Volatility-Whitened Probabilistic Residual Modeling for Long-Term Time Series Forecasting

Fan Zhang
Shandong Technology and Business University
Yantai 264005, China
zhangfan@sdtbu.edu.cn

Shiming Fan
Shandong Technology and Business University
Yantai 264005, China
2024410061@sdtbu.edu.cn

Zexuan Ma
Shandong Technology and Business University
Yantai 264005, China
2024410053@sdtbu.edu.cn

Shijun Chen
Shandong Technology and Business University
Yantai 264005, China
2024410055@sdtbu.edu.cn

Meijia Wang
Shandong Technology and Business University
Yantai 264005, China
2024410048@sdtbu.edu.cn

Hua Wang*
Ludong University
Yantai 264025, China
hwa229@163.com

Abstract

Compared with deterministic methods that output only point estimates, probabilistic forecasting can characterize both future trends and their uncertainty simultaneously, making it more suitable for decision-making in complex real-world scenarios. In recent years, diffusion models, owing to their powerful generative modeling capability, have been introduced into the field of time series forecasting. However, we find that most existing diffusion-based forecasting methods directly construct the diffusion process in the original future space, which is easily affected by heteroscedastic scale imbalance, causing highly volatile dimensions to dominate the denoising learning and leading to unstable corrections near observation boundaries. Motivated by these issues, we propose **VolaRM**, a volatility-whitened probabilistic residual modeling framework for long-term time series forecasting. By constructing a conditional probabilistic base distribution, it represents future targets as whitened residuals relative to this base distribution and performs conditional modeling in the normalized residual space, thereby alleviating scale imbalance across variables and forecasting horizons. Meanwhile, a specific gating mechanism is introduced to enhance boundary continuity and the stability of long-term forecasting. Extensive experiments on eight real-world datasets from different application domains demonstrate the effectiveness of VolaRM.

1 Introduction

Time series forecasting is essential in many real-world applications, such as traffic management [23, 18, 48, 21, 31, 12, 44], energy scheduling [39, 11, 48, 59, 24], finance [20, 25, 57, 32, 43], and healthcare [56, 28, 49, 58]. In these scenarios, accurate point estimation alone is often insufficient, since decision-making depends not only on the expected future trajectory but also on the associated uncertainty [45, 35, 15]. This has motivated increasing interest in probabilistic time series forecasting, which aims to estimate the conditional distribution of future observations given historical context.

*Corresponding author.

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

Recently, diffusion models have shown strong potential for probabilistic forecasting [17, 8, 5, 2]. By gradually perturbing future trajectories and learning the reverse denoising process, diffusion-based methods can model complex output distributions beyond unimodal Gaussian assumptions. However, most existing diffusion-based forecasting methods still construct the diffusion process in the raw observation space or around mean-conditioned endpoints, with limited consideration of scale heterogeneity. As shown in Fig. 1(a), this becomes problematic in multivariate forecasting, where different variables and prediction horizons often exhibit substantially different volatility levels [42, 54, 40].

A key yet underexplored issue is that heteroscedastic scale imbalance can distort diffusion learning [29, 47, 55]. When denoising is performed directly in the raw signal space, as illustrated in Fig. 1(b) and Fig. 1(c), high-variance channels tend to dominate the noise-matching objective, while low-volatility components receive much weaker supervision. As a result, the model may fit large-amplitude fluctuations well but underfit fine-grained stable patterns. Moreover, directly refining raw future values may introduce unstable corrections near the observation-forecast boundary, leading to discontinuity or excessive deviation from a reliable forecast center, as shown in Fig. 1(d).

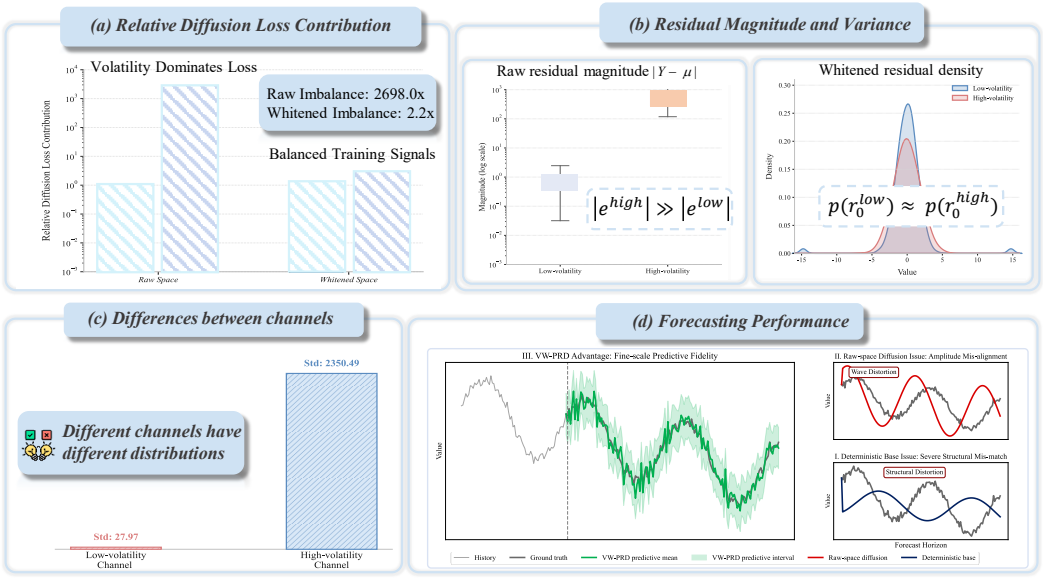


Figure 1: **Motivation of VolaRM.** (a) Multivariate time series exhibit strong heteroscedasticity across channels. (b) Raw residuals are severely scale-imbalanced, while volatility whitening produces a more aligned residual space. (c) Raw-space diffusion is dominated by high-volatility dimensions, whereas the whitened space yields more balanced supervision. (d) VolaRM mitigates boundary instability by performing conservative residual refinement around a probabilistic base.

To address these limitations, we propose **VolaRM**, a **Vola**tility-Whitened Probabilistic **R**esidual **M**odeling framework that performs generation in a whitened residual space rather than the raw future space. Specifically, VolaRM first estimates a probabilistic base distribution with horizon-wise mean and scale, and then performs diffusion on the normalized residual between the target and the base forecast. By explicitly whitening the residual with the predicted volatility, the diffusion branch operates in a more balanced latent space, which alleviates the dominance of high-volatility dimensions and improves learning stability across channels and horizons.

Based on this idea, VolaRM consists of three key components: a boundary-conditioned probabilistic base branch for estimating the forecast center and uncertainty scale, a volatility-whitened residual diffusion process for modeling structured stochastic deviations, and a conservative gated reconstruction mechanism for suppressing over-correction and improving boundary smoothness. Compared with existing diffusion-based forecasting methods, VolaRM better matches the heteroscedastic nature of real-world time series and provides a more stable diffusion learning framework for multivariate long-term forecasting. The main contributions of this work are summarized as follows:

❶ **Problem Revisit.** We identify a subtle but important limitation of existing diffusion-based forecasting methods: diffusion learning in the raw future space is affected by heteroscedastic scale imbalance, causing high-volatility dimensions to dominate denoising training.

❷ **Technical Contribution.** We propose VolaRM, a volatility-whitened probabilistic residual modeling framework for time series forecasting. The proposed method first estimates horizon-wise conditional mean and uncertainty scale, and then performs diffusion in the volatility-whitened residual space, leading to a more balanced and stable generative process.

❸ **Unified Forecasting Framework.** We develop a boundary-conditioned probabilistic base branch together with a conservative gated reconstruction mechanism, and unify them under a joint optimization objective, thereby forming a practical and effective framework for probabilistic multivariate time series forecasting.

2 Related Work

In recent years, time series forecasting has developed rapidly with the advance of deep learning [13, 19, 64, 64, 61, 52]. For long-term forecasting, a series of models, including Autoformer [51], FEDformer [65], PatchTST [26], TimesNet [50], iTransformer [21], FITS [53], MOFO [23], DUET [32], TimeAlign [38], DecompNet [22], TimeBase [10], DAG [33] and TimePerceiver [14], have continuously advanced the state of the art. However, although these methods can effectively capture temporal dynamics from raw observations, most of them are primarily designed for deterministic end-to-end forecasting and do not explicitly model the predictive distribution of future trajectories.

To characterize uncertainty, another line of research focuses on probabilistic forecasting and generative modeling. Early methods such as DeepAR [36] estimate future distributions through likelihood-based modeling. Recently, diffusion-based methods, including TimeGrad [34], D3VAE [16], TimeDiff [37], Neural MJD [9], TEDM [1], MMPD [62], and TimeDART [46], have shown stronger flexibility in modeling complex future distributions. However, most existing diffusion-based forecasting methods still operate in the raw future space, making them vulnerable to heteroscedastic scale imbalance across variables and prediction horizons. In contrast, our method performs diffusion in a volatility-whitened residual space, leading to more balanced denoising supervision and more stable long-term probabilistic forecasting.

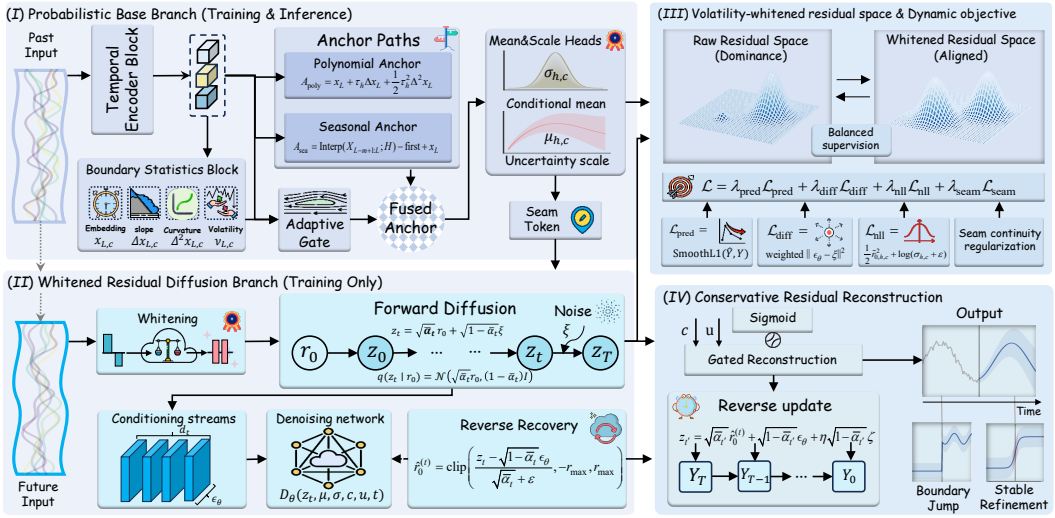


Figure 2: **Overall framework of VolaRM.** VolaRM decouples long-term forecasting into probabilistic base estimation and volatility-whitened residual modeling. The final forecast is reconstructed by conservatively injecting the generated residual into the base prediction, improving stability under heteroscedastic dynamics.

3 Methodology

To address the challenges of heteroscedastic scale imbalance and boundary saltation in multivariate long-term time series forecasting, we propose the **Volatility-Whitened Probabilistic Residual Modeling (VolaRM)** framework. As shown in Fig. 2. This framework decouples the complex generation task into two stable sub-processes: structured probabilistic baseline estimation and constrained residual diffusion within a normalized latent space. Given a historical observation window $X \in \mathbb{R}^{L\times C}$ (where L is the look-back length and C is the number of variables), our objective is to generate the conditional probability distribution of the future window $Y \in \mathbb{R}^{H\times C}$ (where H is the prediction horizon). The decoupled generative process is formulated as:

$$\hat{Y} = \mu(X) + g(X) \odot \sigma(X) \odot \hat{r}_0, \qquad \hat{r}_0 \sim p_\theta(r_0 \mid X). \tag{1}$$

where $\mu(X)$ and $\sigma(X)$ denote the deterministic predictive mean and the adaptive uncertainty scale, respectively; $\hat{r}_0$ represents the whitened residual generated by the diffusion model; and $g(X)$ is a conservative control gate designed to prevent over-correction.

3.1 Historical Context Encoding and Boundary Statistics

Time-series forecasting depends not only on long-range historical dynamics but also on the local geometry at the observation-forecast boundary. We therefore first map the historical sequence to a contextual representation

$$H = \mathrm{Enc}(X) \in \mathbb{R}^{L\times d}, \tag{2}$$

where $\mathrm{Enc}(\cdot)$ denotes a temporal encoder composed of stacked temporal blocks. To retain both global evolution and boundary-sensitive information, we aggregate the encoded sequence through

$$c = \phi_c([\mathrm{MeanPool}(H), H_L]) \in \mathbb{R}^d, \tag{3}$$

where H_L is the hidden state at the last observation step and $\phi_c(\cdot)$ is a learnable projection. The resulting context vector c provides a compact summary of the historical sequence and will be shared by both the probabilistic base branch and the diffusion branch.

To explicitly model the local behavior at the prediction boundary, we extract a channel-wise statistic vector:

$$s_c = \left[x_{L,c}, \Delta x_{L,c}, \Delta^2 x_{L,c}, \nu_{L,c}\right]^\top, \tag{4}$$

where $\Delta x_{L,c} = x_{L,c} - x_{L-1,c}$,$\Delta^2 x_{L,c} = x_{L,c} - 2x_{L-1,c} + x_{L-2,c}$ and $\nu_{L,c} = \sqrt{\mathrm{Var}\left(\{x_{i,c} - x_{i-1,c}\}_{i=L-w+1}^{L}\right) + \varepsilon}$. These quantities correspond to the boundary value, local slope, local curvature, and recent volatility, respectively. They are particularly important in time-series forecasting because the first few future steps are usually governed by short-term continuation, local turning patterns, and near-boundary uncertainty.

3.2 Boundary-aware Probabilistic Base Distribution

A central design of VolaRM is to build a strong probabilistic base before applying diffusion. Rather than asking the diffusion model to recover the entire future from noisy states, we first estimate a structural forecast center and a heteroscedastic scale. This design is motivated by the observation that most future trajectories are not arbitrary samples from an unconstrained latent process, but rather structured deviations around a context-dependent forecast baseline.

We first construct a curvature-aware anchor using local polynomial extrapolation:

$$A_{h,c}^{\mathrm{poly}} = x_{L,c} + \tau_h \Delta x_{L,c} + \frac{1}{2}\tau_h^2 \Delta^2 x_{L,c}, \qquad \tau_h = \frac{h}{H}. \tag{5}$$

This term encourages a smooth transition from the observed history into the forecast horizon. However, many real-world time series also contain short-term oscillatory patterns that cannot be represented well by a low-order extrapolation. To account for such local recurring structures, we further define a shape-preserving anchor as follows:

$$A_{:,c}^{\mathrm{sea}} = \mathrm{Interp}(X_{L-m+1:L,c}; H) - \mathrm{Interp}(X_{L-m+1:L,c}; H)_1 + x_{L,c}, \tag{6}$$

where $\mathrm{Interp}(\cdot; H)$ denotes interpolation of the recent history tail to the target horizon. Since different channels may rely on trend continuation and local pattern reuse to different extents, we introduce an adaptive gate:

$$\omega_c = \mathrm{sigmoid}\big(\phi_a(s_c)\big), \tag{7}$$

and define the final anchor as:

$$A_{h,c} = (1-\omega_c)A^{\mathrm{poly}}_{h,c} + \omega_c A^{\mathrm{sea}}_{h,c}. \tag{8}$$

This anchor provides a boundary-aware prior for the future trajectory. Conditioned on the global context c, the horizon embedding e_h, and a boundary-dependent bias $(b^{\mu}_c, b^{\sigma}_c) = \phi_b(s_c)$, the probabilistic base distribution is given by:

$$\mu_{h,c} = A_{h,c} + W_{\mu}(c + e_h)_c + 0.1\, b^{\mu}_c, \tag{9}$$

$$\sigma_{h,c} = \mathrm{clip}\left(\mathrm{softplus}\big(W_{\sigma}(c+e_h)_c + b^{\sigma}_c\big) + \sigma_{\min},\, \sigma_{\min},\, \sigma_{\max}\right). \tag{10}$$

Here, μ captures the conditional forecast center, while σ captures the horizon-wise and channel-wise uncertainty scale. This branch is therefore responsible for modeling the predictable structure and the coarse uncertainty magnitude before residual generation begins. To avoid reintroducing scale imbalance when aggregating boundary information, we first normalize the channel-wise boundary statistics before constructing the seam token. Specifically, we define

$$\tilde{s}_c = \left[\frac{x_{L,c} - m_c}{q_c + \varepsilon}, \frac{\Delta x_{L,c}}{q_c + \varepsilon}, \frac{\Delta^2 x_{L,c}}{q_c + \varepsilon}, \frac{\nu_{L,c}}{q_c + \varepsilon}\right]^{\top}, \tag{11}$$

where m_c and q_c denote the local mean and volatility scale of channel c estimated from the historical window. The seam token is then obtained by aggregating the normalized boundary statistics:

$$u = \phi_s\left(\frac{1}{C}\sum_{c=1}^{C} \tilde{s}_c\right). \tag{12}$$

This token acts as a compact descriptor of the current boundary regime and is injected into the denoising network.

3.3 Volatility-Whitened Residual Diffusion

The main innovation of VolaRM lies in performing diffusion on a *volatility-whitened residual* instead of directly on the future sequence. Existing raw-space diffusion methods implicitly mix forecast level, scale, and stochastic deviation in one space, which is especially problematic for multivariate forecasting because large-amplitude channels dominate the denoising loss. To remove this imbalance, we define the clean residual variable as:

$$r_0 = \mathrm{clip}\left(\frac{Y - \mu}{\sigma + \varepsilon}, -r_{\max}, r_{\max}\right). \tag{13}$$

This transformation removes the forecast center and normalizes the residual magnitude with the predicted scale. As a result, the latent residual space is much more homogeneous across channels and horizons, allowing the diffusion branch to focus on structured uncertainty rather than raw amplitude differences. We provide a theoretical justification of this balancing effect and the stability induced by conservative reconstruction in Appendix E.

On the whitened residual space, the forward diffusion process is defined as:

$$q(z_t \mid r_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\, r_0,\, (1-\bar{\alpha}_t)I\right), \tag{14}$$

which can be reparameterized as:

$$z_t = \sqrt{\bar{\alpha}_t}\, r_0 + \sqrt{1-\bar{\alpha}_t}\, \xi, \qquad \xi \sim \mathcal{N}(0, I), \tag{15}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$. Compared with direct future-space diffusion, this formulation yields a better-conditioned denoising target because the underlying residual has already been standardized by the probabilistic base distribution.

To predict the injected noise, we construct a denoiser conditioned on both the diffusion state and the probabilistic forecast context. Specifically, the denoising input and the condition vector are defined as:

$$\psi_t = [z_t,\ \tanh(\mu),\ \log(\sigma + \varepsilon)], \tag{16}$$

$$d_t = c + u + e_t, \tag{17}$$

where e_t denotes the diffusion-step embedding. The denoiser is then written as:

$$\epsilon_\theta = D_\theta(z_t, \mu, \sigma, c, u, t) = D_\theta(\psi_t, d_t). \tag{18}$$

This design allows the diffusion branch to access the predicted center, predicted scale, historical context, and boundary state simultaneously. Intuitively, the denoiser no longer needs to infer global signal magnitude from noisy observations; instead, it only needs to refine the residual structure that remains unexplained by the base forecast. Given ϵ_θ, the clean residual estimate at step t is recovered by:

$$\hat{r}_0^{(t)} = \mathrm{clip}\left(\frac{z_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta}{\sqrt{\bar{\alpha}_t} + \varepsilon}, -r_{\max}, r_{\max}\right). \tag{19}$$

3.4 Conservative Residual Reconstruction

Although the whitened residual diffusion branch improves probabilistic expressiveness, directly injecting the full recovered residual into the final forecast may lead to unstable long-horizon corrections. This is a forecasting-specific issue: unlike image generation, time-series forecasting requires continuity with the observed history and should remain anchored to a reasonable structural forecast center.

To stabilize the residual refinement, we introduce a channel-wise conservative gate defined as:

$$g = \rho_{\max} \cdot \mathrm{sigmoid}\big(W_g(c+u)\big) \in \mathbb{R}^C, \qquad 0 < g_c < \rho_{\max}. \tag{20}$$

which is broadcast along the horizon dimension during reconstruction. The final prediction is reconstructed as:

$$\hat{Y} = \mu + g \odot \sigma \odot \hat{r}_0. \tag{21}$$

This formulation can be interpreted as follows. The base branch μ determines the forecast center, σ determines the scale of stochastic variation, and $\hat{r}_0$ represents the normalized residual correction produced by diffusion. The gate g controls how strongly this correction should be applied. When the probabilistic base is already reliable, the gate suppresses unnecessary residual injection; when the future is more uncertain, the gate allows larger residual refinement.

During inference, we adopt a truncated DDIM-inspired reverse process. Let $\mathcal{S} = \{t_K > \cdots > t_1 > 0\}$ denote the reduced sampling schedule. For adjacent steps $t \to t'$, the latent residual is updated as:

$$z_{t'} = \sqrt{\bar{\alpha}_{t'}}\,\hat{r}_0^{(t)} + \sqrt{1-\bar{\alpha}_{t'}}\,\epsilon_\theta + \eta\sqrt{1-\bar{\alpha}_{t'}}\,\zeta, \qquad \zeta \sim \mathcal{N}(0, I), \tag{22}$$

where $\eta = 0$ corresponds to deterministic sampling and $\eta > 0$ enables stochastic generation. This inference design keeps the reverse process efficient while preserving the generative flexibility of diffusion.

3.5 Unified Objective

The objective of VolaRM is designed to jointly optimize forecast accuracy, diffusion quality, uncertainty estimation, and boundary continuity. For clarity, we keep only the essential terms in the main objective:

$$\mathcal{L} = \lambda_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{diff}}\mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{nll}}\mathcal{L}_{\mathrm{nll}} + \lambda_{\mathrm{seam}}\mathcal{L}_{\mathrm{seam}}. \tag{23}$$

The prediction loss directly supervises the final forecast in the observation space:

$$\mathcal{L}_{\mathrm{pred}} = \mathrm{SmoothL1}(\hat{Y}, Y). \tag{24}$$

This term ensures that the final output after residual reconstruction remains faithful to the ground-truth future. The diffusion loss performs noise matching in the whitened residual space:

$$\mathcal{L}_{\mathrm{diff}} = \frac{1}{HC}\sum_{h=1}^{H}\sum_{c=1}^{C} w_h\big(\epsilon_{\theta,h,c} - \xi_{h,c}\big)^2, \tag{25}$$

where $w_h = 1.2 - 0.4\frac{h-1}{H-1}$ is a horizon-dependent weight that slightly emphasizes near-future steps. This design reflects the fact that in time-series forecasting, the first few prediction steps are often the most sensitive to boundary mismatch and therefore deserve stronger denoising supervision.

To ensure that the scale branch captures meaningful heteroscedastic uncertainty, we optimize a Gaussian negative log-likelihood using the *unclipped* whitened residual:

$$\mathcal{L}_{\text{nll}} = \frac{1}{HC}\sum_{h=1}^{H}\sum_{c=1}^{C}\left(\frac{1}{2}\tilde{r}_{0,h,c}^{2} + \log(\sigma_{h,c}+\varepsilon)\right). \tag{26}$$

This term encourages the predicted scale σ to grow in difficult or unstable regions and to remain tight in stable regions, thereby improving the probabilistic calibration of the forecast.

Finally, since a common failure mode in time-series prediction is a discontinuity between the observed history and the predicted horizon, we introduce a seam-aware regularization:

$$\begin{aligned}\mathcal{L}_{\text{seam}} = &\left\|(\hat{Y}_1 - X_L) - (X_L - X_{L-1})\right\|_1 \\ &+ \left\|(\hat{Y}_2 - 2\hat{Y}_1 + X_L) - (X_L - 2X_{L-1} + X_{L-2})\right\|_1 .\end{aligned} \tag{27}$$

This loss aligns the first-order and second-order transitions at the forecasting boundary, which helps suppress abrupt jumps in the first few future steps.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate our method on eight widely used real-world multivariate time series datasets spanning four domains, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, Exchange, Electricity, and Traffic. For more detailed information, please see Appendix A.

Implementation Details. We implement all experiments in PyTorch [27] and run them on a single NVIDIA RTX 3090 GPU (24GB). Model training is performed with Adam, and forecasting accuracy is measured by the standard metrics MSE and MAE. To maintain fairness in evaluation, we disable the "Drop Last" batch-sampling trick [30]. More details about training settings and hyperparameter configurations are provided in Appendix B.

Baselines. We evaluate VolaRM against nine representative forecasting baselines, namely DLinear [60],PatchTST [26], iTransformer [21], PatchMLP [41], Pathformer [3], TimeMosaic [6], TimeMAE [4], SimMTM [7], and TimeDART [46].

Table 1: Long-term forecasting results on eight benchmark datasets. The best results are highlighted in **bold**, and the second-best results are underlined. See Appendix C for complete results.

Model	ETTh1		ETTh2		ETTm1		ETTm2		Exchange		Weather		Electricity		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear (2023)	0.424	0.433	0.352	0.394	0.345	0.374	0.257	0.317	0.332	0.395	0.229	0.274	0.165	0.263	0.418	0.287
PatchTST (2023)	0.419	0.436	0.351	0.395	0.349	0.381	0.256	0.314	0.376	0.413	0.223	0.261	0.171	0.270	0.397	0.275
iTransformer (2024)	0.440	0.445	0.359	0.395	0.347	0.378	0.258	0.318	0.321	0.384	0.232	0.270	0.163	0.258	0.397	0.281
PatchMLP (2025)	0.462	0.460	0.380	0.416	0.374	0.396	0.269	0.324	–	–	0.230	0.267	0.200	0.301	0.524	0.382
Pathformer (2025)	0.417	0.426	0.360	0.395	0.357	0.374	0.253	0.308	0.384	0.414	0.225	0.258	0.168	0.261	0.416	0.264
TimeMosaic (2026)	0.412	0.423	0.363	0.406	0.360	0.377	0.256	0.309	–	–	0.228	0.256	0.170	0.263	0.433	0.286
TimeMAE (2026)	0.434	0.445	0.402	0.431	0.350	0.383	0.269	0.326	0.426	0.446	0.226	0.265	0.196	0.309	0.409	0.275
SimMTM (2023)	0.409	0.428	0.353	0.390	0.347	0.385	0.263	0.320	0.451	0.455	0.230	0.271	0.162	0.256	0.392	0.264
TimeDART (2025)	0.411	0.426	0.345	0.386	0.344	0.379	0.257	0.316	0.358	0.405	0.226	0.263	0.163	0.254	**0.388**	**0.263**
VolaRM (Ours)	**0.400**	**0.417**	**0.321**	**0.370**	**0.340**	**0.370**	**0.247**	**0.306**	**0.320**	**0.383**	**0.218**	**0.253**	**0.157**	**0.249**	0.404	0.277

4.2 Main Results

Table 1 summarizes the long-term forecasting performance on eight benchmark datasets. Overall, VolaRM achieves the best results on seven of them, which strongly validates its effectiveness across diverse scenarios.

❶ ***VolaRM outperforms recent mainstream long-term forecasting architectures.***

Table 2: Ablation study of VolaRM on six benchmark datasets. Variant 1 uses only the probabilistic base without diffusion refinement; Variant 2 uses unwhitened residual diffusion; Variant 3 adopts a fixed uncertainty scale; Variant 4 discards boundary-aware conditioning; and Variant 5 is trained without seam-aware regularization.

Datasets	Ours		Variant 1		Variant 2		Variant 3		Variant 4		Variant 5	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.359**	**0.384**	0.373	0.403	0.372	0.402	0.371	0.401	0.372	0.402	0.373	0.403
ETTh2	**0.259**	**0.323**	0.283	0.333	0.283	0.333	0.271	0.325	0.282	0.332	0.287	0.337
ETTm1	**0.275**	**0.332**	0.318	0.357	0.397	0.345	0.321	0.354	0.298	0.346	0.321	0.361
ETTm2	**0.160**	**0.246**	0.166	0.249	0.168	0.251	0.162	0.251	0.166	0.250	0.164	0.251
Weather	**0.144**	**0.187**	0.186	0.215	0.172	0.207	0.157	0.197	0.166	0.203	0.169	0.203
Electricity	**0.130**	**0.218**	0.151	0.243	0.161	0.251	0.145	0.232	0.151	0.237	0.158	0.251

Across multiple benchmarks, VolaRM consistently surpasses strong baselines such as iTransformer, Pathformer, and TimeMosaic, with average improvements ranging from 3.98% to 12.70%. This suggests that combining probabilistic base estimation with scale-normalized residual diffusion is more effective than relying solely on architectural refinements.

❷ ***Probabilistic residual diffusion is more effective than reconstruction-based learning.***

VolaRM also performs better than reconstruction-based or self-supervised methods such as TimeMAE and SimMTM. These approaches learn time-series representations through reconstruction or masked objectives, but the learned representations are still mainly used for deterministic forecasting and thus cannot explicitly characterize uncertainty in future trajectories. In contrast, VolaRM directly models the predictive distribution through probabilistic residual diffusion. Compared with TimeMAE, it reduces MSE by 10.96% and MAE by 8.45% on average.

❸ ***VW-PRD provides a more stable diffusion-based forecasting paradigm.***

Compared with TimeDART, VolaRM reduces MSE and MAE by 3.44% and 2.24% on average, respectively. This indicates that volatility-whitened residual diffusion can provide more balanced denoising supervision and more stable residual refinement. The only exception is the Traffic dataset, where TimeDART slightly outperforms VolaRM, because Traffic contains strong road-network spatial dependencies, whereas VolaRM mainly focuses on temporal heteroscedasticity and boundary-stable residual generation.

4.3 Ablation Study

To further verify the effectiveness of each component in VolaRM, we conduct ablation studies on six benchmark datasets. Five variants are designed according to the core modules of the proposed framework. **Variant 1** removes the diffusion refinement branch and only keeps the probabilistic base prediction, which is used to examine whether residual diffusion brings additional benefits beyond a strong base forecast. **Variant 2** performs diffusion on the unwhitened residual, aiming to evaluate the importance of volatility whitening for balancing denoising supervision. **Variant 3** replaces the learned probabilistic scale with a fixed uncertainty scale, which tests whether horizon-wise and channel-wise scale estimation is necessary. **Variant 4** discards boundary-aware conditioning, so as to assess the contribution of local boundary statistics to forecast continuity and near-horizon prediction. **Variant 5** removes the seam-aware regularization, which is used to evaluate whether explicitly constraining the transition around the observation-forecast boundary is helpful. As shown in Table 2, all variants consistently perform worse than the full VolaRM model across the six datasets. This demonstrates that each component contributes to the final forecasting performance.

4.4 Qualitative Probabilistic Forecasting

We first visualize the probabilistic forecasting results of VolaRM on six benchmark datasets, including ETTh1, ETTh2, ETTm1, ETTm2, Exchange, and Weather. As shown in Fig. 3, the blue curve denotes the predictive mean μ, while the shaded regions represent the uncertainty bands $\mu \pm 1\sigma$ and $\mu \pm 2\sigma$. The red curve corresponds to the ground truth. It can be observed that VolaRM produces adaptive uncertainty intervals across different datasets and temporal regimes. In relatively smooth regions, the uncertainty band remains narrow, while in highly volatile or structurally changing regions, the

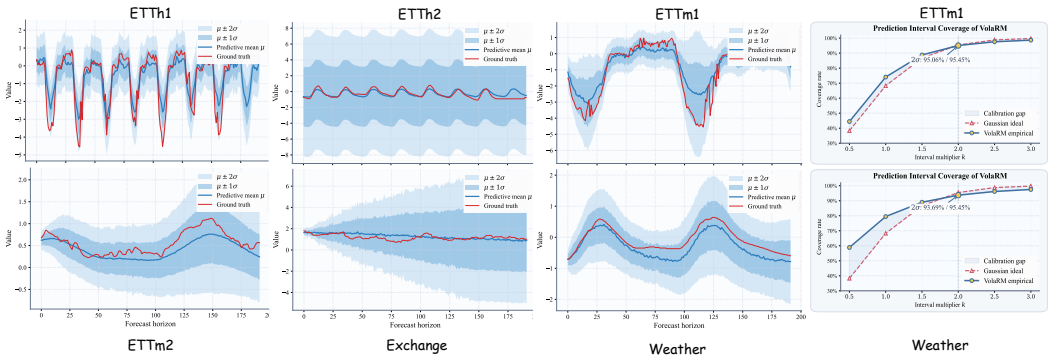


Figure 3: **Qualitative forecasting and uncertainty calibration of VolaRM.** The six panels on the left display the prediction mean and uncertainty range for different datasets. The two panels on the right display the prediction interval coverage curves for ETTm1 and Weather.

interval expands accordingly. Meanwhile, the predictive mean remains close to the target trajectory in most cases. These results indicate that VolaRM is able to jointly provide accurate point prediction and meaningful uncertainty estimation.

4.5 Prediction Interval Coverage

To further evaluate the calibration quality of the learned probabilistic scale, we compute the empirical coverage under different prediction intervals on ETTm1 and Weather. Specifically, we measure the proportion of ground-truth values that fall into the interval $[\mu - k\sigma, \mu + k\sigma]$ as the multiplier k increases. As shown in Fig. 3, the empirical coverage grows monotonically with k and closely follows the Gaussian reference curve. In particular, at the 2σ level, the empirical coverage reaches 95.06% on ETTm1 and 93.69% on Weather, which are close to the ideal Gaussian coverage of 95.45%. This demonstrates that the uncertainty scale learned by VolaRM is reasonably calibrated and can provide reliable confidence intervals for probabilistic forecasting.

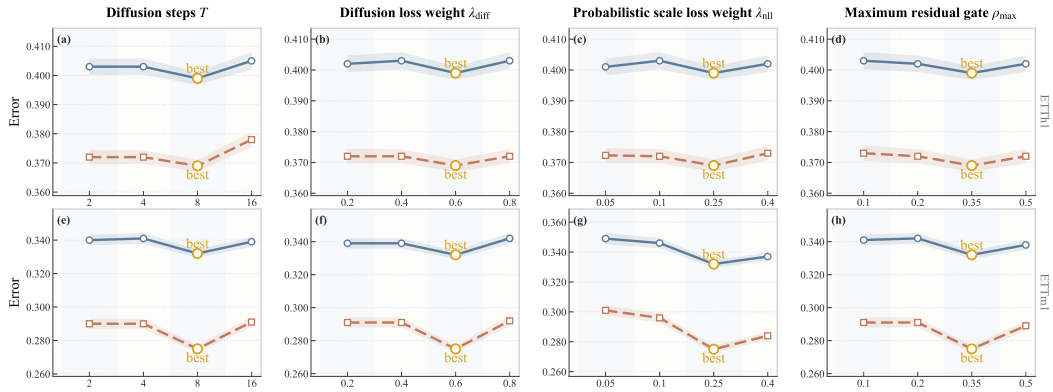


Figure 4: Hyperparameter sensitivity of VolaRM.

4.6 Hyperparameter Sensitivity Experiments

We investigate the sensitivity of VolaRM to four key hyperparameters: the number of diffusion steps T, the diffusion loss weight λ_{diff}, the probabilistic scale loss weight λ_{nll}, and the maximum residual gate $\rho_{\max}$. For each experiment, we keep the other hyperparameters fixed at their default values and vary only the target hyperparameter. As shown in Fig. 4. The default configuration, i.e., $T = 8$, $\lambda_{\text{diff}} = 0.6$, $\lambda_{\text{nll}} = 0.25$, and $\rho_{\max} = 0.35$, achieves the best or near-best results on both ETTh1 and ETTm1. This suggests that moderate diffusion refinement, balanced denoising supervision, proper probabilistic scale calibration, and conservative residual injection are important for stable long-term

Table 3: Probabilistic forecasting results with prediction length $H = 96$. Lower values indicate better performance. The best results are highlighted in **bold**, and the second-best results are underlined.

Dataset	Model	CRPS	$Q_{0.5}$	$Q_{0.6}$	$Q_{0.7}$	$Q_{0.8}$	$Q_{0.9}$
ETTh1	VolaRM (Ours)	**1.57663**	**0.19574**	**0.18846**	**0.17107**	**0.14093**	**0.09212**
	TimeMosaic	1.71695	0.22329	0.19995	0.17769	0.15222	0.10533
	iTransformer	5.88829	0.20196	0.19675	0.18058	0.14996	0.09799
	PatchTST	5.77796	0.19821	0.19302	0.17591	0.14465	0.09489
	DLinear	5.94708	0.19776	0.19108	0.17505	0.14693	0.09918
ETTh2	VolaRM (Ours)	**1.34147**	0.16846	0.16046	0.14418	0.11837	**0.07925**
	TimeMosaic	1.38944	0.17598	0.16571	0.14841	0.12257	0.08206
	iTransformer	5.09726	0.17170	0.16517	0.14943	0.12322	0.08263
	PatchTST	4.92974	**0.16591**	**0.15946**	**0.14393**	**0.11799**	0.07930
	DLinear	5.04570	0.16827	0.16089	0.14503	0.11928	0.08077
ETTm1	VolaRM (Ours)	**1.36926**	**0.16638**	**0.16324**	**0.14988**	**0.12379**	**0.08134**
	TimeMosaic	1.50854	0.18822	0.16913	0.15741	0.14025	0.09927
	iTransformer	5.26276	0.17863	0.17544	0.16155	0.13459	0.08972
	PatchTST	5.10632	0.17135	0.16813	0.15540	0.13104	0.08958
	DLinear	5.61156	0.18168	0.17581	0.16163	0.13614	0.09254
ETTm2	VolaRM (Ours)	**1.00412**	**0.12291**	**0.11884**	**0.10857**	**0.09056**	**0.06119**
	TimeMosaic	1.03677	0.13116	0.12282	0.11021	0.09183	0.06236
	iTransformer	3.79454	0.12657	0.12330	0.11356	0.09624	0.06827
	PatchTST	3.76080	0.12570	0.12259	0.11299	0.09545	0.06676
	DLinear	3.92389	0.12991	0.12699	0.11743	0.09980	0.07104

forecasting. Meanwhile, the performance gains tend to saturate beyond certain values, implying diminishing returns from further increasing diffusion depth or correction strength.

4.7 Probabilistic Forecasting Performance.

We further evaluate the probabilistic forecasting capability of VolaRM on four ETT datasets with $H = 96$, using CRPS and quantile losses at different confidence levels. As shown in Table 3, VolaRM achieves the best CRPS on all four datasets, with average improvement of approximately 6.00% over the strongest competing method.

VolaRM also obtains the best results on all quantile metrics for ETTh1, ETTm1, and ETTm2. On ETTh2, although PatchTST performs slightly better at several intermediate quantiles, VolaRM still achieves the best overall CRPS and $Q_{0.9}$. These results demonstrate the effectiveness of VolaRM in modeling predictive uncertainty and producing reliable probabilistic forecasts.

5 Conclusion

This paper proposes VolaRM, a volatility-whitened probabilistic residual modeling framework for long-term time series forecasting. Starting from the problem of heteroscedastic scale imbalance, VolaRM first estimates the conditional predictive mean and uncertainty scale, and then performs conditional modeling in the whitened residual space, thereby alleviating the dominance of highly volatile dimensions during learning. Meanwhile, the conservative gated reconstruction mechanism enhances continuity and stability near the forecasting boundary. Experiments on eight benchmark datasets show that VolaRM achieves superior performance over strong existing baselines. We hope this work provides new insights into heteroscedastic modeling and stable probabilistic forecasting for long-term time series prediction.

Acknowledgements

This work was supported in part by the following: the National Natural Science Foundation of China under Grant Nos. U24A20219, 62272281, U24A20328, the Yantai Natural Science Foundation under Grant No. 2024JCYJ034, and the Youth Innovation Technology Project of Higher School in Shandong Province under Grant No. 2023KJ212.

Competing Interests. The authors declare no competing interests.

References

[1] Edgardo Solano Carrillo, Sreerag V Naveenachandran, and Julia Niebling. Tedm: Time series forecasting with elucidated diffusion models. In *The Fourteenth International Conference on Learning Representations*.

[2] Hui Chen and Vikas Singh. Single-step operator learning for conditioned time-series diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[3] Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. *arXiv preprint arXiv:2402.05956*, 2024.

[4] Mingyue Cheng, Xiaoyu Tao, Zhiding Liu, Qi Liu, Hao Zhang, Rujiao Zhang, and Enhong Chen. Timemae: Self-supervised representations of time series with decoupled masked autoencoders. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pages 498–508, 2026.

[5] So-Yoon Cho, Jin-Young Kim, Kayoung Ban, Hyeng Keun Koo, and Hyun-Gyoon Kim. Diffolio: A diffusion model for multivariate probabilistic financial time-series forecasting and portfolio construction. *Information Fusion*, page 104286, 2026.

[6] Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. Timemosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. *arXiv preprint arXiv:2509.19406*, 2025.

[7] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36:29996–30025, 2023.

[8] Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11979–11987, 2024.

[9] Yuanpei Gao, Qi Yan, Yan Leng, and Renjie Liao. Neural mjd: Neural non-stationary merton jump diffusion for time series prediction. *arXiv preprint arXiv:2506.04542*, 2025.

[10] Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, Wu Jiang, and Yang Wang. Timebase: The power of minimalism in long-term time series forecasting. 2025.

[11] Songtao Huang, Zhen Zhao, Can Li, and LEI BAI. TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] Zikun Jin, Yuhua Qian, Xinyan Liang, Jiaqian Zhang, and Haijun Geng. Robust signal enhancement via fractional detail views and knowledge guided multi-view fusion. In *Forty-third International Conference on Machine Learning*, 2026.

[13] Zikun Jin, Yuhua Qian, Xinyan Liang, Jiaqian Zhang, Jinpeng Yuan, Shen Hu, Haijun Geng, and Honghong Cheng. Signal enhancement via multi-view dynamic representation and alignment-aware fusion. In *AAAI*, pages 22454–22462, 2026.

[14] Jaebin Lee and Hankook Lee. Timeperceiver: An encoder-decoder framework for generalized time-series forecasting. *arXiv preprint arXiv:2512.22550*, 2025.

[15] Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

[16] 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.

[17] Guojun Liang, Najmeh Abiri, Atiye Sadat Hashemi, Jens Lundström, Stefan Byttner, and Prayag Tiwari. Latent space score-based diffusion model for probabilistic multivariate time series imputation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[18] Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

[19] Zhiming Lin, Kai Zhao, Sophie Zhang, Peilai Yu, and Canran Xiao. Cec-zero: Zero-supervision character error correction with self-generated rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23612–23620, 2026.

[20] Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. *arXiv preprint arXiv:2410.04442*, 2024.

[21] 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*, 2024.

[22] Donghao Luo and Xue Wang. Decompnet: Enhancing time series forecasting models with implicit decomposition. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[23] Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[24] Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. *Advances in Neural Information Processing Systems*, 38:58382–58418, 2026.

[25] 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.

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

[27] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[28] Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2400–2410, 2024.

[29] Yiming Qin, Huangjie Zheng, Jiangchao Yao, Mingyuan Zhou, and Ya Zhang. Class-balancing diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18434–18443, 2023.

[30] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9):2363–2377, 2024.

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

[32] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1185–1196, 2025.

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

[34] 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.

[35] J Rishi, GVS Mothish, and Deepak Subramani. Conditional diffusion model with nonlinear data transformation for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

[36] 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.

[37] 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.

[38] Zhihang Song, Lihui Peng, Jianming Hu, Danya Yao, and Yi Zhang. Timealign: A multi-modal object detection method for time misalignment fusing in autonomous driving. *arXiv preprint arXiv:2412.10033*, 2024.

[39] Artyom Stitsyuk and Jaesik Choi. xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20601–20609, 2025.

[40] Namjoon Suh, Yuning Yang, Din-Yin Hsieh, Qitong Luan, Shirong Xu, Shixiang Zhu, and Guang Cheng. Timeautodiff: A unified framework for generation, imputation, forecasting, and time-varying metadata conditioning of heterogeneous time series tabular data. *arXiv preprint arXiv:2406.16028*, 2024.

[41] Peiwang Tang and Weitai Zhang. Unlocking the power of patch: Patch-based mlp for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 12640–12648, 2025.

[42] 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.

[43] Binwu Wang, Gaoyun Lin, Jiaming Ma, Qihe Huang, Zhengyang Zhou, Xu Wang, Pengkun Wang, and Yang Wang. Crisp: A spectral-based interaction strategy for multivariate time series forecasting. In *Forty-third International Conference on Machine Learning*, 2026.

[44] Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2948–2959, 2024.

[45] Chenxi Wang, Linxiao Yang, Zhixian Wang, Liang Sun, and Yi Wang. A non-isotropic time series diffusion model with moving average transitions. In *Forty-second International Conference on Machine Learning*, 2025.

[46] Daoyu Wang, Mingyue Cheng, Zhiding Liu, and Qi Liu. TimeDART: A diffusion autoregressive transformer for self-supervised time series representation. In *Forty-second International Conference on Machine Learning*, 2025.

[47] Kaibo Wang, Xiaowen Fu, Yuxuan Han, and Yang Xiang. Diffhammer: Rethinking the robustness of diffusion-based adversarial purification. *Advances in Neural Information Processing Systems*, 37:89535–89562, 2024.

[48] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[49] Yulong Wang, Yushuo Liu, Xiaoyi Duan, and Kai Wang. Filterts: Comprehensive frequency filtering for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21375–21383, 2025.

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

[51] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[52] Canran Xiao, Tianxiang Xu, Siyuan Ma, Yiyang Jiang, Haoyu Gao, and Yuhan Wu. Reversible primitive–composition alignment for continual vision–language learning. In *International Conference on Learning Representations*, volume 2026, pages 88700–88722, 2026.

[53] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

[54] Yingnan Yang, Qingling Zhu, and Jianyong Chen. Vcformer: Variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 5335–5343. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track.

[55] Zhen Yao, Bo Lang, Zhihao Zheng, Haoyan Xu, Zhengtao Yao, Zhikang Dong, and Mooi Choo Chuah. From semantic to reasoing: A survey of language-guided segmentation in the multimodal era. 2026.

[56] Jiexia Ye, Weiqi Zhang, Ziyue Li, Jia Li, Meng Zhao, and Fugee Tsung. Medualtime: A dual-adapter language model for medical time series-text multimodal learning. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 7913–7921. International Joint Conferences on Artificial Intelligence Organization, 8 2025. Main Track.

[57] Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. *Advances in Neural Information Processing Systems*, 37:55115–55140, 2024.

[58] Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In *Forty-first International Conference on Machine Learning*, 2024.

[59] Wuqing Yu, Weichen Guo, Jian Zhou, Shuyu Luo, and Jiacai Zhang. Robust inter-series dependency modeling for time series forecasting via information-theoretic alignment. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 150303–150337. PMLR, 06–11 Jul 2026.

[60] 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.

[61] Jiayu Zhang, Chuangxin Zhao, Canran Xiao, Ruibo Duan, Wenyi Mo, Haoyu Gao, and Wenshuo Wang. Pi-cca: Prompt-invariant cca certificates for replay-free continual multimodal learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[62] Yunhao Zhang, Wenyao Hu, Jiale Zheng, Lujia Pan, and Junchi Yan. Mmpd: Diverse time series forecasting via multi-mode patch diffusion loss. In *The Fourteenth International Conference on Learning Representations*.

[63] 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.

[64] Heng Zhou, Jing Tang, Jusheng Zhang, Yanshu Li, Canran Xiao, Liwei Hou, Zong Ke, and Jiawei Yao. Comem: Compositional concept-graph memory for vision–language adaptation. In *International Conference on Learning Representations*, volume 2026, pages 20009–20032, 2026.

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

A Dataset Descriptions

We evaluate VolaRM on widely used real-world multivariate time series benchmarks from multiple domains, including electricity systems, transportation networks, and environmental monitoring. These datasets differ in sampling frequency, sequence length, variable dimensionality, and cross-variable dependency patterns, providing a comprehensive benchmark for assessing long-term forecasting performance under diverse temporal dynamics.

- ❶ **ETT (Electricity Transformer Temperature)**[2] [63]. The ETT benchmark contains two-year operational records from electricity transformers in China, covering oil temperature and load-related variables. We use both hourly subsets (ETTh1 and ETTh2) and 15-minute subsets (ETTm1 and ETTm2), which differ in temporal granularity and exhibit varying short- and long-term temporal dependencies.
- ❷ **Traffic**[3] [51]. The Traffic dataset records hourly road occupancy rates collected by loop detectors on San Francisco Bay Area freeways from 2015 to 2016. Its high dimensionality and network-level traffic propagation patterns make it a challenging benchmark for modeling complex inter-sensor dependencies.
- ❸ **Electricity**[4] [51]. The Electricity dataset consists of hourly electricity consumption records from 321 clients between 2012 and 2014. The correlations among variables are often sparse and time-varying, reflecting shared usage behaviors under specific temporal conditions such as peak demand periods.
- ❹ **Weather**[5] [51]. The Weather dataset includes 21 meteorological variables recorded every 10 minutes in Germany throughout 2020. It contains diverse atmospheric indicators, such as temperature and humidity, and is characterized by nonlinear temporal variations and heterogeneous relationships among variables.

The detailed statistics of these datasets are summarized in Table 4.

[2] https://github.com/zhouhaoyi/ETDataset

[3] https://pems.dot.ca.gov/

[4] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[5] https://www.bgc-jena.mpg.de/wetter/

Table 4: Statistics of the benchmark datasets. Dim. denotes the number of variables. Dataset Size reports the number of samples in the training, validation, and test splits. Series Length lists the prediction horizons used in long-term forecasting. Frequency denotes the sampling interval. The statistics are from iTransformer [21]

Tasks	Dataset	Dim.	Series Length	Dataset Size	Frequency	Domain	Periodicity
Long-term Forecasting	ETTm1	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15 min	Temperature	96
	ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15 min	Temperature	96
	ETTh1	7	{96, 192, 336, 720}	(8545, 2881, 2881)	1 hour	Temperature	24
	ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	1 hour	Temperature	24
	Electricity	321	{96, 192, 336, 720}	(18317, 2633, 5261)	1 hour	Electricity	24
	Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	1 hour	Transportation	24
	Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10 min	Weather	144

B Implementation Details

B.1 Experiment Details

We conduct experiments under the standard multivariate long-term forecasting setting. The prediction horizons are selected from $\{96, 192, 336, 720\}$, and all models are trained, validated, and tested on the same dataset splits for fair comparison. The implementation is based on PyTorch [27], and all experiments are performed on a single NVIDIA RTX 3090 GPU with 24GB memory. We train the models using the Adam optimizer with early stopping. Hyperparameters are selected on the validation set. Specifically, the learning rate is searched over $\{1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}\}$, the dropout rate is selected from $\{0.1, 0.2, 0.3, 0.5\}$, the model dimension d_{model} is chosen from $\{64, 128, 256\}$, and the number of diffusion steps is searched in the range $[4, 10]$. Other model-specific hyperparameters are tuned according to the validation performance. Forecasting performance is measured by MSE and MAE.

B.2 Evaluation Metrics

We use mean squared error (MSE) and mean absolute error (MAE) as the evaluation metrics. MSE emphasizes large deviations, whereas MAE measures the average absolute forecasting error. For multivariate forecasting, they are computed over all prediction steps and variables:

$$\begin{aligned} \text{MSE} &= \frac{1}{HC} \sum_{i=1}^{H} \sum_{j=1}^{C} \left(Y_{i,j} - \hat{Y}_{i,j} \right)^2, \\ \text{MAE} &= \frac{1}{HC} \sum_{i=1}^{H} \sum_{j=1}^{C} \left| Y_{i,j} - \hat{Y}_{i,j} \right|. \end{aligned} \tag{28}$$

where H is the prediction horizon, C is the number of variables, $Y_{i,j}$ denotes the ground truth, and $\hat{Y}_{i,j}$ denotes the corresponding prediction.

C Full Results

Table 6 reports the complete forecasting results on all benchmark datasets and prediction horizons. Overall, VolaRM achieves the best average performance on most datasets, especially on ETT, Weather, Electricity, and Exchange, demonstrating the effectiveness of volatility-whitened residual modeling under diverse temporal dynamics. The consistent gains over strong baselines indicate that estimating a probabilistic base distribution and modeling normalized residuals can better handle heteroscedastic scale variations across variables and forecasting horizons. The only exception is Traffic, where methods that better capture road-network spatial dependencies show slight advantages. This is because Traffic contains strong spatial correlations among sensors, whereas VolaRM mainly focuses on temporal heteroscedasticity modeling and boundary-stable residual generation. Nevertheless, VolaRM remains competitive on Traffic, confirming its general applicability to long-term forecasting tasks.

Table 5: Comparison with TimeDART (2025) over eight datasets. Results are reported as mean ± standard deviation over three random seeds. The best result in each metric is highlighted in **bold color**.

Datasets	VolaRM (Ours)		TimeDART (2025)	
	MSE	MAE	MSE	MAE
ETTh1	**0.400±0.003**	**0.417±0.005**	0.411±0.005	0.426±0.003
ETTh2	**0.321±0.002**	**0.370±0.004**	0.345±0.004	0.386±0.005
ETTm1	**0.340±0.006**	**0.369±0.003**	0.344±0.004	0.379±0.006
ETTm2	**0.247±0.004**	**0.306±0.005**	0.257±0.003	0.316±0.005
Exchange	**0.320±0.003**	**0.383±0.005**	0.358±0.006	0.405±0.005
Weather	**0.218±0.004**	**0.253±0.002**	0.226±0.005	0.263±0.004
Electricity	**0.157±0.005**	**0.249±0.005**	0.163±0.006	0.254±0.005
Traffic	0.404±0.006	0.277±0.007	**0.388±0.005**	**0.263±0.003**
Avg. Rel. Impr. (%)	**3.44**	**2.24**	–	–

C.1 Experiment Statistical Significance

We assess the robustness of VolaRM by running each experiment with three random seeds and reporting the mean ± standard deviation. As shown in Table 5, we perform one-sided paired statistical tests across the eight datasets to compare VolaRM against TimeDART. For MSE, the paired t-test and Wilcoxon signed-rank test produce $p = 0.0518$ and $p = 0.0508$, respectively, indicating a borderline significant improvement. For MAE, the paired t-test yields $p = 0.0274$, while the Wilcoxon test gives $p = 0.0547$, showing that VolaRM achieves statistically significant or near-significant gains depending on the test used. Overall, VolaRM reduces the average MSE and MAE by 3.44% and 2.24%, respectively, and outperforms TimeDART on 7 of the 8 datasets, which supports the effectiveness and stability of our method.

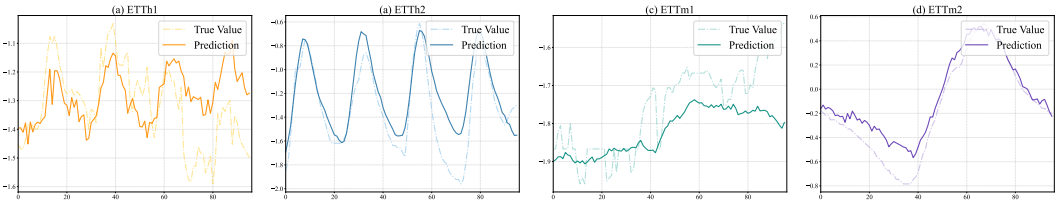


Figure 5: We visualize the forecasting results on ETTh1, ETTh2, ETTm1, and ETTm2 with a prediction horizon of 96. For each dataset, the ground truth is shown as a dashed line and the prediction as a solid line.

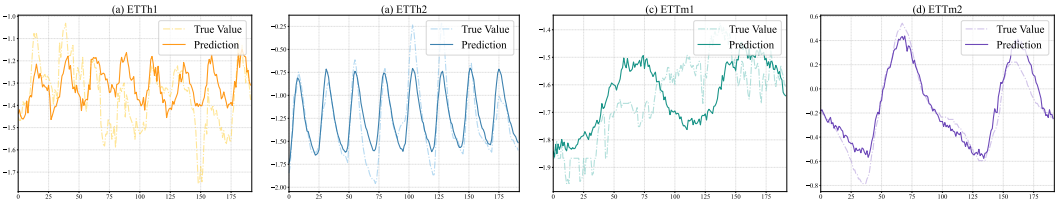


Figure 6: We visualize the forecasting results on ETTh1, ETTh2, ETTm1, and ETTm2 with a prediction horizon of 192. For each dataset, the ground truth is shown as a dashed line and the prediction as a solid line.

Table 6: Full results of different models with longer look-back lengths $L \in \{320, 336, 720\}$. The prediction length set is $H \in \{96, 192, 336, 720\}$. The best result is **bold**, the second best result is underlined.

Models		**VolaRM**		TimeDART		SimMTM		TimeMAE		TimeMosaic		Pathformer		PatchMLP		iTransformer		PatchTST		DLinear	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.359**	**0.384**	0.370	0.394	0.379	0.407	0.387	0.411	0.362	0.387	0.372	0.392	0.411	0.426	0.386	0.405	0.377	0.397	0.379	0.403
	192	**0.400**	0.413	0.402	0.419	0.412	0.424	0.420	0.431	0.400	**0.411**	0.408	0.415	0.443	0.445	0.430	0.435	0.409	0.425	0.431	0.429
	336	**0.420**	**0.424**	0.426	0.427	0.421	0.431	0.453	0.453	0.436	0.435	0.438	0.434	0.468	0.461	0.450	0.452	0.431	0.444	0.440	0.440
	720	**0.422**	**0.448**	0.446	0.462	0.424	0.449	0.476	0.485	0.451	0.460	0.450	0.463	0.525	0.508	0.495	0.487	0.457	0.477	0.445	0.462
	avg	**0.400**	**0.417**	0.411	0.426	0.409	0.428	0.434	0.445	0.412	0.423	0.417	0.426	0.462	0.460	0.440	0.445	0.419	0.436	0.424	0.434
ETTh2	96	**0.259**	**0.323**	0.283	0.340	0.293	0.347	0.325	0.378	0.300	0.387	0.279	0.336	0.311	0.371	0.292	0.347	0.274	0.337	0.275	0.336
	192	0.342	**0.376**	0.343	0.381	0.355	0.386	0.394	0.423	0.354	0.403	0.345	0.380	0.381	0.413	0.348	0.384	0.348	0.384	**0.337**	0.376
	336	**0.312**	**0.370**	0.364	0.399	0.370	0.401	0.424	0.447	0.378	0.439	0.378	0.408	0.401	0.427	0.372	0.407	0.377	0.416	0.365	0.401
	720	**0.372**	0.412	0.390	0.425	0.395	0.427	0.464	0.476	0.421	**0.393**	0.437	0.455	0.426	0.451	0.424	0.444	0.406	0.441	0.429	0.461
	avg	**0.321**	**0.370**	0.345	0.386	0.353	0.390	0.402	0.431	0.363	0.406	0.360	0.395	0.380	0.416	0.359	0.395	0.351	0.394	0.352	0.394
ETTm1	96	**0.275**	**0.332**	0.286	0.342	0.288	0.348	0.289	0.344	0.304	0.344	0.290	0.335	0.305	0.357	0.287	0.342	0.289	0.343	0.287	0.340
	192	**0.321**	**0.360**	0.326	0.367	0.327	0.373	0.330	0.371	0.339	0.364	0.337	0.363	0.351	0.384	0.331	0.371	0.329	0.368	0.324	0.363
	336	**0.356**	**0.379**	0.357	0.388	0.363	0.395	0.366	0.393	0.375	0.385	0.374	0.384	0.387	0.403	0.358	0.384	0.362	0.390	0.358	0.382
	720	0.409	**0.407**	**0.407**	0.417	0.412	0.424	0.416	0.424	0.422	0.413	0.428	0.416	0.454	0.441	0.412	0.416	0.416	0.423	0.410	0.413
	avg	**0.340**	**0.370**	0.344	0.378	0.348	0.385	0.350	0.383	0.360	0.376	0.357	0.374	0.374	0.396	0.347	0.378	0.349	0.381	0.345	0.374
ETTm2	96	**0.160**	**0.246**	0.165	0.256	0.172	0.261	0.174	0.263	0.164	0.247	0.164	0.250	0.179	0.265	0.168	0.262	0.165	0.255	0.164	0.255
	192	**0.215**	**0.285**	0.221	0.294	0.223	0.300	0.233	0.303	0.226	0.291	0.219	0.288	0.232	0.301	0.224	0.295	0.221	0.293	0.223	0.301
	336	**0.266**	**0.318**	0.279	0.330	0.282	0.331	0.291	0.340	0.279	0.324	0.267	0.319	0.285	0.334	0.274	0.330	0.276	0.327	0.273	0.328
	720	**0.349**	**0.375**	0.364	0.385	0.374	0.388	0.380	0.396	0.353	0.376	0.361	0.377	0.380	0.394	0.367	0.385	0.362	0.381	0.366	0.384
	avg	**0.247**	**0.306**	0.257	0.316	0.263	0.320	0.269	0.326	0.256	0.310	0.253	0.308	0.269	0.324	0.258	0.318	0.256	0.314	0.256	0.317
Exchange	96	**0.077**	**0.199**	0.086	0.211	0.100	0.226	0.098	0.226	-	-	0.088	0.208	-	-	0.080	0.201	0.088	0.207	0.080	0.202
	192	0.161	0.289	0.175	0.302	0.210	0.332	0.219	0.340	-	-	0.183	0.304	-	-	**0.155**	**0.287**	0.186	0.308	0.161	0.291
	336	0.319	0.412	0.344	0.431	0.389	0.460	0.400	0.466	-	-	0.354	0.429	-	-	**0.298**	**0.399**	0.374	0.446	0.327	0.434
	720	**0.721**	**0.631**	0.829	0.675	1.104	0.800	0.989	0.751	-	-	0.909	0.716	-	-	0.749	0.650	0.857	0.692	0.758	0.654
	avg	**0.320**	**0.383**	0.358	0.405	0.451	0.454	0.426	0.446	-	-	0.384	0.414	-	-	0.320	0.384	0.376	0.413	0.332	0.395
Weather	96	**0.144**	**0.187**	0.149	0.199	0.158	0.211	0.154	0.203	0.144	0.189	0.148	0.195	0.153	0.206	0.157	0.207	0.149	0.196	0.154	0.210
	192	**0.187**	**0.231**	0.193	0.240	0.199	0.249	0.191	0.241	0.197	0.232	0.191	0.235	0.197	0.245	0.200	0.248	0.191	0.239	0.198	0.252
	336	**0.235**	**0.271**	0.244	0.280	0.246	0.286	0.243	0.282	0.245	0.276	0.243	0.274	0.247	0.282	0.252	0.287	0.242	0.279	0.248	0.292
	720	**0.305**	**0.324**	0.317	0.331	0.317	0.337	0.318	0.334	0.327	0.327	0.318	0.326	0.324	0.334	0.320	0.336	0.312	0.330	0.318	0.344
	avg	**0.218**	**0.253**	0.226	0.262	0.230	0.271	0.226	0.265	0.228	0.256	0.225	0.258	0.230	0.267	0.232	0.270	0.223	0.261	0.229	0.274
Electricity	96	**0.130**	**0.218**	0.132	0.225	0.133	0.223	0.165	0.285	0.140	0.236	0.135	0.222	0.165	0.271	0.134	0.230	0.143	0.247	0.135	0.233
	192	**0.144**	**0.236**	0.150	0.241	0.147	0.237	0.181	0.297	0.156	0.250	0.157	0.253	0.183	0.288	0.154	0.250	0.158	0.260	0.151	0.248
	336	**0.163**	**0.257**	0.166	0.258	0.166	0.265	0.199	0.312	0.173	0.267	0.170	0.267	0.201	0.304	0.169	0.265	0.168	0.267	0.169	0.269
	720	**0.192**	**0.284**	0.203	0.290	0.203	0.297	0.238	0.341	0.212	0.298	0.211	0.302	0.250	0.343	0.194	0.288	0.214	0.307	0.203	0.301
	avg	**0.157**	**0.249**	0.163	0.254	0.162	0.256	0.196	0.309	0.170	0.263	0.168	0.261	0.200	0.302	0.163	0.258	0.171	0.270	0.165	0.263
Traffic	96	0.373	0.264	**0.357**	**0.247**	0.368	0.262	0.382	0.261	0.409	0.274	0.384	0.250	0.486	0.361	0.363	0.265	0.370	0.262	0.395	0.275
	192	0.394	0.275	0.376	0.256	**0.373**	**0.251**	0.399	0.267	0.424	0.281	0.405	0.257	0.515	0.378	0.384	0.273	0.386	0.269	0.407	0.280
	336	0.408	0.276	**0.389**	0.262	0.395	**0.254**	0.411	0.274	0.435	0.287	0.424	0.265	0.533	0.386	0.396	0.277	0.396	0.275	0.417	0.286
	720	0.439	0.294	**0.429**	0.286	0.432	0.290	0.446	0.298	0.464	0.302	0.452	**0.283**	0.564	0.401	0.445	0.308	0.435	0.295	0.454	0.308
	avg	0.404	0.277	**0.388**	**0.263**	0.392	0.264	0.410	0.275	0.433	0.286	0.416	0.264	0.524	0.382	0.397	0.281	0.397	0.275	0.418	0.287

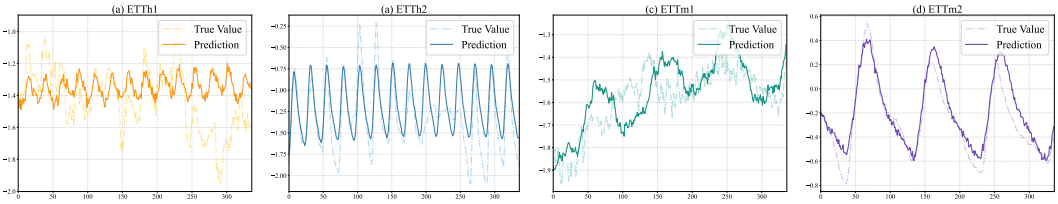


Figure 7: We visualize the forecasting results on ETTh1, ETTh2, ETTm1, and ETTm2 with a prediction horizon of 336. For each dataset, the ground truth is shown as a dashed line and the prediction as a solid line.

D Visualization of prediction results

To further provide an intuitive understanding of the forecasting behavior of VolaRM, we visualize the prediction results on four ETT benchmark datasets, including ETTh1, ETTh2, ETTm1, and ETTm2. Figures 5–8 show the forecasting curves under different prediction horizons, i.e., $H = 96$, $H = 192$, $H = 336$, and $H = 720$, respectively. VolaRM is able to capture both short-term fluctuations and long-term temporal patterns across different datasets. For datasets with clear periodic structures, such

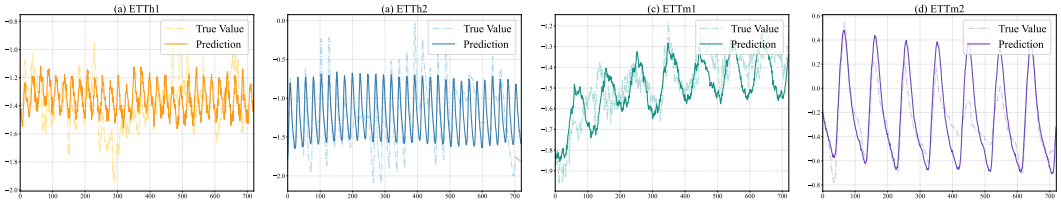


Figure 8: We visualize the forecasting results on ETTh1, ETTh2, ETTm1, and ETTm2 with a prediction horizon of 720. For each dataset, the ground truth is shown as a dashed line and the prediction as a solid line.

as ETTh2 and ETTm2, the predicted curves closely follow the phase and amplitude variations of the ground truth. For more irregular datasets such as ETTh1 and ETTm1, VolaRM still preserves the major trend evolution and avoids severe deviation over longer horizons. These visualizations further demonstrate that the proposed volatility-whitened probabilistic residual modeling strategy can produce stable and reliable long-term forecasts under diverse temporal dynamics. As the prediction horizon increases, the forecasting task becomes more challenging due to accumulated uncertainty and stronger distributional variation. Nevertheless, VolaRM maintains reasonable trajectory consistency even for $H = 336$ and $H = 720$, indicating that the probabilistic base estimation and conservative residual refinement are helpful for stabilizing long-horizon prediction.

E Theoretical Analysis

This section provides a simple theoretical explanation of why volatility-whitened residual modeling helps stabilize diffusion-based forecasting. The goal is not to fully characterize the nonlinear denoising network, but to clarify how whitening affects the scale of diffusion inputs, the signal-to-noise ratio, and the residual reconstruction magnitude.

E.1 Effect of Whitening on Diffusion Input Scale

Let

$$e_{h,c} = Y_{h,c} - \mu_{h,c} \tag{29}$$

denote the raw forecasting residual at horizon h and variable c. In raw-space diffusion, the clean signal is directly related to $e_{h,c}$, whose variance can vary significantly across channels:

$$\mathrm{Var}(e_{h,c}) = v_{h,c}. \tag{30}$$

For the standard noise-prediction objective, the target noise ξ follows $\mathcal{N}(0, I)$ and therefore has unit variance. Thus, the imbalance does not come from the variance of the noise target itself. Instead, it arises because the noisy diffusion input

$$z_t^{\mathrm{raw}} = \sqrt{\bar{\alpha}_t}\, e + \sqrt{1 - \bar{\alpha}_t}\, \xi \tag{31}$$

has a channel-dependent variance:

$$\mathrm{Var}(z_{t,h,c}^{\mathrm{raw}}) = \bar{\alpha}_t v_{h,c} + (1 - \bar{\alpha}_t). \tag{32}$$

When $v_{h,c}$ differs greatly across channels, the denoising network receives inputs with highly different scales and signal-to-noise ratios. This may distort network activations and make the gradient contribution of each channel depend on the raw residual scale.

To see this from the gradient perspective, consider the noise-matching loss

$$\mathcal{L}_{\mathrm{diff}} = \left\| \epsilon_\theta(z_t, t) - \xi \right\|^2. \tag{33}$$

Its gradient can be written as

$$\nabla_\theta \mathcal{L}_{\mathrm{diff}} = 2 \left(\epsilon_\theta(z_t, t) - \xi \right)^\top \nabla_\theta \epsilon_\theta(z_t, t). \tag{34}$$

Although ξ is scale-normalized, the Jacobian term $\nabla_\theta \epsilon_\theta(z_t, t)$ depends on the input z_t. Therefore, large-scale raw residuals can still affect activations and gradient magnitudes through the denoising input distribution.

VolaRM instead defines the whitened residual as

$$r_{h,c} = \frac{e_{h,c}}{\sigma_{h,c} + \varepsilon}. \tag{35}$$

Assume that the predicted scale is a calibrated estimate of the residual standard deviation, i.e.,

$$(1-\delta)v_{h,c} \le \sigma_{h,c}^2 \le (1+\delta)v_{h,c}, \qquad 0 \le \delta < 1. \tag{36}$$

Ignoring the small numerical constant ε for simplicity, the whitened residual variance satisfies

$$\frac{1}{1+\delta} \le \mathrm{Var}(r_{h,c}) = \frac{v_{h,c}}{\sigma_{h,c}^2} \le \frac{1}{1-\delta}. \tag{37}$$

Thus, the variance ratio across channels is reduced from

$$\frac{\max_c v_{h,c}}{\min_c v_{h,c}} \tag{38}$$

in the raw residual space to at most

$$\frac{1+\delta}{1-\delta} \tag{39}$$

in the whitened residual space.

The corresponding diffusion input becomes

$$z_t = \sqrt{\bar{\alpha}_t}\, r + \sqrt{1-\bar{\alpha}_t}\, \xi, \tag{40}$$

whose variance is

$$\mathrm{Var}(z_{t,h,c}) = \bar{\alpha}_t \mathrm{Var}(r_{h,c}) + (1-\bar{\alpha}_t). \tag{41}$$

Since $\mathrm{Var}(r_{h,c})$ is now bounded within a narrow range, the diffusion inputs and their signal-to-noise ratios are more balanced across channels and horizons. This provides a more precise explanation of why volatility whitening stabilizes denoising learning.

E.2 Optimal Scale under the Stabilized Gaussian NLL

We further analyze the probabilistic scale branch optimized by the Gaussian negative log-likelihood. In the main objective, VolaRM uses the unclipped whitened residual:

$$\tilde{r}_{0,h,c} = \frac{Y_{h,c} - \mu_{h,c}}{\sigma_{h,c} + \varepsilon}. \tag{42}$$

Accordingly, the corresponding NLL term can be written as

$$\mathcal{L}_{\mathrm{nll}} = \frac{1}{HC} \sum_{h=1}^{H} \sum_{c=1}^{C} \left[\frac{1}{2} \frac{(Y_{h,c} - \mu_{h,c})^2}{(\sigma_{h,c} + \varepsilon)^2} + \log(\sigma_{h,c} + \varepsilon) \right]. \tag{43}$$

This form is consistent with the stabilized scale term used in the main objective.

Let

$$e_{h,c} = Y_{h,c} - \mu_{h,c}, \qquad s_{h,c} = \sigma_{h,c} + \varepsilon. \tag{44}$$

For a fixed conditional context, the expected NLL for one horizon-variable pair becomes

$$\mathcal{J}(s_{h,c}) = \frac{1}{2} \frac{\mathbb{E}[e_{h,c}^2 \mid X]}{s_{h,c}^2} + \log s_{h,c}. \tag{45}$$

Taking the derivative with respect to $s_{h,c}$ gives

$$\frac{\partial \mathcal{J}}{\partial s_{h,c}} = -\frac{\mathbb{E}[e_{h,c}^2 \mid X]}{s_{h,c}^3} + \frac{1}{s_{h,c}}. \tag{46}$$

Setting the derivative to zero yields

$$s_{h,c}^2 = \mathbb{E}[e_{h,c}^2 \mid X]. \tag{47}$$

Table 7: Ablation study of the conservative gating mechanism on four ETT datasets. Lower values indicate better performance. The best results are highlighted in **bold**, and the second-best results are underlined.

Horizon	Variant	ETTh1		ETTh2		ETTm1		ETTm2	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	VolaRM (Ours)	**0.359**	**0.384**	**0.259**	**0.323**	**0.275**	**0.332**	**0.160**	**0.246**
	Fixed Gate	0.374	0.403	0.282	0.332	0.326	0.362	0.161	**0.246**
	w/o Gate	0.375	0.402	0.290	0.337	0.315	0.345	0.162	0.249
192	VolaRM (Ours)	**0.400**	**0.413**	**0.342**	**0.376**	**0.321**	**0.360**	**0.215**	**0.285**
	Fixed Gate	0.403	0.419	0.378	0.391	0.395	0.388	**0.215**	**0.285**
	w/o Gate	0.409	0.422	0.387	0.402	0.394	0.389	0.217	0.289
336	VolaRM (Ours)	**0.420**	**0.424**	**0.312**	**0.370**	**0.356**	**0.379**	**0.266**	**0.318**
	Fixed Gate	0.435	0.429	0.364	0.397	0.372	0.393	0.269	0.319
	w/o Gate	0.438	0.432	0.363	0.394	0.366	**0.379**	0.268	0.321
720	VolaRM (Ours)	**0.422**	**0.448**	**0.372**	**0.412**	**0.409**	**0.407**	**0.349**	**0.375**
	Fixed Gate	0.429	0.453	0.405	0.426	0.413	0.410	**0.349**	**0.375**
	w/o Gate	0.454	0.468	0.408	0.427	0.428	0.410	**0.349**	0.377

Therefore, the optimal relation is

$$(\sigma_{h,c} + \varepsilon)^2 = \mathbb{E}\left[(Y_{h,c} - \mu_{h,c})^2 \mid X\right]. \tag{48}$$

Equivalently,

$$\sigma^*_{h,c} = \sqrt{\mathbb{E}\left[(Y_{h,c} - \mu_{h,c})^2 \mid X\right]} - \varepsilon. \tag{49}$$

When the scale is clipped to a finite interval, the practical optimum becomes

$$\sigma^*_{h,c} = \mathrm{clip}\left(\sqrt{\mathbb{E}\left[(Y_{h,c} - \mu_{h,c})^2 \mid X\right]} - \varepsilon, \sigma_{\min}, \sigma_{\max}\right). \tag{50}$$

This derivation shows that the NLL encourages the combined stabilized scale $\sigma_{h,c} + \varepsilon$ to match the conditional residual standard deviation, rather than directly forcing $\sigma^2_{h,c}$ to equal the conditional variance.

E.3 Bounded Residual Correction

The final prediction of VolaRM is reconstructed as

$$\hat{Y}_{h,c} = \mu_{h,c} + g_c \sigma_{h,c} \hat{r}_{0,h,c}, \tag{51}$$

where

$$g_c = \rho_{\max} \cdot \mathrm{sigmoid}(W_g(c + u)), \qquad 0 < g_c < \rho_{\max}. \tag{52}$$

Since the recovered residual is clipped by construction,

$$|\hat{r}_{0,h,c}| \le r_{\max}, \tag{53}$$

the correction around the probabilistic base is bounded:

$$|\hat{Y}_{h,c} - \mu_{h,c}| = |g_c \sigma_{h,c} \hat{r}_{0,h,c}| \le \rho_{\max} \sigma_{h,c} r_{\max}. \tag{54}$$

Thus, the diffusion branch cannot arbitrarily perturb the base forecast. The residual correction is controlled by both the conservative gate and the predicted uncertainty scale, which helps suppress over-correction and improve long-horizon stability.

F Ablation Study of the Gating Mechanism.

We further investigate the effectiveness of the proposed gating mechanism by comparing VolaRM with a fixed-gate variant and a variant without gating on four ETT datasets across different prediction horizons. As shown in Table 7, VolaRM achieves the best or tied-best performance in most settings, demonstrating the consistent advantage of adaptive gating over fixed or removed gating strategies.

In particular, the performance degradation becomes more evident on ETTh1, ETTh2, and ETTm1 when the adaptive gate is replaced or removed, while the differences on ETTm2 are relatively small. These results indicate that dynamically controlling the residual correction according to the input context can better balance correction strength and prediction stability, especially under longer forecasting horizons.

F.1 Implementation Details of the Training Objective

For reproducibility, we detail the complete objective used in our implementation. It contains six weighted components:

$$\begin{aligned}\mathcal{L}_{\text{impl}} = \lambda_{\text{base}}\mathcal{L}_{\text{base}} + \lambda_{\text{refine}}\mathcal{L}_{\text{refine}} + \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} \\ + \lambda_{\text{nll}}\mathcal{L}_{\text{nll}} + \lambda_{\Delta}\mathcal{L}_{\Delta} + \lambda_{\text{gate}}\mathcal{L}_{\text{gate}}.\end{aligned} \tag{55}$$

Below, barred quantities denote predictions and targets in the normalized training space. We use $\ell(A, B)$ to denote the element-wise SmoothL1 loss averaged over all elements.

Base and refined forecast supervision. We supervise both the probabilistic base mean and the reconstructed forecast:

$$\begin{aligned}\mathcal{L}_{\text{base}} &= \ell(\bar{\mu}, \bar{Y}), \\ \mathcal{L}_{\text{refine}} &= \ell(\hat{\bar{Y}}, \bar{Y}).\end{aligned} \tag{56}$$

Base supervision encourages the model to retain a useful forecasting center, while refined forecast supervision trains the residual correction and gate jointly.

Diffusion supervision. The implementation adopts a v-prediction parameterization. Given the clipped whitened residual r_0 and injected noise ξ, the training target and recovered residual are

$$\begin{aligned}v_t &= \sqrt{\bar{\alpha}_t}\,\xi - \sqrt{1-\bar{\alpha}_t}\,r_0, \\ \hat{r}_0^{(t)} &= \sqrt{\bar{\alpha}_t}\,z_t - \sqrt{1-\bar{\alpha}_t}\,v_\theta.\end{aligned} \tag{57}$$

The diffusion objective combines velocity prediction with an auxiliary clean-residual reconstruction term:

$$\mathcal{L}_{\text{diff}} = \ell(v_\theta, v_t) + 0.1\,\ell(\hat{r}_0^{(t)}, r_0). \tag{58}$$

The auxiliary term directly supervises residual recovery. The base mean, scale, and conditioning features supplied to the denoiser are detached in this training path; the base branch remains trainable through its forecasting and likelihood objectives.

Probabilistic scale supervision. The scale branch is trained using Gaussian negative log-likelihood, up to an additive constant:

$$\mathcal{L}_{\text{nll}} = \frac{1}{HC}\sum_{h=1}^{H}\sum_{c=1}^{C}\left[\frac{(\bar{Y}_{h,c} - \bar{\mu}_{h,c})^2}{2\bar{\sigma}_{h,c}^2} + \log\bar{\sigma}_{h,c}\right]. \tag{59}$$

This term uses the unclipped prediction error and encourages the estimated scale to reflect residual magnitude.

Temporal increment and gate regularization. To preserve local temporal variations, we match adjacent forecast increments to the corresponding target increments:

$$\mathcal{L}_{\Delta} = \ell\left(\hat{\bar{Y}}_{2:H} - \hat{\bar{Y}}_{1:H-1}, \bar{Y}_{2:H} - \bar{Y}_{1:H-1}\right). \tag{60}$$

This term operates within the future horizon and differs from explicitly matching first- and second-order transitions across the observation–forecast boundary.

We additionally regularize the average residual injection strength:

$$\mathcal{L}_{\text{gate}} = \frac{1}{HC} \sum_{h=1}^{H} \sum_{c=1}^{C} g_{h,c}. \tag{61}$$

This penalty encourages conservative residual injection, while the refined forecast loss determines when additional correction is beneficial.

The model-level default weights are $\lambda_{\text{base}} = 1.0$, $\lambda_{\text{refine}} = 0.4$, $\lambda_{\text{diff}} = 0.6$, $\lambda_{\text{nll}} = 0.25$, $\lambda_{\Delta} = 0.1$, and $\lambda_{\text{gate}} = 0.02$, unless overridden by the experiment configuration. The training wrapper also supports weighted MAE and MSE auxiliary terms; their coefficients default to zero, so they contribute only when explicitly enabled.

G Limitations

Although VolaRM achieves strong performance on multiple long-term forecasting benchmarks, it still has several limitations. First, the proposed framework mainly focuses on temporal heteroscedasticity and boundary-stable residual modeling, while it does not explicitly model complex spatial or graph-structured dependencies among variables. This may limit its advantage on datasets such as Traffic, where road-network topology and sensor interactions play a dominant role.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract accurately reflect our contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of this work in Appendix G.

   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: Every novel theorem provides its full set of assumptions, and complete proofs are provided in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide complete experimental details in Appendix B.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We will release our code once the paper is accepted.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide complete experimental details in Appendix B

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report mean $\pm$ standard deviation over three random seeds and provide additional statistical significance analysis in Appendix C.

   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 specific GPU type used is listed in Appendix B.

   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: Our research aligns with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The paper focuses on advancing the field of machine learning. While our work may have various societal implications, we believe none are significant enough to warrant specific mention here.

    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: This paper poses no such risks.

    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 code and datasets used in the paper are publicly available and properly credited.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We will make the code publicly available upon acceptance of the paper and provide detailed documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    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: Limited to language polishing or writing assistance.

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.