ToLD: Efficient Time Series Forecasting via Tokenized Truncated Latent Diffusion

Jiayi Tian[1,*], **Jiaze Wang**[2,*], **Tian Xia**[1], **Wenzhe Zhao**[1], **Pengju Ren**[1,*]
[1] National State Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University,
Xi'an, Shaanxi, China
[2] Harbin Institute of Technology, Shenzhen, China
tianreg@stu.xjtu.edu.cn, 24b951071@stu.hit.edu.cn
{tian_xia, wenzhe, pengjuren}@xjtu.edu.cn

Abstract

Probabilistic long-term time series forecasting requires estimating a distribution over future trajectories under practical deployment constraints, where inference latency and stability can be as critical as forecast quality. Existing diffusion-based forecasters provide expressive uncertainty modeling, yet their iterative denoising incurs substantial horizon insensitive runtime overhead, which limits real time long-term use. In contrast, one-shot generative models such as VAEs and flows are computationally efficient but often struggle to capture complex multi-modal structure and heavy tailed uncertainty. We propose **ToLD**, an efficient time-series forecasting framework built upon **to**kenized truncated **l**atent **d**iffusion to enable fast and reliable long term generation. ToLD adopts a tokenized conditional latent modeling and performs compact token-wise refinement via a few-step truncated latent diffusion process, where the diffusion noise is injected in a context-dependent manner to model non-stationary uncertainty. To reduce train test mismatch in multi-sample forecasting, we further introduce a score distillation scheme that learns a target-free scoring function for test-time candidate ranking. Experiments on eight real-world datasets show that ToLD improves both point accuracy and probabilistic quality over state-of-the-art, achieving up to **4.3%** MSE improvement and **19.3%** CRPS reduction on long-term tasks with minimal additional inference overhead.

1 Introduction

Long-term time series forecasting (LTSF) underpins decision-making in transportation, energy, healthcare, and finance [36, 22, 16, 34]. Given a historical multivariate sequence $X \in \mathbb{R}^{L\times C}$ with lookback length L and C variables, the goal is to predict a future horizon $Y \in \mathbb{R}^{H\times C}$ of length H. In high-stakes settings, point forecasts alone are insufficient: practitioners also require *calibrated* predictive uncertainty for risk-aware planning [5, 2, 36]. This requirement becomes more challenging at long horizons, where forecast dispersion typically grows and varies with context, resulting in heteroscedastic predictive uncertainty [19, 38, 28].

Most LTSF methods remain deterministic and output point estimates [34, 16]. Classical probabilistic forecasters impose a pre-specified likelihood family at each timestep [25, 21], which can be too restrictive for heavy-tailed, multi-modal, or strongly context-dependent futures. These limitations motivate *generative* probabilistic forecasting that learns flexible conditional distributions $p(Y \mid X)$ and represents uncertainty through samples [36, 35, 12].

*Pengju Ren is the corresponding author. *Equal contribution.

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

Diffusion models are a prominent generative paradigm and often yield high-quality conditional samples [23, 28, 29, 32]. However, our Motivation (Fig. 1) highlights a key deployment barrier for LTSF: an *accuracy efficiency mismatch* across prevailing diffusion paradigms. (i) *Fully iterative diffusion* relies on long reverse chains, making inference prohibitively slow when long horizons or multi-sample evaluation is required. (ii) *Conditional iterative diffusion* further increases per-sample cost due to conditional inputs and still requires many denoising steps. (iii) *Learned-prior diffusion* improves the endpoint prior to better reflect heteroscedasticity, yet inference remains step- and sample-dependent, and uncertainty can still be biased or miscalibrated under limited sampling budgets. As shown in the Pareto analysis (Fig. 1), existing methods either achieve good accuracy at high latency or improve efficiency at the expense of uncertainty quality, leaving a practical gap for *fast and reliable* long-term probabilistic forecasting.

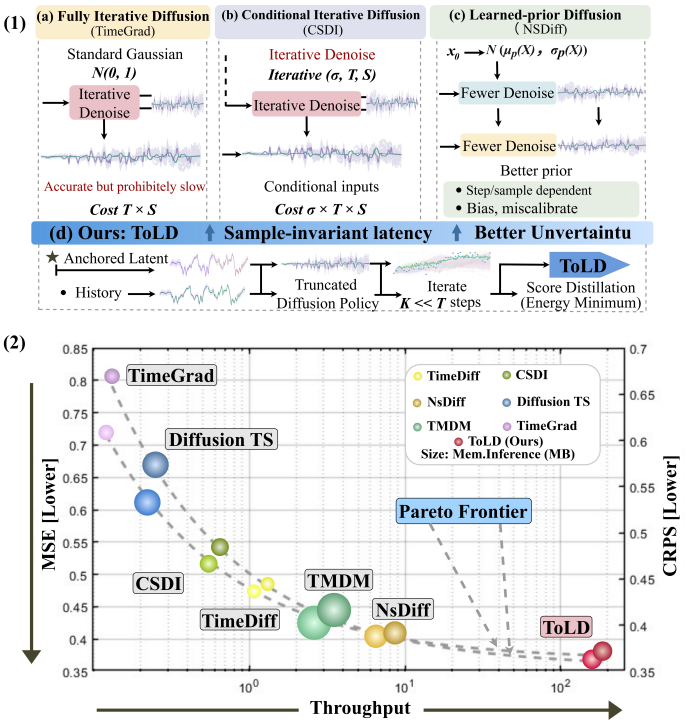


Figure 1: ToLD improves the accuracy–uncertainty–efficiency trade-off through efficient tokenized latent refinement.

We propose **ToLD**, which resolves this tension with *sample-invariant latency* while improving uncertainty modeling. ToLD starts from a token-wise conditional VAE that learns a context-conditional latent prior $p_\theta(z_0 \mid X) = \mathcal{N}(\mu_p(X), \mathrm{diag}(\sigma_p^2(X)))$ and decodes the entire horizon in a single pass. To enhance sample quality without incurring expensive observation-space diffusion, we perform *truncated diffusion refinement* in a compact latent token space $z_t \in \mathbb{R}^{C\times d}$ using only a few deterministic reverse steps ($K \ll T$). Crucially, we inject heteroscedasticity *inside the sampling dynamics* by scaling the latent diffusion noise with the context-dependent prior variance $\sigma_p^2(X)$, so the refinement process adapts to scenario-specific uncertainty rather than assuming a fixed covariance endpoint. Finally, since selecting the optimal sample during testing is not feasible, we replace this process with a learned scoring mechanism. This scorer ranks candidate trajectories based on learned patterns, without relying on future targets, thereby enhancing sample quality under tight sampling budgets by utilizing knowledge gained during training.

Our contributions are as follows:

- We propose **tokenized conditional latent modeling**, which represents each variable with a dedicated context-conditioned latent token. This formulation yields a compact latent forecasting space and decouples the refinement complexity from the forecast horizon.
- We introduce **truncated latent diffusion**, a few-step latent refinement mechanism initialized from a learned conditional prior. By injecting diffusion noise according to the context-dependent prior variance, ToLD incorporates heteroscedastic uncertainty directly into the sampling dynamics while preserving efficient one-pass decoding.
- We present **test-time score distillation** for target-free candidate selection in multi-sample forecasting. The learned scorer distills training-time best-sample selection into an inference-time ranking function, enabling ToLD to select high-quality trajectories without access to future observations.

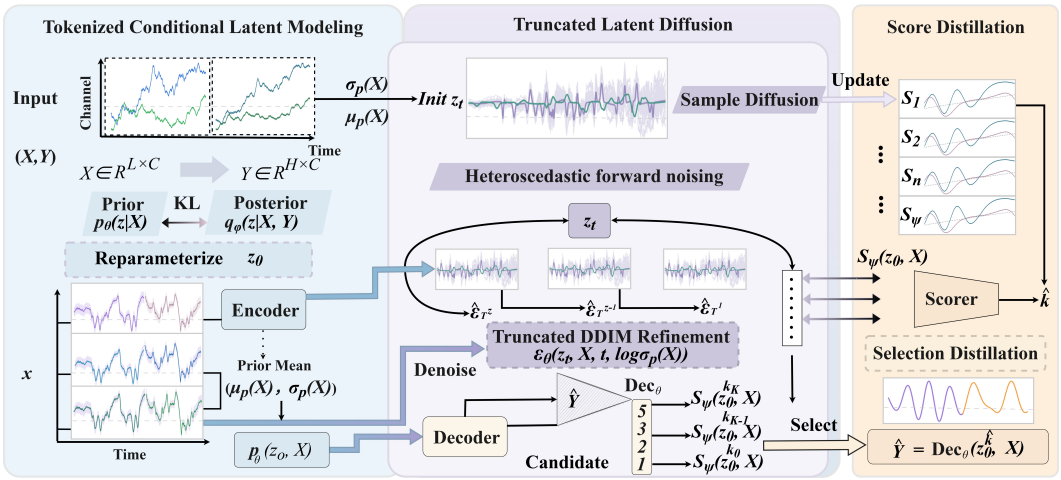


Figure 2: **Overview of ToLD.** Given historical context $\mathbf{X}$, the model generates the full horizon in a single pass using a token-wise conditional VAE, which produces a context-dependent latent prior $p_\theta(\mathbf{z}_0 \mid \mathbf{X})$ and employs a one-step decoder Dec_θ. To improve the quality of the sample with low latency, we refine the latent samples through *truncated latent diffusion*, which is applied for a limited number of steps ($T' \ll T$) in the latent space, ensuring high efficiency. The forward noising process is *heteroscedastic*, where the noise scale is controlled by the prior uncertainty $\boldsymbol{\sigma}_p(\mathbf{X})$, introducing context-dependent dispersion in the diffusion process. Multiple candidates are generated, and the best sample is selected based on a learned scorer. This scorer is trained via score distillation to mimic sample selection at inference time.

2 Related Work

2.1 Diffusion Models for Probabilistic Time-Series Forecasting

Denoising diffusion probabilistic models (DDPMs) have emerged as a powerful paradigm for modeling complex conditional distributions in time series [8, 10]. Early autoregressive approaches like TimeGrad [23] condition the denoiser on RNN hidden states, but their sequential nature leads to prohibitive inference costs for long horizons. Non-autoregressive models, such as CSDI [28] and structured state-space approaches [9], enable parallel generation via masking mechanisms but often assume fixed standard Gaussian priors ($\mathcal{N}(0, I)$), which limits their ability to capture heteroscedastic uncertainty under non-stationary regimes [32]. While recent works like DiffusionTS [33] and NsDiff-fusion [32] attempt to improve sample realism via frequency-aware objectives or learned priors, they typically remain bound by the computational burden of full-step reverse processes [1]. Unlike these full-step diffusion forecasters, our framework leverages a truncated refinement strategy to achieve high-quality uncertainty modeling with minimal sampling overhead.

2.2 Latent Variable Generative Forecasting and Context-Dependent Uncertainty

Latent-variable forecasters provide an efficient alternative by sampling a context-conditional latent $z \sim p(z \mid X)$ and decoding the entire horizon in one pass [37, 20, 4]. To increase expressiveness, flows have been used either as latent priors or posterior approximators [7, 15, 24, 4], and some works improve posterior inference while keeping one-pass generation at test time [4]. However, in long-term settings, purely one-shot latent sampling may under-cover complex predictive distributions, especially when uncertainty is strongly context-dependent (heteroscedastic) [4, 36]. Meanwhile, deterministic forecasting has extensively studied non-stationarity and distribution shift via normalization, attention modification, decomposition, and operator-based modeling [18, 32, 14, 13], but transferring such mechanisms to probabilistic sampling remains non-trivial. While recent work has significantly improved time series forecasting, the treatment of non-stationary uncertainty in probabilistic models has received surprisingly little attention.

3 Preliminary

3.1 Problem Formulation

We study probabilistic long-term forecasting for multivariate time series by learning a conditional predictive distribution $p_\theta(Y \mid X)$ from data:

$$\min_\theta \ \mathbb{E}_{(X,Y)\sim\mathcal{D}}\Big[\, \mathcal{S}\big(p_\theta(\cdot \mid X), Y\big)\,\Big], \tag{1}$$

where θ denotes model parameters, $\mathcal{D}$ denotes the data distribution (or the empirical training set), and $\mathcal{S}(\cdot,\cdot)$ is a proper scoring rule. The input $X \in \mathbb{R}^{L\times C}$ contains L historical steps of C variables, and the target $Y \in \mathbb{R}^{H\times C}$ contains H future steps.

In practice, the distribution $p_\theta(\cdot \mid X)$ is represented by Monte Carlo samples $\tilde{Y}^{(k)} \sim p_\theta(\cdot \mid X)$ for $k = 1, \ldots, K$, where $K \in \mathbb{N}$ is the sampling budget. Training typically employs mini-batches of size B, yielding tensors $X \in \mathbb{R}^{B\times L\times C}$ and $Y \in \mathbb{R}^{B\times H\times C}$. Our framework focuses on heteroscedastic settings where the predictive dispersion of $p_\theta(Y \mid X)$ strongly depends on the input context X. Consequently, accurate forecasting requires not only precise point predictions but also calibrated, input-dependent uncertainty estimates.

3.2 DDPM Basics

A denoising diffusion probabilistic model defines: (i) a fixed forward noising process q; (ii) a learnable reverse denoising process p_θ. Let $T \in \mathbb{N}$ be the total number of diffusion steps, and let $\{\beta_t\}_{t=1}^T$ be a variance schedule with $\beta_t \in (0,1)$. Define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. The standard forward Markov transition is

$$q(y_t \mid y_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\, y_{t-1},\ (1-\alpha_t)I)\,, \tag{2}$$

which implies the closed form

$$q(y_t \mid y_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, y_0,\ (1-\bar{\alpha}_t)I\big)\,. \tag{3}$$

Sampling relies on iteratively applying the reverse transitions from $t = T$ down to $t = 0$, where the reverse model is commonly parameterized by a denoiser that predicts the injected Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$.

3.3 Conditional VAE for One step Forecasting

A conditional variational autoencoder introduces a latent variable z and models

$$p_\theta(Y \mid X) = \int p_\theta(Y \mid z, X)\, p_\theta(z \mid X)\, dz, \tag{4}$$

where $p_\theta(z \mid X)$ is the conditional prior. Training uses a variational posterior $q_\phi(z \mid X, Y)$ with parameters ϕ. A key advantage of conditional VAEs is efficient generation: one samples z from the conditional prior and decodes once to obtain a full horizon forecast. However, one shot latent sampling may under represent complex long horizon predictive distributions, and standard Gaussian latents can be insufficient for heavy tailed or multi modal uncertainty.

4 Method

This section provides a brief overview of our framework. As illustrated in Fig. 2, our method begins with a tokenized conditional VAE backbone that generates a context-dependent latent prior to anchor the prediction; this is subsequently refined by a truncated latent diffusion process, which explicitly models heteroscedastic uncertainty by scaling the diffusion noise with the prior variance, and finally, a score distillation mechanism is employed to rank and select the optimal candidate trajectory at inference time without access to future targets.

4.1 Tokenized Conditional Latent Modeling

We adopt a *token-wise conditional VAE* where each variate is modeled by one latent token, $z_0 \in \mathbb{R}^{C\times d}$. This tokenization aligns with multivariate forecasting and keeps refinement lightweight because diffusion scales with C rather than HC.

Let $\mathrm{Emb}(\cdot)$ denote a series embedding module that maps X to token features $h_X \in \mathbb{R}^{C\times D}$, where D is the embedding dimension.

We parameterize the conditional prior and posterior as diagonal Gaussians:

$$p_\theta(z_0 \mid X) = \mathcal{N}\big(\mu_p(X),\ \mathrm{diag}(\sigma_p^2(X))\big)\,, \tag{5}$$

$$q_\phi(z_0 \mid X, Y) = \mathcal{N}\big(\mu_q(X,Y),\ \mathrm{diag}(\sigma_q^2(X,Y))\big)\,, \tag{6}$$

where $\mu_p(X), \sigma_p(X) \in \mathbb{R}^{C\times d}$ are the prior mean and prior standard deviation, and $\mu_q(X,Y), \sigma_q(X,Y) \in \mathbb{R}^{C\times d}$ are the corresponding posterior parameters. We sample using the reparameterization trick:

$$z_0 = \mu_q(X,Y) + \sigma_q(X,Y) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

Let $\mathrm{Dec}_\theta(\cdot,\cdot)$ be a decoder that outputs the full horizon forecast in a single pass:

$$\hat{Y} = \mathrm{Dec}_\theta(z, X), \quad \hat{Y} \in \mathbb{R}^{H\times C}. \tag{8}$$

We refine the sampled latent via a context-conditioned reparameterization:

$$z = f_\theta(z_0; X), \tag{9}$$

which augments the latent representation while keeping the sampling interface and the single-pass decoder unchanged. We optimize a reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ and a KL regularizer. To improve readability in two column format, we define

$$\mathcal{L}_{\mathrm{KL}} \triangleq \mathrm{KL}(q_\phi(z_0 \mid X, Y) \,\|\, p_\theta(z_0 \mid X))\,, \tag{10}$$

and write the cVAE objective as

$$\mathcal{L}_{\mathrm{cVAE}} = \mathcal{L}_{\mathrm{rec}}(\hat{Y}, Y) + \lambda_{\mathrm{KL}}\,\mathcal{L}_{\mathrm{KL}}, \tag{11}$$

where $\lambda_{\mathrm{KL}} \geq 0$ is a weighting coefficient, commonly annealed for training stability.

4.2 Truncated Latent Diffusion

Let $\{z_t\}_{t=0}^{T}$ denote the latent diffusion trajectory, where z_0 is the clean latent and z_t is the noisy latent at step t. We use a truncation depth $T' \in \{1,\dots,T\}$ and only apply diffusion for $t \in \{1,\dots,T'\}$. This is the main efficiency device, since refinement is performed by a small number of steps in latent space rather than by many steps in observation space.

To integrate heteroscedastic uncertainty into diffusion, we set the forward noise scale to the conditional prior standard deviation $\sigma_p(X)$. This admits an equivalent context-whitened view (Appendix A.2). Define the context dependent variance

$$g(X) \triangleq \sigma_p^2(X). \tag{12}$$

We define the forward noising distribution as

$$q(z_t \mid z_0, X) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, z_0,\ (1-\bar{\alpha}_t)\,\mathrm{diag}\big(g(X)\big)\big)\,, \tag{13}$$

which admits the sampling form

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\,\sigma_p(X) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{14}$$

When $\sigma_p(X) \equiv \mathbf{1}$, Eq. (14) reduces to the stationary latent diffusion used in standard formulations.

Let $\epsilon_\theta(\cdot)$ be a denoiser that predicts the injected noise from (z_t, X, t). We optionally condition on $\log \sigma_p(X)$ for numerical stability, but this is not required by the formulation. We define

$$\hat{\epsilon} = \epsilon_\theta\big(z_t, X, t, \log \sigma_p(X)\big), \tag{15}$$

and train the denoiser using a noise matching objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\big(z_t, X, t, \log \sigma_p(X)\big)\right\|_2^2\right], \tag{16}$$

where t is sampled uniformly from $\{1, \ldots, T'\}$ and $\epsilon \sim \mathcal{N}(0, I)$.

Given z_t and $\hat{\epsilon}$, we estimate the clean latent as

$$\hat{z}_0 = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\,\sigma_p(X) \odot \hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}. \tag{17}$$

At inference, we initialize the refinement chain at $t_0 = T'$ using an anchored distribution centered at the prior mean:

$$z_{t_0} = \sqrt{\bar{\alpha}_{t_0}}\,\mu_p(X) + \sqrt{1-\bar{\alpha}_{t_0}}\,\sigma_p(X) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{18}$$

We then apply deterministic DDIM style updates for $t = t_0, t_0 - 1, \ldots, 2$:

$$\begin{aligned} z_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\hat{z}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\sigma_p(X) \odot \\ \epsilon_\theta\big(z_t, X, t, \log \sigma_p(X)\big), \end{aligned} \tag{19}$$

where $\hat{z}_0$ is computed by Eq. (17). After refinement, we decode once to obtain the forecast $\hat{Y} = \mathrm{Dec}_\theta(z_0, X)$.

The context dependent scale $\sigma_p(X)$ appears in the forward distribution of Eq. (13), which changes the induced reverse dynamics and the sampling distribution of z_0. Therefore heteroscedastic uncertainty is integrated into the generative mechanism rather than introduced by an auxiliary weighting scheme.

4.3 Test-Time Score Distillation

Multi-sample generation often improves probabilistic forecasts, but selecting the best sample by comparing with ground truth is infeasible at inference time. This creates a train–test mismatch: training can leverage the task loss to determine the best candidate, while test-time selection must rely solely on the inputs. We resolve this mismatch with *test-time score distillation* by distilling the training-time loss-based selection behavior into a scorer that ranks candidates using only (z_0, X).

Given K refined latent candidates $\{z_0^{(k)}\}_{k=1}^K$ and decoded forecasts $\{\hat{Y}^{(k)}\}_{k=1}^K$, we define a task loss $\ell(\hat{Y}, Y)$ such as mean squared error. The optimal selection index is

$$k^\star = \arg\min_{k \in \{1,\ldots,K\}} \ell(\hat{Y}^{(k)}, Y). \tag{20}$$

Let $s_\psi(z_0, X) \in \mathbb{R}$ be a scorer distillation network with parameters ψ that outputs a scalar score for a candidate latent. We compute scores $\{s_\psi(z_0^{(k)}, X)\}_{k=1}^K$ and train the scorer using a classification objective:

$$\mathcal{L}_{\text{score}} = \mathrm{CE}\Big(\{s_\psi(z_0^{(k)}, X)\}_{k=1}^K, k^\star\Big), \tag{21}$$

where CE denotes the cross entropy loss after applying a softmax over the K scores. At inference, we select

$$\hat{k} = \arg\max_{k \in \{1,\ldots,K\}} s_\psi(z_0^{(k)}, X), \tag{22}$$

and output $\hat{Y} = \mathrm{Dec}_\theta(z_0^{(\hat{k})}, X)$.

4.4 Joint Learning Objective

We train the model by minimizing a weighted sum of four terms, each corresponding to one component in our framework: (i) a reconstruction term for accurate forecasting, (ii) a variational regularization term for latent distribution learning, (iii) a latent diffusion noise matching term for refinement, and (iv) a scorer distillation term for test-time sample selection. For each training pair (X, Y), we define the following losses.

Table 1: Multivariate long-term forecasting results with horizon $H \in \{96, 192, 336, 720\}$ and fixed lookback window length $L = 96$. Results are averaged from all prediction horizons. (Red: Best, Blue: Second Best)

Models	**ToLD** Ours		TimePro (2025)		S-Mamba (2025)		SOFTS (2024)		iTransformer (2024)		PatchTST (2023)		Crossformer (2023)		TiDE (2023)		TimesNet (2023)		DLinear (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	**0.168**	0.263	0.169	**0.262**	0.170	0.265	0.174	0.264	0.178	0.270	0.189	0.276	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300
Exchange	**0.348**	**0.398**	0.352	0.399	0.367	0.408	0.361	0.402	0.360	0.403	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414
Weather	**0.246**	**0.272**	0.251	0.276	0.251	0.276	0.255	0.278	0.258	0.278	0.256	0.279	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317
Solar-Energy	**0.229**	**0.248**	0.232	0.266	0.240	0.273	**0.229**	0.256	0.233	0.262	0.236	0.266	0.641	0.639	0.347	0.417	0.301	0.319	0.330	0.401
ETTm1	**0.384**	**0.393**	0.391	0.400	0.398	0.405	0.393	0.403	0.407	0.410	0.396	0.406	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407
ETTm2	**0.269**	**0.314**	0.281	0.326	0.288	0.332	0.287	0.330	0.288	0.332	0.287	0.330	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401
ETTh1	**0.428**	**0.431**	0.438	0.438	0.455	0.450	0.449	0.442	0.454	0.447	0.453	0.446	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452
ETTh2	**0.357**	**0.386**	0.377	0.403	0.381	0.405	0.373	0.400	0.383	0.407	0.385	0.410	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515

Reconstruction loss. Given a refined latent sample z and decoder $\mathrm{Dec}_\theta(\cdot,\cdot)$, the forecast is $\hat{Y} = \mathrm{Dec}_\theta(z, X)$. In our implementation we use the mean squared error

$$\mathcal{L}_{\mathrm{rec}}(\hat{Y}, Y) \triangleq \frac{1}{HC} \left\| \hat{Y} - Y \right\|_F^2, \tag{23}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. Other proper scoring rules such as negative log-likelihood can be used when an explicit likelihood model is available.

Latent diffusion refinement loss. To train the denoiser ϵ_θ, we sample a diffusion step $t \sim \mathrm{Uniform}(\{1, \dots, T'\})$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, then construct $z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\, \sigma_p(X) \odot \epsilon$ as in Eq. (14). We minimize the noise-matching loss

$$\mathcal{L}_{\mathrm{diff}}(X, Y) \triangleq \mathbb{E}_{t,\epsilon} \left[\frac{1}{Cd} \left\| \epsilon - \epsilon_\theta\big(z_t, X, t, \log \sigma_p(X)\big) \right\|_F^2 \right], \tag{24}$$

where C is the number of variables and d is the latent dimension. This term directly trains ϵ_θ and enables truncated DDIM refinement at inference.

Score distillation loss. When multi-sample training is enabled, we generate K refined candidates $\{z_0^{(k)}\}_{k=1}^K$ and decode $\hat{Y}^{(k)} = \mathrm{Dec}_\theta(z_0^{(k)}, X)$. We define the optimal selection index using a task loss ℓ (MSE in our implementation):

$$k^\star \triangleq \arg \min_{k \in \{1,\dots,K\}} \ell\Big(\hat{Y}^{(k)}, Y\Big), \tag{25}$$

$$\ell(\hat{Y}, Y) \triangleq \frac{1}{HC} \|\hat{Y} - Y\|_F^2. \tag{26}$$

Let the scorer output logits $r^{(k)} = s_\psi(z_0^{(k)}, X)$. We distill score selection by a cross-entropy loss

$$\mathcal{L}_{\mathrm{score}}(X, Y) \triangleq -\log \frac{\exp(r^{(k^\star)})}{\sum_{j=1}^K \exp(r^{(j)})}. \tag{27}$$

If the scorer is disabled, we set its weight to zero.

The overall training loss is

$$\mathcal{L}(X, Y) = \mathcal{L}_{\mathrm{cVAE}} + \lambda_{\mathrm{diff}}\, \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{score}}\, \mathcal{L}_{\mathrm{score}}, \tag{28}$$

where $\lambda_{\mathrm{diff}} \geq 0$ and $\lambda_{\mathrm{score}} \geq 0$ control the contributions of latent refinement and score distillation, respectively.

5 Experiment

5.1 Setup

Datasets. We evaluate our method on eight widely-used real-world multivariate time-series datasets with diverse temporal patterns and noise levels, including Electricity (ECL), four ETT subsets (ETTh1, ETTh2, ETTm1, ETTm2), Exchange (EXG), Solar-Energy (Solar), and Weather. Following the

Table 2: MSE/CRPS results on seven real-world datasets. Lower is better. Bold indicates the best result.

Model	ETTh1		ETTh2		ETTm1		ETTm2		ECL		EXG		Solar	
	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS	MSE	CRPS
TimeGrad[23]	0.813	0.606	1.496	1.212	0.831	0.647	0.967	0.775	0.504	0.397	1.058	0.826	0.446	0.293
CSDI [28]	0.708	0.492	0.900	0.647	0.752	0.524	1.069	0.817	0.822	0.577	1.081	0.855	0.675	0.432
TimeDiff [26]	0.479	0.465	0.485	0.471	0.477	0.464	0.333	0.316	0.764	0.750	0.446	0.433	0.713	0.700
DiffusionTS [33]	0.774	0.603	1.411	1.168	0.744	0.574	1.232	1.035	0.856	0.633	1.564	1.251	0.740	0.470
TMDM [11]	0.607	0.452	0.490	0.383	0.455	0.375	0.395	0.289	0.359	0.461	0.430	0.336	0.316	0.350
NsDiff [32]	0.523	0.392	0.490	0.358	0.455	0.346	0.352	0.256	0.306	0.290	0.412	0.324	0.307	0.300
ToLD (Ours)	**0.428**	**0.367**	**0.357**	**0.338**	**0.384**	**0.320**	**0.269**	**0.228**	**0.168**	**0.234**	**0.348**	**0.318**	**0.229**	**0.227**

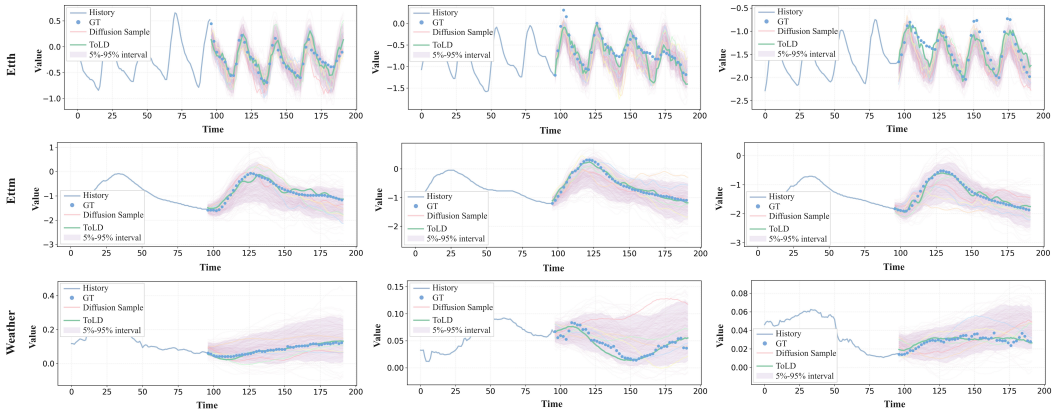


Figure 3: Examples of long-term time series forecasting on the ETTh (top row) and Exchange (bottom row) datasets. The green curves represent the predictions of ToLD, while the blue dots denote the ground truth.

standard long-term forecasting protocol [10, 12], the ETT datasets are split by {12/4/4} months for train/validation/test, while the other datasets are split by a {7:1:2} ratio. Detailed statistics and preprocessing are provided in the Appendix B.1.

Baselines. We compare against a broad set of strong baselines, covering both deterministic forecasters and probabilistic/generative models. Specifically, we include Linear/MLP-based methods (SOFTS [6], TiDE [3], TimesNet [31], DLinear [34]) and Transformer-based methods (iTransformer [16], PatchTST [20], Crossformer [37], Stationary [15]). When available, we additionally report results of recent competitive models such as TimePro [17] and S-Mamba[30]. For probabilistic forecasting, we compare with representative diffusion-based and score-based approaches including TimeGrad[23], CSDI[28], TimeDiff[26], DiffusionTS[33], TMDM[11], and NsDiff [32]. Notably, TMDM denoises from $\mathcal{N}(f_\phi(X), I)$, while most diffusion-based baselines start from $\mathcal{N}(0, I)$.

Evaluation Metrics. We report both point and distributional forecasting quality using a unified metric suite: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Continuous Ranked Probability Score (CRPS). Lower values indicate better performance for all metrics. Results are averaged over all prediction horizons unless otherwise stated. Detailed definitions and implementation are provided in the Appendix B.3.

5.2 Main Results

The comprehensive empirical evaluation presented in Table 1 and Table 2 substantiates the superior performance of **ToLD** across a wide array of multivariate time series benchmarks. In the realm of deterministic forecasting, our method consistently surpasses strong baselines and yields an average accuracy improvement of approximately 4.3% in error metrics. This consistent gain across varying prediction horizons suggests that the model effectively mitigates the error accumulation typically observed in long-term forecasting tasks. Beyond point accuracy, **ToLD** provides consistent improvements in probabilistic forecasting metrics while maintaining substantially lower inference

Table 4: Ablation study on various datasets. Results are averaged over forecast horizons $H \in \{96, 192, 336, 720\}$ with fixed lookback $L = 96$. We report MSE/MAE (point accuracy) and CRPS (probabilistic quality).

Variant	ETTh1			ETTh2			ETTm1			ETTm2			Weather		
	MSE↓	MAE↓	CRPS↓	MSE↓	MAE↓	CRPS↓	MSE↓	MAE↓	CRPS↓	MSE↓	MAE↓	CRPS↓	MSE↓	MAE↓	CRPS↓
Full (ToLD)	0.428	0.431	0.367	0.357	0.386	0.338	0.384	0.393	0.320	0.269	0.314	0.228	0.246	0.272	0.259
w/o Latent Diff	0.437	0.438	0.381	0.358	0.388	0.351	0.386	0.397	0.329	0.274	0.321	0.243	0.248	0.274	0.261
HomoDiff	0.433	0.432	0.375	0.359	0.391	0.341	0.388	0.395	0.332	0.276	0.323	0.249	0.249	0.273	0.263
w/o Anchor	0.429	0.432	0.385	0.358	0.395	0.344	0.389	0.395	0.338	0.270	0.318	0.236	0.247	0.275	0.262
w/o DDIM	0.430	0.433	0.386	0.374	0.397	0.342	0.389	0.398	0.335	0.278	0.324	0.252	0.250	0.274	0.264
w/o Token-wise	0.433	0.436	0.379	0.369	0.398	0.339	0.389	0.397	0.336	0.275	0.321	0.246	0.247	0.274	0.264
w/o Truncation	0.434	0.435	0.370	0.361	0.388	0.340	0.387	0.394	0.328	0.273	0.320	0.241	0.249	0.275	0.261
w/o score distillation (random)	0.431	0.434	0.377	0.371	0.402	0.343	0.390	0.399	0.337	0.277	0.322	0.250	0.248	0.273	0.263
w/o score distillation (mean)	0.432	0.435	0.388	0.363	0.392	0.342	0.386	0.395	0.333	0.272	0.319	0.239	0.250	0.275	0.262

cost. The results indicate a substantial reduction in the CRPS metric of roughly 19.3% compared to state-of-the-art generative methods. This significant advancement reveals that **ToLD** does not merely approximate the conditional mean but suggesting improved distributional fit under the evaluated sampling protocol. Overall, **ToLD** successfully bridges the gap between precise trajectory prediction and reliable uncertainty quantification.

To better illustrate the performance of both forecast quality and uncertainty estimation, Fig. 3 visualizes representative examples from the ETTh, ETTm and Weather datasets. These visualizations reveal that even when there is a noticeable shift between the historical context and the forecasted horizon, **ToLD** consistently generates diverse sample trajectories. Among these, the median prediction (highlighted in green) aligns closely with the ground truth (represented by the blue dots), demonstrating the model's ability to track the true future distribution. In addition, the empirical 5–95% uncertainty interval is well-calibrated, encompassing most of the future observations and reflecting time-varying uncertainty throughout the forecast horizon. This indicates that **ToLD** not only provides accurate point predictions but also delivers reliable and dynamically adaptive uncertainty estimates, effectively capturing the evolving stochasticity of the underlying time series and offering practitioners a more comprehensive understanding of forecast confidence across the entire prediction horizon. More examples and analysis of uncertainty modeling can be found in Appendix C.

Table 3: The inference time (in ms) of various time series generative models with different prediction lengths on the univariate ETTh1 dataset. Baselines are taken from mr-Diff [27]. We report **ToLD** in ms/series computed from our ms/batch divided by batch size (32).

Method	Inference time (ms)			
	H=96	H=192	H=336	H=720
ToLD (samples = 1)	**5.14**	**5.31**	**5.44**	**5.25**
ToLD (samples = 50)	5.39	6.26	5.87	6.62
ToLD (samples = 200)	5.57	6.29	5.97	6.71
mr-Diff	8.3	9.8	11.9	21.6
TimeGrad	870.2	1854.5	3119.7	6724.1
TimeDiff	16.2	17.6	26.5	34.6
CSDI	90.4	142.8	398.9	513.1
SSSD	418.6	645.3	1054.2	2516.9

5.3 Computation Efficiency

We evaluate inference efficiency on ETTh1 with varying horizons $H \in \{96, 192, 336, 720\}$. As shown in Table 3, **ToLD** achieves *near-constant latency* across horizons: the per-series inference time stays around $5 \sim 6.7$ ms even when H increases to 720. Moreover, the latency grows only marginally with the sampling budget (from samples $= 1$ to 200), indicating that most computation is amortized by one-pass decoding and lightweight truncated latent refinement. In contrast, iterative diffusion baselines exhibit substantially higher cost and stronger scaling with H or non-negligible overhead, highlighting the practicality of our design under long-term and multi-sample evaluation. More detailed efficiency results are provided in Appendix C.

Table 5: Sensitivity to refinement steps T' and sample size K on ETTh2.

T'	K	MSE↓	MAE↓	CRPS ↓	Infer.Time ↓
1	1	0.369	0.390	0.342	3.95
2	1	0.360	0.389	0.340	3.83
4	1	0.379	0.395	0.355	4.33
8	1	0.381	0.401	0.358	4.48
2	10	0.375	0.402	0.372	4.28
2	50	0.367	0.397	0.361	5.61
2	200	**0.357**	**0.386**	**0.338**	**6.52**
4	10	0.375	0.404	0.371	4.20
4	50	0.367	0.395	0.361	5.41
4	200	0.372	0.396	0.358	8.01

5.4 Ablation Experiments

Importance of Each Component. Table 4 shows that the full **ToLD** consistently achieves the best overall performance. Removing latent diffusion or using homoscedastic noise increases CRPS, suggesting that heteroscedastic latent refinement is important for uncertainty modeling. Removing anchored initialization, DDIM-style sampling, or token-wise refinement also degrades MSE/MAE and CRPS, indicating that these components jointly improve both accuracy and probabilistic quality.

Sensitivity to Refinement and Sampling. Table 5 studies the effects of refinement depth T' and sample size K on ETTh2. Increasing T' beyond a small value yields limited gains while increasing inference cost. By contrast, a larger sampling budget generally improves MSE, MAE, and CRPS with only moderate latency growth. These results suggest that shallow latent refinement combined with multi-sample selection provides an effective accuracy–efficiency trade-off.

6 Conclusion

In this work, we introduced ToLD, a latency-aware generative forecasting framework that unifies high-fidelity uncertainty modeling with practical long-term deployment. By operating in an anchored latent space and applying a truncated diffusion refinement strategy, ToLD achieves near-constant inference latency while preserving strong generative expressiveness. The proposed score-distillation ranking mechanism further mitigates the train–test mismatch in multi-sample forecasting, enabling reliable test-time selection without access to future targets. Across a broad set of real-world benchmarks, ToLD consistently delivers improvements in both point accuracy and uncertainty metrics, while maintaining substantially lower and more stable inference cost than iterative diffusion baselines. These results highlight the framework's suitability for real-time, long-term forecasting scenarios. Overall, ToLD provides a generalizable foundation for efficient, uncertainty-aware time-series forecasting under practical latency constraints.

7 Acknowledgments

This work was supported in part by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM504, and National Natural Science Foundation of China No.62302381, No.52441602. The Authors are with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, China.

References

[1] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022.

[2] Borui Cai, Shuiqiao Yang, Longxiang Gao, and Yong Xiang. Hybrid variational autoencoder for time series forecasting. *Knowledge-Based Systems*, 281:111079, 2023.

[3] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

[4] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[6] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[7] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[9] Jiaxi Hu, Disen Lan, Ziyu Zhou, Qingsong Wen, and Yuxuan Liang. Time-ssm: Simplifying and unifying state space models for time series forecasting. *arXiv preprint arXiv:2405.16312*, 2024.

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

[11] Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024.

[13] Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

[14] Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. Time-series anomaly detection: Overview and new trends. *Proceedings of the VLDB Endowment (PVLDB)*, 17(12): 4229–4232, 2024.

[15] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35:9881–9893, 2022.

[16] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2023.

[17] Xiaowen Ma, Zhenliang Ni, Shuai Xiao, and Xinghao Chen. Timepro: Efficient multivariate long-term time series forecasting with variable-and time-aware hyper-state. *arXiv preprint arXiv:2505.20774*, 2025.

[18] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8983–8991, 2021.

[19] Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29, 2023.

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

[21] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9):3760–3765, 2019.

[22] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. *arXiv preprint arXiv:2412.10859*, 2024.

[23] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pages 8857–8868. PMLR, 2021.

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[25] 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.

[26] Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. In *International Conference on Machine Learning*, pages 31016–31029. PMLR, 2023.

[27] Lifeng Shen, Weiyu Chen, and James Kwok. Multi-resolution diffusion models for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

[29] Chengsen Wang, Zirui Zhuang, Qi Qi, Jingyu Wang, Xingyu Wang, Haifeng Sun, and Jianxin Liao. Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection. *Advances in neural information processing systems*, 36:10758–10774, 2023.

[30] Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025.

[31] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[32] Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. *arXiv preprint arXiv:2505.04278*, 2025.

[33] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. *arXiv preprint arXiv:2403.01742*, 2024.

[34] 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, pages 11121–11128, 2023.

[35] Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024.

[36] Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. Probts: Benchmarking point and distributional forecasting across diverse prediction horizons. *Advances in Neural Information Processing Systems*, 37:48045–48082, 2024.

[37] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=vSVLM2j9eie`.

[38] 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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 core contribution as an efficient probabilistic long-term time-series forecasting framework based on tokenized conditional latent modeling, truncated latent diffusion, and test-time score distillation. These claims are supported by the method description, main experimental results, efficiency analysis, calibration evaluation, and ablation studies. The scope is limited to probabilistic long-term time-series forecasting on the evaluated benchmark datasets, rather than claiming universal improvements across all forecasting settings.

   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: The paper discusses limitations of ToLD, including the potential dependence of test-time score distillation on the training distribution, possible robustness issues under severe distribution shifts, and additional memory or sampling overhead when scaling to extremely high-dimensional or very long-horizon settings. The paper also clarifies that the empirical conclusions are based on standard long-term forecasting benchmarks and should be further validated in broader deployment scenarios.

   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: The paper provides the assumptions and derivations for the proposed latent diffusion formulation in the appendix, including the context-dependent forward noising process and its equivalent whitened view. These derivations support the methodological claims about heteroscedastic latent refinement, while the paper does not rely on additional unproven theoretical guarantees.

   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: The paper and appendix describe the datasets, data splits, forecasting horizons, lookback window, model architecture, training objective, optimization settings, evaluation metrics, ablation variants, and inference protocol needed to reproduce the main results. Additional implementation details, including the latent dimensions, refinement steps, sampling budget, and training configuration, are provided in the appendix and supplemental material.

   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 paper provides an anonymized implementation repository in the supplemental material, together with instructions for environment setup, data preparation, training, evaluation, and reproduction of the main experiments. The released materials include training scripts, configuration files, evaluation scripts, and guidance for reproducing the reported results on public benchmark datasets. Open-source code link: `https://anonymous.4open.science/r/ToLD-C801/`.

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: The paper specifies the evaluated datasets, train/validation/test splits, prediction horizons, lookback length, evaluation metrics, optimizer settings, loss terms, sampling budget, refinement depth, and major hyperparameters. The appendix further provides implementation details for the embedding module, conditional prior and posterior, decoder, denoiser, scorer, and ablation settings.

   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: The paper reports mean $\pm$ standard deviation over multiple independent runs with different random seeds for key experiments, including mechanism-attribution, calibration, and robustness analyses. The paper also clarifies that the reported deviations reflect run-to-run variation rather than dataset-level uncertainty or confidence intervals.

   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: The paper reports the main compute resources used for training and evaluation, including the GPU type and practical runtime configuration. It also provides inference-time and memory comparisons for ToLD and representative baselines, allowing readers to estimate the computational scale required to reproduce the main experiments.

   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: To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics. The work uses public benchmark time-series datasets for methodological research and does not involve deceptive practices, unsafe deployment claims, or undisclosed human-subject experimentation by the authors.

   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: The paper discusses the potential benefits of efficient and uncertainty-aware long-term forecasting for applications such as energy, transportation, healthcare, and finance, where calibrated uncertainty can support risk-aware decision-making. It also acknowledges potential negative impacts, including risks from over-reliance on automated forecasts, poor

calibration under distribution shift, and possible misuse in high-stakes decision pipelines without appropriate human oversight.

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 high-risk generative models, scraped datasets, or sensitive assets that would require special safeguards for responsible release. The work focuses on probabilistic time-series forecasting methods evaluated on existing public benchmark datasets.

    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: The paper cites the original datasets and prior forecasting methods used for comparison. The supplemental material provides corresponding access references where applicable, and the experiments are conducted on public benchmark datasets and properly attributed baseline implementations.

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: [N/A]

    Justification: The paper does not introduce a new dataset, benchmark, or externally released pretrained model asset. The main contribution is a methodological framework and implementation code evaluated on existing public datasets.

    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: The paper does not involve crowdsourcing experiments or new research with human subjects conducted by the authors. It uses pre-existing public benchmark time-series datasets.

    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: The paper does not report new human-subject data collection, intervention, or crowdsourcing conducted by the authors. Accordingly, IRB approval or equivalent review is not applicable to the research presented here.

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: LLMs are not an important, original, or non-standard component of the proposed method, experiments, or scientific claims. Any possible use of LLMs for writing, editing, or formatting assistance does not affect the core methodology and therefore does not require declaration under this policy.

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.

A Derivation and Theoretical Justification

A.1 Definitions and Notation

We consider multivariate long-term forecasting with context–target pairs. Let $\mathbf{X} \in \mathbb{R}^{L\times C}$ be the historical context (lookback length L, number of variables C), and $\mathbf{Y} \in \mathbb{R}^{H\times C}$ be the future horizon (length H).

We use token-wise latents $\mathbf{z}_0 \in \mathbb{R}^{C\times d}$ with latent dimension d. The conditional prior and variational posterior are diagonal Gaussians: $\boldsymbol{\mu}_p(\mathbf{X}), \boldsymbol{\sigma}_p(\mathbf{X}) \in \mathbb{R}^{C\times d}$ and $\boldsymbol{\mu}_q(\mathbf{X}, \mathbf{Y}), \boldsymbol{\sigma}_q(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{C\times d}$.

Let $\mathrm{Diag}(\cdot)$ map a tensor in $\mathbb{R}^{C\times d}$ to a diagonal covariance in $\mathbb{R}^{(Cd)\times(Cd)}$ (after flattening tokens). Define the context-dependent latent variance

$$\boldsymbol{v}(\mathbf{X}) \triangleq \boldsymbol{\sigma}_p^2(\mathbf{X}) \in \mathbb{R}^{C\times d}, \tag{29}$$

where the square is elementwise.

For truncated diffusion refinement, let $T \in \mathbb{N}$ be the full diffusion depth and let $T' \in \{1, \dots, T\}$ be the truncation depth used in our refinement. We use a schedule $\{\beta_t\}_{t=1}^{T'}$ with $\beta_t \in (0, 1)$ and define

$$\alpha_t \triangleq 1 - \beta_t, \qquad \bar{\alpha}_t \triangleq \prod_{i=1}^{t} \alpha_i, \qquad t \in \{1, \dots, T'\}. \tag{30}$$

Let $\odot$ and $\oslash$ denote elementwise multiplication and division, respectively. We use two independent standard Gaussians:

$$\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{31}$$

where $\mathbf{I}$ is the identity matrix of dimension Cd. We reserve $\boldsymbol{\xi}$ for VAE reparameterization and $\boldsymbol{\varepsilon}$ for diffusion noise.

A.2 Closed-form Forward Process and Context Whitening

Forward transition. We define the heteroscedastic latent forward process:

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{X}) = \mathcal{N}\Big(\sqrt{\alpha_t}\, \mathbf{z}_{t-1}, (1 - \alpha_t)\, \mathrm{Diag}(\boldsymbol{v}(\mathbf{X}))\Big). \tag{32}$$

Closed-form marginal. By unrolling the Markov chain (same algebra as standard DDPM but with scaled diagonal covariance), we obtain the closed-form marginal:

$$q(\mathbf{z}_t \mid \mathbf{z}_0, \mathbf{X}) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_t}\, \mathbf{z}_0, (1 - \bar{\alpha}_t)\, \mathrm{Diag}(\boldsymbol{v}(\mathbf{X}))\Big). \tag{33}$$

Sampling form. Equivalently,

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\sigma}_p(\mathbf{X}) \odot \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{34}$$

Context whitening. Define the whitened latent variables

$$\widetilde{\mathbf{z}}_t \triangleq \mathbf{z}_t \oslash \boldsymbol{\sigma}_p(\mathbf{X}), \qquad \widetilde{\mathbf{z}}_0 \triangleq \mathbf{z}_0 \oslash \boldsymbol{\sigma}_p(\mathbf{X}). \tag{35}$$

Dividing the sampling form by $\boldsymbol{\sigma}_p(\mathbf{X})$ yields

$$\widetilde{\mathbf{z}}_t = \sqrt{\bar{\alpha}_t}\, \widetilde{\mathbf{z}}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\varepsilon}. \tag{36}$$

Thus, conditioned on $\mathbf{X}$, the heteroscedastic diffusion in $\mathbf{z}$-space becomes a *standard* diffusion in $\widetilde{\mathbf{z}}$-space.

A.3 Reverse Posterior Distribution (Closed-form)

The reverse posterior $q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{z}_0, \mathbf{X})$ is Gaussian because it is proportional to a product of Gaussians:

$$q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{z}_0, \mathbf{X}) \propto q(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \mathbf{X})\, q(\mathbf{z}_{t-1} \mid \mathbf{z}_0, \mathbf{X}). \tag{37}$$

Using the two Gaussian factors from Appendix A.2, and standard Gaussian product identities, the posterior takes the form

$$q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{z}_0, \mathbf{X}) = \mathcal{N}\Big(\boldsymbol{\mu}_{t-1}(\mathbf{z}_t, \mathbf{z}_0, \mathbf{X}), \mathrm{Diag}(\boldsymbol{\Sigma}_{t-1}(\mathbf{X}))\Big), \tag{38}$$

where the diagonal covariance is

$$\boldsymbol{\Sigma}_{t-1}(\mathbf{X}) = \frac{(1-\bar{\alpha}_{t-1})(1-\alpha_t)}{1-\bar{\alpha}_t}\, \boldsymbol{v}(\mathbf{X}), \tag{39}$$

and the posterior mean is

$$\boldsymbol{\mu}_{t-1}(\mathbf{z}_t, \mathbf{z}_0, \mathbf{X}) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\, \mathbf{z}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\, \mathbf{z}_0. \tag{40}$$

This mirrors the standard DDPM posterior, except that the covariance is scaled by the context-dependent $\boldsymbol{v}(\mathbf{X})$.

A.4 Truncated Deterministic Refinement (DDIM Subsampling View)

Our inference uses a *deterministic* DDIM-style refinement in latent space and truncates the chain to T' steps. The key justification is: in the whitened space, our refinement is exactly the standard DDIM ODE sampling with a reduced set of time indices.

DDIM in whitened space. Let $\widehat{\boldsymbol{\varepsilon}}_\theta(\widetilde{\mathbf{z}}_t, \mathbf{X}, t)$ be a predictor of $\boldsymbol{\varepsilon}$ in whitened space. Define the clean-latent estimator

$$\widehat{\widetilde{\mathbf{z}}}_0 = \frac{\widetilde{\mathbf{z}}_t - \sqrt{1-\bar{\alpha}_t}\, \widehat{\boldsymbol{\varepsilon}}_\theta(\widetilde{\mathbf{z}}_t, \mathbf{X}, t)}{\sqrt{\bar{\alpha}_t}}. \tag{41}$$

The DDIM deterministic update from t to $t-1$ is

$$\widetilde{\mathbf{z}}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, \widehat{\widetilde{\mathbf{z}}}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\, \widehat{\boldsymbol{\varepsilon}}_\theta(\widetilde{\mathbf{z}}_t, \mathbf{X}, t). \tag{42}$$

Mapping back to heteroscedastic latent space. Using $\mathbf{z}_t = \boldsymbol{\sigma}_p(\mathbf{X}) \odot \widetilde{\mathbf{z}}_t$, the corresponding update in original latent space becomes

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, \widehat{\mathbf{z}}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\, \boldsymbol{\sigma}_p(\mathbf{X}) \odot \widehat{\boldsymbol{\varepsilon}}_\theta\Big(\mathbf{z}_t \oslash \boldsymbol{\sigma}_p(\mathbf{X}), \mathbf{X}, t\Big), \tag{43}$$

which is exactly the form used by our method.

Exact subsampling consistency under perfect predictor. Assume $\widehat{\boldsymbol{\varepsilon}}_\theta(\widetilde{\mathbf{z}}_t, \mathbf{X}, t) = \mathbb{E}[\boldsymbol{\varepsilon} \mid \widetilde{\mathbf{z}}_t, \mathbf{X}]$ for all t (Bayes-optimal predictor). Then the deterministic DDIM trajectory in whitened space is a unique solution to a well-defined probability-flow ODE, and sampling with any subsequence of time indices yields the same terminal distribution as the full deterministic path evaluated at the same end time.

Under the Bayes-optimal predictor, the DDIM update corresponds to an ODE discretization of the probability-flow dynamics (standard diffusion theory). Evaluating the ODE solution at a coarser set of times is *exactly* a form of subsampling of the same deterministic flow; therefore truncation does not change the underlying target flow, but only changes the numerical integration budget. In our case, this argument holds in whitened space, and the mapping $\mathbf{z}_t = \boldsymbol{\sigma}_p(\mathbf{X}) \odot \widetilde{\mathbf{z}}_t$ is an invertible linear transform conditioned on $\mathbf{X}$, which preserves distributional correctness when mapping back.

When the predictor is imperfect, truncation introduces an integration/approximation error; however, (i) truncation happens in a compact latent space (dimension Cd instead of HC), and (ii) the initialization (Appendix A.5) starts from a prior-anchored distribution, so only a small number of refinement steps is typically sufficient in practice.

A.5 Anchored Initialization as a Valid Forward Marginal

We initialize refinement at $t_0 \triangleq T'$ by

$$\mathbf{z}_{t_0} = \sqrt{\bar{\alpha}_{t_0}}\, \boldsymbol{\mu}_p(\mathbf{X}) + \sqrt{1-\bar{\alpha}_{t_0}}\, \boldsymbol{\sigma}_p(\mathbf{X}) \odot \boldsymbol{\varepsilon}_0, \qquad \boldsymbol{\varepsilon}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{44}$$

By Appendix A.2, this is exactly sampling from $q(\mathbf{z}_{t_0} \mid \mathbf{z}_0 = \boldsymbol{\mu}_p(\mathbf{X}), \mathbf{X})$. Hence, our truncated refinement chain starts from a *valid* forward marginal induced by the learned prior mean.

A.6 Optimality of the Noise-Matching Loss

We train the denoiser by minimizing an ℓ_2 noise-matching objective in the standardized (whitened) space:

$$\mathcal{L}_{\text{diff}} \triangleq \mathbb{E}_{t,\boldsymbol{\varepsilon}}\Big[\big\|\boldsymbol{\varepsilon} - \widehat{\boldsymbol{\varepsilon}}_\theta(\widetilde{\mathbf{z}}_t, \mathbf{X}, t)\big\|_2^2\Big]. \tag{45}$$

For any fixed $(\widetilde{\mathbf{z}}_t, \mathbf{X}, t)$, the minimizer of mean squared error is the conditional expectation:

$$\widehat{\boldsymbol{\varepsilon}}^\star_\theta(\widetilde{\mathbf{z}}_t, \mathbf{X}, t) = \mathbb{E}[\boldsymbol{\varepsilon} \mid \widetilde{\mathbf{z}}_t, \mathbf{X}, t]. \tag{46}$$

Therefore, $\mathcal{L}_{\text{diff}}$ is a statistically consistent objective for learning the Bayes-optimal noise predictor. This provides the key assumption used in Appendix A.4.

A.7 CVAE Objective (ELBO)

The conditional predictive model is

$$p_\theta(\mathbf{Y} \mid \mathbf{X}) = \int p_\theta(\mathbf{Y} \mid \mathbf{z}_0, \mathbf{X})\, p_\theta(\mathbf{z}_0 \mid \mathbf{X})\, d\mathbf{z}_0. \tag{47}$$

Using a variational posterior $q_\phi(\mathbf{z}_0 \mid \mathbf{X}, \mathbf{Y})$, we obtain the ELBO:

$$\log p_\theta(\mathbf{Y} \mid \mathbf{X}) \geq \mathbb{E}_{q_\phi}\big[\log p_\theta(\mathbf{Y} \mid \mathbf{z}_0, \mathbf{X})\big] - \mathrm{KL}\big(q_\phi(\mathbf{z}_0 \mid \mathbf{X}, \mathbf{Y}) \,\|\, p_\theta(\mathbf{z}_0 \mid \mathbf{X})\big). \tag{48}$$

Thus minimizing

$$\mathcal{L}_{\text{cVAE}} \triangleq \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\, \mathrm{KL}\big(q_\phi(\mathbf{z}_0 \mid \mathbf{X}, \mathbf{Y}) \,\|\, p_\theta(\mathbf{z}_0 \mid \mathbf{X})\big) \tag{49}$$

is equivalent to maximizing a (weighted) lower bound of $\log p_\theta(\mathbf{Y} \mid \mathbf{X})$.

A.8 Score Distillation as Consistent Selection Learning

Given K candidates $\{\mathbf{z}_0^{(k)}\}_{k=1}^K$ and decoded forecasts $\{\widehat{\mathbf{Y}}^{(k)}\}_{k=1}^K$, define an optimal selection index

$$k^\dagger \triangleq \arg\min_{k \in \{1,\dots,K\}} \ell(\widehat{\mathbf{Y}}^{(k)}, \mathbf{Y}), \tag{50}$$

where $\ell(\cdot,\cdot)$ is a task loss. Let $s_\psi(\mathbf{z}_0, \mathbf{X}) \in \mathbb{R}$ output logits and define the softmax probability

$$\pi_\psi(k \mid \mathbf{X}) \triangleq \frac{\exp(s_\psi(\mathbf{z}_0^{(k)}, \mathbf{X}))}{\sum_{j=1}^K \exp(s_\psi(\mathbf{z}_0^{(j)}, \mathbf{X}))}. \tag{51}$$

Training with cross-entropy

$$\mathcal{L}_{\text{score}} \triangleq -\log \pi_\psi(k^\dagger \mid \mathbf{X}) \tag{52}$$

is the maximum-likelihood objective for predicting the optimal-selected index from $\mathbf{X}$ and candidates. Under standard classification consistency assumptions, minimizing $\mathcal{L}_{\text{score}}$ yields a scorer that ranks candidates in agreement with the optimal rule, thereby reducing train–test mismatch when only a limited K is available.

B Reproducibility

B.1 Datasets

We evaluate our method-**ToLD**, on eight widely-used real-world multivariate time series datasets, which cover a wide range of uncertainty levels, from relatively stable to highly non-stationary, and from low to significant heteroscedasticity in various domains: Four ETT subsets (ETTh1, ETTh2, ETTm1, ETTm2) from electricity transformer operational data, including variables such as load and oil temperature, with ETTh sampled hourly and ETTm sampled every 15 minutes (2016/07–2018/07); Electricity, which records hourly electricity usage of 321 customers from 2012 to 2014; Exchange, which logs daily exchange rates of eight countries from 1990 to 2016; Weather, containing multi-dimensional meteorological observations for testing model robustness to seasonality and noisy fluctuations; Solar-Energy, a dataset from the National Renewable Energy Laboratory, containing solar power output data from multiple photovoltaic plants, exhibiting distinct daily cycles and weather-driven uncertainties. These datasets vary significantly in terms of dimensionality, sampling frequency, and uncertainty types, providing comprehensive coverage to evaluate the model's accuracy and calibration capabilities over long forecasting horizons.

B.2 Implementation Details.

To ensure consistency with existing literature and guarantee reproducibility, we strictly adhere to the standard evaluation protocol for multivariate long-term forecasting across all datasets unless otherwise specified. Specifically, we fix the historical lookback window length at $L = 96$ and evaluate model performance across four distinct prediction horizons, where $H \in \{96, 192, 336, 720\}$. Regarding the experimental environment, all models are implemented using the PyTorch framework. Training uses NVIDIA A800 GPUs, whereas controlled latency benchmarking uses one NVIDIA A6000. For the inference phase of probabilistic models, predictions are derived by drawing samples from the learned distribution. To balance computational efficiency with estimation accuracy, we utilize 200 Monte Carlo samples to approximate distributional metrics unless stated otherwise. Furthermore, to maintain a fair comparison, we adopt the official default hyperparameters recommended in the original papers or codebases for all baseline methods whenever applicable.

B.3 Metrics

Given an input context $\mathbf{X} \in \mathbb{R}^{L_{\text{ctx}} \times C_{\text{var}}}$ and future target $\mathbf{Y} \in \mathbb{R}^{H_{\text{pred}} \times C_{\text{var}}}$, our probabilistic forecaster outputs (i) a point forecast $\hat{\mathbf{Y}}$ and (ii) S_{samp} Monte-Carlo samples $\{\mathbf{Y}^{(s)}\}_{s=1}^{S_{\text{samp}}}$. For a test mini-batch of size B_{batch}, we denote the scalar ground truth as $y_{b,h,c}$ and the point forecast as $\hat{y}_{b,h,c}$, where $b \in \{1, \ldots, B_{\text{batch}}\}$ indexes the batch element, $h \in \{1, \ldots, H_{\text{pred}}\}$ indexes the horizon step, and $c \in \{1, \ldots, C_{\text{var}}\}$ indexes the variable/channel. The sampled prediction is $y^{(s)}_{b,h,c}$ for sample index $s \in \{1, \ldots, S_{\text{samp}}\}$.

MAE: Mean absolute error (MAE) is computed on the point forecast:

$$\text{MAE} = \frac{1}{B_{\text{batch}}\, H_{\text{pred}}\, C_{\text{var}}} \sum_{b=1}^{B_{\text{batch}}} \sum_{h=1}^{H_{\text{pred}}} \sum_{c=1}^{C_{\text{var}}} |\hat{y}_{b,h,c} - y_{b,h,c}|\,. \tag{53}$$

MSE: Mean squared error (MSE) is computed on the point forecast:

$$\text{MSE} = \frac{1}{B_{\text{batch}}\, H_{\text{pred}}\, C_{\text{var}}} \sum_{b=1}^{B_{\text{batch}}} \sum_{h=1}^{H_{\text{pred}}} \sum_{c=1}^{C_{\text{var}}} (\hat{y}_{b,h,c} - y_{b,h,c})^2\,. \tag{54}$$

CRPS (quantile-based approximation with constant extrapolation). The continuous ranked probability score (CRPS) evaluates the compatibility between a predictive CDF $F(\cdot)$ and an observation y:

$$\text{CRPS}(F, y) = \int_{-\infty}^{+\infty} \left(F(z) - \mathbb{I}\{y \le z\}\right)^2 dz, \tag{55}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

We compute CRPS via an equivalent quantile-integral form. Let $\tau \in (0, 1)$ be a quantile level and $Q(\tau)$ be the predictive quantile function. Then:

$$\text{CRPS}(Q, y) = \int_0^1 2\left(Q(\tau) - y\right)\left(\mathbb{I}\{y \le Q(\tau)\} - \tau\right) d\tau. \tag{56}$$

In practice, we choose J_{qt} quantile levels $\{\tau_j\}_{j=1}^{J_{\text{qt}}}$ with $\tau_1 < \cdots < \tau_{J_{\text{qt}}}$, and estimate quantiles from samples:

$$Q_j(b, h, c) = \text{Quantile}_{\tau_j}\left(\left\{y^{(s)}_{b,h,c}\right\}_{s=1}^{S_{\text{samp}}}\right). \tag{57}$$

We set a uniform step size $\Delta\tau = \tau_{j+1} - \tau_j$ (constant in our implementation).

We adopt constant extrapolation outside $[\tau_1, \tau_{J_{\text{qt}}}]$:

$$Q(\tau) = \begin{cases} Q_1, & \tau < \tau_1, \\ Q(\tau), & \tau_1 \le \tau \le \tau_{J_{\text{qt}}}, \\ Q_{J_{\text{qt}}}, & \tau > \tau_{J_{\text{qt}}}. \end{cases} \tag{58}$$

With this approximation, CRPS for each scalar (b, h, c) is computed as the sum of three parts:

$$\mathrm{CRPS}_{b,h,c} = \mathrm{CRPS}^{\mathrm{low}}_{b,h,c} + \mathrm{CRPS}^{\mathrm{mid}}_{b,h,c} + \mathrm{CRPS}^{\mathrm{high}}_{b,h,c}. \tag{59}$$

(1) Lower tail term:

$$\mathrm{CRPS}^{\mathrm{low}}_{b,h,c} = 2\left(Q_1(b,h,c) - y_{b,h,c}\right) \cdot \begin{cases} \left(\tau_1 - \frac{1}{2}\tau_1^2\right), & y_{b,h,c} \le Q_1(b,h,c), \\ \left(-\frac{1}{2}\tau_1^2\right), & y_{b,h,c} > Q_1(b,h,c). \end{cases} \tag{60}$$

(2) Middle term via Riemann sum:

$$\mathrm{CRPS}^{\mathrm{mid}}_{b,h,c} = 2\sum_{j=1}^{J_{\mathrm{qt}}} \left(Q_j(b,h,c) - y_{b,h,c}\right)\left(\mathbb{I}\{y_{b,h,c} \le Q_j(b,h,c)\} - \tau_j\right)\Delta\tau. \tag{61}$$

(3) Upper tail term:

$$\mathrm{CRPS}^{\mathrm{high}}_{b,h,c} = 2\left(Q_{J_{\mathrm{qt}}}(b,h,c) - y_{b,h,c}\right) \cdot \begin{cases} \frac{1}{2}(1-\tau_{J_{\mathrm{qt}}})^2, & y_{b,h,c} \le Q_{J_{\mathrm{qt}}}(b,h,c), \\ -\frac{1}{2}\left(1 - \tau_{J_{\mathrm{qt}}}^2\right), & y_{b,h,c} > Q_{J_{\mathrm{qt}}}(b,h,c). \end{cases} \tag{62}$$

Finally, we aggregate CRPS across batch, horizon, and variables:

$$\mathrm{CRPS} = \frac{1}{B_{\mathrm{batch}}\, H_{\mathrm{pred}}\, C_{\mathrm{var}}} \sum_{b=1}^{B_{\mathrm{batch}}} \sum_{h=1}^{H_{\mathrm{pred}}} \sum_{c=1}^{C_{\mathrm{var}}} \mathrm{CRPS}_{b,h,c}. \tag{63}$$

QICE (decile-based quantile interval calibration error). We evaluate calibration by quantile interval coverage error (QICE) using $M_{\mathrm{bin}} = 10$ decile intervals. Using the same quantile boundaries $\{Q_j\}_{j=1}^{9}$ (with $\tau_j = 0.1j$), we form 10 intervals: $(-\infty, Q_1)$, $[Q_1, Q_2)$, $\ldots$, $[Q_8, Q_9)$, $[Q_9, +\infty)$.

For each (b, h, c), we compute the interval index:

$$k_{b,h,c} = \sum_{j=1}^{9} \mathbb{I}\left\{y_{b,h,c} \ge Q_j(b,h,c)\right\}, \qquad k_{b,h,c} \in \{0, 1, \ldots, 9\}. \tag{64}$$

Let $r_m(h)$ be the empirical proportion of ground-truth values falling into interval m at horizon step h:

$$r_m(h) = \frac{1}{B_{\mathrm{batch}}\, C_{\mathrm{var}}} \sum_{b=1}^{B_{\mathrm{batch}}} \sum_{c=1}^{C_{\mathrm{var}}} \mathbb{I}\left\{k_{b,h,c} = m\right\}, \qquad m \in \{0, 1, \ldots, 9\}. \tag{65}$$

Under perfect calibration, $r_m(h)$ should be close to $1/M_{\mathrm{bin}} = 0.1$ for all m.

We define per-horizon QICE as:

$$\mathrm{QICE}(h) = \frac{1}{M_{\mathrm{bin}}} \sum_{m=0}^{M_{\mathrm{bin}}-1} \left| r_m(h) - \frac{1}{M_{\mathrm{bin}}} \right|. \tag{66}$$

The final QICE score averages over horizon steps:

$$\mathrm{QICE} = \frac{1}{H_{\mathrm{pred}}} \sum_{h=1}^{H_{\mathrm{pred}}} \mathrm{QICE}(h). \tag{67}$$

Smaller QICE indicates better calibrated quantile intervals.

C Other Results and Discussions

Across the additional quantitative results in Appendix C, we observe a consistent pattern: **ToLD** remains competitive on *point accuracy* (MSE/MAE) while providing strong *probabilistic quality* (CRPS), and it achieves this with substantially improved *inference efficiency*. Notably, the gains are most pronounced on long-term and highly non-stationary cases , suggesting that the proposed anchored latent space and truncated refinement policy can better cope with distribution shift and time-varying uncertainty, without incurring the heavy iterative cost of standard diffusion sampling.

Table 6: CRPS/QICE results on eight real-world datasets. Lower is better. Bold indicates the best result.

Model	Metric	ETTh1	ETTh2	ETTm1	ETTm2	ECL	EXG	Solar
TimeGrad (2021)	QICE	6.731	9.488	6.693	6.962	7.118	9.464	7.378
CSDI (2022)	QICE	3.107	5.331	2.828	8.106	7.506	7.864	9.957
TimeDiff (2023)	QICE	14.931	14.813	14.795	13.385	15.466	14.556	14.914
DiffusionTS (2024)	QICE	6.423	9.577	5.605	9.959	8.205	10.411	6.627
ToLD (Ours)	QICE	**3.537**	**5.160**	**5.558**	**5.434**	**7.134**	**10.105**	**6.032**

C.1 Mechanism Attribution across Different Backbones.

To further understand where the gains of ToLD come from, we conduct a mechanism-attribution study over two probabilistic generative backbones: a cVAE-based backbone and TMDM. Our central hypothesis is that once a probabilistic forecasting model provides a conditional latent representation, short-chain, stable, and input-uncertainty-aware refinement in that latent space can improve accuracy, uncertainty estimation, and efficiency simultaneously. This is because the latent representation already compresses the main semantic structure of plausible future trajectories, so refinement no longer needs to reconstruct the full future sequence from noise in the observation space. Instead, it acts as a local correction around a context-dependent solution manifold.

As shown in Table 7, adding truncated latent refinement consistently improves both Avg. MSE and Avg. CRPS over the base models, indicating that short-chain latent refinement is effective beyond a single backbone. Further incorporating anchored initialization and DDIM-style refinement brings additional improvements, suggesting that stable few-step refinement is important for reliable latent correction. Heteroscedastic noise provides further gains by adapting the refinement dynamics to input-dependent uncertainty. Finally, the full ToLD achieves the best results on both backbones, showing that tokenized latent refinement, anchored truncated diffusion, heteroscedastic noise, and test-time score distillation provide cumulative and complementary benefits rather than acting as isolated architectural add-ons.

Table 7: Mechanism attribution over two probabilistic generative backbones. Results are reported as mean $\pm$ standard deviation over 5 independent runs.

Backbone	Variant	Avg. MSE ↓	Avg. CRPS ↓
cVAE	Base	0.385 ± 0.007	0.340 ± 0.006
	+ latent truncated refinement	0.343 ± 0.005	0.316 ± 0.005
	+ latent truncated refinement + Anchor + DDIM	0.340 ± 0.005	0.309 ± 0.004
	+ latent truncated refinement + Anchor + DDIM + Heteroscedastic noise	0.339 ± 0.004	0.306 ± 0.004
	Full ToLD	$\mathbf{0.337 \pm 0.004}$	$\mathbf{0.302 \pm 0.004}$
TMDM	Base	0.436 ± 0.009	0.378 ± 0.008
	+ truncated refinement	0.430 ± 0.004	0.369 ± 0.005
	+ truncated refinement + Anchor + DDIM	0.420 ± 0.006	0.355 ± 0.004
	+ truncated refinement + Anchor + DDIM + Heteroscedastic noise	0.416 ± 0.005	0.349 ± 0.006
	Full ToLD	$\mathbf{0.409 \pm 0.004}$	$\mathbf{0.344 \pm 0.004}$

C.2 Quantitative Analysis

Table 6 reports QICE on representative real-world datasets. Overall, **ToLD** achieves *competitive* calibration error, and in some cases improves upon diffusion baselines. Meanwhile, we also note that the best QICE is not always attained by our model. This suggests that while our method substantially improves CRPS and accuracy in the main results, *calibration measured by QICE may still leave room for improvement* on certain datasets. We attribute this to the fact that QICE emphasizes coverage-frequency mismatch across quantile intervals, whereas our training objective is primarily aligned with CRPS-style distributional quality and task-level forecasting fidelity. Future work could incorporate explicit calibration regularizers or post-hoc recalibration to further reduce QICE without sacrificing sharpness.

Table 8 summarizes multivariate long-term forecasting performance over $H \in \{96, 192, 336, 720\}$ (averaged across horizons). We find that **ToLD** achieves the best (or tied-best) MSE/MAE on most

Table 8: Multivariate long-term forecasting results with prediction lengths $H \in \{96, 192, 336, 720\}$ and fixed lookback window length $L = 96$. Results are taken from TimePro. Lower is better.

Dataset	H	**ToLD Ours**		TimePro 2025		S-Mamba 2025		SOFTS 2024		iTransformer 2024		PatchTST 2023		Crossformer 2023		TiDE 2023		TimesNet 2023		DLinear 2023	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.319**	**0.359**	0.326	0.364	0.333	0.368	0.325	0.361	0.334	0.368	0.329	0.365	0.404	0.426	0.364	0.387	0.338	0.375	0.345	0.372
	192	**0.361**	**0.378**	0.367	0.383	0.376	0.390	0.375	0.389	0.377	0.391	0.380	0.394	0.450	0.451	0.398	0.404	0.374	0.387	0.380	0.389
	336	**0.399**	**0.400**	0.402	0.409	0.408	0.413	0.405	0.412	0.426	0.420	0.400	0.410	0.532	0.515	0.428	0.425	0.410	0.411	0.413	0.413
	720	**0.460**	**0.438**	0.469	0.446	0.475	0.448	0.466	0.447	0.491	0.459	0.475	0.453	0.666	0.589	0.487	0.461	0.478	0.450	0.474	0.453
	Avg	**0.384**	**0.393**	0.391	0.400	0.398	0.405	0.393	0.403	0.407	0.410	0.396	0.406	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407
ETTm2	96	**0.165**	**0.245**	0.178	0.260	0.179	0.263	0.180	0.261	0.180	0.264	0.184	0.264	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.292
	192	**0.230**	**0.291**	0.242	0.303	0.250	0.309	0.246	0.306	0.250	0.309	0.246	0.306	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.362
	336	**0.290**	**0.331**	0.303	0.342	0.312	0.349	0.319	0.352	0.311	0.348	0.308	0.346	0.597	0.542	0.377	0.422	0.321	0.351	0.369	0.427
	720	**0.392**	**0.391**	0.400	0.399	0.411	0.406	0.405	0.401	0.412	0.407	0.409	0.402	1.730	1.042	0.558	0.524	0.408	0.403	0.554	0.522
	Avg	**0.269**	**0.314**	0.281	0.326	0.288	0.332	0.287	0.330	0.288	0.332	0.287	0.330	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401
ETTh1	96	**0.365**	**0.389**	0.375	0.398	0.386	0.405	0.381	0.399	0.386	0.405	0.394	0.406	0.423	0.448	0.479	0.464	0.384	0.402	0.386	0.400
	192	**0.416**	**0.419**	0.427	0.429	0.443	0.437	0.435	0.431	0.441	0.436	0.440	0.435	0.471	0.474	0.525	0.492	0.436	0.429	0.437	0.432
	336	**0.463**	**0.448**	0.472	0.450	0.489	0.468	0.480	0.452	0.487	0.458	0.491	0.462	0.570	0.546	0.565	0.515	0.491	0.469	0.481	0.459
	720	**0.471**	**0.470**	0.476	0.474	0.502	0.489	0.499	0.488	0.503	0.491	0.487	0.479	0.653	0.621	0.594	0.558	0.521	0.500	0.519	0.516
	Avg	**0.428**	**0.431**	0.438	0.438	0.455	0.450	0.449	0.442	0.454	0.447	0.453	0.446	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452
ETTh2	96	**0.273**	**0.325**	0.293	0.345	0.296	0.348	0.297	0.347	0.297	0.349	0.288	0.340	0.745	0.584	0.400	0.440	0.340	0.374	0.333	0.387
	192	**0.350**	**0.379**	0.367	0.394	0.376	0.396	0.373	0.394	0.380	0.400	0.376	0.395	0.877	0.656	0.528	0.509	0.402	0.414	0.477	0.476
	336	**0.401**	**0.413**	0.419	0.431	0.424	0.431	0.410	0.426	0.428	0.432	0.440	0.451	1.043	0.731	0.643	0.571	0.452	0.452	0.594	0.541
	720	**0.405**	**0.428**	0.427	0.445	0.426	0.444	0.411	0.433	0.427	0.445	0.436	0.453	1.104	0.763	0.874	0.679	0.462	0.468	0.831	0.657
	Avg	**0.357**	**0.386**	0.377	0.403	0.381	0.405	0.373	0.400	0.383	0.407	0.385	0.410	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515
ECL	96	**0.139**	0.235	**0.139**	0.234	**0.139**	0.235	0.143	**0.233**	0.148	0.240	0.164	0.251	0.219	0.314	0.237	0.329	0.168	0.272	0.197	0.282
	192	0.159	0.254	**0.156**	0.249	0.159	0.255	0.158	**0.248**	0.162	0.253	0.173	0.262	0.231	0.322	0.236	0.330	0.184	0.289	0.196	0.285
	336	0.173	0.270	**0.172**	**0.267**	0.176	0.272	0.178	0.269	0.178	0.269	0.190	0.279	0.246	0.337	0.249	0.344	0.198	0.300	0.209	0.301
	720	**0.204**	**0.293**	0.209	0.299	**0.204**	0.298	0.218	0.305	0.225	0.317	0.230	0.313	0.280	0.363	0.284	0.373	0.220	0.320	0.245	0.333
	Avg	**0.168**	0.263	0.169	**0.262**	0.170	0.265	0.174	0.264	0.178	0.270	0.189	0.276	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300
Exchange	96	0.086	**0.204**	**0.085**	**0.204**	0.086	0.207	0.091	0.209	0.086	0.206	0.088	0.205	0.256	0.367	0.094	0.218	0.107	0.234	0.088	0.218
	192	**0.172**	**0.294**	0.178	0.299	0.182	0.304	0.176	0.303	0.177	0.299	0.176	0.299	0.470	0.509	0.184	0.307	0.226	0.344	0.176	0.315
	336	0.338	0.420	0.328	0.414	0.332	0.418	0.329	0.416	0.331	0.417	**0.301**	**0.397**	1.268	0.883	0.349	0.431	0.367	0.448	0.313	0.427
	720	**0.797**	**0.676**	0.817	0.679	0.867	0.703	0.848	0.680	0.847	0.691	0.901	0.714	1.767	1.068	0.852	0.698	0.964	0.746	0.839	0.695
	Avg	**0.348**	**0.398**	0.352	0.399	0.367	0.408	0.361	0.402	0.360	0.403	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414
Weather	96	**0.157**	**0.201**	0.166	0.207	0.165	0.210	0.166	0.208	0.174	0.214	0.176	0.217	0.158	0.230	0.202	0.261	0.172	0.220	0.196	0.255
	192	0.208	**0.250**	0.216	0.254	0.214	0.252	0.217	0.253	0.221	0.254	0.221	0.256	**0.206**	0.277	0.242	0.298	0.219	0.261	0.237	0.296
	336	**0.270**	**0.293**	0.273	0.296	0.274	0.297	0.282	0.300	0.278	0.296	0.275	0.296	0.272	0.335	0.287	0.335	0.280	0.306	0.283	0.335
	720	0.350	**0.345**	0.351	0.346	0.350	**0.345**	0.356	0.351	0.358	0.347	0.352	0.346	0.398	0.418	0.351	0.386	0.365	0.359	**0.345**	0.381
	Avg	**0.246**	**0.272**	0.251	0.276	0.251	0.276	0.255	0.278	0.258	0.278	0.256	0.279	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317
Solar-Energy	96	0.205	**0.220**	**0.196**	0.237	0.205	0.244	0.200	0.230	0.203	0.237	0.205	0.246	0.310	0.331	0.312	0.399	0.250	0.292	0.290	0.378
	192	**0.226**	**0.242**	0.231	0.263	0.237	0.270	0.229	0.253	0.233	0.261	0.237	0.267	0.734	0.725	0.339	0.416	0.296	0.318	0.320	0.398
	336	**0.239**	**0.262**	0.250	0.281	0.258	0.288	0.243	0.269	0.248	0.273	0.250	0.276	0.750	0.735	0.368	0.430	0.319	0.330	0.353	0.415
	720	0.248	**0.269**	0.253	0.285	0.260	0.288	**0.245**	0.272	0.249	0.275	0.252	0.275	0.769	0.765	0.370	0.425	0.338	0.337	0.356	0.413
	Avg	**0.229**	**0.248**	0.232	0.266	0.240	0.273	**0.229**	0.256	0.233	0.262	0.236	0.266	0.641	0.639	0.347	0.417	0.301	0.319	0.330	0.401

datasets. The improvements are generally consistent rather than extreme, which is expected in mature long-term forecasting benchmarks where strong transformer baselines are already highly optimized. Nevertheless, the gains become clearer on the ETT family: for example, on **ETTh2** our MSE improves from 0.373 (best competing baseline) to 0.357 (a relative reduction of $\approx 4.3\%$), and on **ETTm2** our MSE improves from 0.281 to 0.269 ($\approx 4.3\%$). On **ETTh1**, the MSE decreases from 0.438 to 0.428 ($\approx 2.3\%$), and on **Weather** from 0.251 to 0.246 ($\approx 2.0\%$). These results indicate that the proposed design is particularly beneficial in settings with strong seasonality and noticeable distribution shift, where a purely deterministic backbone may struggle to represent multi-modal futures robustly.

Table 2 compares **ToLD** against representative *generative probabilistic* forecasting methods. Here, the improvement trend is more substantial and more uniform: **ToLD** achieves the lowest CRPS across all reported datasets, while simultaneously lowering MSE. For instance, on **Solar**, CRPS drops from 0.300 (best baseline NsDiff) to 0.227 (a relative reduction of $\approx 24.3\%$), and on **ECL**, CRPS decreases from 0.290 to 0.234 ($\approx 19.3\%$). On **ETTm2**, CRPS improves from 0.256 to 0.228 ($\approx 10.9\%$), and on **ETTh2**, from 0.358 to 0.338 ($\approx 5.6\%$). These results support our main claim that anchoring diffusion in a structured latent space, combined with short-horizon refinement, can provide both *sharper* and *better-calibrated* predictive distributions compared with classical diffusion sampling or purely learned-prior approaches.

Table 9 highlights that **ToLD** offers a favorable accuracy–latency trade-off. Compared to NsDiff, the inference time is reduced from 208.07ms to 5.14ms (**about** $\times 40.5$ **speedup**), and compared to TimeGrad or DiffusionTS, the speedup is three orders of magnitude. Importantly, this efficiency does

Table 9: Computation efficiency comparison (lower time and memory are better; lower MSE/MAE are better). Bold face indicates best result.

Model	Mem.Inference (MB)	Tim.Inference (ms)	MSE	MAE
TimeGrad	8.61	8319.29	0.813	1.062
CSDI	22.61	446.70	0.708	0.949
TimeDiff	3.40	238.78	0.479	0.517
DiffusionTS	79.23	8214.53	0.774	1.089
TMDM	213.46	237.37	0.607	0.696
NsDiff	57.75	208.07	0.523	0.594
Ours (ToLD)	**18.30**	**5.14**	**0.365**	**0.389**

not come at the cost of accuracy: **ToLD** also achieves the best MSE/MAE in the same table. We also observe that the inference memory footprint remains modest (18.30MB), substantially smaller than heavy transformer-diffusion baselines, which further supports the feasibility of real-time deployment scenarios.

C.3 Qualitative Analysis

Qualitative analysis (Fig. 4). Fig. 4 visualizes multiple representative cases from ETTh, ETTm, and Weather, for horizons $H=96$ and $H=192$. We highlight three qualitative observations. First, **tracking fidelity**: the median forecast of **ToLD** (green) generally follows the ground truth trajectory (blue dots) more closely than individual diffusion samples, especially near turning points and phase transitions (common in ETTh/ETTm seasonal patterns). Second, **uncertainty adaptivity**: the empirical 5–95% interval expands or contracts in a time-dependent manner, widening in regions where the future becomes less predictable and remaining tighter in smoother segments; this behavior is desirable for non-stationary forecasting. Third, **robustness under distribution shift**: in several panels, the historical context exhibits a different amplitude/offset from the forecast horizon; despite this shift, the predicted interval still covers most future observations while maintaining reasonable sharpness, suggesting that the model captures both aleatoric variability and temporal uncertainty growth. We note that a few cases still show mild under-/over-coverage, consistent with the QICE results, indicating that further calibration improvements may be beneficial. Overall, these visualizations qualitatively corroborate the quantitative gains in CRPS and the practical reliability of the generated predictive distribution.

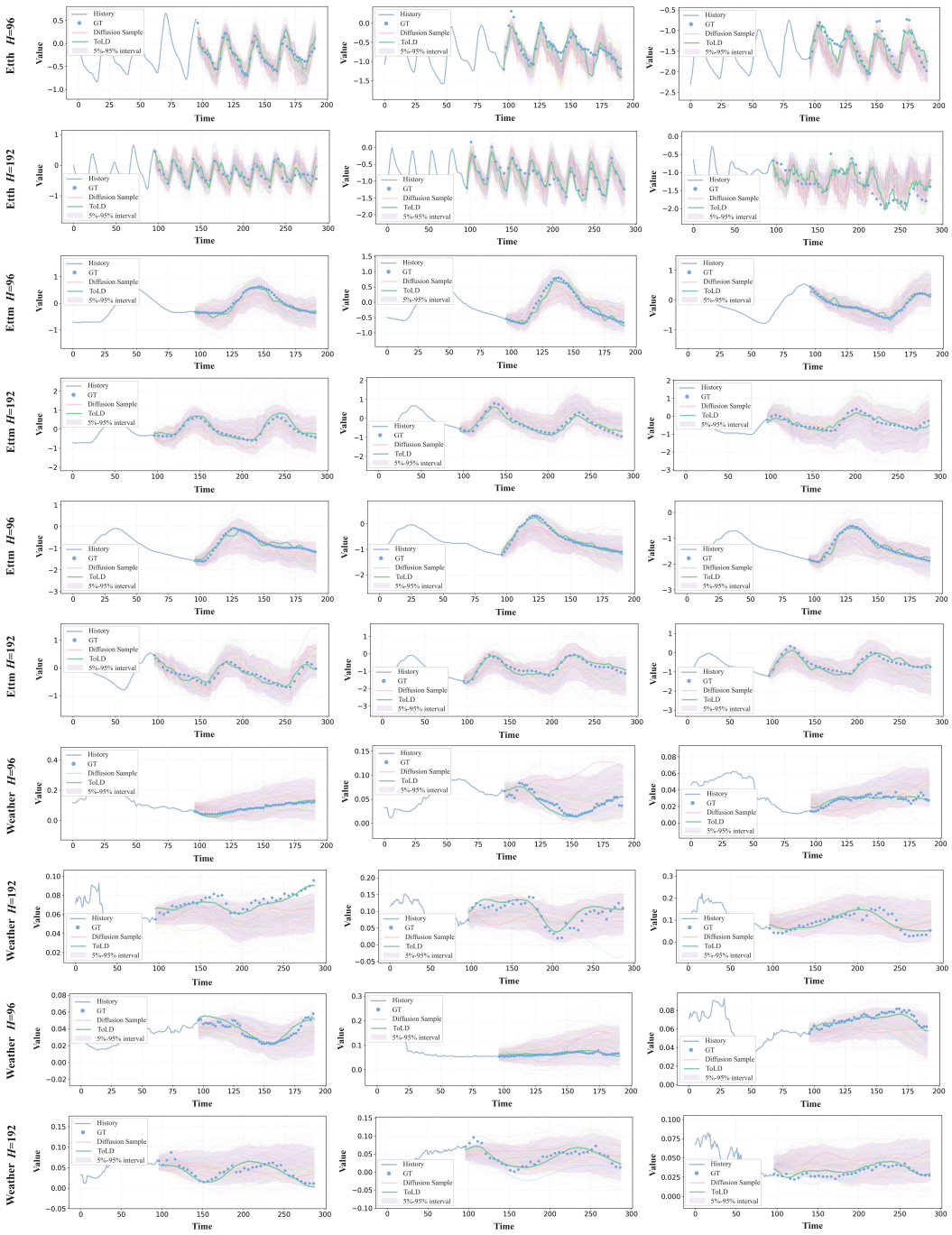


Figure 4: Qualitative Results.