequency-domain artifact [Sun et al., 2025, Guen and Thome, 2019, LE GUEN and THOME, 2020], but can also be understood as the compression of non-convex latent dynamical modes.

This perspective suggests that mitigating over-smoothing requires more than generating multiple future samples. If the samples are drawn from an implicit output distribution without exposing their latent mode structure, they may still fail to preserve distinct dynamical regimes. A mode-preserving forecaster should instead separate two questions: what mode-conditioned futures are possible, and how strongly the current history supports each mode. This requires explicitly maintaining multiple mode-conditioned predictive distributions and modeling uncertainty over their mode-selection probabilities [Klushyn et al., 2026, Liu et al., 2025a]. However, representation alone is not sufficient. Since each input is supervised by only one realized future, directly optimizing all candidate modes toward the same ground truth can collapse different mode-conditioned distributions back to the observed trajectory. Therefore, a mode-preserving forecaster also needs an optimization signal that keeps generated trajectories accurate while encouraging dynamical consistency and valid diversity.

To this end, we propose Dirichlet-Guided Group Forecasting (DGF), a mode-preserving forecasting framework for mitigating over-smoothing. DGF learns multiple mode-conditioned predictive distributions to represent candidate future dynamics. It further places an input-dependent Dirichlet distribution over the mode-selection probability vector, separating the prediction of candidate futures from their selection. Forecasts are generated by sampling mode-selection probabilities and drawing trajectories from selected mode-conditioned distributions, avoiding deterministic aggregation of incompatible modes. To train this hierarchical stochastic process, DGF uses a GRPO-based objective [Shao et al., 2024] with trajectory-level accuracy and dynamical consistency rewards, group-level credit assignment, and a separate component-diversity regularizer. Experiments on real-world benchmarks demonstrate that DGF improves forecasting performance while producing more diverse and dynamically consistent future trajectories. In summary, our main contributions are as follows:

- We formulate over-smoothing in time series forecasting as latent dynamical mode compression under single-realization supervision, where multiple plausible future modes may be weakened, merged, or averaged in the learned forecasting process.

- We propose DGF, a mode-preserving probabilistic forecasting framework that jointly learns mode-conditioned predictive distributions and models second-order uncertainty over their mode-selection probability vector with a Dirichlet distribution.
- We develop a GRPO-based training objective that optimizes the hierarchical sampling process with accuracy, dynamical consistency, and diversity rewards.
- Experiments on real-world forecasting benchmarks show that DGF reduces over-smoothing while improving forecasting accuracy, diversity, and dynamical consistency.

2 Related Works

Probabilistic and Generative Forecasting Probabilistic and generative forecasting methods aim to represent the uncertainty of future time series beyond a single deterministic trajectory. Instead of producing only a point forecast, these methods model the conditional predictive distribution of future states given historical observations. Early parametric approaches typically assume predefined likelihood families, such as Gaussian or Student's t distributions. For example, DeepAR [Salinas et al., 2020] estimates likelihood parameters with autoregressive recurrent networks, while DeepState [Rangapuram et al., 2018] combines state space models with deep neural networks to learn probabilistic latent states. To relax restrictive parametric assumptions, quantile-based and non-parametric methods such as MQ-RNN [Wen et al., 2017] and SQF-RNN [Gasthaus et al., 2019] directly estimate multiple quantiles or spline-based quantile functions. More expressive generative forecasting methods further improve distributional flexibility: normalizing flow based methods [Rasul et al., 2021b] transform simple base distributions into complex predictive distributions, while diffusion-based approaches such as TimeGrad [Rasul et al., 2021a] and CSDI [Tashiro et al., 2021] generate diverse future trajectories through iterative denoising. These methods provide important advances in representing non-Gaussian and multi-modal predictive uncertainty; however, they usually model uncertainty directly in the observation space, with latent dynamical modes and their selection structure remaining implicit. In contrast, we explicitly learn multiple mode-conditioned predictive distributions and use a Dirichlet-guided group mechanism to model and optimize uncertainty over their mode composition, thereby targeting over-smoothing from the perspective of latent mode compression.

Deep Learning for Time Series Forecasting Deep learning has substantially reshaped time series forecasting (TSF) by enabling flexible representation learning from complex temporal data [LeCun et al., 2015, Lim and Zohren, 2021]. Early neural forecasting methods mainly relied on recurrent neural networks to model sequential dependencies [Hewamalage et al., 2021, Tang et al., 2021], while convolutional architectures were later adopted to capture local temporal patterns through causal or dilated convolutions [Bai et al., 2018, Zhan et al., 2023]. With the success of attention mechanisms, Transformer-based models became widely used for capturing long-range dependencies in TSF [Wen et al., 2023]. Recent studies have also revisited simpler yet effective architectures, including linear and MLP-based models such as DLinear [Zeng et al., 2023] and TiDE [Das et al., 2023], as well as models with stronger temporal inductive biases, such as PatchTST [Nie et al., 2023], TimesNet [Wu et al., 2023], FEDformer [Zhou et al., 2022], FreDF [Zhang et al., 2024], and iTransformer [Liu et al., 2024b]. These methods have significantly improved forecasting accuracy and temporal representation learning, but they mainly focus on how to encode historical dependencies, while the preservation and optimization of multiple plausible future modes remains less directly addressed.

Large Language Models and Foundation Models for TSF Inspired by the strong generalization ability of large language models (LLMs), recent studies have explored adapting pre-trained language models to time series forecasting by reformulating numerical sequences as textual prompts or token sequences. PromptCast [Xue and Salim, 2023] converts time series into sentence-like prompts, LLMTime [Gruver et al., 2024] represents numerical sequences as strings for zero-shot forecasting, and AutoTimes [Liu et al., 2025b] leverages autoregressive generation by conditioning LLMs on historical sequences. Other works align time series with the latent space of pre-trained models through reprogramming, prompt tuning, or lightweight fine-tuning, such as FPT [Zhou et al., 2023], Time-LLM [Jin et al., 2024], TEMPO [Cao et al., 2024], and UniTime [Liu et al., 2024a]. Beyond adapting general-purpose LLMs, recent time-series foundation models are trained directly on large-scale temporal corpora to support zero-shot or few-shot forecasting across domains, including decoder-only or tokenized forecasting models such as TimeGPT [Garza et al., 2024], TimesFM [Das et al., 2024], and Chronos [Ansari et al., 2024], as well as unified or patch-based foundation models such as

MOMENT [Goswami et al., 2024], MOIRAI [Woo et al., 2024], Timer [Liu et al., 2024c], and Timer-XL [Liu et al., 2025c]. These works mainly focus on representation transfer, scaling behavior, long-context modeling, or cross-domain generalization.

3 Preliminaries

3.1 Time Series Forecasting

Let $X = [x_1, \ldots, x_T] \in \mathbb{R}^T$ denote a historical series window, and let $Y = [x_{T+1}, \ldots, x_{T+S}] \in \mathbb{R}^S$ denote the future trajectory over a prediction horizon S. Given a training dataset $\mathcal{D} = \{(X^{(n)}, Y^{(n)})\}_{n=1}^N$, time series forecasting aims to infer the future trajectory from the observed history. A common deterministic formulation learns a single-valued forecasting function f_θ and outputs $\hat{Y} = f_\theta(X)$. The model is typically trained by empirical risk minimization:

$$\min_\theta \mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \ell \left(f_\theta(X^{(n)}), Y^{(n)} \right), \tag{1}$$

where $\ell(\cdot, \cdot)$ measures the discrepancy between the predicted trajectory and the observed future. In contrast, probabilistic forecasting represents a conditional predictive distribution $p_\theta(Y \mid X)$, from which forecast samples or summary statistics can be obtained.

3.2 The Dirichlet Distribution

The Dirichlet distribution is a distribution over probability vectors on the simplex. Let $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K] \in \Delta^{K-1}$, where $\pi_k \geq 0$ and $\sum_{k=1}^K \pi_k = 1$. Given concentration parameters $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$ with $\alpha_k > 0$, the Dirichlet density is

$$\mathrm{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}, \boldsymbol{\pi} \in \Delta^{K-1}, \text{where } B(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^K \alpha_k)}. \tag{2}$$

Let $\alpha_0 = \sum_{k=1}^K \alpha_k$. The expected probability of component k is $\mathbb{E}[\pi_k] = \alpha_k / \alpha_0$. The normalized parameters determine the mean probability vector, while the total concentration α_0 controls dispersion around this mean. Thus, Dirichlet distributions with the same mean can express different levels of variability in mode-selection probabilities. In DGF, π denotes a sampled mode-selection probability vector, and α parameterizes its input-dependent distribution.

4 Problem Analysis and Motivation

4.1 Over-Smoothing as Dynamic Structure Degradation

Over-smoothing in TSF refers to the degradation of dynamic structures in predicted future trajectories. A smoothed forecast may follow the coarse trend of the observed future, yet suppress local peaks, abrupt changes, oscillations, or regime transitions. Such forecasts can be acceptable under average value-level metrics while being dynamically uninformative. Therefore, over-smoothing is better understood as a mismatch between value-level accuracy and dynamic-structure preservation.

Let $\hat{Y}$ be a predicted trajectory and Y_{gt} be the observed future. We denote a conventional value-level discrepancy by $d_{\mathrm{val}}(\hat{Y}, Y_{gt})$. To describe dynamic preservation, let $\Phi : \mathcal{Y} \to \mathcal{Z}$ be a dynamic feature mapping that summarizes trajectory-level properties such as trend, volatility, frequency pattern, turning points, and transition behavior. The corresponding dynamic-structure discrepancy is $d_{\mathrm{dyn}}(\hat{Y}, Y_{gt}) = D(\Phi(\hat{Y}), \Phi(Y_{gt}))$, where $D(\cdot, \cdot)$ measures discrepancy in the dynamic feature space. The key observation is $d_{\mathrm{val}}(\hat{Y}, Y_{gt})$ small $\not\Rightarrow$ $d_{\mathrm{dyn}}(\hat{Y}, Y_{gt})$ small. Thus, a forecast may be close to the observed future in value space while losing the structural properties that make the trajectory dynamically plausible. This motivates analyzing over-smoothing through dynamic-structure preservation rather than only through average prediction error.

4.2 Latent-Mode Structure of Future Uncertainty

Future uncertainty in time series is often structured rather than purely random. For a given historical context X, the future trajectory Y may depend on an unobserved discrete latent mode M, which represents possible dynamical regimes such as trend directions, volatility states, oscillatory patterns, abrupt changes, or transition behaviors. We use M to describe future-relevant structure that is partially informed by the observed history, but not fully determined by it.

Assumption 4.1 (Partially observed latent future modes) *For a historical context X, future trajectory Y, and discrete latent dynamical mode M, the mode M satisfies*

$$I(X;M) > 0, H(M \mid X) > 0, I(M;Y \mid X) > 0. \tag{3}$$

This assumption states that the historical context contains information about the future mode, but does not determine it completely. Meanwhile, the latent mode provides additional information about the future trajectory beyond X alone. Thus, M is neither arbitrary noise independent of the input nor a redundant variable absorbed by the observed history; it captures partially observed future structure. Under this assumption, the conditional future distribution can be decomposed as

$$p(Y \mid X) = \sum_{m=1}^{K} p(Y \mid X, M = m)p(M = m \mid X). \tag{4}$$

Here, $p(Y \mid X, M = m)$ denotes the mode-conditioned future distribution, while $p(M = m \mid X)$ denotes the probability that the future follows the m-th latent mode under the observed history. This decomposition separates within-mode uncertainty, i.e., variability of trajectories under a fixed mode, from mode-selection uncertainty, i.e., uncertainty over which latent mode will be realized after conditioning on X. The key training difficulty is that the latent mode M is not observed. For each historical window, the dataset usually provides only one realized future trajectory, which corresponds to one realized mode. Other dynamically plausible modes under the same observed history are not directly supervised. Therefore, even if the conditional distribution is multi-modal, the available supervision for each input is a single realization. This single-realization supervision is the setting in which mode-conditioned structures can be compressed or collapsed during training.

4.3 Mode Averaging and Non-Convex Dynamic Structures

The latent-mode decomposition shows that the conditional future distribution may contain multiple mode-conditioned trajectory structures. However, estimating a predictive distribution and producing forecast outputs are different problems. A predictive distribution describes possible futures, whereas a forecasting system must eventually output either a trajectory or a finite set of trajectories. If the output decision reduces incompatible modes to a centralized forecast, the resulting trajectory may lose the dynamic structure of every individual mode.

To formalize this issue, let $\mathcal{Y} \subseteq \mathbb{R}^S$ denote the trajectory space over the prediction horizon. For each latent mode m, let $\mathcal{S}_m \subseteq \mathcal{Y}$ denote the set of dynamically valid trajectories under that mode, and let $\mathcal{S} = \bigcup_{m=1}^{K} \mathcal{S}_m$ denote the union of all mode-specific valid trajectory sets. Such a set is generally non-convex: averaging an increasing and a decreasing trajectory may produce a flat trajectory, and averaging oscillatory trajectories with different phases may suppress their amplitudes. Therefore, a forecast obtained by averaging across modes may no longer belong to any valid mode-specific structure. The following theorem connects this geometric observation to point-risk minimization.

Theorem 4.1 (Point-risk minimization can induce dynamically invalid forecasts) *Assume that the dynamically valid trajectory set $\mathcal{S} = \bigcup_{m=1}^{K} \mathcal{S}_m$ is non-convex. Then there exists a conditional distribution $p(Y \mid X = x)$ supported entirely on $\mathcal{S}$ such that the Bayes-optimal predictor under squared loss,*

$$f^\star(x) = \arg\min_{f(x)} \mathbb{E}[\|f(x) - Y\|_2^2 \mid X = x], \tag{5}$$

is dynamically invalid

$$Y \sim p(Y \mid X = x) \Rightarrow Y \in \mathcal{S} \text{almost surely}, f^\star(x) \notin \mathcal{S}. \tag{6}$$

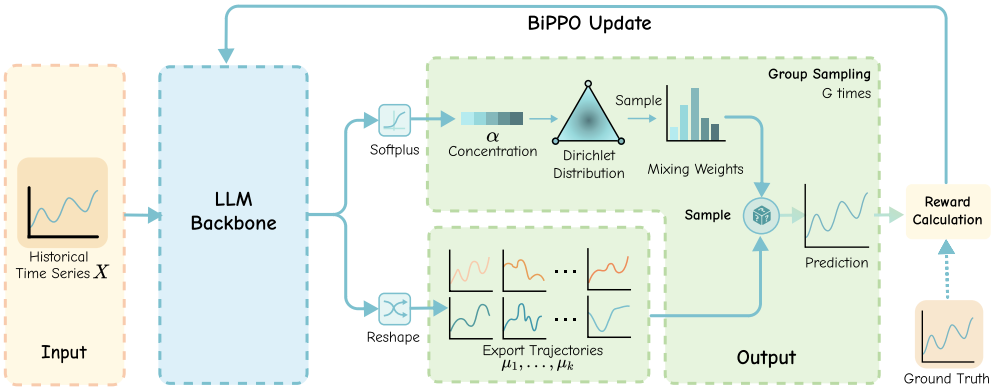


Figure 2: Overview of DGF. The backbone produces a contextual representation used to parameterize predictive components and a Dirichlet distribution over component-selection probabilities. For each input, B probability vectors are sampled. Within each mini-group, one sampled vector is held fixed while G forecasts are generated by sampling component indices and their corresponding predictive distributions. Trajectory-level and group-level credit assignment optimize the sampling process, while a separate diversity regularizer encourages component separation.

The proof is provided in Appendix D.1. Theorem 4.1 shows that over-smoothing can arise from the interaction between non-convex dynamic mode structures and centralized point decisions. The possible futures may all be dynamically valid, but the squared-loss optimal point forecast can still lie outside the valid dynamic-mode set. In this case, the forecast is not merely inaccurate in value space; it is dynamically degraded because it compresses incompatible latent modes into a trajectory that preserves none of them. However, simply increasing the number of prediction heads does not remove this compression. Under single-realization supervision, if all heads are optimized against the same observed future without mode-specific assignment or separation, the heads receive identical supervision and can collapse to the same solution.

Proposition 4.1 (Naive multi-head supervision admits a collapsed optimum) *Consider a multi-head forecaster with K deterministic heads $\{\mu_k(X)\}_{k=1}^K$ trained on a single realized target Y_{gt} for each input X using $\mathcal{L}_{\text{naive}}(X, Y_{gt}) = \sum_{k=1}^K \|\mu_k(X) - Y_{gt}\|_2^2$. For any fixed training sample (X, Y_{gt}), the unique minimizer satisfies $\mu_1(X) = \mu_2(X) = \cdots = \mu_K(X) = Y_{gt}$. For the population risk under squared loss, the minimizer satisfies $\mu_1^\star(X) = \mu_2^\star(X) = \cdots = \mu_K^\star(X) = \mathbb{E}[Y \mid X]$.*

The proof is provided in Appendix D.2. Proposition 4.1 explains why multiple predictive heads alone are insufficient for mode-preserving forecasting. If every head is directly pulled toward the same realized trajectory, then the model has no incentive to assign different heads to different latent modes. This is precisely the collapse mechanism that a mode-preserving method must counteract.

Together, Theorem 4.1 and Proposition 4.1 identify two potential sources of mode compression: averaging incompatible futures at prediction time and training all predictive heads toward the same target. These observations motivate separating component prediction from component selection, together with training signals that encourage dynamic consistency and component separation. DGF implements this design through mode-conditioned predictive distributions, Dirichlet-guided group sampling, and hierarchical reward-based optimization.

5 Method

Overview. As shown in Figure 2, DGF learns K mode-conditioned predictive distributions and an input-dependent Dirichlet distribution over their selection probabilities. A sampled probability vector is shared by multiple forecasts, forming a mini-group. Hierarchical GRPO assigns credit to individual forecasts and sampled probability vectors, while a separate diversity regularizer encourages separation among predictive components.

5.1 Dirichlet-Guided Group Forecasting

To preserve latent future modes, DGF separates mode-conditioned prediction from mode selection. Given a historical input X, an encoder first extracts a contextual representation $h = F_{\text{enc}}(X)$. Based on h, the model produces K mode-conditioned predictive distributions $q_{\theta,k}(Y \mid X) = g_{\theta,k}(h), k = 1, \ldots, K$, where each distribution is intended to capture one candidate latent dynamical mode. The distribution family can be chosen according to the task, such as Gaussian, Student-t, or other probabilistic output heads. In parallel, DGF predicts a Dirichlet concentration vector $\alpha_\theta(X) = [\alpha_{\theta,1}(X), \ldots, \alpha_{\theta,K}(X)], \alpha_{\theta,k}(X) > 0$. In practice, we parameterize it by evidence scores $e_\theta(X) = g_\alpha(h)$ and set $\alpha_{\theta,k}(X) = \alpha_{min} + \text{softplus}(e_{\theta,k}(X))$. A deterministic softmax gate outputs one mode-selection probability vector for each input. DGF instead parameterizes a distribution over such vectors, $p(\pi \mid X) = \text{Dir}(\pi \mid \alpha_\theta(X))$. The normalized parameters specify the expected component probabilities, while the total concentration controls variability around them. Sampled probability vectors represent alternative mode-composition hypotheses, which are evaluated through the group-level mechanism below.

For each input X, DGF samples B independent mode-selection probability vectors, $\pi_b \sim \text{Dir}(\alpha_\theta(X))$, where $b = 1, \ldots, B$. Each vector is shared by a mini-group of G forecasts. For the i-th forecast in group b, we sample

$$z_{b,i} \sim \text{Cat}(\pi_b), \qquad \hat{Y}_{b,i} \sim q_{\theta,z_{b,i}}(Y \mid X), \qquad i = 1, \ldots, G. \tag{7}$$

Here, $z_{b,i}$ is the selected predictive-component index. The joint density for a probability vector, component index, and forecast is

$$p_\theta(\pi_b, z_{b,i}, \hat{Y}_{b,i} \mid X) = \text{Dir}(\pi_b \mid \alpha_\theta(X)) \, \text{Cat}(z_{b,i} \mid \pi_b) \, q_{\theta,z_{b,i}}(\hat{Y}_{b,i} \mid X). \tag{8}$$

If each trajectory independently samples its own Dirichlet vector, its marginal component-selection probability is

$$P(z_i = k \mid X) = \mathbb{E}[\pi_{i,k}] = \frac{\alpha_{\theta,k}(X)}{\alpha_0(X)}, \qquad \alpha_0(X) = \sum_{k=1}^{K} \alpha_{\theta,k}(X). \tag{9}$$

Group sharing preserves this single-trajectory marginal but induces dependence among component selections within a mini-group. It also allows each sampled composition to be evaluated using multiple forecasts. These mini-groups are credit-assignment units for the same input, distinct from optimization mini-batches.

5.2 Reward Design and Hierarchical GRPO

The hierarchical sampling process in DGF involves three different optimization targets. Sampled trajectories should be accurate and dynamically plausible; the Dirichlet sampler should favor mode compositions that produce high-quality forecast groups; and the K mode-conditioned predictive distributions should remain separated to avoid mode collapse. Therefore, we separate trajectory-level rewards, Dirichlet-level credit assignment, and distribution-level diversity regularization.

For each sampled trajectory $\hat{Y}_{b,i}$ in the b-th mini-group, we define the trajectory reward as $r^{\text{traj}}_{b,i} = \lambda_1 r_{\text{acc},b,i} + \lambda_2 r_{\text{dyn},b,i}$. The accuracy term grounds the forecast in the realized future, e.g., $r_{\text{acc},b,i} = -\text{MSE}(\hat{Y}_{b,i}, Y_{\text{gt}})$, or other forecasting scores such as MAE, CRPS, or log-likelihood. The dynamics term uses a fixed set of temporal descriptors

$$\begin{aligned} r_{\text{dyn},b,i} = &- \eta_1 \|\Delta \hat{Y}_{b,i} - \Delta Y_{\text{gt}}\|_1 - \eta_2 \|\Delta^2 \hat{Y}_{b,i} - \Delta^2 Y_{\text{gt}}\|_1 \\ &- \eta_3 \|A(\hat{Y}_{b,i}) - A(Y_{\text{gt}})\|_1 - \eta_4 d_{\text{trans}}(X, \hat{Y}_{b,i}), \end{aligned} \tag{10}$$

where Δ and Δ^2 capture local slope and curvature, $A(\cdot)$ is the normalized FFT amplitude spectrum, and d_{trans} penalizes discontinuity between the historical window and the predicted horizon. Thus, the trajectory reward evaluates both value-level accuracy and dynamic consistency.

We encourage diversity among the K predictive components through a separate regularizer. Distances between sampled trajectories alone do not establish component separation: even identical predictive distributions can generate different samples. Moreover, a diversity bonus shared by all trajectories in

a mini-group is removed by within-group advantage centering. We therefore regularize the predictive components directly:

$$\mathcal{R}_{\mathrm{div}}^{K}(X) = \frac{1}{K(K-1)} \sum_{k \neq l} \min(D_{\mathrm{mode}}(q_{\theta,k}(\cdot \mid X), q_{\theta,l}(\cdot \mid X)), m), \tag{11}$$

Here, D_{mode} measures separation between predictive components, and the margin m caps the reward for each pair. Let $\mu_k(X)$ denote the mean trajectory of $q_{\theta,k}(Y \mid X)$. We define component separation using value and dynamic-feature distances:

$$D_{\mathrm{mode}}(q_{\theta,k}, q_{\theta,l}) = D_{\mathrm{val}}(\mu_k, \mu_l) + \rho D_{\mathrm{dyn}}(\Phi(\mu_k), \Phi(\mu_l)). \tag{12}$$

This regularizer encourages distinct component means and dynamic features. Once a pairwise distance reaches m, further separation yields no additional diversity reward. We optimize the hierarchical sampling process using separate trajectory-level and Dirichlet-level credit assignment. For sample-level credit assignment, forecasts are compared within the same mini-group. Given trajectory rewards $\{r_{b,i}^{\mathrm{traj}}\}_{i=1}^{G}$, we compute $\mu_b = \frac{1}{G}\sum_{j=1}^{G} r_{b,j}^{\mathrm{traj}}$, $\sigma_b = \mathrm{Std}(\{r_{b,j}^{\mathrm{traj}}\}_{j=1}^{G})$, and $A_{b,i}^{\mathrm{sample}} = \frac{r_{b,i}^{\mathrm{traj}} - \mu_b}{\sigma_b + \epsilon}$. The sample-level objective is

$$\mathcal{J}_{\mathrm{sample}} = \sum_{b} \sum_{i=1}^{G} A_{b,i}^{\mathrm{sample}} [\log p(z_{b,i} \mid \pi_b) + \log q_{\theta, z_{b,i}}(\hat{Y}_{b,i} \mid X)]. \tag{13}$$

This term reinforces better trajectories and updates the selected mode-conditioned predictive distribution. For Dirichlet-level credit assignment, π_b must be evaluated by the quality of the whole mini-group. Directly applying normalized sample-level advantages to $\log p_\theta(\pi_b \mid \alpha_\theta(X))$ would cancel out because π_b is shared within the group. We therefore define the mini-group reward as $R_b = \frac{1}{G}\sum_{i=1}^{G} r_{b,i}^{\mathrm{traj}}$. Across B mini-groups, we compute $\mu_R = \frac{1}{B}\sum_{b=1}^{B} R_b$, $\sigma_R = \mathrm{Std}(\{R_b\}_{b=1}^{B})$, and $A_b^{\mathrm{dir}} = \frac{R_b - \mu_R}{\sigma_R + \epsilon}$. The Dirichlet-level objective is

$$\mathcal{J}_{\mathrm{dir}} = \sum_{b=1}^{B} A_b^{\mathrm{dir}} \log p_\theta(\pi_b \mid \alpha_\theta(X)), \tag{14}$$

which favors mode-composition hypotheses that generate better forecast groups. The final objective is $\mathcal{J}_{\mathrm{DGF}} = \mathcal{J}_{\mathrm{sample}} + \gamma \mathcal{J}_{\mathrm{dir}} + \lambda_3 \mathcal{R}_{\mathrm{div}}^{K}$. Here, $\mathcal{J}_{\mathrm{sample}}$ optimizes individual trajectories and selected predictive heads, $\mathcal{J}_{\mathrm{dir}}$ optimizes the Dirichlet mode-composition distribution, and $\mathcal{R}_{\mathrm{div}}^{K}$ encourages separation among the K predictive components.

Point forecasting. For deterministic LSF evaluation, we use the mean trajectory of the component with the largest expected selection probability:

$$\bar{\pi}_k(X) = \frac{\alpha_{\theta,k}(X)}{\alpha_0(X)}, \qquad k^* = \arg\max_k \bar{\pi}_k(X), \qquad \hat{Y}_{\mathrm{point}} = \mu_{k^*}(X). \tag{15}$$

We refer to this rule as MAP-mode output. It depends only on the input and model predictions and avoids averaging across component means. All main LSF results use this rule; alternative decision protocols are examined in Appendix F.

6 Experiment

We evaluate DGF on deterministic and probabilistic forecasting benchmarks. The experiments examine whether DGF preserves standard forecasting accuracy, improves probabilistic forecast quality, and mitigates over-smoothing and dynamic-structure degradation. Dataset details, finetuning baselines, metric definitions, full results, and component ablations are provided in Appendix E and F.

6.1 Main Results

Long Sequence Forecasting We evaluate deterministic forecasting performance on standard LSF benchmarks following [Woo et al., 2024]. Since LSF evaluation requires a single reported trajectory, DGF reports the mode-selected forecast; alternative decision protocols are analyzed in Appendix F. As shown in Table 1, DGF consistently improves both Moirai-small and Moirai-base across datasets and achieves the best average performance on most benchmarks. These results indicate that the proposed mode-preserving generation mechanism improves standard point-forecasting accuracy without relying on deterministic aggregation of all mode-conditioned distributions.

Table 1: Long sequence forecasting results averaged over prediction lengths $\{96, 192, 336, 720\}$. The best results for each metric are highlighted in **bold**.

Method	ETTm1		ETTm2		ETTh1		ETTh2		Electricity		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear[2023]	0.403	0.407	0.350	0.401	0.456	0.452	0.559	0.515	0.212	0.300	0.265	0.317
PatchTST[2023]	0.387	0.400	0.281	0.326	0.469	0.455	0.387	0.407	0.216	0.304	0.259	0.281
iTransformer[2024b]	0.407	0.410	0.288	0.332	0.454	0.448	0.383	0.407	0.178	0.270	0.258	0.278
TimeMixer[2024]	0.381	0.396	0.275	0.323	0.447	0.440	0.365	0.395	0.182	0.273	0.240	0.272
SimpleTM [2025]	0.381	0.396	0.275	0.322	0.423	0.428	0.353	0.391	0.166	0.261	0.243	0.271
Moirai-small	0.448	0.410	0.300	0.341	0.416	0.427	0.355	0.381	0.233	0.320	0.268	0.279
+ *Full finetuning*	0.367	0.382	0.273	0.316	0.415	0.428	0.352	0.378	0.193	0.279	0.228	0.254
+ *Prompt tuning*	0.384	0.391	0.291	0.334	0.414	0.428	0.354	0.381	0.217	0.304	0.235	0.258
+ *LoRA*	0.370	0.383	0.272	0.314	0.414	0.427	0.354	0.380	0.192	0.279	0.224	0.252
+ *MSFT [2026]*	0.353	0.377	0.250	0.301	0.412	0.426	0.349	0.375	0.187	0.275	0.215	0.248
Moirai-base	0.382	0.388	0.281	0.326	0.412	0.424	0.356	0.388	0.188	0.274	0.246	0.265
+ *Full finetuning*	0.368	0.371	0.258	0.307	0.408	0.424	0.357	0.385	0.173	0.264	0.231	0.258
+ *Prompt tuning*	0.378	0.386	0.279	0.325	0.411	0.423	0.360	0.387	0.183	0.271	0.230	0.255
+ *LoRA*	0.361	0.370	0.259	0.307	0.408	0.423	0.356	0.387	0.172	0.263	0.230	0.260
+ *MSFT [2026]*	0.332	0.369	0.247	0.305	0.407	0.422	0.352	0.383	0.169	0.260	0.213	0.244
DGF with Moirai-small	0.338	0.369	0.238	0.292	0.399	0.420	**0.337**	**0.368**	0.177	0.265	0.207	0.235
DGF with Moirai-base	**0.317**	**0.357**	**0.234**	**0.290**	**0.395**	**0.410**	0.340	0.374	**0.157**	**0.246**	**0.203**	**0.232**

Probabilistic Forecasting We further evaluate distributional forecast quality using CRPS and MSIS, which measure the accuracy and calibration of the predictive distribution. As shown in Table 2, DGF achieves the best or competitive results across most datasets, suggesting that mode-conditioned sampling improves not only individual forecasts but also the quality of the generated predictive distribution. Additional probabilistic forecasting metrics are reported in Table 11.

Table 2: Probabilistic forecasting results measured by CRPS and MSIS. Lower values are better, and the best results are highlighted in **bold**.

Method	Electricity		Solar		Weather		Istanbul Traffic		Turkey Power	
	CRPS	MSIS	CRPS	MSIS	CRPS	MSIS	CRPS	MSIS	CRPS	MSIS
DeepAR[2020]	0.065	6.893	0.431	11.181	0.132	21.651	0.108	4.094	0.066	13.520
PatchTST[2023]	0.052	5.744	0.518	8.447	0.059	7.759	0.112	3.813	0.054	8.978
TiDE[2023]	0.048	5.672	0.420	13.754	0.054	8.095	0.110	4.752	0.046	8.579
Moirai-small	0.072	7.999	0.471	8.425	0.049	5.236	0.173	5.937	0.048	7.127
+ *Full finetuning*	0.055	6.009	0.395	6.947	0.039	4.477	0.151	6.735	0.040	6.887
+ *Prompt tuning*	0.066	6.595	0.421	6.936	0.050	4.901	0.154	4.733	0.045	7.042
+ *LoRA*	0.064	6.753	0.372	6.582	0.039	4.386	0.154	4.753	0.042	7.051
+ *MSFT [2026]*	0.047	5.327	0.353	7.706	0.036	4.178	0.141	4.447	0.038	6.810
Moirai-base	0.055	6.172	0.419	7.011	0.041	5.136	0.116	4.461	0.040	6.766
+ *Full finetuning*	0.049	5.414	0.188	4.292	0.038	5.282	0.120	7.272	0.036	6.712
+ *Prompt tuning*	0.054	6.024	0.412	6.885	0.040	5.274	0.105	3.987	0.040	6.698
+ *LoRA*	0.051	5.651	0.382	6.745	0.037	4.904	0.113	4.752	0.036	6.744
+ *MSFT [2026]*	0.046	5.199	**0.142**	3.464	0.035	4.603	0.098	**3.685**	0.034	6.419
DGF with Moirai-small	0.039	5.298	0.348	6.632	0.033	**4.057**	0.128	4.389	0.031	6.759
DGF with Moirai-base	**0.031**	**5.168**	0.164	**3.219**	**0.030**	4.159	**0.072**	3.702	**0.028**	**6.264**

6.2 Over-Smoothing Analysis

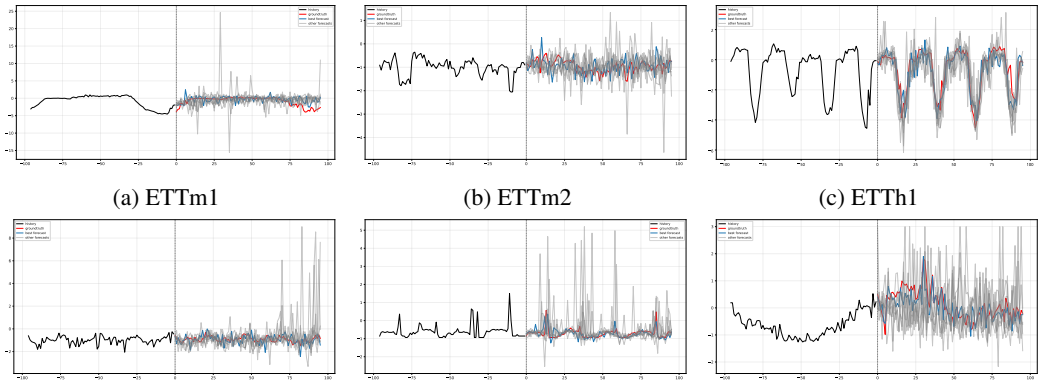
To examine dynamic-structure preservation, we evaluate generated forecast sets using the STRIPE protocol [LE GUEN and THOME, 2020]. We report mean-sample and best-sample MSE and DILATE. Mean-sample scores measure average forecast quality, while best-sample scores measure how closely the best available candidate matches the observed future under a fixed sampling budget. DILATE additionally evaluates shape discrepancy and temporal distortion. As shown in Table 3, DGF improves both mean-sample and best-sample scores over the Moirai-base backbone. The mean-sample improvements indicate better overall sample quality, while the best-sample improvements indicate that the generated set contains candidates closer to the observed future. Lower DILATE scores provide evidence of improved dynamic fidelity.

Table 3: Forecasting results on the Traffic and Electricity datasets, averaged over 5 runs (mean ± std). Metrics are scaled for readability. Lower values are better.

Method	Traffic				Electricity			
	MSE		DILATE		MSE		DILATE	
	mean	best	mean	best	mean	best	mean	best
N-BEATS MSE	–	7.8 ± 0.3	–	22.1 ± 0.8	–	24.6 ± 0.9	–	29.3 ± 1.3
N-BEATS DILATE	–	17.1 ± 0.8	–	17.8 ± 0.3	–	38.9 ± 1.9	–	20.7 ± 0.5
DeepAR	15.1 ± 1.7	6.6 ± 0.7	30.3 ± 1.9	16.9 ± 0.6	67.6 ± 5.1	25.6 ± 0.4	59.8 ± 5.2	17.2 ± 0.3
cVAE-DILATE	10.0 ± 1.7	8.8 ± 1.6	19.1 ± 1.2	17.0 ± 1.1	28.9 ± 0.8	27.8 ± 0.8	24.6 ± 1.4	22.4 ± 1.3
Variety Loss	9.8 ± 0.8	7.9 ± 0.8	18.9 ± 1.4	15.9 ± 1.2	29.4 ± 1.0	27.7 ± 1.0	24.7 ± 1.1	21.6 ± 1.0
Entropy Regularization	11.4 ± 1.3	10.3 ± 1.4	19.1 ± 1.4	16.8 ± 1.3	34.4 ± 4.1	32.9 ± 3.8	29.8 ± 3.6	25.6 ± 3.1
Diverse DPP	11.2 ± 1.8	6.9 ± 1.0	20.5 ± 1.0	14.7 ± 1.0	31.5 ± 0.8	25.8 ± 1.3	26.6 ± 1.0	19.4 ± 1.0
STRIPE S+T	10.1 ± 0.4	6.5 ± 0.2	19.2 ± 0.8	14.2 ± 0.2	29.7 ± 0.3	23.4 ± 0.2	24.4 ± 0.3	16.9 ± 0.2
Moirai-base	0.420 ± 0.01	0.375 ± 0.01	0.232 ± 0.01	0.208 ± 0.01	0.202 ± 0.01	0.182 ± 0.01	0.111 ± 0.01	0.046 ± 0.01
Full finetuning	0.407 ± 0.01	0.353 ± 0.01	0.174 ± 0.01	0.158 ± 0.01	0.171 ± 0.01	0.148 ± 0.01	0.085 ± 0.01	0.039 ± 0.01
DGF	0.295 ± 0.01	0.247 ± 0.01	0.142 ± 0.01	0.120 ± 0.01	0.128 ± 0.01	0.103 ± 0.01	0.052 ± 0.01	0.021 ± 0.01

6.3 Prediction Result Visualization

Figure 3 visualizes eight forecast samples for the same input on six datasets, using a prediction horizon of 96. Each panel includes the last 96 historical observations and the observed future. The samples illustrate variation in predicted levels and local temporal patterns. This visualization complements the quantitative sample-quality and dynamic-fidelity evaluation in Section 6.2.



(a) ETTm1 (b) ETTm2 (c) ETTh1

(d) ETTh2 (e) Electricity (f) Weather

Figure 3: Qualitative visualization of DGF forecasts on six datasets with prediction horizon 96. Each panel shows the last 96 historical observations, the observed future, and eight sampled forecasts, illustrating variation among the generated trajectories.

7 Conclusion

In this paper, we study over-smoothing in time series forecasting from the perspective of latent dynamical mode compression. We propose DGF, which combines mode-conditioned predictive distributions, Dirichlet-guided group sampling, and hierarchical reward-based optimization with component-diversity regularization. For point forecasting, DGF selects a component mean rather than averaging across components. Experiments on long-sequence and probabilistic forecasting benchmarks demonstrate improved forecasting accuracy and sample quality, with DILATE results providing evidence of better dynamic-structure preservation.

Acknowledgments and Disclosure of Funding

The authors sincerely thank the anonymous reviewers for their valuable comments and constructive feedback. This work was supported by the National Natural Science Foundation of China under Grant No. 62506355.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

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

Muhammad Bilal, Hyeok Kim, Muhammad Fayaz, and Pravin Pawar. Comparative analysis of time series forecasting approaches for household electricity consumption prediction, 2022.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting, 2024. URL https://arxiv.org/abs/2310.04948.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

Abhimanyu Das, Weihao Kong, Andrew Leach, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder, 2023.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024. URL https://arxiv.org/abs/2310.10688.

Claude Duchon and Robert Hale. Time series analysis in meteorology and climatology: An introduction, 2012.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1, 2024. URL https://arxiv.org/abs/2310.03589.

Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. Probabilistic forecasting with spline quantile function rnns. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1901–1910. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/gasthaus19a.html.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters, 2024. URL https://arxiv.org/abs/2310.07820.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models, 2019. URL https://arxiv.org/abs/1909.09020.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, January 2021. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2020.06.008. URL http://dx.doi.org/10.1016/j.ijforecast.2020.06.008.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

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, 2024. URL https://arxiv.org/abs/2310.01728.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep state-space models, 2026. URL https://arxiv.org/abs/2602.23050.

Vincent LE GUEN and Nicolas THOME. Probabilistic time series forecasting with shape and temporal diversity. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4427–4440. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/2f2b265625d76a6704b08093c652fd79-Paper.pdf.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539. Number: 7553 Publisher: Nature Publishing Group.

Li Li, Xiaonan Su, Yi Zhang, Yuetong Lin, and Zhiheng Li. Trend modeling for traffic time series analysis: An integrated study. *IEEE Transactions on Intelligent Transportation Systems*, 16(6): 3430–3439, 2015.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting, 2024a. URL https://arxiv.org/abs/2310.09751.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=SrEOUSyJcR.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024b. URL https://arxiv.org/abs/2310.06625.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models, 2024c. URL https://arxiv.org/abs/2402.02368.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: autoregressive time series forecasters via large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025b. Curran Associates Inc. ISBN 9798331314385.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=KMCJXjlDDr.

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

Adam Paszke, S. Gross, Francisco Massa, A. Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Z. Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach 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. *NeurIPS*, 2019.

Zhongzheng Qiao, Chenghao Liu, Yiming Zhang, Ming Jin, Quang Pham, Qingsong Wen, Ponnuthurai Nagaratnam Suganthan, Xudong Jiang, and Savitha Ramasamy. Multi-scale finetuning for encoder-based time series foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=OPOBV0zXu7.

Syama Sundar Rangapuram, Matthias Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7796–7805, Red Hook, NY, USA, 2018. Curran Associates Inc.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, 2021a. URL https://arxiv.org/abs/2101.12072.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=WiGQBFuVRv.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.07.001. URL https://www.sciencedirect.com/science/article/pii/S0169207019301888.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Yanru Sun, Zongxia Xie, Dongyue Chen, Emadeldeen Eldele, and Qinghua Hu. Hierarchical classification auxiliary network for time series forecasting. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i19.34286. URL https://doi.org/10.1609/aaai.v39i19.34286.

Yuqing Tang, Fusheng Yu, Witold Pedrycz, Xiyang Yang, Jiayin Wang, and Shihu Liu. Building trend fuzzy granulation-based lstm recurrent neural network for long-term time-series forecasting. *IEEE transactions on fuzzy systems*, 30(6):1599–1613, 2021.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: conditional score-based diffusion models for probabilistic time series imputation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting, 2024. URL https://arxiv.org/abs/2405.14616.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey, 2023.

Ruofeng Wen, Kari Torkkola, and Balakrishnan Narayanaswamy. A multi-horizon quantile recurrent forecaster. 11 2017. doi: 10.48550/arXiv.1711.11053.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis, 2023.

Hao Xue and Flora D. Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting, 2023. URL https://arxiv.org/abs/2210.08964.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2023.

Tianxiang Zhan, Yuanpeng He, Yong Deng, and Zhen Li. Differential convolutional fuzzy time series forecasting, 2023.

Xingyu Zhang, Siyu Zhao, Zeen Song, Huijie Guo, Jianqi Zhang, Changwen Zheng, and Wenwen Qiang. Not all frequencies are created equal:towards a dynamic fusion of frequencies in time-series forecasting, 2024. URL https://arxiv.org/abs/2407.12415.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022.

Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all:power general time series analysis by pretrained lm, 2023. URL https://arxiv.org/abs/2302.11939.

A Limitations and Broader Impact

Limitations Although DGF is a model-agnostic framework for mode-preserving forecasting, it still has several limitations. First, the reward-based optimization stage benefits from a reasonably good initialization. Since the rewards are defined over sampled trajectories, starting from a poorly trained forecasting model may lead to inefficient exploration and unstable optimization. In this work, we use pretrained time series backbones as a warm start to make the hierarchical sampling process stable and reproducible. This choice is not required by the DGF formulation itself, and the framework can in principle be applied to other forecasting backbones or trained after a supervised pretraining stage. However, the quality of the initialization may affect optimization efficiency and final performance. Second, DGF introduces additional sampling cost compared with deterministic single-trajectory forecasting. During training, the model samples Dirichlet mode-composition vectors, mode indices, and mode-conditioned trajectories, and hierarchical GRPO requires multiple samples or mini-groups to estimate relative advantages. This increases computation and memory usage. We view this overhead as the cost of preserving multiple latent future modes rather than collapsing them into a single averaged forecast. In practice, this cost can be controlled by the number of modes, samples, and mini-groups, and future work can further reduce it through efficient sampling, mode pruning, or distillation.

Broader Impact. This work proposes a mode-preserving forecasting paradigm for time series prediction. Instead of forcing a model to output a single averaged future, DGF encourages the model to represent multiple plausible future modes and reason about their selection uncertainty. This perspective may benefit applications where dynamic events, regime shifts, or rare but important future patterns matter, such as energy management, weather forecasting, traffic control, and financial risk monitoring. By producing forecasts that are more dynamically consistent and mode-diverse, the proposed framework may help downstream decision-makers better understand uncertainty and avoid over-reliance on over-smoothed predictions. At the same time, forecasts generated by DGF should still be interpreted with caution in high-stakes domains. The generated candidates reflect model-based estimates rather than guaranteed future outcomes, and incorrect or overconfident use of multi-modal forecasts may still lead to poor decisions. Appropriate calibration, domain validation, and human oversight remain necessary when deploying such forecasting systems in practice.

B Use of Large Language Models

Large language models (LLMs) were used solely as a general-purpose tool for grammar and language polishing. They were not involved in research ideation, methodology, analysis, or substantive writing. The authors take full responsibility for all contents of the paper.

C Notation List

We summarize the used notation in this paper as

- $X = [x_1, \ldots, x_T] \in \mathbb{R}^T$: historical input time series window with length T.
- $Y = [x_{T+1}, \ldots, x_{T+S}] \in \mathbb{R}^S$: future trajectory over prediction horizon S.
- Y_{gt}: observed ground-truth future trajectory paired with the historical input X.
- $\hat{Y}$: predicted future trajectory generated by the forecasting model.
- $\mathcal{D} = \{(X^{(n)}, Y^{(n)})\}_{n=1}^{N}$: training dataset containing N historical-future pairs.
- T: length of the historical input window.
- S: length of the prediction horizon.
- $M \in \{1, \ldots, K\}$: discrete latent dynamical mode that represents a possible future regime or dynamic pattern.
- K: number of latent modes or mode-conditioned predictive distributions.
- $p(Y \mid X)$: conditional future distribution given historical input X.
- $p(Y \mid X, M = m)$: mode-conditioned future distribution under latent mode m.

- $p(M = m \mid X)$: probability that the future follows the m-th latent mode given X.
- $q_{\theta,k}(Y \mid X)$: the k-th mode-conditioned predictive distribution parameterized by model parameters θ.
- $h = F_{\text{enc}}(X)$: contextual representation extracted from the historical input by the encoder.
- $g_{\theta,k}(\cdot)$: prediction head that maps the contextual representation h to the k-th mode-conditioned predictive distribution.
- $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$: mode-selection probability vector on the $(K-1)$-simplex.
- Δ^{K-1}: $(K-1)$-dimensional probability simplex, defined by $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$.
- $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$: Dirichlet concentration parameter vector.
- $\alpha_\theta(X) = [\alpha_{\theta,1}(X), \ldots, \alpha_{\theta,K}(X)]$: input-dependent Dirichlet concentration vector predicted by DGF.
- $\alpha_0 = \sum_{k=1}^{K} \alpha_k$: total concentration of the Dirichlet distribution, controlling dispersion around its mean.
- $e_\theta(X)$: evidence score vector used to parameterize the Dirichlet concentration parameters.
- $\text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$: Dirichlet distribution over probability vector $\boldsymbol{\pi}$ with concentration parameter $\boldsymbol{\alpha}$.
- $\pi_b \sim \text{Dir}(\alpha_\theta(X))$: sampled mode-selection probability vector for the b-th mini-group.
- B: number of mini-groups sampled for the same input.
- G: number of forecast samples generated within each mini-group.
- $z_{b,i} \sim \text{Cat}(\pi_b)$: latent mode index sampled for the i-th trajectory in the b-th mini-group.
- $\hat{Y}_{b,i}$: sampled forecast trajectory generated from the selected mode-conditioned predictive distribution in the b-th mini-group.
- $p_\theta(\pi_b, z_{b,i}, \hat{Y}_{b,i} \mid X)$: joint probability of the hierarchical sampling process for a sampled mode-selection vector, mode index, and forecast trajectory.
- $\Phi(\cdot)$: dynamic feature mapping that extracts trajectory-level structural properties such as trend, volatility, frequency pattern, turning points, and transition behavior.
- $\Psi(X, \hat{Y})$: transition feature mapping that characterizes the dynamic transition from historical input X to generated future $\hat{Y}$.
- $d_{\text{val}}(\hat{Y}, Y_{\text{gt}})$: value-level discrepancy between a forecast and the observed future.
- $d_{\text{dyn}}(\Phi(\hat{Y}), \Phi(Y_{\text{gt}}))$: dynamic-structure discrepancy between the generated trajectory and the observed future in dynamic feature space.
- $d_{\text{trans}}(\Psi(X, \hat{Y}))$: transition discrepancy that penalizes dynamically implausible transitions from history to future.
- $\mathcal{Y} \subseteq \mathbb{R}^S$: trajectory space over the prediction horizon.
- $\mathcal{S}_m \subseteq \mathcal{Y}$: set of dynamically valid trajectories under latent mode m.
- $\mathcal{S} = \bigcup_{m=1}^{K} \mathcal{S}_m$: union of all dynamically valid trajectory sets.
- $r_{b,i}^{\text{traj}}$: trajectory-level reward assigned to the i-th sampled forecast in the b-th mini-group.
- $r_{\text{acc},b,i}$: accuracy reward measuring closeness between $\hat{Y}_{b,i}$ and Y_{gt}.
- $r_{\text{dyn},b,i}$: dynamics reward measuring dynamic consistency and transition plausibility of $\hat{Y}_{b,i}$.
- $\mathcal{R}_{\text{div}}^{K}(X)$: distribution-level diversity regularizer encouraging the K mode-conditioned predictive distributions to remain distinct.
- $D_{\text{mode}}(q_{\theta,k}, q_{\theta,l})$: distance between two mode-conditioned predictive distributions.
- $D_{\text{val}}(\mu_k, \mu_l)$: value-space distance between mean trajectories of two mode-conditioned predictive distributions.
- $D_{\text{dyn}}(\Phi(\mu_k), \Phi(\mu_l))$: dynamic-feature-space distance between two mode-conditioned mean trajectories.

- $\mu_k(X) = \mathbb{E}_{q_{\theta,k}}[Y]$: mean trajectory of the k-th mode-conditioned predictive distribution.
- m: margin used in the distribution-level diversity regularizer to prevent rewarding arbitrary divergence.
- $\lambda_1, \lambda_2, \lambda_3$: weights for accuracy reward, dynamics reward, and distribution-level diversity regularization.
- $\eta_1, \eta_2, \eta_3, \eta_4$: weights controlling the two components of the dynamics reward.
- ρ: coefficient balancing value-space and dynamic-feature-space distances in D_{mode}.
- μ_b: mean trajectory-level reward within the b-th mini-group.
- σ_b: standard deviation of trajectory-level rewards within the b-th mini-group.
- $A_{b,i}^{\text{sample}}$: sample-level advantage for the i-th forecast in the b-th mini-group.
- R_b: average trajectory-level reward of the b-th mini-group.
- μ_R: mean mini-group reward across all B mini-groups.
- σ_R: standard deviation of mini-group rewards across all B mini-groups.
- A_b^{dir}: Dirichlet-level advantage for the b-th mini-group.
- $\mathcal{J}_{\text{sample}}$: sample-level GRPO objective for optimizing sampled modes and selected predictive distributions.
- $\mathcal{J}_{\text{dir}}$: Dirichlet-level GRPO objective for optimizing mode-selection probability distribution.
- $\mathcal{J}_{\text{DGF}}$: final DGF training objective combining sample-level optimization, Dirichlet-level optimization, and distribution-level diversity regularization.
- γ: weight controlling the strength of the Dirichlet-level objective.
- ϵ: small constant used for numerical stability in advantage normalization.
- $I(\cdot;\cdot)$: mutual information.
- $H(M \mid X)$: conditional entropy of the latent mode given the historical input.
- $\ell(\cdot,\cdot)$: generic forecasting loss function.
- θ: learnable parameters of DGF.

D Proofs

D.1 Proof of Theorem 4.1

Proof D.1 *Since the dynamically valid trajectory set $\mathcal{S}$ is non-convex, by definition there exist two trajectories $Y_a, Y_b \in \mathcal{S}$ and a coefficient $\lambda \in (0,1)$ such that*

$$\bar{Y} = \lambda Y_a + (1-\lambda) Y_b \notin \mathcal{S}.$$

We construct a conditional distribution supported only on these two dynamically valid trajectories:

$$p(Y \mid X = x) = \lambda \delta(Y - Y_a) + (1-\lambda)\delta(Y - Y_b),$$

where $\delta(\cdot)$ denotes the Dirac delta distribution. Since $Y_a, Y_b \in \mathcal{S}$, any trajectory sampled from this conditional distribution belongs to $\mathcal{S}$ almost surely:

$$Y \sim p(Y \mid X = x) \Rightarrow Y \in \mathcal{S} \quad \textit{almost surely}.$$

Under squared loss, the Bayes-optimal point predictor is

$$f^\star(x) = \arg\min_{f(x)} \mathbb{E}\left[\|f(x) - Y\|_2^2 \mid X = x\right].$$

For any candidate prediction $a \in \mathcal{Y}$, the conditional risk is

$$\mathcal{R}(a) = \mathbb{E}\left[\|a - Y\|_2^2 \mid X = x\right].$$

Expanding the squared norm gives

$$\mathcal{R}(a) = \|a\|_2^2 - 2a^\top \mathbb{E}[Y \mid X = x] + \mathbb{E}[\|Y\|_2^2 \mid X = x].$$

The last term does not depend on a. Taking the gradient with respect to a and setting it to zero yields

$$2a - 2\mathbb{E}[Y \mid X = x] = 0,$$

so the unique minimizer is

$$f^\star(x) = \mathbb{E}[Y \mid X = x].$$

For the constructed conditional distribution,

$$\mathbb{E}[Y \mid X = x] = \lambda Y_a + (1 - \lambda)Y_b = \bar{Y}.$$

By construction, $\bar{Y} \notin \mathcal{S}$. Therefore,

$$f^\star(x) \notin \mathcal{S}.$$

Thus, there exists a conditional distribution whose possible futures are all dynamically valid, while the squared-loss optimal point forecast is dynamically invalid.

D.2 Proof of Proposition 4.1

Proof D.2 *For a fixed training sample (X, Y_{gt}), the naive multi-head loss is*

$$\mathcal{L}_{\text{naive}}(X, Y_{gt}) = \sum_{k=1}^{K} \|\mu_k(X) - Y_{gt}\|_2^2.$$

Each term is nonnegative and depends only on the corresponding head output $\mu_k(X)$. The loss is minimized if and only if every term is minimized. For squared Euclidean loss, the unique minimizer of the k-th term is

$$\mu_k(X) = Y_{gt}.$$

Therefore,

$$\mu_1(X) = \mu_2(X) = \cdots = \mu_K(X) = Y_{gt}.$$

For the population risk, consider the conditional risk at a fixed input $X = x$:

$$\mathcal{R}_{\text{naive}}(x) = \sum_{k=1}^{K} \mathbb{E}\left[\|\mu_k(x) - Y\|_2^2 \mid X = x\right].$$

Again, the objective decomposes over heads. For each k, the squared-loss Bayes predictor is

$$\mu_k^\star(x) = \mathbb{E}[Y \mid X = x].$$

Hence,

$$\mu_1^\star(x) = \mu_2^\star(x) = \cdots = \mu_K^\star(x) = \mathbb{E}[Y \mid X = x].$$

This holds for every x where the conditional expectation exists. Therefore, naive multi-head supervision admits a collapsed optimum in which all heads represent the same conditional mean.

E Implementation Details

This section provides dataset descriptions, finetuning baseline configurations, and metric definitions omitted from the main text.

E.1 Dataset details

For long sequence forecasting (LSF), we conduct experiments on six well-established datasets, including the ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, and Electricity. We note that these datasets are not included in the pretraining datasets of the TSFMs we evaluated. The key properties of these LSF datasets are detailed in Table 4. Following Moirai [Woo et al., 2024], we use 5 out-of-distribution datasets for probabilistic forecasting: Electricity, Solar-Power, Jena Weather, Istanbul Traffic[2], and Turkey Power[3]. Detailed descriptions of these datasets are provided in Table 5.

[2]https://www.kaggle.com/datasets/leonardo00/istanbul-traffic-index

[3]https://www.kaggle.com/datasets/dharanikra/electrical-power-demand-in-turkey

E.2 Implementation Details

All experiments are implemented in PyTorch [Paszke et al., 2019] and trained on NVIDIA H100 80GB GPUs. DGF is built on pretrained Moirai-small and Moirai-base [Woo et al., 2024], which are used as warm-start time series backbones. The Dirichlet concentration parameters are obtained by $\alpha_k = \alpha_{min} + \text{softplus}(e_k)$ to ensure positivity. We use the Adam optimizer [Kingma and Ba, 2015], with learning rate selected from $\{10^{-4}, 5 \times 10^{-5}, 10^{-5}\}$ and batch size selected from $\{32, 64, 128\}$ according to validation performance. Training proceeds for up to 200 epochs, with early stopping applied when validation performance does not improve for 15 epochs. The main DGF hyperparameters include the number of mode-conditioned distributions K, the number of Dirichlet mini-groups B, the number of samples per mini-group G, the Dirichlet-level objective weight γ, the dynamical consistency weight λ_2, and the distribution-level diversity weight λ_3. These values are selected on the validation set from $K \in \{4, 8, 16, 32\}$, $B \in \{2, 4, 8, 16\}$, $G \in \{4, 8, 16, 32\}$, $\gamma \in \{0, 0.1, 0.3, 1, 3\}$, $\lambda_2 \in (0, 0.1)$, and $\lambda_3 \in (0, 0.01)$. We fix $\lambda_1 = 1$ for the accuracy reward. Inputs are normalized following the Moirai preprocessing protocol, and predictions are rescaled back to the original scale before evaluation. For deterministic LSF evaluation, DGF reports the MAP-mode output by default; other decision protocols are studied in Appendix F. Each experiment is repeated 5 times with different random seeds.

E.3 Finetuning baselines

Full finetuning involves updating all parameters of the pretrained model. We use a small learning rate for stable adaptation.

LoRA [Hu et al., 2021] introduces trainable rank-decomposition matrices into the attention layers, enabling parameter-efficient finetuning by injecting updates into a low-rank subspace. For Moirai, we use the PEFT library to implement LoRA. We apply LoRA to the query, key, and value projection layers. In addition to the LoRA modules, we also allow the output prediction head to be trainable. The LoRA configuration follows standard settings with rank $r = 16$ and scaling factor $\alpha = 32$.

For Moirai, we implement prompt fine-tuning by introducing trainable soft prompt embeddings, which are prepended to the input tokens in the embedding space. We avoid inserting them into the patch token space, as doing so can interfere with the statistical computation of RevIN and offers less expressive capacity compared to the high-dimensional embedding space. During inference, we discard the prompt embeddings from the encoder output and use only the time series embeddings as final representation for prediction. Similarly, only the output head and the prompt embeddings are finetuned, while all other parameters remain frozen. Prompt length is set to 2 by default.

E.4 Metric details

For long sequence forecasting, we follow the standard protocols to use mean square error (MSE) and mean absolute error for evaluation. For probabilistic forecasting, we include Continuous Ranked Probability Score (CRPS), Mean Scaled Interval Score (MSIS), absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE), mean absolute scaled error (MASE), normalized deviation (ND), and normalized root mean squared error (NRMSE) as metrics. The main text reports CRPS and MSIS, while additional probabilistic metrics are reported in Appendix Table 11. The definitions and calculations of probabilistic forecasting metrics are as follows. Note that the notations used here are independent of those in the main text.

Continuous Ranked Probability Score Given a predicted distribution with c.d.f. F and ground truth $\mathbf{Y}$, the CRPS is defined as:

$$\text{CRPS} = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha), \mathbf{Y}) d\alpha$$
$$\Lambda_\alpha(q, \mathbf{Y}) = (\alpha - \mathbf{1}_{\mathbf{Y}<\text{q}})(\mathbf{Y} - q),$$

where Λ_α is the α-quantile loss, also known as the pinball loss at quantile level α. In practice, the CRPS is intractable or computationally expensive to compute, and we also want to compute a normalized metric, thus we compute a normalized discrete approximation, the mean weighted sum

quantile loss, defined as the average of K quantiles:

$$\text{CRPS} \approx \frac{1}{K}\sum_{k=1}^{K}\text{wQL}[\alpha_k]$$
$$\text{wQL}[\alpha] = 2\frac{\sum_i \Lambda_\alpha(\hat{q}_i(\alpha), \mathbf{Y}_i)}{\sum_i |\mathbf{Y}_i|},$$

where $\mathbf{Y}_i$ is the ground truth at at time step i and $\hat{q}_t(\alpha)$ is the predicted α-quantile at time step i. We take $K = 9, \alpha_1 = 0.1, \alpha_2 = 0.2, \ldots, \alpha_9 = 0.9$ in practice.

Mean Scaled Interval Score The MSIS is a metric to evaluate uncertainty around point forecasts. Given an upper bound prediction, U_i, and lower bound prediction L_i, the MSIS is defined as:

$$\text{MSIS} = \frac{1}{H \cdot \left(\frac{1}{n-m}\sum_{i=m+1}^{n}|\mathbf{Y}_i - \mathbf{Y}_{i-m}|\right)} \cdot \left[\sum_{i=1}^{H}(U_i - L_i) + \frac{2}{a}(L_i - \mathbf{Y}_i)\mathbb{1}_{\{\mathbf{Y}_i < L_i\}} + \frac{2}{a}(\mathbf{Y}_i - U_i)\mathbb{1}_{\{\mathbf{Y}_i > U_i\}}\right]$$

where $a = 0.05$ is the significance level for a 95% prediction interval, over a forecast horizon of length H, and m is the seasonal factor.

symmetric Mean Absolute Percentage Error The sMAPE is an accuracy measure based on percentage errors, treating over- and under-predictions symmetrically, commonly used in forecasting.

$$\text{SMAPE} = \frac{200}{H}\sum_{i=1}^{H}\frac{|\mathbf{Y}_i - \widehat{\mathbf{Y}}_i|}{|\mathbf{Y}_i| + |\widehat{\mathbf{Y}}_i|},$$

Mean Absolute Scaled Error The MASE is a metric for forecasting accuracy, scaling errors by the in-sample mean absolute error of a naive forecast, ensuring interpretability and comparability.

$$\text{MASE} = \frac{1}{H}\sum_{i=1}^{H}\frac{|\mathbf{Y}_i - \widehat{\mathbf{Y}}_i|}{\frac{1}{H-s}\sum_{j=s+1}^{H}|\mathbf{Y}_j - \mathbf{Y}_{j-s}|},$$

where s is the periodicity of the data. $\mathbf{Y}, \widehat{\mathbf{Y}} \in \mathbb{R}^{H\times D}$ are the ground truth and prediction results of the future with H time points and D dimensions. $\mathbf{Y}_i$ means the i-th future time point.

Normalized Deviation The ND measures prediction accuracy by standardizing deviations between predicted and actual values, aiding model evaluation and optimization.

$$ND = \frac{\sum_i |Y_i - \hat{Y}_i|}{\sum_i |Y_i|},$$

Normalized Root Mean Squared Error The NRMSE quantifies prediction error, enables model comparison, aids optimization, and provides interpretable results in time series forecasting.

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(\mathbf{Y}_i - \hat{\mathbf{Y}}_i\right)^2}}{\max(\mathbf{Y}) - \min(\mathbf{Y})}.$$

E.5 Computational Cost Analysis

We further analyze the computational cost of DGF to examine whether the proposed mode-preserving forecasting mechanism introduces additional overhead. All cost benchmarks are implemented in PyTorch [Paszke et al., 2019] and run with MOIRAI-1.0-small as the backbone. We use context length 96, prediction horizon 24, and `num_samples=8`. All results are obtained with seed 1. We

Table 4: Summary of datasets used in the long sequence forecasting evaluation.

Task	Dataset	Variate	Dataset Size	Predict Length	Frequency	Information
Long Sequence Forecasting	ETTh1	7	17420	{96, 192, 336, 720}	Hourly	Temperature
	ETTh2	7	17420	{96, 192, 336, 720}	Hourly	Temperature
	ETTm1	7	69680	{96, 192, 336, 720}	15 min	Temperature
	ETTm2	7	69680	{96, 192, 336, 720}	15 min	Temperature
	Electricity	321	26304	{96, 192, 336, 720}	Hourly	Electricity
	Weather	21	52696	{96, 192, 336, 720}	10 min	Weather

Table 5: Summary of datasets used in the probabilistic forecasting evaluation setting.

Task	Dataset	Variate	Dataset Size	Predict Length	Rolling Evaluation	Frequency	Information
Probabilistic Forecasting	Electricity	321	26304	24	7	H	Energy
	Solar	137	8760	24	7	H	Energy
	Weather	21	52696	144	7	10T	Climate
	Istanbul Traffic	3	14244	24	7	H	Transport
	Turkey Power	18	26304	24	7	H	Energy

compare three settings: zero-shot Moirai, supervised full finetuning, and DGF full finetuning. For both supervised full finetuning and DGF, the trainable ratio is 1.0, i.e., the backbone is not frozen.

The timing protocol is designed to isolate the computational cost of training and inference. The reported training wall-clock time measures only the optimizer-update stage, including forward propagation, loss computation, backward propagation, and optimizer step; model loading, data loading, and initialization are excluded. Inference time excludes model loading and is measured after two warm-up batches. CUDA synchronization is applied before and after timing to reduce measurement noise. Table 6 shows the accuracy-cost trade-off on ETTh1, ETTm1, and Weather, including training time, inference time, and GPU memory usage. The average MSE values are taken from the main forecasting experiments, whereas the cost measurements use context length 96, prediction horizon 24, and eight samples. Under this configuration, DGF achieves lower average forecasting MSE and comparable or lower measured computation time than supervised full finetuning. These timing results characterize the specified benchmark and do not estimate the total cost of the full experimental campaign.

F Additional Ablation Study

Component Analysis We conduct ablation studies to examine the contribution of each major component in DGF. The variants remove or replace the components corresponding to mode-conditioned representation, Dirichlet-guided mode selection, and reward-based optimization. As shown in Table 7, removing any major component degrades performance. The drops from *w/o multiple heads* and *softmax routing* confirm the importance of explicitly representing multiple latent modes and modeling uncertainty over their selection probabilities. Removing the dynamics reward or diversity regularization weakens performance, indicating that dynamic feedback and mode-level separation are both necessary for mode-preserving forecasting. Finally, removing the Dirichlet-level advantage leads to degradation, showing that group-level credit assignment is important for learning effective mode-composition hypotheses.

Forecast decision protocols. DGF defines a stochastic forecasting policy, while LSF benchmarks require a single reported trajectory. We compare four protocols: *Random sample*, which evaluates one trajectory sampled from the full DGF process; *MAP-mode output*, which reports the mean trajectory of the mode with the largest expected Dirichlet probability; *Sample-and-rerank*, which selects from sampled candidates using a history-to-future dynamical consistency score without ground-truth access; and *Best-of-N*, an oracle protocol that selects the candidate with the lowest ground-truth error. MAP-mode is used in the main table, while Best-of-N is reported only as a mode-coverage upper bound.

Table 6: Accuracy-cost trade-off with MOIRAI-1.0-small. We report average MSE from the main forecasting results and computational cost from the seed-1 cost benchmark with context length 96, prediction horizon 24, and `num_samples=8`. Training time excludes model loading, data loading, and initialization, and measures only optimizer-update computation. Inference time excludes model loading and is measured after two warm-up batches. Lower values are better.

Dataset	Method	Avg. MSE	Train Total (s)	Infer Total (s)	Train Mem. (GB)	Infer Mem. (GB)
ETTh1	Moirai-small zero-shot	0.416	–	1.0763	–	0.0852
	Supervised full finetuning	0.415	10.8496	1.1620	0.3236	0.1694
	DGF full finetuning	0.399	9.4667	1.0560	0.2943	0.1620
ETTm1	Moirai-small zero-shot	0.448	–	1.0516	–	0.0852
	Supervised full finetuning	0.367	10.5942	1.0754	0.3236	0.1694
	DGF full finetuning	0.338	9.3407	1.0467	0.2943	0.1620
Weather	Moirai-small zero-shot	0.268	–	1.0551	–	0.0862
	Supervised full finetuning	0.228	10.6123	1.0946	0.3237	0.1704
	DGF full finetuning	0.207	8.8507	1.0629	0.2943	0.1631

Table 7: Component ablation of DGF using Moirai-base as the backbone. The ablation is conducted on representative LSF datasets with four prediction lengths.

Dataset	Len.	Metric	DGF	w/o multiple heads	softmax routing	w/o dynamics reward	w/o diversity regularization	w/o Dirichlet-level advantage	supervised adaptation only
ETTm1	96	MSE	0.265	0.299	0.315	0.289	0.278	0.300	0.312
		MAE	0.322	0.330	0.336	0.326	0.326	0.330	0.334
	192	MSE	0.302	0.339	0.356	0.324	0.317	0.344	0.355
		MAE	0.347	0.352	0.362	0.351	0.351	0.360	0.361
	336	MSE	0.330	0.370	0.379	0.356	0.342	0.373	0.380
		MAE	0.365	0.378	0.383	0.372	0.369	0.375	0.380
	720	MSE	0.372	0.410	0.423	0.401	0.389	0.416	0.426
		MAE	0.392	0.406	0.408	0.400	0.399	0.405	0.409
ETTm2	96	MSE	0.152	0.170	0.177	0.162	0.160	0.175	0.176
		MAE	0.229	0.242	0.254	0.240	0.240	0.245	0.249
	192	MSE	0.202	0.224	0.228	0.215	0.211	0.225	0.230
		MAE	0.273	0.283	0.285	0.278	0.277	0.283	0.288
	336	MSE	0.251	0.273	0.280	0.264	0.264	0.274	0.282
		MAE	0.308	0.319	0.328	0.320	0.312	0.322	0.325
	720	MSE	0.330	0.341	0.343	0.337	0.336	0.340	0.344
		MAE	0.352	0.363	0.371	0.360	0.356	0.365	0.367
Electricity	96	MSE	0.127	0.140	0.138	0.135	0.133	0.137	0.144
		MAE	0.219	0.230	0.239	0.226	0.223	0.234	0.236
	192	MSE	0.142	0.160	0.169	0.155	0.151	0.159	0.166
		MAE	0.239	0.249	0.252	0.247	0.244	0.256	0.256
	336	MSE	0.165	0.175	0.176	0.169	0.169	0.173	0.176
		MAE	0.251	0.263	0.271	0.258	0.259	0.266	0.267
	720	MSE	0.192	0.203	0.207	0.197	0.198	0.201	0.207
		MAE	0.274	0.290	0.298	0.284	0.280	0.292	0.295

F.1 Parameter Sensitivity

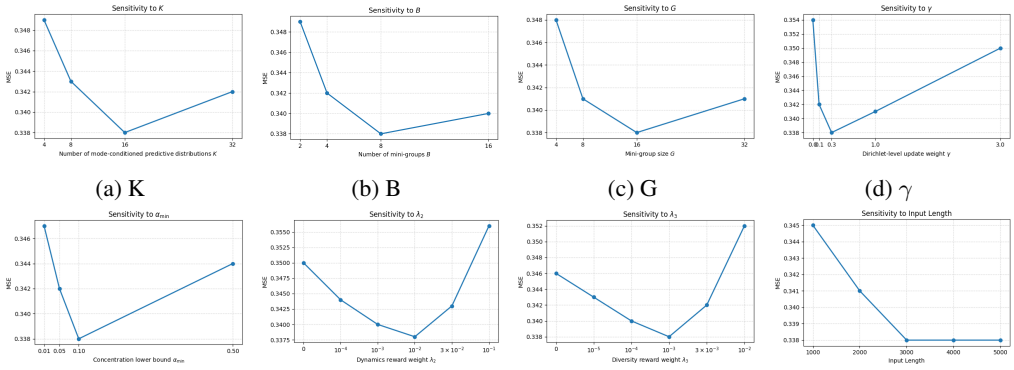
We analyze the sensitivity of DGF to major hyperparameters. All experiments use Moirai-small on ETTh1 with prediction length 96. We vary one hyperparameter at a time and report MSE in Figure 4. Overall, DGF remains stable across a reasonable range of hyperparameter values.

G Full results

We provide the full MSE and MAE results in Tables 9 and 10, and the full probabilistic forecasting results in Table 11.

Table 8: Ablation on deterministic forecast decision protocols for LSF using Moirai-base as the backbone. MAP-mode output is the reported protocol in the main table. Best-of-N is reported only as an oracle coverage indicator and is not used as the default deterministic protocol.

Dataset	Len.	Metric	MAP-mode output	Random sample	Sample-and-rerank	Best-of-N oracle
ETTm1	96	MSE	0.265	0.278	0.258	0.238
		MAE	0.322	0.331	0.317	0.303
	192	MSE	0.302	0.319	0.294	0.271
		MAE	0.347	0.358	0.341	0.326
	336	MSE	0.330	0.347	0.321	0.297
		MAE	0.365	0.376	0.359	0.342
	720	MSE	0.372	0.391	0.362	0.336
		MAE	0.392	0.407	0.385	0.368
ETTm2	96	MSE	0.152	0.160	0.148	0.136
		MAE	0.229	0.236	0.225	0.216
	192	MSE	0.202	0.214	0.197	0.181
		MAE	0.273	0.282	0.269	0.258
	336	MSE	0.251	0.265	0.244	0.225
		MAE	0.308	0.318	0.303	0.291
	720	MSE	0.330	0.347	0.322	0.298
		MAE	0.352	0.364	0.347	0.333
Electricity	96	MSE	0.127	0.134	0.124	0.114
		MAE	0.219	0.226	0.216	0.207
	192	MSE	0.142	0.151	0.138	0.127
		MAE	0.239	0.247	0.236	0.226
	336	MSE	0.165	0.175	0.160	0.147
		MAE	0.251	0.259	0.247	0.237
	720	MSE	0.192	0.204	0.186	0.171
		MAE	0.274	0.284	0.269	0.257



(a) K (b) B (c) G (d) γ

(e) $\alpha_{\min}$ (f) λ_2 (g) λ_3 (h) Input Length

Figure 4: Parameter Sensitivity Analysis.

Table 9: Full MSE results of long sequence forecasting experiments.

		Baselines					Moirai-small					Moirai-base					DGF	
Dataset	Len.	DLinear [2023]	PatchTST [2023]	iTransformer [2024b]	TimeMixer [2024]	SimpleTM [2025]	Moirai-small	+ *Full finetuning*	+ *Prompt tuning*	+ *LoRA*	+ *MSFT* [2026]	Moirai-base	+ *Full finetuning*	+ *Prompt tuning*	+ *LoRA*	+ *MSFT* [2026]	DGF with Moirai-small	DGF with Moirai-base
ETTm1	96	0.345	0.329	0.334	0.320	0.321	0.404	0.303	0.335	0.302	0.295	0.335	0.312	0.330	0.311	0.284	0.280	**0.265**
	192	0.380	0.367	0.377	0.361	0.360	0.435	0.352	0.368	0.357	0.338	0.366	0.355	0.363	0.345	0.317	0.321	**0.302**
	336	0.413	0.399	0.426	0.390	0.390	0.462	0.388	0.405	0.389	0.371	0.391	0.380	0.389	0.373	0.343	0.358	**0.330**
	720	0.474	0.454	0.491	0.454	0.454	0.490	0.425	0.428	0.431	0.409	0.434	0.426	0.431	0.414	0.382	0.392	**0.372**
ETTm2	96	0.193	0.175	0.180	0.175	0.173	0.205	0.179	0.197	0.179	0.165	0.197	0.176	0.197	0.177	0.166	0.154	**0.152**
	192	0.284	0.241	0.250	0.237	0.238	0.261	0.234	0.252	0.234	0.218	0.250	0.230	0.247	0.230	0.217	0.205	**0.202**
	336	0.369	0.305	0.311	0.298	0.296	0.319	0.291	0.304	0.288	0.267	0.301	0.282	0.300	0.280	0.265	0.257	**0.251**
	720	0.554	0.402	0.412	0.391	0.393	0.415	0.388	0.413	0.387	0.349	0.375	0.344	0.374	0.347	0.339	0.338	**0.330**
ETTh1	96	0.386	0.414	0.386	0.375	0.366	0.387	0.382	0.384	0.382	0.380	0.375	0.372	0.375	0.373	0.372	0.364	**0.361**
	192	0.437	0.460	0.441	0.429	0.422	0.418	0.419	0.415	0.418	0.416	0.406	0.404	0.406	0.404	0.404	0.401	**0.391**
	336	0.481	0.501	0.487	0.484	0.440	0.431	0.434	0.429	0.431	0.428	0.426	0.423	0.425	0.423	0.422	0.419	**0.414**
	720	0.519	0.500	0.503	0.498	0.463	0.427	0.426	0.427	0.426	0.423	0.440	0.434	0.440	0.434	0.429	**0.413**	0.416
ETTh2	96	0.333	0.302	0.297	0.289	0.281	0.287	0.286	0.286	0.286	0.279	0.284	0.283	0.284	0.284	0.280	**0.274**	**0.274**
	192	0.477	0.388	0.380	0.372	0.355	0.350	0.349	0.349	0.349	0.347	0.350	0.355	0.354	0.351	0.350	**0.336**	0.339
	336	0.594	0.426	0.428	0.386	0.365	0.378	0.376	0.378	0.377	0.376	0.378	0.387	0.392	0.379	0.379	**0.360**	0.367
	720	0.831	0.431	0.427	0.412	0.413	0.403	0.396	0.403	0.402	0.392	0.412	0.403	0.411	0.411	0.400	**0.378**	0.380
Electricity	96	0.197	0.195	0.148	0.153	0.141	0.205	0.154	0.191	0.152	0.150	0.158	0.144	0.155	0.142	0.139	0.141	**0.127**
	192	0.196	0.199	0.162	0.166	0.151	0.220	0.172	0.205	0.176	0.172	0.174	0.166	0.168	0.160	0.159	0.158	**0.142**
	336	0.209	0.215	0.178	0.185	0.173	0.236	0.203	0.219	0.197	0.193	0.191	0.176	0.185	0.178	0.176	0.185	**0.165**
	720	0.245	0.256	0.225	0.225	0.201	0.270	0.242	0.252	0.243	0.234	0.229	0.207	0.226	0.210	0.203	0.224	**0.192**
Weather	96	0.196	0.177	0.174	0.163	0.162	0.183	0.154	0.163	0.153	0.147	0.163	0.152	0.159	0.151	0.144	0.142	**0.140**
	192	0.237	0.225	0.221	0.208	0.208	0.229	0.200	0.207	0.197	0.189	0.207	0.198	0.199	0.198	0.184	0.181	**0.171**
	336	0.283	0.278	0.278	0.251	0.263	0.288	0.246	0.254	0.243	0.234	0.264	0.250	0.248	0.249	0.229	0.225	**0.220**
	720	0.345	0.354	0.358	0.339	0.340	0.371	0.311	0.315	0.305	0.292	0.350	0.326	0.314	0.322	0.296	**0.280**	0.282

Table 10: Full MAE results of long sequence forecasting experiments.

		Baselines					Moirai-small					Moirai-base					DGF	
Dataset	Len.	DLinear [2023]	PatchTST [2023]	iTransformer [2024b]	TimeMixer [2024]	SimpleTM [2025]	Moirai-small	+ *Full finetuning*	+ *Prompt tuning*	+ *LoRA*	+ *MSFT* [2026]	Moirai-base	+ *Full finetuning*	+ *Prompt tuning*	+ *LoRA*	+ *MSFT* [2026]	DGF with Moirai-small	DGF with Moirai-base
ETTm1	96	0.372	0.367	0.368	0.357	0.361	0.383	0.345	0.359	0.344	0.341	0.360	0.334	0.355	0.337	0.335	0.330	**0.322**
	192	0.389	0.385	0.391	0.381	0.380	0.402	0.372	0.380	0.374	0.367	0.379	0.361	0.377	0.359	0.359	0.358	**0.347**
	336	0.413	0.410	0.420	0.404	0.404	0.416	0.393	0.403	0.394	0.387	0.394	0.380	0.393	0.379	0.378	0.379	**0.365**
	720	0.453	0.439	0.459	0.441	0.438	0.437	0.419	0.423	0.421	0.414	0.419	0.409	0.417	0.407	0.404	0.410	**0.392**
ETTm2	96	0.292	0.259	0.264	0.258	0.257	0.282	0.251	0.273	0.250	0.242	0.271	0.249	0.271	0.248	0.246	0.234	**0.229**
	192	0.362	0.302	0.309	0.299	0.299	0.318	0.292	0.309	0.290	0.281	0.306	0.288	0.301	0.289	0.285	**0.271**	0.273
	336	0.427	0.343	0.348	0.340	0.338	0.355	0.329	0.343	0.327	0.314	0.339	0.325	0.339	0.327	0.320	**0.301**	0.308
	720	0.522	0.400	0.407	0.396	0.395	0.410	0.390	0.409	0.390	0.368	0.388	0.367	0.387	0.366	0.369	0.361	**0.352**
ETTh1	96	0.400	0.419	0.405	0.400	0.392	0.402	0.400	0.402	0.399	0.401	0.398	0.396	0.397	0.397	0.395	0.399	**0.384**
	192	0.432	0.445	0.436	0.421	0.421	0.423	0.423	0.423	0.423	0.421	0.417	0.416	0.416	0.416	0.415	0.415	**0.401**
	336	0.459	0.466	0.458	0.458	0.438	0.435	0.438	0.435	0.435	0.433	0.429	0.429	0.428	0.429	0.429	0.426	**0.418**
	720	0.516	0.488	0.491	0.482	0.462	0.450	0.453	0.451	0.450	0.449	0.452	0.454	0.452	0.451	0.450	0.440	**0.437**
ETTh2	96	0.387	0.348	0.349	0.341	0.338	0.334	0.332	0.337	0.333	0.326	0.334	0.330	0.335	0.334	0.327	0.325	**0.320**
	192	0.476	0.400	0.400	0.392	0.387	0.374	0.372	0.373	0.373	0.369	0.380	0.378	0.378	0.378	0.374	**0.357**	0.362
	336	0.541	0.433	0.432	0.414	0.401	0.395	0.392	0.394	0.394	0.391	0.405	0.402	0.404	0.405	0.404	**0.389**	0.398
	720	0.657	0.446	0.445	0.434	0.436	0.421	0.416	0.420	0.420	0.413	0.432	0.429	0.432	0.431	0.427	**0.401**	0.415
Electricity	96	0.282	0.285	0.240	0.247	0.235	0.299	0.242	0.285	0.244	0.241	0.248	0.236	0.246	0.234	0.230	0.232	**0.219**
	192	0.285	0.289	0.253	0.256	0.247	0.310	0.265	0.294	0.266	0.262	0.263	0.256	0.258	0.252	0.252	0.252	**0.239**
	336	0.301	0.305	0.269	0.277	0.267	0.323	0.289	0.307	0.285	0.282	0.278	0.267	0.274	0.269	0.266	0.268	**0.251**
	720	0.333	0.337	0.317	0.310	0.293	0.347	0.319	0.331	0.321	0.316	0.307	0.295	0.305	0.296	0.293	0.307	**0.274**
Weather	96	0.255	0.218	0.214	0.209	0.207	0.216	0.189	0.199	0.189	0.185	0.198	0.186	0.196	0.186	0.182	0.177	**0.167**
	192	0.296	0.260	0.254	0.250	0.248	0.258	0.236	0.241	0.233	0.229	0.240	0.235	0.235	0.235	0.224	0.215	**0.214**
	336	0.335	0.297	0.296	0.287	0.290	0.297	0.272	0.276	0.271	0.266	0.282	0.278	0.272	0.278	0.261	0.253	**0.251**
	720	0.381	0.348	0.349	0.341	0.341	0.346	0.317	0.317	0.315	0.311	0.338	0.333	0.318	0.342	0.311	0.297	**0.295**

Table 11: Full results for probabilistic forecasting experiments.

		Baselines			Moirai-small					Moirai-base					DGF	
Dataset	Metric	DeepAR [2020]	PatchTST [2023]	TiDE [2023]	Moirai-small	+ *Full finetuning*	+ *Prompt tuning*	+ *LoRA*	+ *MSFT* [2026]	Moirai-base	+ *Full finetuning*	+ *Prompt tuning*	+ *LoRA*	+ *MSFT* [2026]	DGF with Moirai-small	DGF with Moirai-base
Electricity	sMAPE	0.118	0.107	0.102	0.134	0.112	0.125	0.123	0.095	0.111	0.100	0.109	0.103	0.094	0.087	**0.081**
	MASE	0.844	0.753	0.706	0.981	0.810	0.887	0.872	0.664	0.792	0.716	0.783	0.746	0.653	0.650	**0.641**
	ND	0.080	0.065	0.061	0.092	0.070	0.084	0.079	0.059	0.069	0.063	0.069	0.064	0.058	0.054	**0.051**
	NRMSE	0.704	0.506	0.514	0.840	1.260	0.698	0.650	0.478	0.551	0.517	0.583	0.508	0.471	0.469	**0.459**
Solar	sMAPE	1.385	1.501	1.400	1.445	1.400	1.413	1.391	1.381	1.410	1.282	1.407	1.387	**1.264**	1.354	1.296
	MASE	1.222	1.607	1.265	1.465	1.181	1.331	1.160	1.113	1.292	0.552	1.285	1.184	**0.422**	1.089	0.459
	ND	0.520	0.685	0.538	0.624	0.504	0.567	0.495	0.475	0.551	0.239	0.548	0.505	**0.184**	0.468	0.201
	NRMSE	1.033	1.408	1.093	1.135	1.000	1.081	0.953	0.949	1.034	0.554	1.053	0.967	0.452	0.932	**0.447**
Weather	sMAPE	0.776	0.668	0.636	0.686	0.612	0.685	0.617	0.605	0.623	0.626	0.613	0.610	0.622	0.609	**0.591**
	MASE	3.170	0.844	0.832	0.521	0.466	0.520	0.472	0.451	0.487	0.511	0.484	0.465	0.474	**0.449**	0.465
	ND	0.163	0.072	0.066	0.063	0.043	0.063	0.043	0.043	0.048	0.045	0.046	0.043	0.044	0.042	**0.040**
	NRMSE	0.486	0.260	0.214	0.229	0.200	0.232	0.197	0.198	0.417	2.980	0.659	0.717	0.636	**0.192**	0.398
Istanbul Traffic	sMAPE	**0.249**	0.287	0.280	0.359	0.319	0.302	0.327	0.295	0.284	0.251	0.288	0.263	0.289	0.289	0.281
	MASE	0.613	0.653	0.618	0.990	0.827	0.815	0.907	0.815	0.644	0.620	0.573	0.621	0.568	0.801	**0.542**
	ND	0.139	0.148	0.140	0.224	0.188	0.185	0.206	0.182	0.146	0.140	0.130	0.141	0.129	0.179	**0.121**
	NRMSE	0.181	0.190	0.185	0.294	0.298	0.284	0.282	0.252	0.194	0.251	0.170	0.219	0.160	0.247	**0.149**
Turkey Power	sMAPE	0.404	0.416	0.389	0.389	0.378	0.387	0.382	0.377	0.378	0.372	0.377	0.371	0.372	0.373	**0.367**
	MASE	1.395	1.234	0.904	0.948	0.863	0.947	0.887	0.864	0.888	0.816	0.866	0.825	0.814	0.866	**0.810**
	ND	0.083	0.071	0.059	0.061	0.048	0.058	0.055	0.051	0.051	0.045	0.052	0.045	0.045	0.053	**0.042**
	NRMSE	0.181	0.158	0.139	0.149	0.124	0.142	0.131	0.122	0.118	0.101	0.120	0.101	0.099	0.125	**0.093**

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 abstract and introduction state the main claims of the paper: we reinterpret over-smoothing in time series forecasting as non-convex latent dynamical mode compression under single-realization supervision, and propose DGF to preserve mode-conditioned futures through Dirichlet-guided sampling and reward-based optimization. The experimental sections evaluate forecasting accuracy, probabilistic forecast quality, over-smoothing mitigation, and component effectiveness.

   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 have discussed limitations in Appendix A.

   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: We have provided the detailed proofs and analyses in Appendix D.

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 provide benchmark descriptions, implementation details, and evaluation protocols in Section 6 and Appendix E, with additional ablations and full results in Appendices F and G.

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: The implementation is available at `https://github.com/Zh-XY22/DGF`. Experimental settings and evaluation protocols are described in Appendices E and F.

   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 specify datasets, evaluation protocols, finetuning baselines, optimizer, learning-rate search space, batch-size search space, DGF hyperparameters, and deterministic decision protocols in Section 6, Appendix E, and Appendix F.

   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: [Yes]

   Justification: We repeat the main experiments with multiple random seeds and report mean and standard deviation for the over-smoothing analysis in Table 3.

   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 discuss in Appendix E.5.

   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: This work conforms, in every respect, 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 have discussed this in Appendix A.

    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: The paper does not release a high-risk generative model, scraped dataset, or model intended for dual-use applications. The method is evaluated on public time series forecasting benchmarks.

    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 have cited all used data and provided the licenses properly.

    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: We provide documentation for the released implementation, including training scripts, evaluation scripts, configuration files, and instructions for reproducing the main experiments.

    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: This work does not involve crowdsourcing, user studies, or human-subject research.

    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: This work does not involve crowdsourcing, user studies, or human-subject research, so IRB approval is not applicable.

    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: [N/A]

    Justification: We only use LLM for editing, which does not involve LLMs as any important, original, or non-standard components.

    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.