TimePre: Bridging Accuracy, Efficiency, and Stability in Probabilistic Time-Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

We propose **TimePre**, a simple framework that unifies the efficiency of Multilayer Perceptron (MLP)-based models with the distributional flexibility of Multiple Choice Learning (MCL) for Probabilistic Time-Series Forecasting (PTSF). **Stabilized Instance Normalization (SIN)**, the core of TimePre, is a normalization layer that explicitly addresses the trade-off among accuracy, efficiency, and stability. SIN stabilizes the hybrid architecture by correcting channel-wise statistical shifts, thereby resolving the catastrophic hypothesis collapse. Extensive experiments on six benchmark datasets demonstrate that TimePre achieves the best Distortion on all six benchmarks and the best or highly competitive CRPS-Sum on most benchmarks. Critically, TimePre achieves inference speeds that are orders of magnitude faster than sampling-based models, and is more stable than prior MCL approaches.

1 Introduction

Probabilistic time-series forecasting (PTSF) Kim et al. (2025) aims to model the conditional distribution of future trajectories given historical observations. It is important in a wide range of applications, including weather prediction Wu et al. (2023), energy management Rajagukguk et al. (2020), and finance. Recent advances have introduced several generative paradigms for PTSF LeCun et al. (2015); Hu et al. (2025); Wang et al. (2019), including diffusion-based models such as TimeGrad Rasul et al. (2021), flow-based models such as TempFlow Rasul et al. (2020), and copula-based approaches such as TACTiS-2 Ashok et al. (2024).

However, these approaches typically rely on costly multi-step sampling to represent uncertainty, which limits their efficiency and practical scalability. To address this issue, TimeMCL Cortes et al. (2025); Perera et al. (2024a) introduced a non-sampling formulation for PTSF based on Multiple Choice Learning (MCL), modeling uncertainty with a finite set of discrete hypotheses within an autoregressive RNN D YAMADA et al. (2021); Lyu et al. (2021). In this framework, each prediction head is trained under a winner-takes-all (WTA) objective, where only the hypothesis with the smallest loss for each sample receives gradient updates. While this competitive mechanism encourages specialization among hypotheses, it also induces highly uneven gradient allocation Rodriguez Domínguez et al. (2025), often resulting in limited hypothesis diversity and unstable optimization.

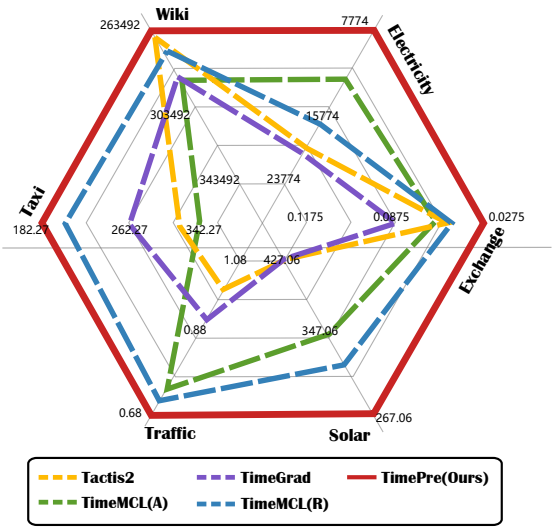

Figure 1: Model performance comparison on the Distortion metric across six real-world benchmark datasets. Lower is better.

At the same time, recent progress in long-term time-series forecasting (LTSF) has shown that lightweight architectures, such as linear models and multilayer perceptron (MLP)-based backbones Rumelhart et al. (1986), can out-

perform more complex Transformer-based methods Nie
et al. (2023); Zhou et al. (2021) in both accuracy and efficiency. Representative examples include DLinear Zeng et al. (2023), TiDE Das et al. (2023), and TimeMixer Wang et al. (2024). These results suggest that simple forecasting backbones can be highly effective when properly designed.

Motivated by this trend, a natural question is whether the efficiency of lightweight forecasting backbones can be combined with the distributional flexibility of the MCL paradigm. Our investigation reveals a fundamental incompatibility between the two. When MCL is directly applied to lightweight architectures, training becomes unstable and hypotheses quickly collapse. In particular, linear projections lack the implicit regularization and manifold constraints provided by nonlinear encoders such as LSTMs Hochreiter & Schmidhuber (1997); Neyshabur et al. (2017); Zhang et al. (2021; 2025; 2018). As a result, scale disparities in real-world data are directly exposed and can be rapidly amplified during optimization. Under the competitive WTA objective, only a small subset of scale-aligned hypotheses tends to receive consistent gradient updates, while the others stagnate, eventually leading to irreversible hypothesis collapse.

To address this issue, we propose **TimePre**, a probabilistic forecasting framework built on Stabilized Instance Normalization (SIN) and a direct multi-hypothesis predictor. SIN performs adaptive channel-wise rescaling before the input reaches the lightweight encoder, mitigating statistical shifts Quionero-Candela et al. (2009) and improving optimization stability under the WTA objective. This design preserves hypothesis diversity while maintaining the efficiency advantages of direct forecasting. As shown in Figure 1, TimePre achieves strong forecasting performance across six benchmark datasets while being substantially faster than existing sampling-based methods.

Our main contributions are summarized as follows:

- We identify a fundamental incompatibility between lightweight forecasting backbones and the MCL paradigm, and analyze how this mismatch leads to hypothesis collapse under competitive training.
- We propose TimePre, which combines the efficiency of lightweight forecasting models with the distributional modeling capability of MCL. Its core component, Stabilized Instance Normalization, improves optimization stability and supports balanced competition among hypotheses.
- Extensive experiments on six benchmark datasets show that TimePre achieves state-of-the-art or highly competitive performance across probabilistic forecasting metrics, while offering substantially faster inference than sampling-based baselines.

2 Related Work

2.1 Multiple Choice Learning and Multi-Hypothesis Forecasting

Multiple Choice Learning (MCL) provides a principled framework for modeling diverse outcomes under uncertainty. It was originally introduced as an assignment-based multi-model training scheme by Guzmán-Rivera et al. Guzman-Rivera et al. (2012), and later reformulated as a differentiable winner-takes-all (WTA) objective for multi-head neural networks Rupprecht et al. (2017). Under this formulation, different prediction heads specialize in different modes of the target distribution. From the perspective of vector quantization, MCL can also be interpreted as learning a finite set of representative codevectors that approximate a conditional distribution Gersho & Gray (1992); Loubes & Pelletier (2017); Letzelter et al. (2024). This view provides a useful theoretical basis for multi-hypothesis prediction.

A central challenge in MCL is the instability induced by hard competition among hypotheses. Because only the winning head receives dominant gradient updates, optimization can be sensitive to initialization and prone to mode collapse or poor local minima. Several studies have therefore proposed relaxed or stabilized variants of the WTA objective. For example, annealed or stochastic formulations smooth the optimization landscape and improve diversity during training Lee et al. (2016); Perera et al. (2024a), while more recent extensions introduce learned scoring or resilient assignment mechanisms to better handle ambiguous outputs and unknown numbers of valid modes Perera et al. (2024b).

In probabilistic time-series forecasting, TimeMCL adapts the MCL framework to generate a discrete set of plausible future trajectories Cortes et al. (2025). Compared with sampling-based probabilistic forecasters, this line of work offers an attractive trade-off among forecast quality, diversity, and inference efficiency. However, existing MCL-based forecasting methods mainly rely on autoregressive or heavier sequential architectures, and the interaction between MCL and lightweight linear forecasting backbones remains underexplored. Our work focuses on this gap and studies how to stabilize multi-hypothesis learning in this setting.

2.2 Time-Series Forecasting Backbones

Time-series forecasting has evolved from traditional statistical and machine learning methods Ho & Xie (1998); Friedman (2000) to deep neural architectures that can model complex temporal dependencies. Early deep forecasting models were largely based on recurrent neural networks such as GRU and LSTM, which process temporal dynamics through sequential hidden states. Representative probabilistic methods such as DeepAR further combine autoregressive recurrent backbones with parametric likelihood modeling, enabling uncertainty estimation in practical forecasting tasks Salinas et al. (2020); Kim et al. (2025). Despite their success, recurrent models often suffer from limited parallelism and reduced efficiency in long-horizon forecasting.

To address these limitations, Transformer-based architectures were later introduced for time-series forecasting. Models such as Informer, Autoformer, and FEDformer use self-attention or frequency-domain modules to capture long-range temporal dependencies, and they established Transformers as a major paradigm in the field Vaswani et al. (2017); Zhou et al. (2021); Wu et al. (2021); Zhou et al. (2022). At the same time, subsequent studies showed that part of their empirical gains can often be attributed to design choices such as normalization, decomposition, scaling, or data preprocessing, rather than attention alone.

This observation motivated a new line of lightweight forecasting models based on linear or MLP-style architectures. Methods such as DLinear, TiDE, and TimeMixer demonstrate that simple backbones can achieve strong performance while being substantially more efficient than recurrent or attention-based models Zeng et al. (2023); Das et al. (2023); Wang et al. (2024). Related studies have also revisited the trade-off between channel-independence and channel-mixing strategies in multivariate forecasting, showing that lightweight architectures can remain competitive when their inductive biases are carefully designed. These advances make lightweight backbones an appealing foundation for probabilistic forecasting.

Nevertheless, most existing lightweight models are designed for deterministic prediction rather than multi-hypothesis probabilistic forecasting. As a result, it remains unclear how to combine their efficiency with the distributional flexibility of MCL without causing instability during training. Our method is motivated by this question and targets the compatibility between lightweight forecasting backbones and multi-hypothesis learning.

3 Approach

3.1 Preliminaries

We consider a multivariate stochastic process $\{x_t\}_{t=1}^{T}$, where $x_t \in \mathbb{R}^D$ denotes a D-dimensional observation at time t. Given a look-back window of length L and a forecast horizon of length H, we define the input–output pair as

$$\mathbf{X}_t = [x_{t-L},\, x_{t-L+1},\, \ldots,\, x_{t-1}]^\top \in \mathbb{R}^{L\times D}, \tag{1}$$

$$\mathbf{Y}_t = [x_t,\, x_{t+1},\, \ldots,\, x_{t+H-1}]^\top \in \mathbb{R}^{H\times D}. \tag{2}$$

The forecasting objective is to learn a mapping $f_\Theta : \mathbb{R}^{L\times D} \to \mathbb{R}^{H\times D}$ that minimizes the conditional risk

$$\min_{\Theta}\ \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}}\big[\ell(f_\Theta(\mathbf{X}),\mathbf{Y})\big], \tag{3}$$

where $\ell(\cdot,\cdot)$ is a task-specific loss induced by the forecasting likelihood and $\mathcal{D}$ denotes the underlying data distribution.

However, the objective in Equation (3) defines a deterministic mapping, whereas real-world temporal processes are often stochastic and may admit multiple plausible futures. A single predictor is therefore insufficient to approximate the full conditional distribution $p(\mathbf{Y} \mid \mathbf{X})$. To model such multi-modality in a tractable way, we adopt the functional quantization formulation Guzman-Rivera et al. (2012), in which a finite set of K hypothesis functions $\{f_\Theta^{(k)}\}_{k=1}^K$ jointly approximates the conditional manifold by minimizing

$$\min_{\{f_\Theta^{(k)}\}} \mathbb{E}_{(\mathbf{X},\mathbf{Y})} \left[\min_{k=1,\dots,K} d\big(f_\Theta^{(k)}(\mathbf{X}), \mathbf{Y}\big) \right], \tag{4}$$

where $d(\cdot,\cdot)$ denotes a trajectory-level discrepancy measure, typically the ℓ_2 distance. Under this view, each $f_\Theta^{(k)}$ serves as a representative centroid of a distinct mode of $p(\mathbf{Y} \mid \mathbf{X})$, providing a bridge between deterministic regression and probabilistic forecasting.

This formulation follows the multi-hypothesis learning paradigm introduced in Multiple Choice Learning Guzman-Rivera et al. (2012); Lee et al. (2016) and later extended to structured prediction Rupprecht et al. (2017). Probabilistic forecasting approaches such as DeepAR Salinas et al. (2020), MQRNN Wen et al. (2017), and Deep Ensembles Lakshminarayanan et al. (2017) are also related in that they approximate $p(\mathbf{Y} \mid \mathbf{X})$ through multiple predictive hypotheses, although they differ substantially in training and inference mechanisms.

3.2 Diagnosis: Instability of Linear Backbones under MCL

Our empirical study reveals a consistent instability when modern linear backbones are combined with the multi-hypothesis formulation in Equation (4). Rather than causing numerical divergence, this combination tends to produce learning stagnation and hypothesis collapse Rupprecht et al. (2017), where only a small subset of heads remains active while the others fail to learn meaningful predictions. We attribute this behavior to the interaction of two factors: inter-variable scale imbalance in real-world multivariate data, and the absence of implicit regularization in linear mappings.

Impact of scale imbalance on the WTA objective. Real-world multivariate time-series, such as Electricity and Wiki, contain variables with substantially different physical units and magnitudes. Standard global normalization, such as dataset-level z-score normalization, is often insufficient in this setting because it does not correct per-instance, per-variable scale imbalance. This is particularly problematic under the winner-takes-all (WTA) objective in Equation (4), which is highly sensitive to relative error scale. A hypothesis $f_\Theta^{(k)}$ that is initialized slightly closer to a high-magnitude variable may repeatedly win the arg min operation, even if it performs poorly on other variables. As a result, gradient updates become concentrated on a small subset of heads, while the remaining hypotheses receive little useful supervision. This imbalance leads to unstable optimization and eventually to hypothesis collapse.

Linear backbones as unconstrained amplifiers. This failure mode becomes more severe when the forecasting backbone is linear. Nonlinear encoders such as LSTMs Hochreiter & Schmidhuber (1997) and Transformers Vaswani et al. (2017) provide a form of implicit regularization by coupling features through shared nonlinear representations. In contrast, linear backbones directly propagate input-scale disparities and initialization bias into the WTA competition. Without a shared manifold to regularize head behavior, the K hypotheses compete in a poorly conditioned feature space, and small initial differences can quickly grow into large functional disparities. This effect makes inactive heads difficult to recover and impedes stable optimization.

Requirements for stabilization. This diagnosis suggests that a stabilization mechanism for linear MCL forecasting should satisfy three requirements:

1. **Per-instance, per-variable operation.** It should correct inter-variable scale imbalance at the channel level, rather than relying on batch-level statistics.
2. **Robustness to non-stationarity and outliers.** It should remain stable under spikes, distribution shift, and heavy-tailed temporal observations.

3. **Analytical reversibility.** It should allow predictions to be mapped back to the original data scale without losing physical interpretability.

To satisfy these requirements, we introduce Stabilized Instance Normalization (SIN), a robust, channel-wise, and reversible preconditioning layer for multi-hypothesis forecasting with lightweight backbones.

3.3 TimePre

Overall architecture. TimePre is a three-stage pipeline designed to combine the computational efficiency of lightweight forecasting backbones with the probabilistic output structure of Multiple Choice Learning. Given an input context window $\mathbf{X} \in \mathbb{R}^{L\times D}$, TimePre generates K candidate future trajectories $\{\widehat{\mathbf{Y}}^{(k)}\}_{k=1}^{K}$ through three components: Stabilized Instance Normalization ϕ, a linear temporal encoder Enc, and a multi-hypothesis decoder Dec. The overall pipeline is

$$\{\widehat{\mathbf{Y}}^{(1)}, \dots, \widehat{\mathbf{Y}}^{(K)}\} = \mathrm{Dec}\big(\mathrm{Enc}(\phi(\mathbf{X}))\big). \tag{5}$$

Here, ϕ preconditions the non-stationary input by reducing the scale imbalance diagnosed in Section 3.2. The stabilized representation is then processed by the linear encoder and finally mapped to a set of diverse trajectory hypotheses by the decoder. This design follows the multi-hypothesis forecasting formulation of Guzman-Rivera et al. (2012); Lee et al. (2016); Rupprecht et al. (2017); Perera et al. (2024b), while replacing heavy recurrent or Transformer-based backbones with a lightweight encoder and a reversible normalization step.

Stabilized Instance Normalization (SIN). SIN is introduced as a minimal preconditioning mechanism to stabilize WTA optimization under lightweight linear backbones. Its goal is not to increase model expressiveness, but to provide robust, channel-wise, and reversible normalization.

Standard Instance Normalization Ulyanov et al. (2016) satisfies the per-instance requirement, but it is sensitive to outliers and distribution shift, both of which are common in non-stationary time-series. A single extreme observation can substantially distort the empirical mean and variance, which in turn destabilizes normalization.

To improve robustness, SIN computes channel-wise statistics using a trimmed estimator. For each variable (channel) d, let $\mathbf{x}^{(d)} = (s_1^{(d)}, \dots, s_L^{(d)})^\top \in \mathbb{R}^L$, and let $s_{(i)}^{(d)}$ denote its i-th order statistic. Given a trimming ratio $p \in [0, 0.5)$, we set $k = \lfloor pL \rfloor$ and compute the robust mean and variance over the central $L - 2k$ values:

$$\mu_r^{(d)} = \frac{1}{L-2k} \sum_{i=k+1}^{L-k} s_{(i)}^{(d)}, \tag{6}$$

$$v_r^{(d)} = \frac{1}{L-2k} \sum_{i=k+1}^{L-k} \big(s_{(i)}^{(d)} - \mu_r^{(d)}\big)^2, \tag{7}$$

$$\sigma_r^{(d)} = \sqrt{v_r^{(d)} + \epsilon}. \tag{8}$$

The normalization and its exact inverse are then applied to the original, unsorted sequence:

$$\tilde{\mathbf{x}}^{(d)} = \frac{\mathbf{x}^{(d)} - \mu_r^{(d)} \mathbf{1}_L}{\sigma_r^{(d)}}, \qquad \mathbf{x}^{(d)} = \tilde{\mathbf{x}}^{(d)} \sigma_r^{(d)} + \mu_r^{(d)} \mathbf{1}_L. \tag{9}$$

Because SIN operates independently on each channel and each instance, it directly addresses the scale imbalance that distorts the WTA competition. At the same time, the use of trimmed statistics improves robustness under non-stationarity, while the closed-form inverse preserves interpretability in the original data space.

Linear temporal encoder. Following recent linear-centric forecasting models Zeng et al. (2023), we use a simple linear layer as the temporal encoder. For each normalized channel $\tilde{\mathbf{x}}^{(d)} \in \mathbb{R}^L$, the encoder projects the look-back window directly to the forecast horizon:

$$\mathbf{z}^{(d)} = W^{(d)}\tilde{\mathbf{x}}^{(d)} + b^{(d)}, \qquad W^{(d)} \in \mathbb{R}^{H\times L},\ b^{(d)} \in \mathbb{R}^H. \tag{10}$$

The latent representation is then formed as

$$\mathbf{Z} = [\mathbf{z}^{(1)} \mid \cdots \mid \mathbf{z}^{(D)}] \in \mathbb{R}^{H\times D}. \tag{11}$$

This encoder is computationally efficient and preserves the channel-wise forecasting structure used in lightweight time-series models.

Multi-hypothesis decoder. The decoder contains K parallel prediction heads. Each head consists of two components:

1. a trajectory head $f_\theta^{(k)}$ that predicts a trajectory hypothesis $\widehat{\mathbf{Y}}^{(k)} \in \mathbb{R}^{H\times D}$ from the shared latent representation $\mathbf{Z}$; and
2. a confidence head $g_\theta^{(k)}$ that estimates the confidence score $\gamma^{(k)}$ of the corresponding hypothesis.

Formally,

$$\widehat{\mathbf{Y}}^{(k)} = f_\theta^{(k)}(\mathbf{Z}), \qquad \gamma^{(k)} = \sigma\big(g_\theta^{(k)}(\mathrm{vec}(\mathbf{Z}))\big), \tag{12}$$

where $\mathrm{vec}(\cdot)$ denotes vectorization and $\sigma(\cdot)$ is the sigmoid function.

3.4 Training and Inference

For a training pair $(\mathbf{X}, \mathbf{Y})$, we first compute the per-head reconstruction loss

$$\mathcal{L}^{(k)} = \frac{1}{HD}\left\|\widehat{\mathbf{Y}}^{(k)} - \mathbf{Y}\right\|_F^2. \tag{13}$$

The winning hypothesis is defined as

$$k^* = \arg\min_k \mathcal{L}^{(k)}. \tag{14}$$

Relaxed WTA objective. To mitigate hypothesis starvation, we adopt an ε-relaxed WTA objective Rupprecht et al. (2017); Perera et al. (2024b); Seo et al. (2020). This objective assigns most of the gradient to the winner while preserving a smaller training signal for the remaining heads:

$$\mathcal{L}_{\text{R-WTA}} = (1-\varepsilon)\mathcal{L}^{(k^*)} + \frac{\varepsilon}{K-1}\sum_{j\neq k^*}\mathcal{L}^{(j)}. \tag{15}$$

Confidence calibration. The confidence scores $\gamma^{(k)}$ are trained to identify the winning hypothesis. We use a binary cross-entropy objective in which the winner is treated as the positive class:

$$\mathcal{L}_{\text{score}} = -\frac{1}{K}\left[\log\gamma^{(k^*)} + \sum_{k\neq k^*}\log\big(1-\gamma^{(k)}\big)\right]. \tag{16}$$

Total objective. The final objective combines the relaxed WTA loss and the calibration loss:

$$\min_{\Theta}\ \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\left[\mathcal{L}_{\text{R-WTA}} + \beta\,\mathcal{L}_{\text{score}}\right], \tag{17}$$

where β controls the trade-off between trajectory accuracy and confidence calibration.

Inference. At inference time, the model performs a single forward pass to produce all K hypotheses $\{\widehat{\mathbf{Y}}^{(k)}\}$ and their confidence scores $\{\gamma^{(k)}\}$. Together, these outputs form the final probabilistic forecast. Under squared-error distortion, the framework can also be interpreted as learning a soft Voronoi partition of future trajectory space, where each hypothesis head approximates the centroid of a conditional mode of $p(\mathbf{Y} \mid \mathbf{X})$.

4 Experiment

4.1 Experimental Setup

Data Sets. We follow the standard evaluation protocol in multivariate probabilistic forecasting and benchmark our model on six well-established real-world benchmark data sets from the GluonTS library Alexandrov et al. (2019). These data sets cover multiple domains, including energy Wu et al. (2019), finance Lai et al. (2017), and transportation Li et al. (2018). All data are preprocessed following prior work to ensure fair and consistent comparison across baselines. More details are provided in the supplementary material.

Metrics. We follow the standard evaluation protocol used in prior MCL-based probabilistic forecasting work Cortes et al. (2025) and report four metrics: Distortion Lee et al. (2016), CRPS-Sum Gneiting & Raftery (2007), FLOPs, and runtime. Distortion is the primary metric, measuring the mean Euclidean distance between each target sequence and its closest predicted hypothesis under the winner-takes-all objective:

$$D_2 = \frac{1}{N} \sum_{i=1}^{N} \min_{k=1,\dots,K} d\big(\mathcal{F}_\theta^k(x^i_{1:t_0-1}),\, x^i_{t_0:T}\big), \tag{18}$$

where K is the number of hypotheses and N is the number of test samples. This metric directly evaluates hypothesis coverage and is therefore well suited for MCL-based forecasting. CRPS-Sum complements Distortion by assessing the overall quality of probabilistic forecasts via the distance between the predicted distribution and the ground truth.

Baselines. We compare with six representative probabilistic forecasting methods covering a wide range of paradigms, resulting in eight baseline variants. The selected baselines include ETS Hyndman et al. (2002), DeepAR Salinas et al. (2020), TimeGrad Rasul et al. (2021), TempFlow Rasul et al. (2020), Tactis2 Ashok et al. (2024), and TimeMCL Cortes et al. (2025). Among them, TempFlow is implemented in two versions based on LSTM and Transformer backbones. To evaluate multi-hypothesis forecasting methods under consistent conditions, we additionally include two WTA-based training variants, Relaxed-WTA and Annealed-MCL.

Training Details. Following prior work, all models are trained using the Adam optimizer with an initial learning rate of 10^{-3} for 200 epochs. Each epoch consists of 30 batches of size 200, sampled from historical data through random windows, where the context length equals the prediction length. Early stopping is applied with a patience of 10 epochs. We reproduced all baseline models under identical configurations and training settings for fairness. All experiments are conducted on a single NVIDIA RTX A6000 GPU with 48 GB memory, and results are averaged over five random seeds.

4.2 Main Results

Tables 1 and 2 summarize the quantitative results of Distortion and CRPS-Sum, comparing the proposed TimePre with baseline models under 16 hypotheses. Figure 2 illustrates the computation–performance trade-off on the Exchange data set. Figure 3 provides qualitative comparisons among TimePre, TimeMCL (R.), and TimeMCL (A.).

Distortion. Table 1 shows that TimePre achieves the best Distortion on all six data sets. It achieves a 38.8% reduction compared to TimeMCL (R.) on Electricity and a 27.6% improvement on Exchange. Similar trends are observed on Solar and Taxi. On Wiki, TimePre still achieves the lowest Distortion, though the margin over the second-best Tactis2 is relatively small (263,492 vs. 263,975). In addition to accuracy, TimePre exhibits notably lower variance across runs (e.g., $\pm$203 on Electricity vs. $\pm$1772 for TimeMCL (R.)).

Table 1: Distortion risk under 16 hypotheses. We report mean ± standard deviation over five random seeds. The best results are in **bold**, and the second-best results are underlined. Lower is better.

Model	Electricity	Exchange	Solar	Traffic	Taxi	Wiki
ETS	23590 ± 2474	0.0796 ± 0.0030	692.32 ± 22.16	2.73 ± 0.02	609.67 ± 1.89	835095 ± 37871
TempFlow (Trf.)	17521 ± 2691	0.1150 ± 0.0290	466.25 ± 23.57	1.38 ± 0.06	308.62 ± 21.75	561226 ± 26593
Tactis2	13972 ± 917	0.0396 ± 0.0026	405.74 ± 17.19	0.87 ± 0.02	243.63 ± 9.10	263975 ± 11178
TimeGrad	14255 ± 1682	0.0576 ± 0.0090	406.91 ± 16.08	0.83 ± 0.02	221.32 ± 7.37	275437 ± 2645
DeepAR	184424 ± 19957	0.1320 ± 0.0204	865.61 ± 36.02	2.55 ± 0.12	477.93 ± 15.22	382340 ± 6592
TempFlow	17429 ± 1131	0.1168 ± 0.0325	424.24 ± 15.91	1.33 ± 0.02	293.76 ± 17.29	395996 ± 21535
TimeMCL (R.)	12693 ± 1772	0.0380 ± 0.0025	292.15 ± 11.68	0.71 ± 0.01	191.23 ± 5.34	268832 ± 9439
TimeMCL (A.)	10335 ± 767	0.0443 ± 0.0051	308.16 ± 14.87	0.72 ± 0.02	252.84 ± 30.62	276315 ± 9782
TimePre (Ours)	7774 ± 203	0.0275 ± 0.0004	267.06 ± 1.55	0.68 ± 0.02	182.27 ± 1.86	263492 ± 2368

Table 2: CRPS-Sum comparison on six benchmark data sets. All values are scaled by 10^2. We report mean ± standard error over five runs. Lower is better. The best results are in **bold**, and the second-best results are underlined.

Data Set	Tactis2	TempFlow	TimeMCL (R.)	TimePre (Ours)
Electricity	5.36 ± 0.34	7.08 ± 0.62	5.46 ± 0.85	3.15 ± 0.32
Exchange	0.82 ± 0.15	2.65 ± 0.91	1.05 ± 0.12	0.72 ± 0.03
Solar	40.58 ± 2.48	52.64 ± 3.28	41.12 ± 4.23	39.79 ± 0.66
Traffic	13.19 ± 1.24	49.15 ± 1.15	8.68 ± 1.10	11.81 ± 1.22
Taxi	22.52 ± 1.60	44.55 ± 6.93	46.19 ± 11.79	21.72 ± 0.53
Wiki	6.24 ± 0.87	14.49 ± 2.00	14.50 ± 3.84	6.14 ± 0.24

CRPS-Sum. Table 2 reports CRPS-Sum across the six data sets. TimePre achieves the best performance on five of six data sets and remains highly competitive on Traffic, where it ranks second behind TimeMCL (R.). The gains on Electricity, Exchange, and Taxi are especially substantial.

Computational Cost. To evaluate computational cost, we measure both runtime and FLOPs on the Exchange data set. As shown in Figure 2, TimePre avoids iterative autoregressive sampling and generates all future predictions in a single forward pass, resulting in strong inference efficiency. Its inference time is the fastest among all compared models, taking only 0.03s per batch. In terms of computational load, TimePre requires 4.28×10^4 FLOPs, second only to DeepAR (2.90×10^4 FLOPs), while achieving substantially lower distortion.

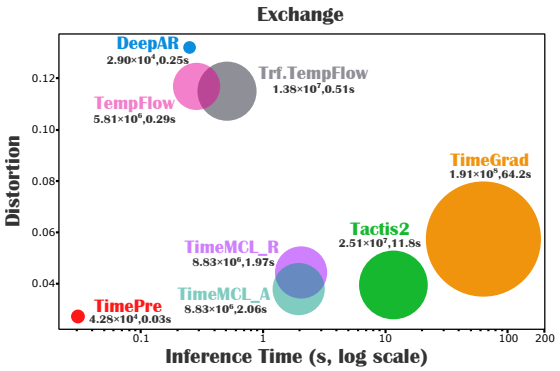

Figure 2: Computation–performance trade-off on the Exchange data set under 16 hypotheses. Lower is better. Circle size indicates FLOPs.

Visualization and Qualitative Analysis. We qualitatively compare TimePre with TimeMCL on the Electricity, Solar, Traffic, Taxi, and Wiki data sets. As shown in Figure 3, TimePre produces more stable predictions, whereas the annealed variant exhibits scale drift and the relaxed variant shows only partial improvement. On Wiki, TimeMCL collapses into nearly constant outputs and fails to produce meaningful forecasts, while TimePre remains stable and captures realistic temporal variations.

4.3 Analysis of Different Normalizations

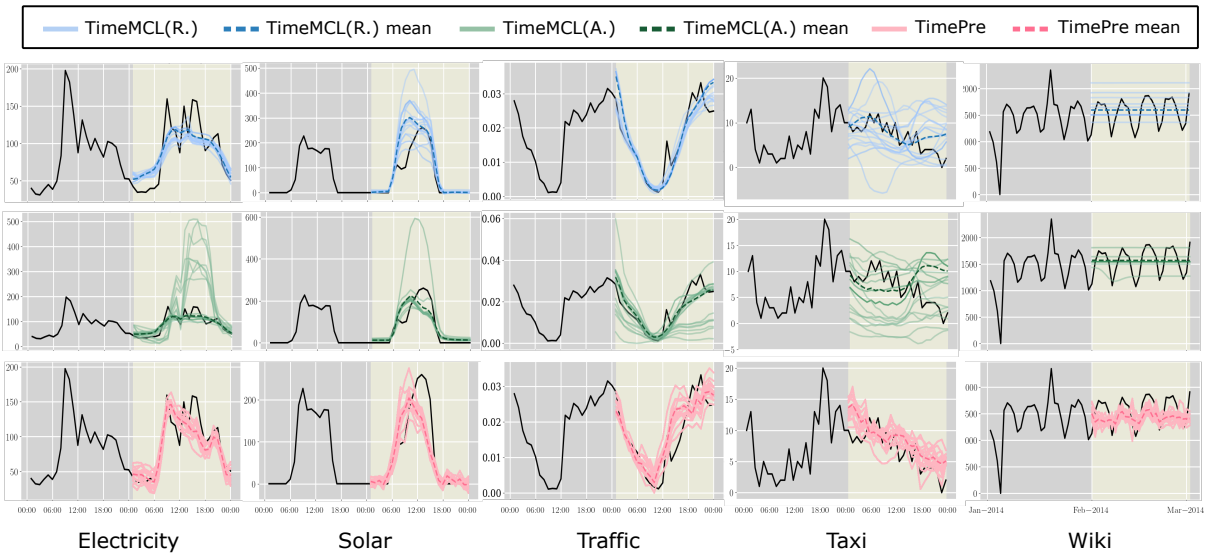

Figure 3: Qualitative forecasting results on five public data sets, comparing three models under the multi-hypothesis paradigm: TimeMCL (R.), TimeMCL (A.), and TimePre, all with 16 hypotheses.

Table 3: Comparison of normalization methods on the Electricity data set. Lower is better for both metrics.

Method	Distortion	CRPS-Sum
BatchNorm Ioffe & Szegedy (2015)	12447 ± 442	0.0649 ± 0.0076
LayerNorm Ba et al. (2016)	9453 ± 334	0.0510 ± 0.0039
GroupNorm Wu & He (2018)	8452 ± 523	0.0530 ± 0.0058
InstanceNorm Ulyanov et al. (2016)	9712 ± 482	0.0623 ± 0.0084
SIN (Ours)	7774 ± 203	0.0315 ± 0.0032

To assess the impact of normalization within TimePre, we replace the Stabilized Instance Normalization module with four commonly used alternatives: BatchNorm Ioffe & Szegedy (2015), LayerNorm Ba et al. (2016), GroupNorm Wu & He (2018), and InstanceNorm Ulyanov et al. (2016). Table 3 and Figure 4 show that the choice of normalization is a decisive factor for training stability and representational quality under the MCL paradigm.

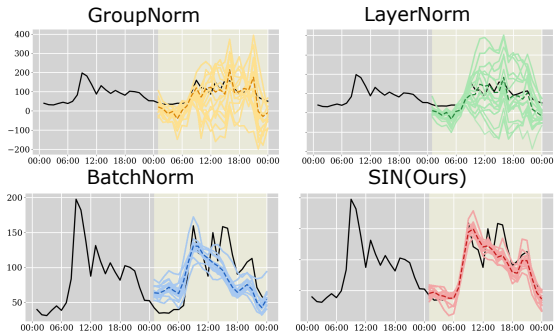

Figure 4: Forecasting comparison across normalization layers on Electricity. LayerNorm and GroupNorm use $[-200, 400]$, while BatchNorm and SIN use $[0, 200]$.

Training Instability. LayerNorm and GroupNorm exhibit pronounced instability during optimization. As shown in Figure 4, both methods produce latent trajectories with inconsistent amplitude and severe scale drift.

Representational Inaccuracy. BatchNorm avoids divergence but yields inferior forecasting accuracy. Since BatchNorm aggregates statistics across a batch, it disrupts the per-sample temporal structure essential for precise forecasting.

Table 4: Effect of the number of hypotheses on Distortion risk on the Electricity data set. Lower is better. Results are averaged over five runs.

#Hypotheses	TimeMCL (R.)	TimeMCL (A.)	TimePre (Ours)
2	16012 ± 2310	15349 ± 2702	9201 ± 336
4	14311 ± 1234	13513 ± 1698	8864 ± 441
6	14173 ± 1432	11999 ± 624	8688 ± 783
8	13618 ± 793	12503 ± 1104	8590 ± 552
10	17216 ± 9112	12597 ± 2155	8066 ± 553
12	17277 ± 7886	12507 ± 820	8202 ± 259
14	13657 ± 2137	10902 ± 1152	7758 ± 480
16	12693 ± 1772	10335 ± 767	7774 ± 182

Table 5: Distortion risk comparison on three data sets. Lower is better. We report mean ± standard error over five runs.

Model	Exchange	Solar	Taxi
TimePre	$\underline{0.0276 \pm 0.0005}$	261.06 ± 1.55	182.27 ± 1.86
TimePre (D.)	0.0271 ± 0.0005	267.21 ± 2.79	$\underline{179.82 \pm 0.83}$
TimePre (M.)	0.0311 ± 0.0017	$\underline{262.07 \pm 5.55}$	169.22 ± 2.00
TimePre (T.)	0.0412 ± 0.0016	478.26 ± 73.69	174.18 ± 14.47

4.4 Ablation Study

Effect of the Number of Hypotheses. To evaluate the stability and scalability of TimePre, we analyze performance under different numbers of hypotheses K. Distortion results on the Electricity data set are compared against baseline models with both Relaxed and Annealed variants.

As shown in Table 4, TimePre consistently outperforms both TimeMCL variants across all values of K, and its performance improves monotonically as K increases from 2 to 16. In contrast, TimeMCL (R.) exhibits non-monotonic behavior and high variance, with Distortion spiking at K=10 ($17{,}216 \pm 9{,}112$) and K=12 ($17{,}277 \pm 7{,}886$), indicating unstable optimization when more hypotheses compete under the WTA objective. TimeMCL (A.) is more stable but still shows considerably higher variance than TimePre at most settings. Notably, TimePre's standard deviation remains consistently low (below 800 across all K), confirming that SIN effectively stabilizes multi-hypothesis learning regardless of the number of competing heads.

Effect of Different Backbones. To assess the robustness and flexibility of TimePre, we replace its original single-layer linear backbone with three representative MLP-based models: DLinear Zeng et al. (2023), TimeMixer Wang et al. (2024), and TiDE Das et al. (2023).

Table 5 reveals two key findings. First, a more complex backbone does not necessarily improve performance: TimePre (T.) with the TiDE backbone suffers a dramatic degradation on Solar (478.26 vs. 261.06) and exhibits high variance (±73.69), suggesting that heavier architectures can amplify instability under the MCL objective. Second, the default single-layer linear backbone achieves the best or near-best result on two of three data sets (Solar and Exchange), indicating that TimePre's gains primarily stem from the SIN preconditioning rather than backbone complexity. TimePre (M.) with the TimeMixer backbone achieves the best Taxi result, suggesting that dataset-specific architectural choices can yield further improvements; however, the default linear backbone offers the most consistent performance across diverse domains.

5 Conclusion

We drew inspiration from recent MLP-based models in LTSF and extended their design to the MCL paradigm. To resolve the inherent incompatibility between linear architectures and the MCL framework, we proposed the Stabilized Instance Normalization mechanism to harmonize feature scales and stabilize optimization.

By integrating a linear backbone with a direct forecasting paradigm, TimePre achieves the best Distortion on all six benchmark datasets and the best or highly competitive CRPS-Sum on most benchmarks, while maintaining extremely fast inference. Overall, TimePre unifies the strengths of deterministic and probabilistic forecasting, bridging accuracy, efficiency, and stability.

Limitations and future work. Our current evaluation focuses on six GluonTS benchmarks that primarily consist of fixed-horizon multivariate time series. While the results are consistent across diverse domains, further validation on datasets with richer covariate structures, varying prediction horizons, or substantially different value ranges would strengthen the generality of the conclusions. Additionally, TimePre's direct forecasting paradigm does not explicitly enforce temporal dependencies across the prediction horizon; incorporating temporal consistency constraints may further improve long-horizon smoothness without sacrificing efficiency.

References

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. Gluonts: Probabilistic time series models in python, 2019. URL https://arxiv.org/abs/1906.05264.

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. TACTis-2: Better, faster, simpler attentional copulas for multivariate time series. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xtOydkE1Ku.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Adrien Cortes, Remi Rehm, and Victor Letzelter. Winner-takes-all for multivariate probabilistic time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4QcFfTu6UT.

Kazunori D YAMADA, Fangzhou Lin, and Tsukasa Nakamura. Developing a novel recurrent neural network architecture with fewer parameters and good learning performance. *Interdisciplinary information sciences*, 27(1):25–40, 2021.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=pCbC3aQB5W.

Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 11 2000. doi: 10.1214/aos/1013203451.

Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Springer, 1992.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pp. 1799–1807, 2012.

S.L. Ho and M. Xie. The use of arima models for reliability forecasting and analysis. *Computers Industrial Engineering*, 35(1):213–216, 1998. ISSN 0360-8352. doi: https://doi.org/10.1016/S0360-8352(98)00066-7. URL https://www.sciencedirect.com/science/article/pii/S0360835298000667.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Yang Hu, Xiao Wang, Zezhen Ding, Lirong Wu, Huatian Zhang, Stan Z. Li, Sheng Wang, Jiheng Zhang, Ziyun Li, and Tianlong Chen. Flowts: Time series generation via rectified flow, 2025. URL https://arxiv.org/abs/2411.07506.

Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18 (3):439–454, 2002. ISSN 0169-2070. doi: https://doi.org/10.1016/S0169-2070(01)00110-8. URL https://www.sciencedirect.com/science/article/pii/S0169207001001108.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges, 2025. URL https://arxiv.org/abs/2411.05793.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2017. URL https://api.semanticscholar.org/CorpusID:4922476.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.

Seungjun Lee, S. P. Purkayastha, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 2119–2127, 2016.

Victor Letzelter, David Perera, Cédric Rommel, Mathieu Fontaine, Slim Essid, Gael Richard, and Patrick Pérez. Winner-takes-all learners are geometry-aware conditional density estimators, 2024. URL https://arxiv.org/abs/2406.04706.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Jean-Michel Loubes and Bertrand Pelletier. A functional view of quantization and clustering. *ESAIM: Probability and Statistics*, 21:93–114, 2017.

Yecheng Lyu, Ming Li, Xinming Huang, Ulkuhan Guler, Patrick Schaumont, and Ziming Zhang. Treernn: Topology-preserving deep graph embedding and learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7493–7499. IEEE, 2021.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Implicit regularization in deep learning: A view from function space. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

David Perera, Victor Letzelter, Theo Mariotte, Adrien Cortes, Mickael Chen, Slim Essid, and Gaël Richard. Annealed multiple choice learning: Overcoming limitations of winner-takes-all with annealing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=WEs4WMzndY.

Ruwan Perera, Dhruv Batra, David Crandall, and Zsolt Kira. Multi-choice learning for multimodal sequence prediction. *Transactions on Machine Learning Research (TMLR)*, 2024b.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.

Rial A. Rajagukguk, Raden A. A. Ramadhan, and Hyun-Jin Lee. A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power. *Energies*, 13(24), 2020. ISSN 1996-1073. doi: 10.3390/en13246623. URL https://www.mdpi.com/1996-1073/13/24/6623.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multi-variate probabilistic time series forecasting via conditioned normalizing flows. *CoRR*, abs/2002.06103, 2020. URL https://arxiv.org/abs/2002.06103.

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

Alejandro Rodriguez Domínguez, Muhammad Shahzad, and Xia Hong. Structured basis function networks: Loss-centric multi-hypothesis ensembles with controllable diversity. 09 2025. doi: 10.48550/arXiv.2509.02792.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3591–3600, 2017.

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

Younggyo Seo, Kimin Lee, Ignasi Clavera, Thanard Kurutach, Jinwoo Shin, and Pieter Abbeel. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 17672–17683, 2020.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *ArXiv*, abs/1607.08022, 2016. URL https://api.semanticscholar.org/CorpusID:16516553.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Yuyang Wang, Alex Smola, Danielle C. Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting, 2019. URL https://arxiv.org/abs/1905.12417.

Ruofeng Wen, Kari Torkkola, and Balakrishnan Narayanaswamy. A multi-horizon quantile recurrent forecaster. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nat. Mac. Intell.*, 5(6):602–611, June 2023. URL https://doi.org/10.1038/s42256-023-00667-9.

Yuxin Wu and Kaiming He. Group normalization. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, pp. 3–19, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01260-1. doi: 10.1007/978-3-030-01261-8_1. URL https://doi.org/10.1007/978-3-030-01261-8_1.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pp. 1907–1913. AAAI Press, 2019. ISBN 9780999241141.

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Ziming Zhang, Fangzhou Lin, Haotian Liu, Jose Morales, Haichong Zhang, Kazunori Yamada, Vijaya B Kolachalama, and Venkatesh Saligrama. Gps: A probabilistic distributional similarity with gumbel priors for set-to-set matching. In *The Thirteenth International Conference on Learning Representations*, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021. URL https://arxiv.org/abs/2012.07436.

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

A Implementation Details

A.1 Hyperparameters

Table 6 lists the key hyperparameters of TimePre used across all experiments. Unless otherwise stated, the same configuration is applied to all six benchmark datasets.

Table 6: Key hyperparameters of TimePre. All values are shared across the six benchmark datasets unless otherwise noted.

Hyperparameter	Symbol	Value
Number of hypotheses	K	16
Trimming ratio (SIN)	p	0.1
Relaxed WTA coefficient	ε	0.05
Confidence loss weight	β	1.0
Trajectory head layers	–	1 (linear)
Trajectory head hidden size	–	$H \times D$
Confidence head layers	–	2 (MLP)
Confidence head hidden size	–	128
Optimizer	–	Adam
Learning rate	–	10^{-3}
Batch size	–	200
Batches per epoch	–	30
Training epochs	–	200
Early stopping patience	–	10 epochs
Numerical stability constant	ϵ	10^{-5}

A.2 Runtime and FLOPs Measurement Protocol

All runtime measurements are conducted on a single NVIDIA RTX A6000 GPU with 48 GB memory. We report wall-clock inference time averaged over 100 forward passes after a 10-pass warm-up, using a fixed batch size of 200 samples. FLOPs are computed using `torch.profiler` with `profile_memory=False`, covering all linear, normalization, and activation operations in a single forward pass. Both metrics are measured on the Exchange dataset under identical conditions for all models.

A.3 CRPS-Sum Computation

Our CRPS-Sum implementation follows the GluonTS evaluation protocol. For MCL-based models that produce K discrete trajectory hypotheses rather than parametric distributions, we construct an empirical CDF from the K predicted trajectories, weighted by the confidence scores $\{\gamma^{(k)}\}$. CRPS is then computed via the standard integral formulation applied to the aggregated (summed across variables) forecast at each horizon step.

B Dataset Details

We evaluate our method on six widely used probabilistic time-series forecasting benchmarks from the GluonTS library, namely *Solar*, *Electricity*, *Exchange*, *Traffic*, *Taxi*, and *Wikipedia*. All datasets contain strictly positive real-valued sequences and come with standard train–test splits defined in prior work. An overview of their main characteristics is provided in Table 7.

Solar. The Solar dataset contains hourly aggregated power production from 137 photovoltaic plants over roughly 7,000 time steps. The series exhibit strong daily seasonality induced by the day–night cycle, making them a canonical benchmark for modeling periodic but weather-dependent generation patterns.

Table 7: Summary of the benchmark datasets used in our experiments. N denotes the number of time series, T the length of each series, and "Freq." the sampling frequency. The prediction horizon H follows the standard settings in prior work.

Dataset	N	T	Freq.	Horizon H
Solar	137	7009	Hourly	24
Electricity	370	5833	Hourly	24
Exchange	8	6071	Daily	30
Traffic	963	4001	Hourly	24
Taxi	1214	1488	30-min	24
Wikipedia	2000	792	Daily	30

Electricity. The Electricity dataset consists of hourly electricity consumption for 370 customers over 5,833 time steps. Demand typically follows both daily and weekly cycles driven by human activity and business operations, and it can be affected by holidays and load spikes, which pose challenges for probabilistic forecasting models.

Exchange. The Exchange dataset contains daily foreign exchange rates for 8 major currency pairs, each with 6,071 observations. Unlike energy or traffic data, these financial series seldom show clear periodicity; instead, they reflect macroeconomic conditions and market events, providing a non-seasonal and highly stochastic forecasting scenario.

Traffic. The Traffic dataset records road occupancy rates (bounded in $[0, 1]$) from 963 loop sensors, sampled hourly for approximately 4,000 time steps. The series display pronounced rush-hour peaks as well as systematic differences between weekdays and weekends, which makes them a representative benchmark for high-dimensional, strongly seasonal traffic flows.

Taxi. The Taxi dataset is based on taxi ride counts in New York City, aggregated at 1,214 spatial locations every 30 minutes. We use the standard preprocessed version from GluonTS, which includes data from January 2015 (training) and January 2016 (testing). The resulting series capture complex spatial–temporal patterns and irregular spikes in demand.

Wikipedia. The Wikipedia dataset contains daily page-view counts for 2,000 popular Wikipedia pages. These series exhibit a mixture of long-term trends, weekly seasonality, and occasional bursts due to external events or campaigns. Following prior work, we adopt the official split and treat this dataset as a challenging benchmark for high-dimensional, event-driven demand forecasting.

Across all datasets, we follow the official train–test splits provided by GluonTS and prior benchmarks. For validation, we reserve the last few time steps before the forecast horizon within the training portion, as summarized in Table 7.

C Evaluation Metrics

We evaluate our approach using four metrics that capture both probabilistic forecasting quality and computational efficiency: Distortion, CRPS-Sum, FLOPs, and Runtime. Among them, Distortion serves as our primary evaluation metric.

Distortion. Distortion measures how well the set of predicted hypotheses covers the true target distribution. Given K predicted trajectories $\{\hat{\mathbf{y}}^{(k)}\}_{k=1}^{K}$ for each ground-truth sequence $\mathbf{y}$, distortion is defined as the minimum Euclidean distance between the target and the closest hypothesis:

$$\text{Distortion}(\mathbf{y}) = \min_{1 \le k \le K} \left\| \mathbf{y} - \hat{\mathbf{y}}^{(k)} \right\|_2 . \tag{19}$$

The final score is obtained by averaging over all test samples:

$$\text{Distortion} = \frac{1}{N}\sum_{i=1}^{N}\min_{1\le k\le K}\left\|\mathbf{y}_i - \hat{\mathbf{y}}_i^{(k)}\right\|_2. \tag{20}$$

Lower distortion indicates better probabilistic coverage and sharper forecasting quality.

CRPS-Sum. The Continuous Ranked Probability Score (CRPS) Gneiting & Raftery (2007) evaluates the accuracy of a predictive distribution. Following prior work and the GluonTS implementation, CRPS-Sum is computed by first aggregating all individual time series and then applying CRPS to the resulting summed distribution at each forecast horizon. Formally, let $\tilde{y}_t = \sum_{d=1}^{D} y_{t,d}$ denote the aggregated target at time step t, and let $\tilde{F}_t$ be the predictive CDF of the corresponding aggregated forecast. CRPS-Sum is then defined as:

$$\text{CRPS-Sum} = \sum_{t=1}^{H}\text{CRPS}\big(\tilde{F}_t, \tilde{y}_t\big). \tag{21}$$

Lower CRPS-Sum indicates better calibrated probabilistic forecasts on the aggregated series.

FLOPs. Floating Point Operations measure the computational cost of a single forward pass. We compute FLOPs using the standard profiling tools in PyTorch, covering all linear, convolutional, normalization, and activation operations. Lower FLOPs indicate better computational efficiency and scalability.

Runtime. Runtime records the wall-clock time required for one forward pass on a single GPU under identical batch size and sequence length. This metric reflects real-world inference latency and complements FLOPs by capturing implementation overhead and hardware-level optimizations.

D Baseline Details

We compare our method against a wide range of strong probabilistic forecasting baselines, including both classical likelihood-based models and recent deep learning approaches. All baseline implementations follow the official code from GluonTS or the authors' releases, and we use the standard hyperparameters recommended in their original papers to ensure fair comparison.

DeepAR. DeepAR Salinas et al. (2020) is an autoregressive probabilistic model based on LSTM networks. It predicts future values by estimating a parametric likelihood (e.g., Gaussian or Negative Binomial) and sampling from the learned distribution. As a widely adopted baseline, DeepAR captures temporal dependencies through recurrent structures but may struggle with long-term dependencies due to sequential recurrence.

TimeGrad. TimeGrad applies a conditional variational autoencoder (CVAE) framework to time-series forecasting. By learning latent stochastic dynamics through diffusion-based variational inference, it provides expressive probabilistic forecasts. Its training, however, requires sampling latent variables per time step, leading to slower inference.

TimeMCL. TimeMCL Perera et al. (2024b) extends the Multiple Choice Learning (MCL) framework to time-series forecasting by training K parallel prediction heads with a winner-takes-all assignment scheme. It generates diverse hypotheses that better cover the multimodal distribution of future trajectories. Although effective, the dense latent representations in TimeMCL lead to high cross-channel covariance and less stable training on some datasets.

TimeMixer. TimeMixer is a multi-period decomposition architecture that mixes temporal patterns across fine-to-coarse granularities. It performs especially well on seasonal datasets due to its frequency-aware decomposition and learned periodic mixing kernels.

TiDE. TiDE uses a two-stage architecture: an encoder that extracts temporal representations and a decoder that predicts the full future horizon in one shot. Its MLP-based structure allows for efficient training, although it can be sensitive to hyperparameter choices and normalization strategies.

DLinear. DLinear is a highly efficient linear modeling baseline that decomposes the input into trend and seasonal components using two simple linear layers. Despite its simplicity, it achieves strong performance on many long-term forecasting benchmarks and is widely used as a lightweight baseline.

For all baselines, we use the standard forecasting horizon defined in prior benchmarks and measure probabilistic performance using Distortion and CRPS-Sum (Section C). Computational efficiency is compared via FLOPs and Runtime under identical batch size and hardware settings.