EASY YET EFFECTIVE: DIRECT PROBABILISTIC FORECASTING WITH RECURRENT VOLATILITY MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent probabilistic forecasters have made sequence generation increasingly expressive, but this alone does not resolve how predictive uncertainty should evolve over a long horizon. We identify a design mismatch: future locations can be generated in parallel, whereas predictive scale benefits from memory of recent volatility. VolDy-VAE separates these computations. A patchwise variational encoder and direct latent projection produce all future latents in one pass; a shared decoder predicts location, while a single GRU carries a scale state from historical to future patches. The model requires neither iterative denoising nor recursive generation of predicted values. Heteroscedastic Gaussian likelihoods train historical reconstruction and future prediction, allowing learned scale to moderate the influence of variable observations on location learning. Across nine long-horizon benchmarks, VolDy-VAE improves CRPS over the listed probabilistic baselines in most settings, with ablations and efficiency results examining the role of recurrent volatility memory.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Probabilistic time series forecasting must predict both where a series may go and how uncertain that prediction is. Recent work has pursued increasingly expressive generators, including diffusion-based forecasters that construct samples through repeated denoising Rasul et al. (2021a); Ye et al. (2025). Yet greater generative capacity alone does not settle a more basic question in long-horizon prediction: how should the uncertainty of future values inherit and evolve the volatility observed in the past? This question matters in applications where the width of a forecast interval affects decisions, including energy dispatch and financial risk control Gneiting & Raftery (2007); Hong & Fan (2016); Sezer et al. (2020).

We trace this difficulty to a mismatch between the temporal requirements of location and scale. Future locations can be projected together from the observed history; their generation need not depend on previously generated values. Predictive scale has a different job: it should respond to the recent volatility level and carry that information across successive future steps. A fixed variance discards this variation, while an independently computed scale at each step has no explicit state connecting the history to the forecast horizon. Autoregressive models retain temporal

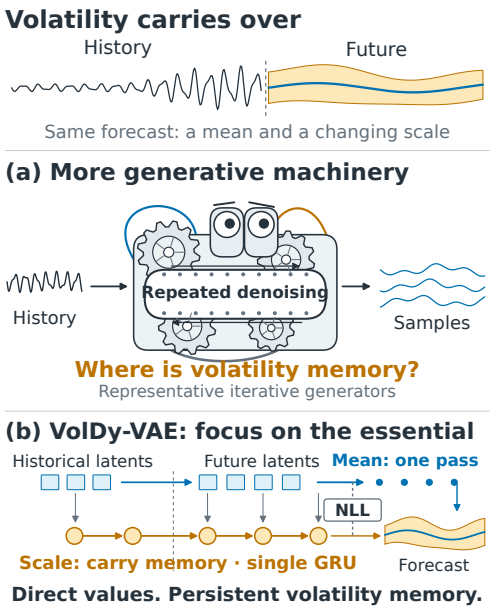

Figure 1: **Complexity versus volatility memory.** Iterative generation is illustrated as a complex machine. Dashed NLL links couple mean and scale learning in VolDy-VAE. Curves are schematic.

state but generate observations sequentially Salinas et al. (2020); Rasul et al. (2021a). The two requirements need not be tied to the same generation procedure.

Figure 1 illustrates the resulting division of labor: generate the future representation in one pass, and use recurrence to carry volatility information. A global mapping produces future latent patches together. A scale state reads the historical patches, crosses the history-to-future boundary, and evolves over the projected future patches. The location path remains direct, while scale retains temporal memory without feedback from predicted observations.

We realize this principle with the Volatility Dynamics Variational Autoencoder (VolDy-VAE). A patchwise variational encoder represents the observed history, and a direct latent projection produces the entire future horizon. A shared decoder predicts historical reconstruction and future location. In parallel, a single GRU decodes scale: it processes historical latent patches and carries its final hidden state into future prediction. Each state emits positive scales for the observations in its patch. A simple heteroscedastic Gaussian negative log-likelihood (NLL) couples the two paths: scales informed by historical volatility weight mean-prediction residuals by inverse variance, while those residuals also train the scale estimates. This makes historical volatility memory part of mean learning within a single probabilistic objective. Thus, the additional temporal machinery consists of one recurrent scale decoder, while forecast values are generated without iterative sampling.

VolDy-VAE retains volatility memory while generating forecast values in parallel. We examine this design through forecasting comparisons, scale-path ablations, synthetic heteroscedasticity diagnostics, and efficiency measurements. Our contributions are:

- We identify a location-scale design mismatch in long-horizon probabilistic forecasting: direct generation suits future location, while predictive scale benefits from memory of historical volatility.
- We address this mismatch with a direct latent forecast and a single recurrent scale decoder whose state continues from historical reconstruction into future prediction, coupling volatility memory and mean learning through Gaussian NLL.
- We evaluate the resulting simple architecture through forecast quality, scale-path ablations, synthetic diagnostics, and inference efficiency across long-horizon benchmarks.

2 Related Work

Sequential probabilistic forecasting. Autoregressive models such as DeepAR Salinas et al. (2020) maintain a recurrent state and emit distribution parameters as observations are generated. TimeGrad Rasul et al. (2021a) combines autoregressive forecasting with diffusion-based sampling. These approaches preserve temporal context, but generating future values sequentially adds horizon-dependent inference work. VolDy-VAE uses recurrence only to carry scale information. Its future latent representation and locations are produced directly.

Expressive and direct generators. Diffusion forecasters, including TimeGrad and NsDiff Rasul et al. (2021a); Ye et al. (2025), obtain flexible samples through repeated denoising. Direct latent forecasters such as K^2VAE Wu et al. (2025) generate the horizon in one pass. The recent Parametric Prior Mapping framework (PPM) Li et al. (2026) maps an input-conditioned parametric prior into a predictive distribution without iterative denoising. PPM adapts the prior under non-stationarity; VolDy-VAE instead transfers historical volatility information into future scale estimates through a recurrent decoder.

Time-varying uncertainty. Classical GARCH models Bollerslev (1986) describe volatility persistence through variance recursions, while heteroscedastic regression allows scale to vary with inputs Nix & Weigend (1994). Appendix Section C.1 explains the connection between this persistence principle and our recurrent scale state. Neural approaches such as HeTVAE Shukla & Marlin (2022) and VHVM Yin & Barucca (2022) also model nonconstant uncertainty, respectively for irregular-series reconstruction and financial volatility estimation. Our setting is multivariate long-horizon forecasting: the model directly generates future locations while a dedicated recurrent decoder evolves predictive scale. Appendix Section A develops these comparisons, and Appendix Table 7 summarizes the architectural distinctions.

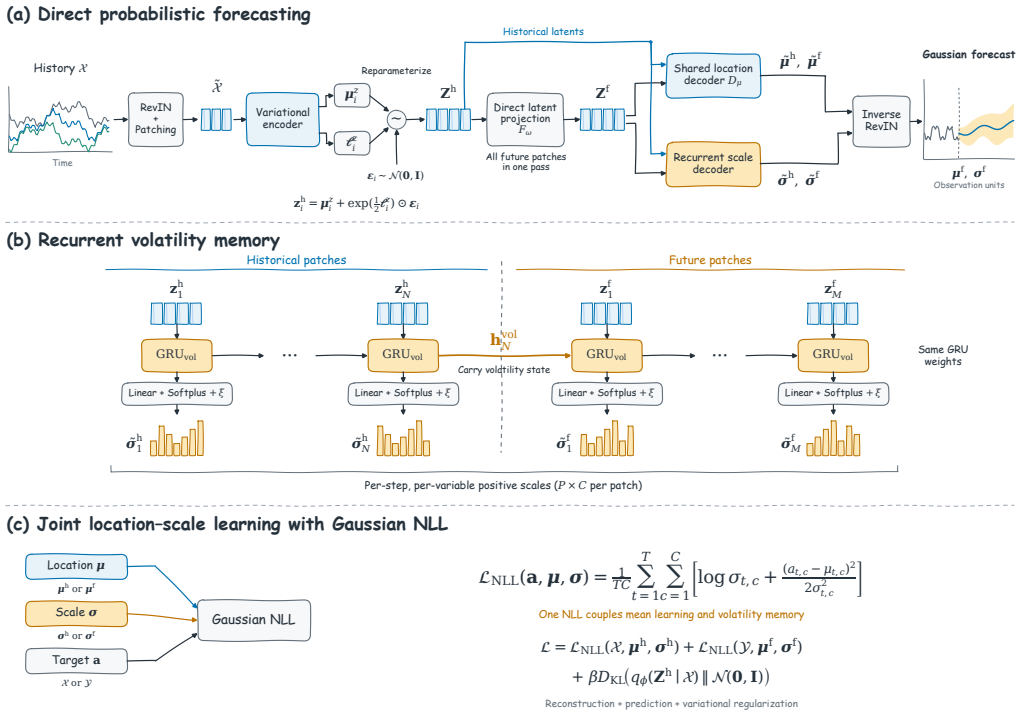

Figure 2: VolDy-VAE. RevIN-normalized history is encoded into variational patch latents and directly projected to future latents. A shared location decoder maps historical and future latents to reconstruction and forecast means. A separate GRU-based scale decoder reads historical latents, carries its hidden state across the forecast boundary, and emits positive scale blocks for future latents. Location prediction is direct; recurrence is confined to the scale path. The Gaussian NLL jointly trains location and scale, weighting squared location residuals by inverse predicted variance.

3 Methodology

VolDy-VAE separates the computation used to generate future locations from the computation used to evolve their predictive scales. A variational encoder represents the look-back window as stochastic patch latents. A direct mapping generates all future latents in one pass, while a recurrent scale decoder carries volatility information from historical patches to future patches. Figure 2 shows the full computation.

3.1 Forecasting formulation and latent generation

Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{L \times C}$ denote a look-back window of length L with C variables, and let $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{H \times C}$ be its future horizon. We model the conditional predictive distribution of $\boldsymbol{\mathcal{Y}}$ given $\boldsymbol{\mathcal{X}}$. RevIN computes a channel-wise mean $m_{\mathcal{X},c}$ and standard deviation $v_{\mathcal{X},c}$ from the look-back window, then maps it to normalized values $\tilde{\boldsymbol{\mathcal{X}}}$ Kim et al. (2022). With affine RevIN, $\tilde{x}_{t,c} = \gamma_c(x_{t,c} - m_{\mathcal{X},c})/v_{\mathcal{X},c} + \beta_c$; without the affine transform, $\gamma_c = 1$ and $\beta_c = 0$. The normalized sequence is divided into $N = \lceil L/P \rceil$ patches of length P Nie et al. (2023). Each patch is flattened before encoding.

Following the variational autoencoder framework Kingma & Welling (2014), the encoder maps historical patches to a diagonal Gaussian latent posterior. For historical patch i,

$$q_\phi(\mathbf{z}_i^h \mid \tilde{\boldsymbol{\mathcal{X}}}) = \mathcal{N}(\boldsymbol{\mu}_i^z, \mathrm{diag}(\exp(\boldsymbol{\ell}_i^z))), \qquad \mathbf{z}_i^h = \boldsymbol{\mu}_i^z + \exp\left(\tfrac{1}{2}\boldsymbol{\ell}_i^z\right) \odot \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1}$$

The reparameterized latent sequence $\mathbf{Z}^h \in \mathbb{R}^{N \times D}$ captures uncertainty in the representation of the historical window. An MLP transforms its flattened representation into all $M = \lceil H/P \rceil$ future latent patches:

$$\mathbf{Z}^f = \mathrm{reshape}\big(F_\omega(\mathrm{vec}(\mathbf{Z}^h))\big) \in \mathbb{R}^{M \times D}. \tag{2}$$

The projection does not consume previously generated observations. The entire future latent horizon is therefore available to both decoder paths after one forward mapping.

3.2 Location and recurrent scale decoding

A shared MLP location decoder is applied to both latent sequences. It maps each historical latent to a length-$P \times C$ reconstruction block and each future latent to a forecast block:

$$\tilde{\boldsymbol{\mu}}^{\mathrm{h}} = D_{\mu}(\mathbf{Z}^{\mathrm{h}}), \qquad \tilde{\boldsymbol{\mu}}^{\mathrm{f}} = D_{\mu}(\mathbf{Z}^{\mathrm{f}}). \tag{3}$$

After reshaping the blocks into time steps and restoring the observation scale, $\boldsymbol{\mu}^{\mathrm{f}}$ gives the full future location without autoregressive value feedback. Sharing D_{μ} also makes historical reconstruction and future prediction use the same latent-to-observation map.

The scale decoder has a different temporal role. A single-layer GRU Cho et al. (2014) first reads historical latents and then continues over future latents from the final historical hidden state:

$$\begin{aligned}(\mathbf{H}^{\mathrm{h}}, \mathbf{h}_{N}^{\mathrm{vol}}) &= \mathrm{GRU}_{\mathrm{vol}}(\mathbf{Z}^{\mathrm{h}}),\\ (\mathbf{H}^{\mathrm{f}}, \mathbf{h}_{N+M}^{\mathrm{vol}}) &= \mathrm{GRU}_{\mathrm{vol}}(\mathbf{Z}^{\mathrm{f}}; \mathbf{h}_{N}^{\mathrm{vol}}).\end{aligned} \tag{4}$$

The same GRU parameters serve both periods. Its state at the history-to-future boundary carries information accumulated from the look-back window, so future scales depend on the projected future latent context and on preceding scale-state updates. Appendix Section C.2 discusses how this temporal sharing affects scale estimation under persistent volatility. A linear head with a Softplus activation maps each recurrent output to a positive $P \times C$ scale block:

$$\tilde{\boldsymbol{\sigma}}_i = \mathrm{Softplus}(W_{\sigma}\mathbf{h}_i^{\mathrm{vol}} + \mathbf{b}_{\sigma}) + \xi, \qquad \xi > 0. \tag{5}$$

The block is reshaped to per-step, per-variable standard deviations; the model does not force every observation in a patch to share one scale. Both decoder heads produce parameters in the same RevIN-normalized coordinates. For either historical or future step, the inverse transformation gives

$$\mu_{t,c} = m_{\mathcal{X},c} + \frac{v_{\mathcal{X},c}}{\gamma_c}(\tilde{\mu}_{t,c} - \beta_c), \qquad \sigma_{t,c} = \frac{v_{\mathcal{X},c}}{|\gamma_c|}\tilde{\sigma}_{t,c}. \tag{6}$$

The standard deviation uses the same look-back scale as the location, but no additive mean or affine bias. The resulting $\boldsymbol{\mu}^{\mathrm{h}}, \boldsymbol{\sigma}^{\mathrm{h}}$ and $\boldsymbol{\mu}^{\mathrm{f}}, \boldsymbol{\sigma}^{\mathrm{f}}$ are all in observation units.

The recurrent scale state is a deterministic function of sampled latents. It introduces temporal memory into scale estimation while leaving the future location path direct. Conditional on a latent sample, the observation likelihood factorizes over forecast steps and variables; marginalizing the shared stochastic latents induces dependence across forecast steps.

3.3 Learning and prediction

The historical and future observations are modeled with heteroscedastic Gaussian likelihoods in observation units. For either period, with target $a_{t,c}$ and predicted parameters $(\mu_{t,c}, \sigma_{t,c})$, the average negative log-likelihood is

$$\mathcal{L}_{\mathrm{NLL}}(\mathbf{a}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{TC}\sum_{t=1}^{T}\sum_{c=1}^{C}\left[\log \sigma_{t,c} + \frac{(a_{t,c} - \mu_{t,c})^2}{2\sigma_{t,c}^2}\right]. \tag{7}$$

The scale determines interval width and the contribution of each residual to location learning. In particular, the location gradient is weighted by the inverse predicted variance (Appendix Section C.6), coupling mean estimation to learned volatility. The complete training objective combines reconstruction, prediction, and variational regularization:

$$\mathcal{L} = \mathcal{L}_{\mathrm{NLL}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mu}^{\mathrm{h}}, \boldsymbol{\sigma}^{\mathrm{h}}) + \mathcal{L}_{\mathrm{NLL}}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mu}^{\mathrm{f}}, \boldsymbol{\sigma}^{\mathrm{f}}) + \beta D_{\mathrm{KL}}\big(q_{\phi}(\mathbf{Z}^{\mathrm{h}} \mid \boldsymbol{\mathcal{X}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big). \tag{8}$$

Appendix Section C.3 gives the separate reconstruction and prediction likelihood terms, including the per-observation Gaussian NLL in Appendix Eq. 10.

Theoretical rationale. Appendix C derives the heteroscedastic likelihood from the ELBO and analyzes how scale weights location residuals. In a constant-mean regime-switching model with known variances, Theorem 1 establishes the efficiency of inverse-variance weighting; Appendix Section C.8 discusses estimated variances. When learned scales track observation noise, attenuation reduces noisy updates. When they absorb mean-fitting error, it can instead leave predictable structure underfit Seitzer et al. (2022). Appendix Eq. 19 separates these sources, while Sections C.6–C.7 analyze gradient weighting and locally optimized scale. These results characterize the local objective; forecast accuracy and calibration assess the jointly trained model.

At inference, we draw historical latents from the encoder posterior, map them to future latents, decode a location-scale forecast, and sample Gaussian observations. Sampling requires no recursive generation of predicted values. The latent projection provides the full horizon directly, while the GRU updates only at patch resolution to evolve predictive scale.

4 EXPERIMENTS

We evaluate VolDy-VAE on real-world forecasting benchmarks, with ablations, calibration diagnostics, and efficiency measurements. Appendix Section B compares learned scales with known volatility trajectories in controlled synthetic data.

4.1 EVALUATION ON REAL-WORLD BENCHMARKS

4.1.1 DATASETS

We conduct experiments on nine widely-recognized real-world benchmarks from ProbTS Zhang et al. (2024), spanning diverse domains: Energy (*ETTh1*, *ETTh2*, *ETTm1*, *ETTm2*, *Electricity*), Transportation (*Traffic*), Weather (*Weather*), Economics (*Exchange*), and Health (*ILI*). Following the standard protocol for long-term forecasting, we set the look-back window length to $L = 96$ for all datasets, except for *ILI* where $L = 36$. The forecasting horizons are set to $H \in \{96, 192, 336, 720\}$ for the general datasets and $H \in \{24, 36, 48, 60\}$ for *ILI*. Dataset definitions and statistics are given in Appendix Section D.5 and Table 11.

4.1.2 BASELINES

We compare VolDy-VAE with 13 baselines, covering both deterministic and probabilistic forecasting models. The selected baselines include deep point forecasters such as FITS Xu et al. (2024), PatchTST Nie et al. (2023), iTransformer Liu et al. (2024), and Koopa Liu et al. (2023), which are originally designed for deterministic prediction. For fair probabilistic evaluation, these models are augmented with a Gaussian output head to predict distribution parameters (μ, σ). All augmented point forecasters are trained with the same Gaussian NLL objective used for probabilistic evaluation, and the reported numbers are obtained from six independent retraining-and-evaluation runs under the same data splits and forecasting horizons. We also include probabilistic and generative models, including TSDiff Kollovieh et al. (2023), TimeGrad Rasul et al. (2021a), CSDI Tashiro et al. (2021), NsDiff Ye et al. (2025), PPM Li et al. (2026), GRU NVP, GRU MAF, Trans MAF Rasul et al. (2021b), as well as K^2VAE Wu et al. (2025). Appendix Section D.1 details these baseline families and the probabilistic adaptation of point forecasters. Among these baselines, TimeGrad and the GRU-flow variants provide autoregressive recurrent probabilistic controls: they propagate recurrent sequence states while emitting likelihood parameters recursively, whereas VolDy-VAE keeps value generation non-autoregressive and reserves recurrence for scale-state transfer. Appendix Section D.7 compares VolDy-VAE with Gaussian DeepAR Salinas et al. (2020), forecasting-adapted HeTVAE Shukla & Marlin (2022), and forecasting-adapted VHVM Yin & Barucca (2022) to examine the role of a dedicated recurrent scale state.

4.1.3 IMPLEMENTATION DETAILS

VolDy-VAE adopts the same patching strategy (patch length $P = 24$) and MLP-based encoder as K^2VAE Wu et al. (2025) for fair comparison. The decoder features a dual-head structure: a Location Head with linear projection and a Scale Head with a single-layer GRU (hidden dimension $= D$) followed by `Softplus` activation. The patch length controls the trade-off between temporal

Table 1: Comparison on long-term probabilistic forecasting across nine real-world datasets ($H = 720$, except ILI where $H = 60$). Lower CRPS and NMAE are better. Values are means and standard deviations over independent runs. Red and blue denote the lowest and second-lowest distinct displayed means; ties after rounding share a rank. Full results across four horizons appear in Appendix Tables 12, 13.

Model	Metric	ETTm1	ETTm2	ETTh1	ETTh2	Electricity	Traffic	Weather	Exchange	ILI
FITS ICLR'24	CRPS	$0.305_{\pm.024}$	$0.449_{\pm.034}$	$0.348_{\pm.025}$	$0.314_{\pm.022}$	$0.115_{\pm.024}$	$0.374_{\pm.004}$	$0.267_{\pm.003}$	$0.074_{\pm.011}$	$0.211_{\pm.011}$
	NMAE	$0.406_{\pm.072}$	$0.540_{\pm.052}$	$0.468_{\pm.012}$	$0.401_{\pm.022}$	$0.149_{\pm.012}$	$0.453_{\pm.022}$	$0.317_{\pm.021}$	$0.097_{\pm.011}$	$0.245_{\pm.017}$
PatchTST ICLR'23	CRPS	$0.304_{\pm.029}$	$0.229_{\pm.036}$	$0.323_{\pm.020}$	$0.304_{\pm.018}$	$0.127_{\pm.015}$	$0.214_{\pm.001}$	$0.142_{\pm.005}$	$0.097_{\pm.007}$	$0.233_{\pm.019}$
	NMAE	$0.382_{\pm.066}$	$0.288_{\pm.034}$	$0.428_{\pm.024}$	$0.371_{\pm.021}$	$0.164_{\pm.024}$	$0.253_{\pm.012}$	$0.152_{\pm.029}$	$0.126_{\pm.001}$	$0.287_{\pm.023}$
iTransformer ICLR'24	CRPS	$0.455_{\pm.021}$	$0.311_{\pm.024}$	$0.350_{\pm.019}$	$0.542_{\pm.015}$	$0.109_{\pm.044}$	$0.284_{\pm.004}$	$0.133_{\pm.004}$	$0.087_{\pm.023}$	$0.222_{\pm.020}$
	NMAE	$0.490_{\pm.038}$	$0.385_{\pm.042}$	$0.449_{\pm.022}$	$0.667_{\pm.012}$	$0.140_{\pm.009}$	$0.361_{\pm.030}$	$0.147_{\pm.019}$	$0.113_{\pm.015}$	$0.278_{\pm.017}$
Koopa NeurIPS'23	CRPS	$0.295_{\pm.027}$	$0.233_{\pm.025}$	$0.318_{\pm.009}$	$0.293_{\pm.026}$	$0.113_{\pm.018}$	$0.358_{\pm.022}$	$0.140_{\pm.007}$	$0.091_{\pm.012}$	$0.228_{\pm.022}$
	NMAE	$0.377_{\pm.037}$	$0.290_{\pm.033}$	$0.412_{\pm.008}$	$0.286_{\pm.042}$	$0.149_{\pm.025}$	$0.432_{\pm.032}$	$0.162_{\pm.009}$	$0.116_{\pm.022}$	$0.288_{\pm.031}$
TSDiff NeurIPS'23	CRPS	$0.478_{\pm.027}$	$0.344_{\pm.046}$	$0.516_{\pm.027}$	$0.406_{\pm.056}$	$0.478_{\pm.005}$	$0.391_{\pm.002}$	$0.152_{\pm.003}$	$0.082_{\pm.010}$	$0.263_{\pm.022}$
	NMAE	$0.622_{\pm.045}$	$0.416_{\pm.065}$	$0.657_{\pm.017}$	$0.482_{\pm.022}$	$0.622_{\pm.142}$	$0.478_{\pm.006}$	$0.141_{\pm.026}$	$0.142_{\pm.009}$	$0.272_{\pm.020}$
GRU NVP ICLR'21	CRPS	$0.546_{\pm.036}$	$0.561_{\pm.273}$	$0.502_{\pm.039}$	$0.539_{\pm.090}$	$0.114_{\pm.013}$	$0.211_{\pm.004}$	$0.110_{\pm.004}$	$0.079_{\pm.009}$	$0.307_{\pm.005}$
	NMAE	$0.707_{\pm.050}$	$0.749_{\pm.385}$	$0.643_{\pm.046}$	$0.688_{\pm.161}$	$0.144_{\pm.017}$	$0.264_{\pm.006}$	$0.135_{\pm.008}$	$0.103_{\pm.009}$	$0.333_{\pm.005}$
GRU MAF ICLR'21	CRPS	$0.536_{\pm.033}$	$0.272_{\pm.029}$	$0.393_{\pm.043}$	$0.990_{\pm.023}$	$0.106_{\pm.007}$	-	$0.122_{\pm.006}$	$0.160_{\pm.019}$	$0.172_{\pm.034}$
	NMAE	$0.711_{\pm.081}$	$0.355_{\pm.048}$	$0.496_{\pm.019}$	$1.092_{\pm.019}$	$0.136_{\pm.098}$	-	$0.149_{\pm.034}$	$0.182_{\pm.010}$	$0.216_{\pm.014}$
Trans MAF ICLR'21	CRPS	$0.688_{\pm.043}$	$0.355_{\pm.043}$	$0.363_{\pm.053}$	$0.327_{\pm.033}$	-	-	$0.113_{\pm.004}$	$0.148_{\pm.017}$	$0.155_{\pm.018}$
	NMAE	$0.822_{\pm.034}$	$0.475_{\pm.029}$	$0.455_{\pm.025}$	$0.412_{\pm.020}$	-	-	$0.148_{\pm.040}$	$0.191_{\pm.006}$	$0.183_{\pm.019}$
TimeGrad ICML'21	CRPS	$0.621_{\pm.037}$	$0.470_{\pm.054}$	$0.523_{\pm.027}$	$0.445_{\pm.016}$	$0.108_{\pm.003}$	$0.220_{\pm.002}$	$0.113_{\pm.011}$	$0.099_{\pm.015}$	$0.295_{\pm.083}$
	NMAE	$0.793_{\pm.034}$	$0.561_{\pm.044}$	$0.672_{\pm.015}$	$0.550_{\pm.018}$	$0.134_{\pm.004}$	$0.263_{\pm.001}$	$0.136_{\pm.020}$	$0.113_{\pm.016}$	$0.325_{\pm.068}$
CSDI NeurIPS'21	CRPS	$0.448_{\pm.038}$	$0.239_{\pm.035}$	$0.528_{\pm.012}$	$0.302_{\pm.040}$	-	-	$0.087_{\pm.003}$	$0.143_{\pm.020}$	$0.283_{\pm.012}$
	NMAE	$0.578_{\pm.051}$	$0.306_{\pm.040}$	$0.657_{\pm.014}$	$0.382_{\pm.030}$	-	-	$0.102_{\pm.005}$	$0.173_{\pm.020}$	$0.299_{\pm.013}$
NsDiff ICML'25	CRPS	$0.323_{\pm.030}$	$0.238_{\pm.026}$	$0.360_{\pm.018}$	$0.289_{\pm.020}$	$0.096_{\pm.008}$	$0.207_{\pm.003}$	$0.090_{\pm.004}$	$0.076_{\pm.008}$	$0.161_{\pm.010}$
	NMAE	$0.410_{\pm.036}$	$0.295_{\pm.034}$	$0.455_{\pm.020}$	$0.322_{\pm.025}$	$0.126_{\pm.014}$	$0.256_{\pm.010}$	$0.106_{\pm.008}$	$0.094_{\pm.012}$	$0.190_{\pm.012}$
PPM ICML'26	CRPS	$0.311_{\pm0.002}$	$0.171_{\pm0.001}$	$0.325_{\pm0.003}$	$0.187_{\pm0.003}$	$0.094_{\pm0.001}$	$0.226_{\pm0.004}$	$0.096_{\pm0.000}$	$0.067_{\pm0.001}$	$0.163_{\pm0.013}$
	NMAE	$0.362_{\pm0.002}$	$0.211_{\pm0.002}$	$0.405_{\pm0.013}$	$0.241_{\pm0.003}$	$0.120_{\pm0.002}$	$0.284_{\pm0.007}$	$0.102_{\pm0.001}$	$0.080_{\pm0.002}$	$0.182_{\pm0.011}$
K^2VAE ICML'25	CRPS	$0.294_{\pm.026}$	$0.221_{\pm.023}$	$0.314_{\pm.011}$	$0.280_{\pm.014}$	$0.087_{\pm.005}$	$0.200_{\pm.001}$	$0.084_{\pm.003}$	$0.069_{\pm.005}$	$0.142_{\pm.008}$
	NMAE	$0.373_{\pm.032}$	$0.275_{\pm.035}$	$0.396_{\pm.012}$	$0.278_{\pm.020}$	$0.117_{\pm.019}$	$0.248_{\pm.010}$	$0.099_{\pm.009}$	$0.084_{\pm.017}$	$0.167_{\pm.007}$
VolDy-VAE	CRPS	$0.272_{\pm.002}$	$0.164_{\pm.002}$	$0.310_{\pm.001}$	$0.174_{\pm.001}$	$0.080_{\pm.006}$	$0.192_{\pm.003}$	$0.073_{\pm.003}$	$0.065_{\pm.004}$	$0.130_{\pm.005}$
	NMAE	$0.358_{\pm.002}$	$0.205_{\pm.003}$	$0.391_{\pm.001}$	$0.221_{\pm.002}$	$0.111_{\pm.001}$	$0.236_{\pm.009}$	$0.086_{\pm.006}$	$0.077_{\pm.001}$	$0.155_{\pm.006}$

resolution and efficiency; we therefore evaluate its sensitivity in Appendix Table 16 and discuss the sub-patch volatility limitation in Appendix Section E. In Appendix Table 16, the layer number N denotes the depth of the MLP encoder, not the number of GRU layers in the scale head. We implement the model in PyTorch and train on a single NVIDIA RTX 4090 GPU using the Adam optimizer Kingma & Ba (2015) (learning rate $\in [10^{-4}, 10^{-3}]$, batch size 32, max 50 epochs with early stopping patience of 10). Appendix Section D.3 gives the full architecture and training settings.

4.1.4 METRICS

We employ two complementary metrics: Continuous Ranked Probability Score (CRPS) to measure the distributional fit and Normalized Mean Absolute Error (NMAE) to assess point prediction accuracy. CRPS is a strictly proper scoring rule Gneiting & Raftery (2007) and is applicable to both closed-form Gaussian outputs and sample-based probabilistic baselines, making it suitable for cross-family comparison where likelihood values are not uniformly available. Lower values indicate better performance for both metrics. Detailed descriptions of these metrics can be found in Appendix Section D.2.

4.1.5 RESULTS AND ANALYSIS

Table 1 compares methods at the longest forecasting horizon ($H = 720$, or $H = 60$ for ILI). VolDy-VAE achieves the lowest CRPS and NMAE on all nine datasets, ahead of both Gaussian-head point forecasters and probabilistic baselines, including K^2VAE and PPM. Appendix Section D.6 reports the results for all forecasting horizons in Tables 12 and 13.

The NMAE gains show that improved distributional forecasts accompany more accurate central predictions in these experiments. Their relationship to adaptive attenuation depends on scale accuracy: larger scales reduce the location gradient, but can reflect either observation noise or mean-fitting error (Appendix Section C.6). The ablations below examine the role of the recurrent scale head,

Table 2: Component-wise ablation on *ETTh1* and *Electricity*. Van.: vanilla VAE (MSE); w/o P: no prediction loss; w/o R: no reconstruction loss. Lower is better; red marks the best.

Data	H	Metric	Van.	w/o P	w/o R	Ours
ETTh1	96	CRPS	0.287	0.374	0.277	0.243
		NMAE	0.364	0.464	0.354	0.323
	192	CRPS	0.323	0.378	0.282	0.266
		NMAE	0.388	0.470	0.370	0.352
	336	CRPS	0.352	0.386	0.305	0.284
		NMAE	0.409	0.486	0.395	0.376
	720	CRPS	0.355	0.393	0.318	0.310
		NMAE	0.416	0.504	0.398	0.391
Electricity	96	CRPS	0.363	0.259	0.082	0.069
		NMAE	0.362	0.363	0.107	0.089
	192	CRPS	0.363	0.261	0.086	0.075
		NMAE	0.364	0.363	0.112	0.096
	336	CRPS	0.364	0.264	0.099	0.078
		NMAE	0.364	0.366	0.129	0.102
	720	CRPS	0.370	0.269	0.103	0.080
		NMAE	0.370	0.371	0.135	0.111

Table 3: RevIN ablation on *ETTh1* and *Weather*. Lower CRPS and NMAE are better; red marks the best.

Data	H	Metric	w/o RevIN	Ours
ETTh1	96	CRPS	0.480	0.243
		NMAE	0.618	0.323
	192	CRPS	0.476	0.266
		NMAE	0.615	0.352
	336	CRPS	0.470	0.284
		NMAE	0.603	0.376
	720	CRPS	0.567	0.310
		NMAE	0.690	0.391
Weather	96	CRPS	0.084	0.070
		NMAE	0.106	0.077
	192	CRPS	0.097	0.067
		NMAE	0.125	0.080
	336	CRPS	0.092	0.071
		NMAE	0.119	0.086
	720	CRPS	0.098	0.073
		NMAE	0.124	0.086

while Appendix Table 14 compares VolDy-VAE with Gaussian DeepAR, adapted HeTVAE, and forecasting-adapted VHVM.

4.2 Ablation Studies

We evaluate the contribution of the training objectives, non-stationarity handling, likelihood parameterization, and recurrent scale dynamics. Unless otherwise stated, each variant keeps the same data splits, forecasting horizons, encoder, latent dynamics module, and optimization budget as the full model. The no-GRU independent-σ variant in Appendix Table 15 replaces the recurrent scale path with an MLP head while retaining the encoder, latent dynamics, location head, and Gaussian NLL. It tests the contribution of scale-state transfer beyond time-varying marginal scales. Appendix Section D.7 includes Gaussian DeepAR as an autoregressive recurrent comparison.

Effect of objective components We first isolate the reconstruction, prediction, and KL terms in the objective detailed in Appendix Section C.3 (Eq. 9). The w/o Prediction variant removes $\mathcal{L}_{pred}$ and trains only from historical reconstruction; the w/o Reconstruction variant removes $\mathcal{L}_{rec}$; and Vanilla VAE (MSE) replaces the heteroscedastic location-scale NLL with a homoscedastic MSE objective while keeping the variational backbone.

Table 2 shows that the full model achieves the best CRPS and NMAE in every reported setting. Removing prediction supervision causes the largest degradation in point accuracy, especially on long horizons, indicating that direct future supervision is necessary for learning forecast-oriented latent dynamics. Removing reconstruction also consistently hurts performance, which supports the role of historical reconstruction as a regularizer for the latent representation. Finally, the Vanilla VAE (MSE) variant performs worse than the heteroscedastic NLL model on both CRPS and NMAE, showing that the gain does not come only from the VAE backbone but from volatility-aware probabilistic training.

Effect of non-stationarity handling We assess how RevIN affects heteroscedastic forecasting. Table 3 shows that removing RevIN substantially degrades both metrics across all horizons. The degradation is particularly large on *ETTh1*, where the $H = 720$ CRPS increases from 0.310 to 0.567 and NMAE increases from 0.391 to 0.690. This indicates that symmetric normalization and denormalization help the latent and scale modules focus on local volatility dynamics rather than being dominated by global level and variance shifts.

Effect of likelihood family The default VolDy-VAE uses a heteroscedastic Gaussian likelihood. To test whether the results mainly come from a more expressive marginal distribution, we replace this decoder with three alternatives under the same encoder, latent dynamics, RevIN module, training split, and hyperparameter budget: Student-t, Gaussian mixture, and Gaussian plus conditional flow.

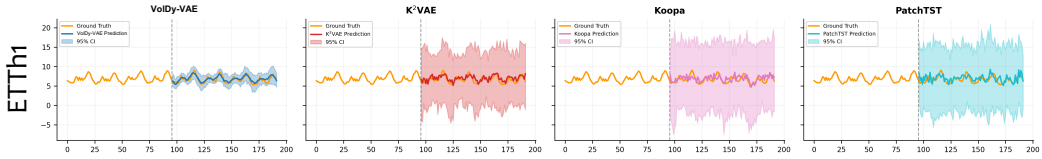

Figure 3: Visual comparison of probabilistic forecasting on *ETTh1*. All subfigures share a unified Y-axis scale. Additional visualization results are provided in Appendix Figures 6 and 7.

Table 4 shows that none of these more flexible likelihoods improves over the Gaussian decoder on the representative long-horizon settings.

Changing the marginal likelihood does not improve the reported CRPS or NMAE in these settings.

We retain the Gaussian location-scale decoder for the evaluated settings and examine scale-head architecture separately in Appendix Section D.8. There, the MLP scale head provides the no-GRU independent-σ comparison under the same encoder, latent dynamics, and likelihood.

Table 4: Likelihood-family ablation at $H = 192$. Lower CRPS and NMAE are better; red marks the best.

Likelihood (parameters)	Metric	ETTh1	Weather
Gaussian (default) (μ, σ)	CRPS	0.266	0.067
	NMAE	0.352	0.080
Student-t (μ, σ, ν)	CRPS	0.286	0.071
	NMAE	0.372	0.084
Gaussian mixture $\{\pi_k, \mu_k, \sigma_k\}_{k=1}^{K}$	CRPS	0.291	0.074
	NMAE	0.377	0.087
Gaussian + flow $(\mu, \sigma, \psi_{\text{flow}})$	CRPS	0.284	0.070
	NMAE	0.370	0.083

4.3 Volatility Dynamics and Calibration

Table 5 compares MLP, LSTM, and GRU scale heads with the remaining architecture and training objective fixed. It reports interval coverage, sharpness, quantile calibration error, and scale autocorrelation; definitions appear in Appendix Section D.2. Across both datasets and horizons, the GRU produces more persistent scale estimates together with better calibrated, narrower intervals. LSTM falls between MLP and GRU on the reported mean values.

Scale persistence alone could reflect excessive smoothing. Here, it accompanies improved coverage and calibration without sacrificing sharpness, supporting the value of recurrent scale memory. Because all heads use the same Gaussian NLL, the comparison isolates the effect of the scale-state architecture within joint location–scale training. Appendix Section B examines predicted scales against known synthetic volatility trajectories, with the forecasts and recovered scales shown in Appendix Figure 5.

Table 5: Volatility and calibration on ETTh1 and ETTh2 ($L = 96$). Each cell shows mean $\pm$ std over six seeds. Coverage is best near 95%; lower sharpness and QICE are better. ρ_σ measures persistence. Red marks the best within each setting.

Data	H	Head	Cov.95	Sharp.	QICE	ρ_σ
ETTh1	96	MLP	$91.2_{\pm.8}$	$1.84_{\pm.05}$	$2.91_{\pm.18}$	$0.61_{\pm.03}$
		LSTM	$93.6_{\pm.6}$	$1.62_{\pm.04}$	$1.94_{\pm.14}$	$0.78_{\pm.02}$
		GRU	$94.4_{\pm.5}$	$1.53_{\pm.03}$	$1.58_{\pm.11}$	$0.84_{\pm.02}$
	192	MLP	$90.5_{\pm.9}$	$1.91_{\pm.06}$	$3.24_{\pm.22}$	$0.58_{\pm.04}$
		LSTM	$93.1_{\pm.7}$	$1.70_{\pm.05}$	$2.18_{\pm.16}$	$0.76_{\pm.03}$
		GRU	$94.1_{\pm.6}$	$1.60_{\pm.04}$	$1.71_{\pm.12}$	$0.83_{\pm.02}$
ETTh2	96	MLP	$90.8_{\pm.7}$	$1.76_{\pm.04}$	$2.84_{\pm.17}$	$0.63_{\pm.03}$
		LSTM	$93.9_{\pm.5}$	$1.55_{\pm.03}$	$1.83_{\pm.13}$	$0.80_{\pm.02}$
		GRU	$94.7_{\pm.4}$	$1.46_{\pm.03}$	$1.42_{\pm.10}$	$0.86_{\pm.01}$
	192	MLP	$89.9_{\pm1.0}$	$1.83_{\pm.05}$	$3.09_{\pm.21}$	$0.59_{\pm.04}$
		LSTM	$93.4_{\pm.6}$	$1.61_{\pm.04}$	$2.01_{\pm.15}$	$0.78_{\pm.03}$
		GRU	$94.5_{\pm.5}$	$1.50_{\pm.03}$	$1.55_{\pm.11}$	$0.85_{\pm.02}$

4.4 Qualitative Analysis

Figure 3 compares forecasts on *ETTh1* ($L = 96, H = 96$) from VolDy-VAE, K^2VAE, Koopa, and PatchTST. VolDy-VAE's intervals expand around local changes and contract in more stable periods, while adjacent forecast steps retain similar widths. The comparison models produce wider, more uniform bands in this example. Appendix Section F shows additional volatile and stable windows.

Table 6: Transferability at $L = 96, H = 96$. Each cell shows mean $\pm$ std over six seeds. Full horizons are in Appendix Table 9.

	CRPS		NMAE	
Dataset	Base	VolDy	Base	VolDy
TimeGAN				
ETTh1	$0.277_{\pm.011}$	$0.262_{\pm.009}$	$0.347_{\pm.012}$	$0.339_{\pm.010}$
Weather	$0.106_{\pm.006}$	$0.068_{\pm.004}$	$0.117_{\pm.007}$	$0.081_{\pm.005}$
Exchange	$0.056_{\pm.004}$	$0.047_{\pm.003}$	$0.071_{\pm.005}$	$0.062_{\pm.004}$
K^2VAE				
ETTh1	$0.264_{\pm.020}$	$0.251_{\pm.017}$	$0.336_{\pm.041}$	$0.332_{\pm.012}$
Weather	$0.080_{\pm.007}$	$0.065_{\pm.004}$	$0.086_{\pm.011}$	$0.076_{\pm.004}$
Exchange	$0.031_{\pm.002}$	$0.029_{\pm.002}$	$0.032_{\pm.002}$	$0.029_{\pm.002}$
PatchTST				
ETTh1	$0.312_{\pm.036}$	$0.268_{\pm.020}$	$0.407_{\pm.022}$	$0.348_{\pm.018}$
Weather	$0.131_{\pm.007}$	$0.074_{\pm.006}$	$0.145_{\pm.016}$	$0.085_{\pm.006}$
Exchange	$0.063_{\pm.006}$	$0.061_{\pm.004}$	$0.079_{\pm.002}$	$0.066_{\pm.004}$

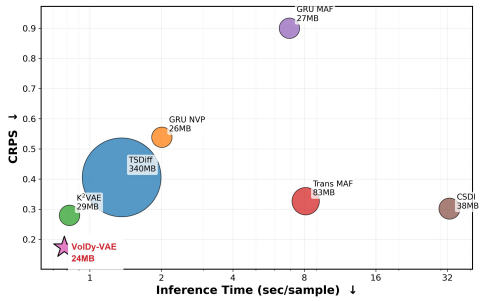

Figure 4: Forecasting quality and inference cost on *ETTh2* ($L = 96, H = 720$). Horizontal position gives inference time per sample and vertical position gives CRPS, so the lower-left region is preferable. Bubble area encodes inference memory in MB; the star marks VolDy-VAE.

4.5 Generalizability Analysis

We add the location-scale uncertainty component to TimeGAN Yoon et al. (2019), K^2VAE, and PatchTST while matching each base model's data split, horizon, seed set, optimizer, and early-stopping protocol. Appendix Table 8 gives implementation details. Table 6 shows lower CRPS and NMAE for all three augmented backbones at $H = 96$; Appendix Table 9 extends the comparison across horizons. The gains span generative, VAE-based, and Transformer-based models in the evaluated settings.

4.6 Efficiency Analysis

We measure inference latency and memory on *ETTh2* with batch size 1 on an NVIDIA RTX 4090, generating 100 samples for probabilistic baselines. Figure 4 places VolDy-VAE in the low-CRPS, low-latency region with a small memory footprint. Its direct forecast path avoids iterative denoising and recursive value generation while transferring a compact scale state. Appendix Section D.4 reports the detailed measurements in Table 10.

5 Conclusions

VolDy-VAE provides an easy yet effective approach to probabilistic long-horizon forecasting. It predicts future locations in parallel and carries historical volatility through a single recurrent scale state, allowing uncertainty to evolve across the forecast horizon without recursive value generation. Historical reconstruction and future prediction share a heteroscedastic likelihood, so the location and scale paths are learned together.

Across nine benchmarks, VolDy-VAE obtains the lowest CRPS and NMAE at the longest evaluated horizons. Component and scale-head ablations, calibration diagnostics, and synthetic volatility analyses support the role of recurrent scale memory. Removing RevIN degrades forecasting quality, whereas the tested more expressive likelihoods do not improve the representative long-horizon results. These ablations locate the gains in volatility modeling and non-stationarity handling. The transferability experiments show that the location-scale component also improves the evaluated backbones, while the efficiency comparison places VolDy-VAE among the low-latency forecasters. Under known variances, the accompanying analysis explains the benefit of weighting observations by their uncertainty. Finer-scale volatility updates for abrupt within-patch changes remain an important direction within this design. Together, these results suggest that direct value generation and explicit scale dynamics form a useful design for probabilistic forecasting.

AI USE STATEMENT

OpenAI's Codex assisted with manuscript language editing, wording refinement, LaTeX formatting, and consistency checks. The authors are responsible for the experiments, numerical values, scientific claims, citations, and final submission.

ETHICS STATEMENT

This work studies probabilistic long-horizon forecasting on public benchmark datasets covering energy, transportation, weather, exchange rates, and influenza-like illness. It does not involve human-subject recruitment, new dataset collection, personally identifiable records, or deployment in a real-world decision system. The main societal consideration is the downstream use of inaccurate or miscalibrated forecasts in consequential decisions.

REPRODUCIBILITY STATEMENT

Section 3 specifies the forecasting formulation, model components, and training objective. Section 4 describes the datasets, forecasting horizons, baselines, metrics, and evaluation protocol. Appendix Section D collects the baseline descriptions, metric definitions, implementation details, and full horizon-wise results; Appendix Section D.9 reports hyperparameter sensitivity. Model code and run scripts accompany the manuscript.

REFERENCES

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. GluonTS: Probabilistic and neural time series modeling in python. *J. Mach. Learn. Res.*, 21(116):1–6, 2020.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

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

Shibo Feng, Chunyan Miao, Ke Xu, Jiaxiang Wu, Pengcheng Wu, Yang Zhang, and Peilin Zhao. Multi-Scale Attention Flow for Probabilistic Time Series Forecasting. *IEEE Trans. Knowl. Data Eng.*, 36(5):2056–2068, 2024a. doi: 10.1109/TKDE.2023.3319672.

Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent Diffusion Transformer for Probabilistic Time Series Forecasting. In *Proc. Conf. AAAI*, pp. 11979–11987, 2024b.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. Int. Conf. Mach. Learn.*, pp. 1050–1059, 2016.

Tilmann Gneiting and Adrian E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *J. Amer. Statist. Assoc.*, 102(477):359–378, 2007.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2): 243–268, 2007.

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

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The Capacity and Robustness Trade-Off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Trans. Knowl. Data Eng.*, 36(11):7129–7142, 2024. doi: 10.1109/TKDE.2024.3400008.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecasting*, 32(3):914–938, 2016.

Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. HDMixer: Hierarchical Dependency with Extendable Patch for Multivariate Time Series Forecasting. In *Proc. Conf. AAAI*, pp. 12608–12616, 2024.

Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate Time Series Forecasting With Dynamic Graph Neural ODEs. *IEEE Trans. Knowl. Data Eng.*, 35(9): 9168–9180, 2023. doi: 10.1109/TKDE.2022.3221989.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Inf. Process. Syst.*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *Proc. Int. Conf. Learn. Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6034.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. Int. Conf. Learn. Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *Proc. Int. Conf. Learn. Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang (Bernie) Wang. Predict, Refine, Synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Advances in Neural Inf. Process. Syst.*, pp. 28341–28364, 2023.

Zhichen Lai, Dalin Zhang, Huan Li, Christian S. Jensen, Hua Lu, and Yan Zhao. LightCTS*: Lightweight correlated time series forecasting enhanced with model distillation. *IEEE Trans. Knowl. Data Eng.*, 36(12):8695–8710, 2024. doi: 10.1109/TKDE.2024.3424451.

Miguel Lázaro-Gredilla and Michalis K Titsias. Variational Heteroscedastic Gaussian Process Regression. In *Proc. Int. Conf. Mach. Learn.*, pp. 841–848, 2011.

Radmir Mishelevich Leushuis. Probabilistic forecasting with VAR-VAE: Advancing time series forecasting under uncertainty. *Information Sciences*, 713:122184, 2025. doi: 10.1016/j.ins.2025.122184.

Jinglin Li, Jun Tan, Qi Fang, and Ning Gui. Parametric Prior Mapping Framework for Non-stationary Probabilistic Time Series Forecasting. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/61351.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Inf. Process. Syst.*, pp. 12271–12290, 2023.

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 *Proc. Int. Conf. Learn. Representations*, pp. 11116–11140, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ea18fdc667e0ef2ad82b2b4d65147ad-Abstract-Conference.html.

Osama Makansi, Eddy Ilg, Özgün Çiçek, and Thomas Brox. Overcoming Limitations of Mixture Density Networks: A Sampling and Fitting Framework for Multimodal Future Prediction. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 7144–7153, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/papers/Makansi_Overcoming_Limitations_of_Mixture_Density_Networks_A_Sampling_and_Fitting_CVPR_2019_paper.pdf.

Nam Nguyen and Brian Quanz. Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting. In *Proc. Conf. AAAI*, pp. 9117–9125, 2021.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *Proc. Int. Conf. Learn. Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/10876.

David A. Nix and Andreas S. Weigend. Estimating the Mean and Variance of the Target Probability Distribution. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pp. 55–60, 1994.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Inf. Process. Syst.*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Pedro Pessoa, Paul Campitelli, Douglas P. Shepherd, S. Banu Ozkan, and Steve Pressé. Mamba time series forecasting with uncertainty quantification. *Machine Learning: Science and Technology*, 6 (3):035012, 2025. doi: 10.1088/2632-2153/adec3b.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Inf. Process. Syst.*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/5cf68969fb67aa6082363a6d4e6468e2-Abstract.html.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *Proc. Int. Conf. Mach. Learn.*, pp. 8857–8868, 2021a.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *Proc. Int. Conf. Learn. Representations*, 2021b. URL https://iclr.cc/virtual/2021/poster/3117.

Robert A Rigby and D Mikis Stasinopoulos. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 54(3):507–554, 2005.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *Int. J. Forecasting*, 36(3):1181–1191, 2020.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antić, and Georg Martius. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *Proc. Int. Conf. Learn. Representations*, 2022. URL https://openreview.net/forum?id=aPOpXlnV1T.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.*, 90:106181, 2020.

Satya Narayan Shukla and Benjamin M. Marlin. Heteroscedastic Temporal Variational Autoencoder For Irregularly Sampled Time Series . In *Proc. Int. Conf. Learn. Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6271.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Inf. Process. Syst.*, pp. 24804–24816, 2021.

Jieting Wang, Huimei Shi, Feijiang Li, and Xiaolei Shang. Beyond MSE: Ordinal Cross-Entropy for Probabilistic Time Series Forecasting. In *Proc. Conf. AAAI*, pp. 26269–26276, 2026.

Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. K^2VAE: A Koopman-Kalman Enhanced Variational AutoEncoder for Probabilistic Time Series Forecasting. In *Proc. Int. Conf. Mach. Learn.*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67562–67583, 2025. URL https://proceedings.mlr.press/v267/wu25w.html.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling Time Series with 10k Parameters. In *Proc. Int. Conf. Learn. Representations*, pp. 26295–26318, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/701251e1db4a2e4dd2ef23f5265d5936-Abstract-Conference.html.

Hao Xue and Flora D. Salim. PromptCast: A new prompt-based learning paradigm for time series forecasting. *IEEE Trans. Knowl. Data Eng.*, 36(11):6851–6864, 2024. doi: 10.1109/TKDE.2023.3342137.

Kaiwen Yan, Chen Long, Huisi Wu, and Zhenkun Wen. Multi-Resolution Expansion of Analysis in Time-Frequency Domain for Time Series Forecasting. *IEEE Trans. Knowl. Data Eng.*, 36(11):6667–6680, 2024. doi: 10.1109/TKDE.2024.3396785.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary Diffusion for Probabilistic Time Series Forecasting. In *Proc. Int. Conf. Mach. Learn.*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72112–72130, 2025. URL https://proceedings.mlr.press/v267/ye25i.html.

Zexuan Yin and Paolo Barucca. Variational heteroscedastic volatility model. *arXiv preprint arXiv:2204.05806*, 2022.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series Generative Adversarial Networks. In *Advances in Neural Inf. Process. Syst.*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html.

Taeho Yoon, Youngsuk Park, Ernest K. Ryu, and Yuyang Wang. Robust probabilistic time series forecasting. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 1336–1358. PMLR, 2022.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In *Proc. Int. Conf. Learn. Representations*, pp. 41582–41610, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b5b66077d016c037576cc56a82f97f66-Abstract-Conference.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? In *Proc. Conf. AAAI*, pp. 11121–11128, 2023.

Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. ProbTS: Benchmarking Point and Distributional Forecasting across Diverse Prediction Horizons. In *Advances in Neural Inf. Process. Syst.*, pp. 48045–48082, 2024.

A Extended Related Work

A.1 Heteroscedasticity and Distributional Forecasting

Probabilistic forecasting often models time-varying volatility, or heteroscedasticity, in future states. Related methods can be grouped into deep autoregressive models, distributional regression, and stochastic process-based approaches. Recent probabilistic forecasting work has further explored multi-scale attention flows to capture uncertainty at different temporal resolutions Feng et al. (2024a) and ordinal objectives that move beyond MSE-style regression Wang et al. (2026).

A.1.1 Deep Autoregressive Models

DeepAR Salinas et al. (2020) uses Recurrent Neural Networks (RNNs) to model hidden state transitions and emit likelihood parameters at each time step. Although DeepAR is often instantiated with Gaussian likelihoods, implementations such as GluonTS Alexandrov et al. (2020) also support heavy-tailed distributions. DeepState Rangapuram et al. (2018) combines state space models with RNNs. In multivariate forecasting, recent TKDE studies also model inter-variable dependencies through dynamic graph neural ODEs Jin et al. (2023) or lightweight correlated-series distillation Lai et al. (2024). These models maintain recurrent sequence states, but the recurrence mainly serves autoregressive value generation and likelihood-parameter emission. It is not used as a dedicated volatility state transferred from the look-back window to a directly generated future horizon. VolDy differs from RNN-based variance emission by combining look-back volatility-state transfer, non-autoregressive horizon generation, and scale-only recurrent memory within a forecasting VAE. Recursive decoding also limits parallelization and can accumulate errors over long forecasting horizons.

A.1.2 Distributional Regression and MDNs

Distributional regression methods have also been adapted for forecasting. GAMLSS Rigby & Stasinopoulos (2005) models location, scale, and shape parameters within a statistical regression framework. Mixture Density Networks (MDNs) Makansi et al. (2019) represent multimodal densities through weighted distributional mixtures. In practice, MDNs can be difficult to train because of numerical instability and mode collapse, and they are usually used in step-wise regression rather than global sequence generation.

A.1.3 Gaussian Processes and Volatility Models

Bayesian non-parametric methods offer principled uncertainty quantification. Heteroscedastic Gaussian Processes Lázaro-Gredilla & Titsias (2011) explicitly model input-dependent volatility. However, GP-based methods generally suffer from cubic computational complexity $\mathcal{O}(N^3)$, limiting their scalability to long multivariate time series. In the econometrics literature, the GARCH family Bollerslev (1986) provides a principled framework for modeling temporal volatility persistence through autoregressive variance recursions. However, classical GARCH models are constrained to linear variance dynamics with globally fixed coefficients, operate on scalar residuals rather than rich latent representations, and are primarily designed for univariate financial return series rather than general-purpose multivariate long-horizon forecasting. Recent volatility-learning approaches further connect probabilistic forecasting with conditional heteroscedasticity. VHVM Yin & Barucca (2022) combines a VAE with recurrent dynamics to estimate time-varying covariance matrices for multivariate financial returns, making it conceptually close to neural volatility modeling. Its target, however, is financial covariance/volatility estimation rather than direct long-horizon sequence forecasting with a non-autoregressive future generator.

A.2 Generative Forecasting Models

Deep generative forecasters model distributional shape more directly than fixed parametric likelihoods. Related non-generative forecasting advances also exploit linear baselines Zeng et al. (2023), hierarchical patch/mixer designs Huang et al. (2024), time-frequency multi-resolution analysis Yan et al. (2024), and prompt-based learning paradigms Xue & Salim (2024). We group the generative

forecasting literature into iterative diffusion models, state space models, and variational latent variable models. PPM Li et al. (2026) offers another direct approach: it constructs an input-conditioned parametric prior and learns a mapping to predictive samples.

A.2.1 Iterative Denoising Models

Denoising Diffusion Probabilistic Models (DDPMs), such as TimeGrad Rasul et al. (2021a) and CSDI Tashiro et al. (2021), formulate forecasting as a conditional denoising process. Models such as Latent Diffusion Transformer Feng et al. (2024b), NsDiff Ye et al. (2025), and DiffusionTS Yuan & Qiao (2024) adapt this framework to non-stationary time series. Diffusion models can represent flexible predictive distributions, but inference requires dozens or hundreds of denoising steps. This iterative sampling cost makes them expensive for real-time forecasting.

A.2.2 State Space Models (SSMs)

Structured State Space Models such as Mamba Gu & Dao (2023) scale linearly with sequence length, $\mathcal{O}(L)$. Approaches such as Mamba-ProbTSF Pessoa et al. (2025) adapt this architecture to probabilistic forecasting. These models are efficient for long sequences, but they typically function as generic sequence encoders. They often do not explicitly separate signal dynamics from uncertainty dynamics, which can make rapid regime changes or extreme volatility harder to represent. Recent multivariate forecasting analysis has also shown that channel-independent designs involve a capacity–robustness trade-off Han et al. (2024), further highlighting the need to distinguish generic sequence modeling choices from mechanisms that explicitly encode uncertainty dynamics.

A.2.3 Variational Autoencoders (VAEs) and Temporal Uncertainty

VAEs support efficient one-step generation. Forecasting VAEs, such as TimeVAE Desai et al. (2021), temporal latent auto-encoders Nguyen & Quanz (2021), and K^2VAE Wu et al. (2025), model long-term nonlinear structure but often rely on MSE-based reconstruction. This objective implicitly assumes a fixed-variance Gaussian likelihood. VAR-VAE Leushuis (2025) also combines VAE-based probabilistic forecasting with an autoregressive latent transition, showing the usefulness of latent temporal structure for uncertainty-aware forecasting under noisy conditions. Its latent VAR component models value dynamics in the latent space, whereas VolDy-VAE introduces a dedicated recurrent scale path for future volatility evolution and evaluates this mechanism under long-horizon probabilistic forecasting benchmarks. While models like HeTVAE Shukla & Marlin (2022) have integrated heteroscedastic modeling into the VAE framework, they are designed for different tasks, namely imputation and interpolation of irregularly sampled data, requiring complex attention mechanisms and ODE solvers that are neither optimized for nor applicable to efficient long-horizon forecasting. To make this distinction empirical rather than only conceptual, we additionally compare against a Gaussian DeepAR baseline and adapt HeTVAE to the long-horizon forecasting setting in Section D.7. This leaves a gap for a framework that explicitly models the temporal dynamics of uncertainty (volatility evolution, regime persistence) while retaining efficient long-horizon generation.

A.3 Positioning: Temporal Uncertainty Dynamics

To clarify the novelty of our approach relative to representative heteroscedastic and probabilistic forecasting families, we provide a structured comparison in Table 7. Existing methods cover different parts of the design space: autoregressive distributional forecasters maintain generic recurrent states, diffusion models provide expressive sample generation, and forecasting VAEs offer efficient direct generation. VolDy targets the less explored intersection of: (1) an explicit volatility state for modeling how uncertainty evolves across time, (2) forecasting-specialized long-horizon prediction, and (3) efficient direct generation.

VolDy-VAE addresses this gap with a recurrent scale head that carries volatility information within a VAE forecasting framework. This positions our work as a forecasting framework for temporal uncertainty dynamics, distinguished from standard heteroscedastic regression by its recurrent scale state. The diagnostic comparisons in Section D.7 further support this distinction: generic autoregressive recurrent likelihood modeling and heteroscedastic likelihood modeling without a forecasting-oriented recurrent scale state remain weaker on the long-horizon benchmarks.

Table 7: Positioning of VolDy relative to representative heteroscedastic and probabilistic forecasting families. "Explicit Vol. State" indicates whether the approach uses a dedicated state for the evolution and persistence of volatility regimes, as opposed to generic sequence states or independent per-step variance estimation. △ denotes partial support through a generic recurrent or generative state rather than an explicit volatility state. "Plug-in Evidence" indicates whether the temporal uncertainty mechanism is evaluated beyond a single base architecture.

Approach	Forecasting Task	Explicit Vol. State	Direct Generation	Plug-in Evidence
Kendall & Gal Kendall & Gal (2017)	Regression	✗	–	✗
HeTVAE Shukla & Marlin (2022)	Imputation	✗	✓	✗
DeepAR/DeepState Salinas et al. (2020); Rangapuram et al. (2018)	Forecasting	△	✗	✗
Diffusion PTSF Rasul et al. (2021a); Tashiro et al. (2021); Ye et al. (2025)	Forecasting	△	✗	✗
K^2VAE Wu et al. (2025)	Forecasting	✗	✓	✗
VolDy principle/plug-in	Forecasting	✓	✓	✓

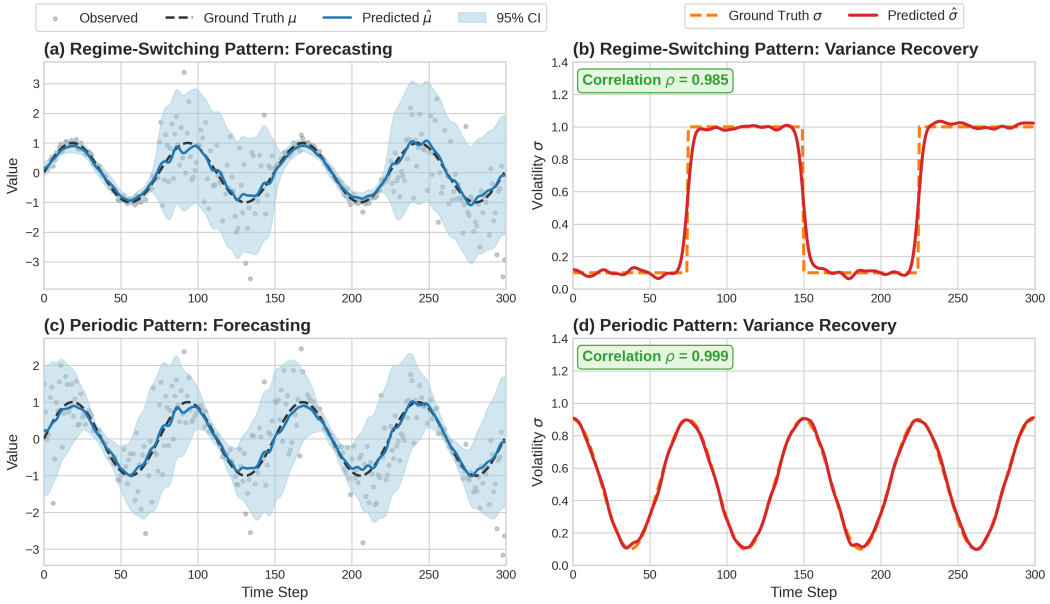

Figure 5: Qualitative results on synthetic datasets. Left: forecasting with adaptive 95% confidence intervals under varying volatility. Right: uncertainty recovery, where the predicted volatility $\hat{\sigma}_t$ closely tracks the ground-truth temporal dynamics.

B Experiments on Synthetic Data

In real-world empirical studies, the true aleatoric uncertainty is inherently latent and unobservable, making it difficult to rigorously verify whether a model captures volatility dynamics or merely overfits residual artifacts Kendall & Gal (2017). We therefore validate the temporal uncertainty modeling capability of VolDy-VAE in a controlled analytical environment where both the underlying trend and the volatility profile are mathematically defined and known Gal & Ghahramani (2016). This experiment tests whether the model can disentangle the deterministic signal (Location Head) from temporally evolving volatility (Scale Head with volatility dynamics).

B.1 Data Generating Process

We analyze univariate sequences governed by the process $y_t = \mu_t + \epsilon_t$, where the underlying deterministic trend is defined as $\mu_t = \sin(t)$ and the stochastic term follows $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$. To stress-test the model under extreme heteroscedasticity, we introduce two distinct volatility profiles as illustrated in Fig. 5. The first is a Regime-Switching scenario (Top Row), which simulates structural breaks by having σ_t alternate abruptly between stable (0.1) and high-volatility (1.0) states, thereby

challenging the model's adaptability to sudden distributional shifts. The second is a Periodic pattern (Bottom Row), defined by a continuous volatility trajectory $\sigma_t = 0.5 + 0.4\cos(t)$; this profile is deliberately phase-shifted relative to the mean to rigorously test the model's ability to decouple correlated but distinct dynamics.

B.2 Results and Analysis

Fig. 5 visualizes the forecasting and variance recovery performance. In the Regime-Switching scenario (a-b), the predicted scale $\hat{\sigma}_t$ closely tracks the ground-truth step function with high correlation ($\rho = 0.985$). This estimation allows the confidence intervals to expand during volatility spikes and contract during stable periods without distorting the trend prediction. Similarly, in the Periodic scenario (c-d), the model achieves near-perfect disentanglement ($\rho = 0.999$), effectively distinguishing the signal's phase from the cyclic volatility profile. These results show that the volatility dynamics module captures the temporal evolution of uncertainty: the predicted $\hat{\sigma}_t$ transitions smoothly between regimes rather than producing noisy per-step estimates.

C Theoretical Analysis

We derive the heteroscedastic reconstruction likelihood from the variational lower bound and analyze the trade-off introduced by scale-dependent gradients. We then examine local scale optimization and establish an inverse-variance weighting result for a constant-mean model with known regime-dependent variances, followed by a discussion of estimated scales.

C.1 Connection to Classical Volatility Models

The recurrent volatility dynamics module draws on the persistence principle of the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) family Bollerslev (1986). A standard GARCH(1,1) model defines the conditional variance as $\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$, where the current variance depends linearly on the previous residual and variance. Our GRU-based formulation borrows this inductive bias in three forecasting-oriented ways: (i) the gating mechanism can learn input-dependent persistence in the scale state rather than globally fixed coefficients; (ii) the nonlinear transition can represent context-dependent volatility responses; and (iii) the volatility state is conditioned on the latent representation $\mathbf{Z}_t$ rather than scalar residuals, enabling the model to leverage multivariate contextual information for uncertainty estimation. Together, these choices define an amortized neural parameterization of the predictive likelihood that captures context-dependent volatility persistence in multivariate long-horizon forecasting. Explicit stochastic-volatility or GARCH priors provide a complementary route when structural volatility inference is the scientific target.

C.2 Role of Temporal Volatility Dynamics

A natural question is why estimating σ_t independently at each time step, as in standard heteroscedastic regression, is insufficient. Let $\{\hat{\sigma}_t^{\text{ind}}\}$ denote variance estimates from an independent per-step estimator and $\{\hat{\sigma}_t^{\text{dyn}}\}$ denote estimates from a recurrent volatility dynamics module. Under the common regime-switching pattern where the true volatility is locally persistent and the recurrent state learns to pool information from adjacent steps without oversmoothing regime boundaries, temporal sharing can reduce the variance of scale estimates within a stable regime. An independent estimator must infer σ_t from a single latent vector $\mathbf{z}_t$, treating each step as an isolated regression problem. In contrast, the recurrent estimator uses the volatility hidden state $\mathbf{h}_t^{\text{vol}}$ to aggregate evidence from neighboring latent states. When adjacent steps share the same volatility regime, this pooling behaves like a learned temporal smoother and reduces step-wise scale noise. At regime boundaries, the GRU gates can limit smoothing lag by adapting how much past volatility state is retained. This conditional intuition is consistent with the synthetic experiments in Appendix Section B, where the predicted $\hat{\sigma}_t$ tracks the ground-truth volatility with high temporal coherence ($\rho > 0.98$) rather than producing noisy per-step estimates.

C.3 Objective Function: Volatility-Aware Forecasting

The training objective combines heteroscedastic reconstruction and prediction likelihoods with KL regularization. Appendix Section C.5 derives the reconstruction likelihood term from the ELBO; Eq. 8 specifies the composite forecasting objective and its KL weight.

The Composite Location-Scale Loss. We adopt the Gaussian Negative Log-Likelihood (NLL) for both the look-back window reconstruction ($\mathcal{L}_{rec}$) and the future horizon prediction ($\mathcal{L}_{pred}$). The total loss function $\mathcal{L}$ is defined as:

$$\begin{aligned}\mathcal{L} = & \underbrace{\mathcal{L}_{rec}(\boldsymbol{\mathcal{X}})}_{\text{Reconstruction}} + \underbrace{\mathcal{L}_{pred}(\boldsymbol{\mathcal{Y}})}_{\text{Prediction}} \\ & + \beta \cdot D_{KL}\left(q_\phi(\mathbf{Z}_{past} \mid \boldsymbol{\mathcal{P}}) \parallel \mathcal{N}(0, \mathbf{I})\right).\end{aligned} \tag{9}$$

Both $\mathcal{L}_{rec}$ and $\mathcal{L}_{pred}$ are computed using the heteroscedastic Gaussian NLL. For a generic sequence $\mathbf{u}$ of length T (where $\mathbf{u}$ is $\mathbf{x}$ or $\mathbf{y}$), the loss is:

$$\mathcal{L}_{\text{NLL}}(\mathbf{u}) = \frac{1}{TC} \sum_{t,c} \left(\log \boldsymbol{\sigma}_{t,c} + \frac{(\mathbf{u}_{t,c} - \boldsymbol{\mu}_{t,c})^2}{2\boldsymbol{\sigma}_{t,c}^2} \right). \tag{10}$$

The two likelihood terms train the latent representation to support historical reconstruction and future prediction, with learned scales weighting the corresponding residuals.

C.4 Adaptive Attenuation and Its Trade-off

For a fixed residual, the Gaussian NLL scales the location gradient by the inverse predicted variance:

$$\frac{\partial \mathcal{L}_{\text{NLL}}}{\partial \mu_{t,c}} = \frac{\mu_{t,c} - u_{t,c}}{TC\,\sigma_{t,c}^2}. \tag{11}$$

When predicted scales reflect observation noise, this weighting reduces the influence of noisy observations on location learning. When scales absorb errors in the current mean predictor, the same mechanism can suppress learning of predictable structure and sustain an underfit mean Seitzer et al. (2022). A recurrent scale state shares information over time, but temporal persistence alone does not distinguish these two sources of residual variation.

Appendix Section C.6 develops this trade-off and its connection to β-NLL. Theorem 1 gives an efficiency result for known variances in a constant-mean model; applying that rationale to learned scales requires checking their accuracy. Joint neural optimization and mean-fitting quality are separate questions from the weighted-estimation result.

C.5 Derivation of the Heteroscedastic ELBO

The standard Variational Autoencoder (VAE) maximizes the Evidence Lower Bound (ELBO) of the marginal likelihood $p_\theta(\boldsymbol{\mathcal{X}})$. For a time series $\boldsymbol{\mathcal{X}}$ and latent variable $\mathbf{Z}$, the ELBO is defined as:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = & \mathbb{E}_{q_\phi(\mathbf{Z}|\boldsymbol{\mathcal{X}})}[\log p_\theta(\boldsymbol{\mathcal{X}}|\mathbf{Z})] \\ & - D_{KL}(q_\phi(\mathbf{Z}|\boldsymbol{\mathcal{X}}) \parallel p(\mathbf{Z})).\end{aligned} \tag{12}$$

We focus on the reconstruction term $\mathcal{L}_{rec} = \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{\mathcal{X}}|\mathbf{Z})]$.

In vanilla VAEs, the decoder assumes a fixed variance $\sigma^2 = c$, implying $p_\theta(\mathbf{x}_t|\mathbf{Z}) = \mathcal{N}(\boldsymbol{\mu}_t, c\mathbf{I})$. This simplifies to the standard MSE loss:

$$\log p_\theta(\boldsymbol{\mathcal{X}}|\mathbf{Z}) \propto -\sum_t \|\mathbf{x}_t - \boldsymbol{\mu}_t\|_2^2 + \text{const.} \tag{13}$$

In VolDy-VAE, we relax this assumption to model a heteroscedastic Gaussian likelihood. The observation model remains conditionally factorized given the latent sequence and decoder states:

$$p_\theta(\boldsymbol{\mathcal{X}}|\mathbf{Z}) = \prod_{t=1}^{T} \prod_{c=1}^{C} p_\theta(\mathbf{x}_{t,c}|\mathbf{Z}, \mathbf{h}_t^{\text{vol}}), \tag{14}$$

where the mean $\boldsymbol{\mu}_t$ is decoded directly from the latents, while the scale $\boldsymbol{\sigma}_t$ is decoded through the recurrent volatility state. Specifically:

$$p_\theta(\mathbf{x}_t|\mathbf{Z}) = \prod_{c=1}^{C} \frac{1}{\sqrt{2\pi}\boldsymbol{\sigma}_{t,c}} \exp\left(-\frac{(\mathbf{x}_{t,c} - \boldsymbol{\mu}_{t,c})^2}{2\boldsymbol{\sigma}_{t,c}^2}\right). \tag{15}$$

Thus, temporal dependence in uncertainty is represented through the deterministic recurrent state that parameterizes $\boldsymbol{\sigma}_t$, rather than through a separate stochastic volatility prior over σ_t. Taking the log-likelihood and omitting constants, with $\Delta_{t,c} = \mathbf{x}_{t,c} - \boldsymbol{\mu}_{t,c}$:

$$\log p_\theta(\mathbf{x}_t|\mathbf{Z}) \propto -\sum_{c=1}^{C} \left(\log \boldsymbol{\sigma}_{t,c} + \frac{\Delta_{t,c}^2}{2\boldsymbol{\sigma}_{t,c}^2}\right). \tag{16}$$

Maximizing this term is equivalent, up to averaging and additive constants, to minimizing the Gaussian NLL in Eq. 10. This derives the heteroscedastic reconstruction term used in the composite objective.

C.6 Gradient Scaling and Mean Underfitting

For a single observation x, omitting subscripts, the Gaussian NLL is

$$J(\mu, \sigma) = \log \sigma + \frac{(x-\mu)^2}{2\sigma^2}, \qquad \sigma > 0. \tag{17}$$

Its partial derivative with respect to the location output is

$$\frac{\partial J}{\partial \mu} = \frac{\mu - x}{\sigma^2}. \tag{18}$$

Proposition 1 (Gradient Scaling). *Holding the residual fixed, the magnitude of the location-output gradient is inversely proportional to the predicted variance.*

Proof. The derivative above multiplies the residual by $1/\sigma^2$. This is the contribution through the location output; parameters shared with the scale path also receive gradients through that path. Thus, larger predicted scales attenuate the location contribution, without by themselves establishing the quality of the resulting mean fit.

Noise and fitting error. To separate their roles, consider a scalar response A conditioned on context $\mathcal{C}$, with a fixed mean predictor $\mu(\mathcal{C})$. Minimizing the conditional expected Gaussian NLL over an unrestricted positive variance gives, when the conditional residual second moment is positive,

$$\begin{aligned} \sigma^2_{\text{opt}}(\mathcal{C}) &= \mathbb{E}[(A - \mu(\mathcal{C}))^2 \mid \mathcal{C}] \\ &= \text{Var}(A \mid \mathcal{C}) + \big(\mathbb{E}[A \mid \mathcal{C}] - \mu(\mathcal{C})\big)^2. \end{aligned} \tag{19}$$

The first term is conditional observation noise; the second is mean-fitting bias. During joint training, a large scale can therefore reflect an inaccurate mean as well as genuine volatility. Down-weighting the former can slow learning in difficult but predictable regions, as analyzed by Seitzer et al. (2022). This identity describes an unrestricted conditional optimum and does not assert that the recurrent decoder attains it.

Connection to β-NLL. Seitzer et al. (2022) moderate this dependence using a detached variance weight. We denote their exponent by $\eta \in [0, 1]$ to distinguish it from the KL weight β in Eq. 8:

$$J_\eta = \text{stopgrad}\big[(\sigma^2)^\eta\big]\, J, \qquad \frac{\partial J_\eta}{\partial \mu} = (\mu - x)(\sigma^2)^{\eta-1}. \tag{20}$$

Here $\eta = 0$ recovers Gaussian NLL, $\eta = 1/2$ gives inverse-standard-deviation weighting, and $\eta = 1$ gives the location-output gradient of half the squared error while still training the scale. This interpolation makes the trade-off between attenuation and mean fitting explicit. VolDy-VAE uses the unweighted Gaussian NLL ($\eta = 0$); the analysis motivates evaluating mean accuracy together with scale calibration.

C.7 Local Scale Optimization and Large Residuals

We examine a scalar loss with freely optimized scale to characterize its response to large residuals. This local calculation concerns the loss geometry; certified robustness to input perturbations, such as the guarantees studied by Yoon et al. (2022), is a separate property.

Fixed scale. With $\sigma = 1$, J grows as $(x-\mu)^2/2$, and its location derivative grows linearly with the residual.

Locally optimized scale. For a fixed nonzero residual magnitude $r = |x-\mu|$, minimizing over $\sigma > 0$ gives

$$\frac{\partial J}{\partial \sigma} = \frac{1}{\sigma} - \frac{r^2}{\sigma^3} = 0, \qquad \sigma^* = r. \tag{21}$$

Substituting this optimum yields the profiled loss

$$\min_{\sigma>0} J(\mu,\sigma) = \log|x-\mu| + \frac{1}{2}, \qquad x \neq \mu. \tag{22}$$

Its growth is logarithmic for large residuals, and the magnitude of its location derivative is $1/r$. This explains how freely adapting scale can attenuate large-residual updates. It applies whether the residual comes from noise or from an underfit mean, which is the trade-off in Appendix Section C.6.

The recurrent model shares scale parameters across observations and enforces a positive scale floor, so its training objective is not the independently profiled loss above. The calculation does not establish that VolDy-VAE is globally more robust than a fixed-variance VAE or that joint training avoids mean underfitting. Those properties require evidence on mean fitting and calibration.

C.8 Efficiency under Regime-Switching Heteroscedasticity

We isolate the statistical effect of accurate variance weights in a constant-mean model. The theorem concerns known variances conditional on a realized regime sequence. Estimated variances and joint neural training are discussed separately below.

Theorem 1 (Efficiency under Regime-Switching Heteroscedasticity)**.** *Let $x_t = \mu^* + \epsilon_t$ for $t = 1,\dots,T$. Conditional on a fixed regime sequence $s_{1:T}$, assume independent errors $\epsilon_t \sim \mathcal{N}(0,\sigma^2_{s_t})$ with known positive variances $\sigma^2_{s_t} \in \{\sigma^2_L, \sigma^2_H\}$. The MSE estimator is the sample mean. The Gaussian NLL estimator is the inverse-variance weighted mean. Both are conditionally unbiased, and*

$$\mathrm{Var}(\hat{\mu}_{\mathrm{NLL}} \mid s_{1:T}) = \frac{1}{\sum_t \sigma^{-2}_{s_t}} \le \frac{\sum_t \sigma^2_{s_t}}{T^2} = \mathrm{Var}(\hat{\mu}_{\mathrm{MSE}} \mid s_{1:T}). \tag{23}$$

Equality holds if and only if all observed variances are equal.

Proof. Minimizing the respective objectives over the scalar mean gives

$$\hat{\mu}_{\mathrm{MSE}} = \frac{1}{T}\sum_t x_t, \qquad \hat{\mu}_{\mathrm{NLL}} = \frac{\sum_t \sigma^{-2}_{s_t} x_t}{\sum_t \sigma^{-2}_{s_t}}. \tag{24}$$

The second expression follows by setting $\sum_t (\mu - x_t)/\sigma^2_{s_t} = 0$. Both estimators have weights summing to one and are conditionally unbiased. Conditional independence gives the variances in the theorem. Cauchy–Schwarz yields

$$\left(\sum_t \sigma^2_{s_t}\right)\left(\sum_t \sigma^{-2}_{s_t}\right) \ge T^2, \tag{25}$$

with equality exactly when the observed variances coincide. This proves the comparison. □

For empirical regime fractions $\pi_L = T^{-1}\sum_t \mathbf{1}\{s_t = L\}$ and $\pi_H = 1 - \pi_L$, let $q = \sigma^2_H/\sigma^2_L$. The exact conditional variance ratio is

$$\frac{\mathrm{Var}(\hat{\mu}_{\mathrm{MSE}} \mid s_{1:T})}{\mathrm{Var}(\hat{\mu}_{\mathrm{NLL}} \mid s_{1:T})} = 1 + \pi_L \pi_H (q + q^{-1} - 2). \tag{26}$$

For fixed positive regime fractions, the gain increases with $q \ge 1$ and grows linearly in q as $q \to \infty$. These fractions describe the realized sequence; replacing them by stationary probabilities requires an additional limiting argument.

Estimated variances and scope. For fixed positive variance estimates, minimizing the Gaussian NLL over the mean gives the plug-in estimator

$$\hat{\mu}_{\text{plug}} = \frac{\sum_t \hat{\sigma}_t^{-2} x_t}{\sum_t \hat{\sigma}_t^{-2}}. \tag{27}$$

For a fixed finite sample, this expression is continuous in the positive variance estimates and approaches the known-variance estimator as all $\hat{\sigma}_t^2$ approach $\sigma_{s_t}^2$. When variance estimates are learned from the same observations, their dependence on the errors prevents direct use of the theorem's finite-sample unbiasedness and variance formulas. Asymptotic efficiency with estimated weights requires additional regularity conditions controlling weight-estimation error.

The theorem characterizes accurate inverse-variance weighting in the stated model. The quality of learned GRU scales, joint optimization, and mean fitting is assessed through forecasting and calibration experiments.

D Experimental Details

D.1 Baselines

We compare VolDy-VAE against 13 baselines under the standard long-horizon ProbTS setting. This setting follows the benchmark splits, context lengths, and forecasting horizons used by prior long-horizon probabilistic forecasting studies, with NsDiff included as an additional diffusion baseline.

The first group consists of Deep Point Forecasters equipped with probabilistic heads. Since standard point forecasters do not naturally output distributions, we augment them with a Gaussian head that predicts mean and variance and is trained with NLL loss. This category includes FITS Xu et al. (2024), a frequency-domain interpolation model; PatchTST Nie et al. (2023), a transformer based on patching and channel independence; iTransformer Liu et al. (2024), which embeds the entire time series as a token; and Koopa Liu et al. (2023), a dynamics-aware model based on Koopman operator theory.

The second group contains Generative and Probabilistic Models, including VAE, flow, diffusion, and prior-mapping architectures. We compare against the VAE-based K^2VAE Wu et al. (2025), which disentangles key and key-response series. For flow-based methods, we include GRU NVP and GRU MAF Rasul et al. (2021b), which combine RNNs with flow-based density estimation, and Trans-MAF Rasul et al. (2021b), a transformer-based flow model. For diffusion-based approaches, we include TimeGrad Rasul et al. (2021a), CSDI Tashiro et al. (2021), NsDiff Ye et al. (2025), and TSDiff Kollovieh et al. (2023). We also include PPM Li et al. (2026), which maps an input-conditioned parametric prior to predictive samples without iterative denoising.

D.2 Evaluation Metrics

We adopt three primary metrics to evaluate forecasting performance and several diagnostic metrics to analyze uncertainty dynamics. Unless otherwise stated, distributional metrics are computed from 100 Monte Carlo samples and reported as mean$\pm$std over six random seeds.

Continuous Ranked Probability Score (CRPS). CRPS measures the compatibility between the predicted distribution and the observation. It is a strictly proper scoring rule Gneiting & Raftery (2007) defined as:

$$\text{CRPS}(F, x) = \int_{\mathbb{R}} \left(F(z) - \mathbb{I}\{x \le z\}\right)^2 \, dz, \tag{28}$$

where F is the predictive Cumulative Distribution Function (CDF) and x is the ground truth. We approximate CRPS using 100 Monte Carlo samples.

Normalized Mean Absolute Error (NMAE). NMAE assesses the accuracy of the point forecast (median or mean of the samples). It is scale-invariant, facilitating comparison across datasets:

$$\text{NMAE} = \frac{\sum_{t,k} |x_{t,k} - \hat{x}_{t,k}|}{\sum_{t,k} |x_{t,k}|}. \tag{29}$$

Quantile Interval Calibration Error (QICE). QICE strictly evaluates the reliability of the confidence intervals. It measures the mean absolute deviation between the empirical coverage $\hat{c}(q)$ and the nominal quantile level q:

$$\mathrm{QICE} = 100 \times \frac{1}{M} \sum_{m=1}^{M} |\hat{c}(q_m) - q_m|, \tag{30}$$

where we use quantile levels $q \in \{0.1, 0.2, \ldots, 0.9\}$. We report QICE in percentage points, and lower QICE indicates better calibration.

Empirical 95% Coverage. For a test set with B forecast windows, horizon length T, and C variables, let $\ell^{95}_{i,t,c}$ and $u^{95}_{i,t,c}$ denote the empirical 2.5% and 97.5% predictive quantiles for instance i, time step t, and variable c. The empirical 95% coverage is:

$$\mathrm{Cov}_{95} = 100 \times \frac{1}{BTC} \sum_{i=1}^{B} \sum_{t=1}^{T} \sum_{c=1}^{C} \mathbb{I}\{\ell^{95}_{i,t,c} \le x_{i,t,c} \le u^{95}_{i,t,c}\}. \tag{31}$$

Values closer to 95 indicate better interval calibration.

Normalized Sharpness. Sharpness measures the average width of the 95% predictive interval. To make widths comparable across variables and datasets, we normalize by the standard deviation s_c of each target variable on the test split:

$$\mathrm{Sharp}_{95} = \frac{1}{BTC} \sum_{i=1}^{B} \sum_{t=1}^{T} \sum_{c=1}^{C} \frac{u^{95}_{i,t,c} - \ell^{95}_{i,t,c}}{s_c + \epsilon}, \tag{32}$$

where ϵ is a small constant for numerical stability. Lower sharpness indicates narrower intervals, but it should be interpreted together with coverage and QICE Gneiting et al. (2007); an overconfident model may obtain low sharpness with poor coverage.

Scale Persistence / Autocorrelation. To quantify whether the predicted uncertainty evolves coherently over time, we compute the lag-1 autocorrelation of the predicted scale series as a scale-persistence score. Let $\hat{\sigma}_{i,t,c}$ be the predicted scale for forecast window i, time step t, and variable c:

$$\rho_\sigma = \frac{1}{BC} \sum_{i=1}^{B} \sum_{c=1}^{C} \mathrm{Corr}\left(\hat{\sigma}_{i,2:T,c}, \hat{\sigma}_{i,1:T-1,c}\right). \tag{33}$$

A larger ρ_σ indicates stronger temporal persistence in the estimated volatility. Because high autocorrelation alone could also reflect excessive smoothing, we interpret ρ_σ jointly with coverage, QICE, and sharpness: a useful volatility-dynamics model should improve temporal coherence without sacrificing calibration or producing overly wide intervals.

D.3 Implementation Details

Model Architecture VolDy-VAE uses an encoder-decoder architecture. For comparison with K^2VAE Wu et al. (2025), we use the same patching strategy (patch length $P = 24$) and a lightweight MLP-based encoder. The main architectural difference is the Location-Scale Decoder with Volatility Dynamics. Unlike standard VAEs that output a single reconstruction vector, this decoder has two heads and a recurrent volatility module. The Location Head uses a linear projection to output a normalized location parameter $\tilde{\boldsymbol{\mu}}_t$, which is mapped back to the original data scale by the inverse RevIN transformation. The Scale Head with Volatility Dynamics uses a single-layer GRU to maintain a volatility hidden state $\mathbf{h}^{\mathrm{vol}}_t$. A linear layer followed by `Softplus` maps this state to a normalized scale parameter $\tilde{\boldsymbol{\sigma}}_t$. The final standard deviation $\boldsymbol{\sigma}_t$ is obtained by multiplying $\tilde{\boldsymbol{\sigma}}_t$ by the instance-wise inverse-RevIN scale from the look-back window, using the stored RevIN standard deviation divided by $|\gamma|$ when affine RevIN is enabled. The GRU hidden dimension is set equal to the latent dimension D, adding minimal computational overhead. For clarity, let $\mathbf{Z}_m \in \mathbb{R}^D$ denote

Table 8: Implementation controls for plug-in transferability experiments.

Backbone	Controlled setting	VolDy modification
TimeGAN	Matched training protocol.	Extend generator output to location/scale and add Gaussian NLL supervision.
K^2VAE	Same backbone and forecasting protocol as the base K^2VAE run.	Replace deterministic objective with Location-Scale NLL.
PatchTST	Matched training protocol.	Replace the point head with a dual-head location-scale decoder trained by Gaussian NLL.

the latent representation of the m-th future patch, where each patch contains P time steps and C variables. The scale head computes $\mathbf{h}_m^{\mathrm{vol}} = \mathrm{GRU}_{\mathrm{vol}}(\mathbf{h}_{m-1}^{\mathrm{vol}}, \mathbf{Z}_m)$ and then projects this hidden state to a vector in $\mathbb{R}^{PC}$:

$$\tilde{\mathbf{s}}_m = \mathrm{Softplus}(\mathrm{Linear}(\mathbf{h}_m^{\mathrm{vol}})) + \xi, \quad \tilde{\mathbf{s}}_m \in \mathbb{R}^{PC}. \tag{34}$$

This vector is reshaped in time-major order and multiplied by the broadcast RevIN scale to obtain $\boldsymbol{\sigma}_m \in \mathbb{R}^{P\times C}$, producing one original-scale standard deviation for each time step and variable inside the patch. Therefore, σ is predicted both per step and per variable; the implementation does not repeat a single scalar or a single channel vector across the whole patch. The same shape convention is used by the location head for $\boldsymbol{\mu}_m$, so the Gaussian likelihood is evaluated element-wise over the resulting $P \times C$ block. This design captures the temporal evolution and persistence of volatility regimes while maximizing the heteroscedastic Gaussian Likelihood $\mathcal{L}_{\mathrm{NLL}}$ as detailed in Eq. 10.

Training Protocol We implement VolDy-VAE using PyTorch Paszke et al. (2019). All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU (24GB). The model is optimized using the Adam optimizer with an initial learning rate of 10^{-3} (dataset-specific adjustments are made within the range $[10^{-4}, 10^{-3}]$). We use a batch size of 32 by default. The training objective is the ELBO, comprising the Location-Scale NLL loss (reconstruction & prediction) and the KL divergence term. We train the model for a maximum of 50 epochs with an early stopping strategy (patience=10). The source code and configuration files are provided in the submitted code archive.

For the plug-in transferability experiments, each base/VolDy pair uses the same data split, forecasting horizon, six-seed evaluation protocol, optimizer family, and early-stopping rule. Table 8 summarizes the controlled component change; parameter counts follow each backbone's native architecture.

The full horizon-wise transferability results referenced in the main text are reported in Table 9.

D.4 Detailed Efficiency Analysis

Following the reporting style of the efficiency analysis in K^2VAE Wu et al. (2025), Table 10 expands the main-text efficiency plot into a model-by-metric table. The measurements are conducted on *ETTh2* with $L = 96$ and $H = 720$ using batch size 1 on a single NVIDIA RTX 4090 GPU. For sample-based probabilistic baselines, we generate 100 samples during inference.

Table 10 shows that VolDy-VAE retains the one-pass inference advantage of VAE-style forecasters while improving forecasting quality. Compared with K^2VAE, VolDy-VAE reduces CRPS from 0.280 to 0.174 and NMAE from 0.278 to 0.221, while also using slightly lower latency and memory. The gain is more pronounced against iterative or autoregressive generative baselines: CSDI requires substantially longer inference time because it samples through repeated denoising steps, while flow-based baselines either incur large sequential likelihood costs or show weaker accuracy on this long-horizon setting. These results support the main efficiency claim: the recurrent scale head adds only a small computational overhead, but the direct latent forecast path avoids the dominant iterative sampling cost of diffusion models and the recursive generation cost of many autoregressive probabilistic forecasters.

D.5 Dataset Statistics

Table 11 summarizes the dimensionality, temporal length, sampling frequency, and application domain of the nine real-world multivariate time series datasets used in the main experiments.

Table 9: Full transferability analysis over four prediction horizons ($L = 96$). Values are mean±std over six seeds.

Framework	Dataset	H	Base CRPS	VolDy CRPS	Base NMAE	VolDy NMAE
TimeGAN	ETTh1	96	$0.277_{\pm.011}$	$0.262_{\pm.009}$	$0.347_{\pm.012}$	$0.339_{\pm.010}$
	ETTh1	192	$0.306_{\pm.014}$	$0.286_{\pm.011}$	$0.382_{\pm.015}$	$0.368_{\pm.012}$
	ETTh1	336	$0.334_{\pm.017}$	$0.309_{\pm.014}$	$0.421_{\pm.019}$	$0.397_{\pm.016}$
	ETTh1	720	$0.381_{\pm.021}$	$0.348_{\pm.018}$	$0.469_{\pm.024}$	$0.436_{\pm.021}$
	Weather	96	$0.106_{\pm.006}$	$0.068_{\pm.004}$	$0.117_{\pm.007}$	$0.081_{\pm.005}$
	Weather	192	$0.118_{\pm.007}$	$0.076_{\pm.005}$	$0.129_{\pm.008}$	$0.089_{\pm.006}$
	Weather	336	$0.129_{\pm.008}$	$0.084_{\pm.006}$	$0.141_{\pm.009}$	$0.098_{\pm.007}$
	Weather	720	$0.145_{\pm.010}$	$0.097_{\pm.007}$	$0.158_{\pm.011}$	$0.112_{\pm.008}$
	Exchange	96	$0.056_{\pm.004}$	$0.047_{\pm.003}$	$0.071_{\pm.005}$	$0.062_{\pm.004}$
	Exchange	192	$0.069_{\pm.005}$	$0.058_{\pm.004}$	$0.087_{\pm.006}$	$0.075_{\pm.005}$
	Exchange	336	$0.083_{\pm.006}$	$0.071_{\pm.005}$	$0.104_{\pm.007}$	$0.091_{\pm.006}$
	Exchange	720	$0.101_{\pm.008}$	$0.087_{\pm.006}$	$0.126_{\pm.009}$	$0.111_{\pm.007}$
K^2VAE	ETTh1	96	$0.264_{\pm.020}$	$0.251_{\pm.017}$	$0.336_{\pm.041}$	$0.332_{\pm.012}$
	ETTh1	192	$0.290_{\pm.016}$	$0.274_{\pm.014}$	$0.372_{\pm.023}$	$0.361_{\pm.018}$
	ETTh1	336	$0.308_{\pm.021}$	$0.298_{\pm.016}$	$0.394_{\pm.022}$	$0.388_{\pm.020}$
	ETTh1	720	$0.314_{\pm.011}$	$0.310_{\pm.012}$	$0.396_{\pm.012}$	$0.389_{\pm.013}$
	Weather	96	$0.080_{\pm.007}$	$0.065_{\pm.004}$	$0.086_{\pm.011}$	$0.076_{\pm.004}$
	Weather	192	$0.079_{\pm.009}$	$0.067_{\pm.006}$	$0.083_{\pm.011}$	$0.074_{\pm.007}$
	Weather	336	$0.082_{\pm.010}$	$0.073_{\pm.007}$	$0.093_{\pm.010}$	$0.080_{\pm.007}$
	Weather	720	$0.084_{\pm.003}$	$0.081_{\pm.008}$	$0.099_{\pm.009}$	$0.091_{\pm.008}$
	Exchange	96	$0.031_{\pm.002}$	$0.029_{\pm.002}$	$0.032_{\pm.002}$	$0.029_{\pm.002}$
	Exchange	192	$0.032_{\pm.010}$	$0.030_{\pm.006}$	$0.040_{\pm.005}$	$0.037_{\pm.004}$
	Exchange	336	$0.048_{\pm.004}$	$0.045_{\pm.005}$	$0.054_{\pm.001}$	$0.050_{\pm.004}$
	Exchange	720	$0.069_{\pm.005}$	$0.062_{\pm.006}$	$0.084_{\pm.017}$	$0.077_{\pm.012}$
PatchTST	ETTh1	96	$0.312_{\pm.036}$	$0.268_{\pm.020}$	$0.407_{\pm.022}$	$0.348_{\pm.018}$
	ETTh1	192	$0.313_{\pm.034}$	$0.282_{\pm.021}$	$0.405_{\pm.088}$	$0.372_{\pm.026}$
	ETTh1	336	$0.319_{\pm.035}$	$0.306_{\pm.024}$	$0.412_{\pm.024}$	$0.397_{\pm.029}$
	ETTh1	720	$0.323_{\pm.020}$	$0.315_{\pm.024}$	$0.428_{\pm.024}$	$0.407_{\pm.030}$
	Weather	96	$0.131_{\pm.007}$	$0.074_{\pm.006}$	$0.145_{\pm.016}$	$0.085_{\pm.006}$
	Weather	192	$0.131_{\pm.014}$	$0.081_{\pm.007}$	$0.144_{\pm.012}$	$0.091_{\pm.007}$
	Weather	336	$0.137_{\pm.008}$	$0.091_{\pm.008}$	$0.149_{\pm.023}$	$0.102_{\pm.008}$
	Weather	720	$0.142_{\pm.005}$	$0.105_{\pm.010}$	$0.152_{\pm.029}$	$0.118_{\pm.010}$
	Exchange	96	$0.063_{\pm.006}$	$0.061_{\pm.004}$	$0.079_{\pm.002}$	$0.066_{\pm.004}$
	Exchange	192	$0.067_{\pm.008}$	$0.066_{\pm.005}$	$0.081_{\pm.002}$	$0.071_{\pm.005}$
	Exchange	336	$0.071_{\pm.017}$	$0.068_{\pm.006}$	$0.085_{\pm.010}$	$0.079_{\pm.006}$
	Exchange	720	$0.097_{\pm.007}$	$0.088_{\pm.007}$	$0.126_{\pm.001}$	$0.094_{\pm.007}$

D.6 Full Results

Table 12 and Table 13 report the full CRPS and NMAE comparisons, including our PPM reproduction across all four horizons. They cover four forecasting horizons ($H \in \{96, 192, 336, 720\}$) across nine real-world datasets. Across these settings, VolDy-VAE has lower CRPS and NMAE than the listed generative and point-forecasting baselines in most cases. These results are consistent with the role of the Location-Scale framework in modeling both long-term dependencies and volatility dynamics.

Table 10: Comparison on model efficiency and forecasting quality on *ETTh2* ($L = 96, H = 720$). Lower values of CRPS, NMAE, inference speed (sec/sample), and memory (MB) indicate better performance or efficiency. The efficiency measurements are obtained with batch size 1. RED: THE BEST, BLUE: THE 2ND BEST.

Model	CRPS ↓	NMAE ↓	Speed ↓	Memory ↓
TSDiff	$0.406_{\pm.056}$	$0.482_{\pm.022}$	1.35	340
GRU NVP	$0.539_{\pm.090}$	$0.688_{\pm.161}$	2.00	26
GRU MAF	$0.990_{\pm.023}$	$1.092_{\pm.019}$	7.10	27
Trans MAF	$0.327_{\pm.033}$	$0.412_{\pm.020}$	8.05	83
CSDI	$0.302_{\pm.040}$	$0.382_{\pm.030}$	32.50	38
K^2VAE	$0.280_{\pm.014}$	$0.278_{\pm.020}$	0.72	29
VolDy-VAE	$0.174_{\pm.001}$	$0.221_{\pm.002}$	0.62	24

Table 11: Statistics of the multivariate time series datasets.

Dataset	**Features (C)**	**Timesteps**	**Frequency**	**Domain**
ETTh1	7	17,420	1 Hour	Energy
ETTh2	7	17,420	1 Hour	Energy
ETTm1	7	69,680	15 Min	Energy
ETTm2	7	69,680	15 Min	Energy
Electricity	321	26,304	1 Hour	Energy
Traffic	862	17,544	1 Hour	Transport
Weather	21	52,696	10 Min	Weather
Exchange	8	7,588	1 Day	Economics
ILI	7	966	1 Week	Health

D.7 Additional Comparison with DeepAR, HeTVAE, and VHVM

We report three diagnostic baselines that separate different explanations for the proposed recurrent scale path. First, DeepAR Salinas et al. (2020) is a canonical autoregressive recurrent probabilistic forecaster: it maintains a recurrent state and emits Gaussian likelihood parameters recursively, but the recurrence is tied to value generation rather than to a dedicated volatility-state transition. Second, HeTVAE Shukla & Marlin (2022) is a closely related heteroscedastic VAE, but its original formulation targets irregular time-series imputation and interpolation rather than long-horizon forecasting. We adapt HeTVAE to the same long-horizon forecasting setting by using the look-back window as observed context and the future horizon as target query points, while preserving its heteroscedastic Gaussian likelihood. Third, VHVM Yin & Barucca (2022) provides a volatility-specific VAE baseline with recurrent heteroscedastic dynamics. We adapt VHVM to the same long-horizon protocol and report the resulting comparison. Together, these baselines test whether VolDy-VAE's gains can be explained by generic autoregressive recurrence, generic heteroscedastic VAE modeling, or a volatility-specific recurrent VAE alone. Table 14 shows that VolDy-VAE remains stronger than these diagnostic baselines on the reported dataset-horizon settings. VHVM is competitive with the other diagnostic controls on several volatility-heavy settings, especially Electricity and Traffic. VolDy-VAE nevertheless has lower scores across the reported table. This pattern is consistent with the proposed combination of historical scale-state transfer and direct future generation.

Table 12: Results of CRPS (mean±std) on long-term forecasting scenarios. The context length is 36 for ILI and 96 for the other datasets. All baselines except PPM use six independent runs; PPM is our reproduction over three seeds under the same data and evaluation protocol. Lower is better. Red and blue denote the lowest and second-lowest distinct displayed means; ties after rounding share a rank.

Dataset	Horizon	Koopa	iTransformer	FITS	PatchTST	GRU MAF	Trans MAF	TSDiff	CSDI	NsDiff	TimeGrad	GRU NVP	K^2VAE	PPM	VolDy-VAE
ETTm1	96	$0.285_{\pm.018}$	$0.301_{\pm.033}$	$0.267_{\pm.023}$	$0.261_{\pm.051}$	$0.295_{\pm.055}$	$0.313_{\pm.045}$	$0.344_{\pm.050}$	$0.236_{\pm.006}$	$0.255_{\pm.014}$	$0.522_{\pm.105}$	$0.383_{\pm.053}$	$0.232_{\pm.010}$	$0.238_{\pm.007}$	$0.220_{\pm.005}$
	192	$0.289_{\pm.024}$	$0.314_{\pm.023}$	$0.261_{\pm.022}$	$0.275_{\pm.030}$	$0.389_{\pm.033}$	$0.424_{\pm.029}$	$0.345_{\pm.035}$	$0.291_{\pm.025}$	$0.281_{\pm.018}$	$0.603_{\pm.092}$	$0.396_{\pm.030}$	$0.259_{\pm.013}$	$0.257_{\pm.008}$	$0.235_{\pm.001}$
	336	$0.286_{\pm.035}$	$0.311_{\pm.029}$	$0.275_{\pm.030}$	$0.285_{\pm.028}$	$0.429_{\pm.021}$	$0.481_{\pm.019}$	$0.462_{\pm.043}$	$0.322_{\pm.033}$	$0.292_{\pm.024}$	$0.601_{\pm.028}$	$0.486_{\pm.032}$	$0.262_{\pm.030}$	$0.272_{\pm.003}$	$0.249_{\pm.002}$
	720	$0.295_{\pm.027}$	$0.455_{\pm.021}$	$0.305_{\pm.024}$	$0.304_{\pm.029}$	$0.536_{\pm.033}$	$0.688_{\pm.043}$	$0.478_{\pm.027}$	$0.448_{\pm.038}$	$0.323_{\pm.030}$	$0.621_{\pm.037}$	$0.546_{\pm.036}$	$0.294_{\pm.026}$	$0.311_{\pm.002}$	$0.272_{\pm.002}$
ETTm2	96	$0.178_{\pm.023}$	$0.181_{\pm.031}$	$0.162_{\pm.053}$	$0.142_{\pm.034}$	$0.177_{\pm.024}$	$0.227_{\pm.013}$	$0.175_{\pm.019}$	$0.115_{\pm.009}$	$0.138_{\pm.010}$	$0.427_{\pm.042}$	$0.319_{\pm.044}$	$0.126_{\pm.007}$	$0.113_{\pm.003}$	$0.109_{\pm.005}$
	192	$0.185_{\pm.014}$	$0.190_{\pm.010}$	$0.185_{\pm.053}$	$0.172_{\pm.023}$	$0.411_{\pm.026}$	$0.253_{\pm.037}$	$0.255_{\pm.029}$	$0.147_{\pm.008}$	$0.162_{\pm.011}$	$0.424_{\pm.061}$	$0.326_{\pm.025}$	$0.148_{\pm.009}$	$0.131_{\pm.000}$	$0.126_{\pm.003}$
	336	$0.198_{\pm.015}$	$0.206_{\pm.055}$	$0.218_{\pm.053}$	$0.195_{\pm.042}$	$0.377_{\pm.023}$	$0.253_{\pm.013}$	$0.328_{\pm.047}$	$0.190_{\pm.018}$	$0.180_{\pm.014}$	$0.469_{\pm.049}$	$0.449_{\pm.145}$	$0.164_{\pm.010}$	$0.145_{\pm.000}$	$0.142_{\pm.001}$
	720	$0.233_{\pm.025}$	$0.311_{\pm.024}$	$0.449_{\pm.034}$	$0.229_{\pm.036}$	$0.272_{\pm.029}$	$0.355_{\pm.043}$	$0.344_{\pm.046}$	$0.239_{\pm.035}$	$0.238_{\pm.026}$	$0.470_{\pm.054}$	$0.561_{\pm.273}$	$0.221_{\pm.023}$	$0.171_{\pm.001}$	$0.164_{\pm.002}$
ETTh1	96	$0.307_{\pm.033}$	$0.292_{\pm.032}$	$0.294_{\pm.023}$	$0.312_{\pm.036}$	$0.293_{\pm.037}$	$0.333_{\pm.045}$	$0.395_{\pm.052}$	$0.437_{\pm.018}$	$0.302_{\pm.018}$	$0.455_{\pm.046}$	$0.379_{\pm.030}$	$0.264_{\pm.020}$	$0.253_{\pm.002}$	$0.243_{\pm.001}$
	192	$0.301_{\pm.014}$	$0.298_{\pm.020}$	$0.304_{\pm.028}$	$0.313_{\pm.034}$	$0.348_{\pm.075}$	$0.351_{\pm.063}$	$0.467_{\pm.044}$	$0.496_{\pm.051}$	$0.330_{\pm.019}$	$0.516_{\pm.038}$	$0.425_{\pm.019}$	$0.290_{\pm.016}$	$0.287_{\pm.003}$	$0.266_{\pm.001}$
	336	$0.312_{\pm.019}$	$0.327_{\pm.043}$	$0.318_{\pm.023}$	$0.319_{\pm.035}$	$0.377_{\pm.026}$	$0.371_{\pm.031}$	$0.450_{\pm.027}$	$0.454_{\pm.025}$	$0.342_{\pm.020}$	$0.512_{\pm.026}$	$0.458_{\pm.054}$	$0.308_{\pm.021}$	$0.303_{\pm.001}$	$0.284_{\pm.002}$
	720	$0.318_{\pm.009}$	$0.350_{\pm.019}$	$0.348_{\pm.025}$	$0.323_{\pm.020}$	$0.393_{\pm.043}$	$0.363_{\pm.053}$	$0.516_{\pm.027}$	$0.528_{\pm.012}$	$0.360_{\pm.018}$	$0.523_{\pm.027}$	$0.502_{\pm.039}$	$0.314_{\pm.011}$	$0.325_{\pm.003}$	$0.310_{\pm.001}$
ETTh2	96	$0.199_{\pm.012}$	$0.185_{\pm.013}$	$0.187_{\pm.011}$	$0.197_{\pm.021}$	$0.239_{\pm.019}$	$0.263_{\pm.020}$	$0.336_{\pm.021}$	$0.164_{\pm.013}$	$0.178_{\pm.012}$	$0.358_{\pm.026}$	$0.432_{\pm.141}$	$0.162_{\pm.009}$	$0.138_{\pm.001}$	$0.141_{\pm.005}$
	192	$0.198_{\pm.022}$	$0.199_{\pm.019}$	$0.195_{\pm.022}$	$0.204_{\pm.055}$	$0.313_{\pm.034}$	$0.273_{\pm.024}$	$0.265_{\pm.043}$	$0.226_{\pm.018}$	$0.205_{\pm.017}$	$0.457_{\pm.081}$	$0.625_{\pm.170}$	$0.186_{\pm.018}$	$0.163_{\pm.002}$	$0.160_{\pm.001}$
	336	$0.262_{\pm.019}$	$0.271_{\pm.033}$	$0.246_{\pm.044}$	$0.277_{\pm.054}$	$0.376_{\pm.034}$	$0.265_{\pm.042}$	$0.350_{\pm.031}$	$0.274_{\pm.022}$	$0.266_{\pm.023}$	$0.481_{\pm.078}$	$0.793_{\pm.319}$	$0.257_{\pm.023}$	$0.178_{\pm.003}$	$0.172_{\pm.002}$
	720	$0.293_{\pm.026}$	$0.542_{\pm.015}$	$0.314_{\pm.022}$	$0.304_{\pm.018}$	$0.990_{\pm.023}$	$0.327_{\pm.033}$	$0.406_{\pm.056}$	$0.302_{\pm.040}$	$0.289_{\pm.020}$	$0.445_{\pm.016}$	$0.539_{\pm.090}$	$0.280_{\pm.014}$	$0.187_{\pm.003}$	$0.174_{\pm.001}$
Electricity	96	$0.110_{\pm.004}$	$0.102_{\pm.004}$	$0.105_{\pm.006}$	$0.126_{\pm.005}$	$0.083_{\pm.009}$	$0.088_{\pm.014}$	$0.344_{\pm.006}$	$0.153_{\pm.137}$	$0.081_{\pm.004}$	$0.096_{\pm.002}$	$0.094_{\pm.003}$	$0.073_{\pm.002}$	$0.069_{\pm.001}$	$0.069_{\pm.002}$
	192	$0.109_{\pm.011}$	$0.104_{\pm.014}$	$0.112_{\pm.104}$	$0.123_{\pm.032}$	$0.093_{\pm.024}$	$0.097_{\pm.009}$	$0.345_{\pm.006}$	$0.200_{\pm.094}$	$0.088_{\pm.006}$	$0.100_{\pm.004}$	$0.097_{\pm.002}$	$0.080_{\pm.004}$	$0.074_{\pm.000}$	$0.075_{\pm.005}$
	336	$0.121_{\pm.011}$	$0.104_{\pm.010}$	$0.111_{\pm.014}$	$0.131_{\pm.024}$	$0.095_{\pm.001}$	-	$0.462_{\pm.054}$	-	$0.092_{\pm.005}$	$0.102_{\pm.007}$	$0.099_{\pm.001}$	$0.084^{*}_{\pm.001}$	$0.081_{\pm.001}$	$0.078_{\pm.001}$
	720	$0.113_{\pm.018}$	$0.109_{\pm.044}$	$0.115_{\pm.024}$	$0.127_{\pm.015}$	$0.106_{\pm.007}$	-	$0.478_{\pm.005}$	-	$0.096_{\pm.008}$	$0.108_{\pm.003}$	$0.114_{\pm.013}$	$0.087^{*}_{\pm.005}$	$0.094_{\pm.001}$	$0.080_{\pm.006}$
Traffic	96	$0.297_{\pm.019}$	$0.256_{\pm.004}$	$0.258_{\pm.004}$	$0.194_{\pm.002}$	$0.215_{\pm.003}$	$0.208_{\pm.004}$	$0.294_{\pm.003}$	-	$0.193_{\pm.002}$	$0.202_{\pm.004}$	$0.187_{\pm.002}$	$0.186^{*}_{\pm.001}$	$0.223_{\pm.005}$	$0.180_{\pm.005}$
	192	$0.308_{\pm.009}$	$0.250_{\pm.002}$	$0.275_{\pm.003}$	$0.198_{\pm.004}$	-	-	$0.306_{\pm.004}$	-	$0.196_{\pm.003}$	$0.208_{\pm.003}$	$0.192_{\pm.001}$	$0.188^{*}_{\pm.002}$	$0.221_{\pm.002}$	$0.187_{\pm.004}$
	336	$0.334_{\pm.017}$	$0.261_{\pm.001}$	$0.327_{\pm.001}$	$0.204_{\pm.002}$	-	-	$0.317_{\pm.006}$	-	$0.202_{\pm.003}$	$0.213_{\pm.003}$	$0.201_{\pm.004}$	$0.195_{\pm.003}$	$0.220_{\pm.002}$	$0.194_{\pm.002}$
	720	$0.358_{\pm.022}$	$0.284_{\pm.004}$	$0.374_{\pm.004}$	$0.214_{\pm.001}$	-	-	$0.391_{\pm.002}$	-	$0.207_{\pm.003}$	$0.220_{\pm.002}$	$0.211_{\pm.004}$	$0.200_{\pm.001}$	$0.226_{\pm.004}$	$0.192_{\pm.003}$
Weather	96	$0.132_{\pm.008}$	$0.131_{\pm.011}$	$0.210_{\pm.013}$	$0.131_{\pm.007}$	$0.139_{\pm.008}$	$0.105_{\pm.011}$	$0.104_{\pm.020}$	$0.068_{\pm.008}$	$0.086_{\pm.006}$	$0.130_{\pm.017}$	$0.116_{\pm.013}$	$0.080_{\pm.007}$	$0.075_{\pm.002}$	$0.070_{\pm.005}$
	192	$0.133_{\pm.017}$	$0.132_{\pm.018}$	$0.205_{\pm.019}$	$0.131_{\pm.014}$	$0.143_{\pm.020}$	$0.142_{\pm.022}$	$0.134_{\pm.012}$	$0.068_{\pm.006}$	$0.085_{\pm.006}$	$0.127_{\pm.019}$	$0.122_{\pm.021}$	$0.079_{\pm.009}$	$0.080_{\pm.001}$	$0.067_{\pm.006}$
	336	$0.136_{\pm.021}$	$0.132_{\pm.010}$	$0.221_{\pm.005}$	$0.137_{\pm.008}$	$0.129_{\pm.012}$	$0.133_{\pm.014}$	$0.137_{\pm.010}$	$0.083_{\pm.002}$	$0.088_{\pm.005}$	$0.130_{\pm.006}$	$0.128_{\pm.011}$	$0.082_{\pm.010}$	$0.087_{\pm.001}$	$0.071_{\pm.008}$
	720	$0.140_{\pm.007}$	$0.133_{\pm.004}$	$0.267_{\pm.003}$	$0.142_{\pm.005}$	$0.122_{\pm.006}$	$0.113_{\pm.004}$	$0.152_{\pm.003}$	$0.087_{\pm.003}$	$0.090_{\pm.004}$	$0.113_{\pm.011}$	$0.110_{\pm.004}$	$0.084_{\pm.003}$	$0.096_{\pm.000}$	$0.073_{\pm.003}$
Exchange	96	$0.063_{\pm.006}$	$0.061_{\pm.003}$	$0.048_{\pm.004}$	$0.063_{\pm.006}$	$0.026_{\pm.010}$	$0.028_{\pm.002}$	$0.079_{\pm.007}$	$0.028_{\pm.003}$	$0.038_{\pm.004}$	$0.068_{\pm.003}$	$0.071_{\pm.006}$	$0.031_{\pm.002}$	$0.020_{\pm.000}$	$0.022_{\pm.003}$
	192	$0.065_{\pm.020}$	$0.062_{\pm.010}$	$0.049_{\pm.011}$	$0.067_{\pm.008}$	$0.034_{\pm.009}$	$0.046_{\pm.017}$	$0.093_{\pm.011}$	$0.045_{\pm.003}$	$0.046_{\pm.006}$	$0.087_{\pm.013}$	$0.068_{\pm.004}$	$0.032_{\pm.010}$	$0.027_{\pm.000}$	$0.030_{\pm.008}$
	336	$0.072_{\pm.008}$	$0.067_{\pm.008}$	$0.052_{\pm.013}$	$0.071_{\pm.017}$	$0.058_{\pm.023}$	$0.045_{\pm.010}$	$0.081_{\pm.007}$	$0.060_{\pm.004}$	$0.061_{\pm.005}$	$0.074_{\pm.009}$	$0.072_{\pm.002}$	$0.048_{\pm.004}$	$0.038_{\pm.001}$	$0.041_{\pm.005}$
	720	$0.091_{\pm.012}$	$0.087_{\pm.023}$	$0.074_{\pm.011}$	$0.097_{\pm.007}$	$0.160_{\pm.019}$	$0.148_{\pm.017}$	$0.082_{\pm.010}$	$0.143_{\pm.020}$	$0.076_{\pm.008}$	$0.099_{\pm.015}$	$0.079_{\pm.009}$	$0.069_{\pm.005}$	$0.067_{\pm.001}$	$0.065_{\pm.004}$
ILI	24	$0.245_{\pm.018}$	$0.212_{\pm.013}$	$0.233_{\pm.015}$	$0.312_{\pm.014}$	$0.097_{\pm.010}$	$0.092_{\pm.019}$	$0.228_{\pm.024}$	$0.250_{\pm.013}$	$0.102_{\pm.007}$	$0.275_{\pm.047}$	$0.257_{\pm.003}$	$0.087_{\pm.003}$	$0.098_{\pm.017}$	$0.082_{\pm.003}$
	36	$0.214_{\pm.008}$	$0.182_{\pm.016}$	$0.217_{\pm.023}$	$0.241_{\pm.021}$	$0.117_{\pm.017}$	$0.115_{\pm.011}$	$0.235_{\pm.010}$	$0.285_{\pm.010}$	$0.130_{\pm.007}$	$0.272_{\pm.057}$	$0.281_{\pm.004}$	$0.113_{\pm.005}$	$0.141_{\pm.014}$	$0.110_{\pm.003}$
	48	$0.271_{\pm.021}$	$0.213_{\pm.012}$	$0.185_{\pm.026}$	$0.242_{\pm.018}$	$0.128_{\pm.019}$	$0.133_{\pm.022}$	$0.265_{\pm.039}$	$0.285_{\pm.036}$	$0.142_{\pm.010}$	$0.295_{\pm.033}$	$0.288_{\pm.008}$	$0.124_{\pm.010}$	$0.147_{\pm.021}$	$0.114_{\pm.008}$
	60	$0.228_{\pm.022}$	$0.222_{\pm.020}$	$0.211_{\pm.011}$	$0.233_{\pm.019}$	$0.172_{\pm.034}$	$0.155_{\pm.018}$	$0.263_{\pm.022}$	$0.283_{\pm.012}$	$0.161_{\pm.010}$	$0.295_{\pm.083}$	$0.307_{\pm.005}$	$0.142_{\pm.008}$	$0.163_{\pm.013}$	$0.130_{\pm.005}$

$*$ denotes entries unavailable or inconsistent in the original report that were reproduced under the official setting. Specifically, the starred K^2VAE CRPS values for *Electricity* ($H = 336, 720$) and *Traffic* ($H = 96, 192$) are reproduced by us.

Table 13: Results of NMAE (mean±std) on long-term forecasting scenarios. The context length is 36 for ILI and 96 for the other datasets. All baselines except PPM use six independent runs; PPM is our reproduction over three seeds under the same data and evaluation protocol. Lower is better. Red and blue denote the lowest and second-lowest distinct displayed means; ties after rounding share a rank.

Dataset	Horizon	Koopa	iTransformer	FITS	PatchTST	GRU MAF	Trans MAF	TSDiff	CSDI	NsDiff	TimeGrad	GRU NVP	K^2VAE	PPM	VolDy-VAE
ETTm1	96	$0.362_{\pm.022}$	$0.369_{\pm.029}$	$0.349_{\pm.032}$	$0.329_{\pm.100}$	$0.402_{\pm.087}$	$0.456_{\pm.042}$	$0.441_{\pm.021}$	$0.308_{\pm.005}$	$0.315_{\pm.016}$	$0.645_{\pm.129}$	$0.488_{\pm.058}$	$0.284_{\pm.011}$	$0.278_{\pm.003}$	$0.286_{\pm.009}$
	192	$0.365_{\pm.032}$	$0.384_{\pm.041}$	$0.341_{\pm.032}$	$0.338_{\pm.022}$	$0.476_{\pm.046}$	$0.553_{\pm.012}$	$0.441_{\pm.019}$	$0.377_{\pm.026}$	$0.354_{\pm.020}$	$0.748_{\pm.084}$	$0.514_{\pm.042}$	$0.323_{\pm.020}$	$0.300_{\pm.004}$	$0.305_{\pm.001}$
	336	$0.364_{\pm.026}$	$0.380_{\pm.020}$	$0.356_{\pm.022}$	$0.344_{\pm.013}$	$0.522_{\pm.019}$	$0.590_{\pm.047}$	$0.571_{\pm.033}$	$0.419_{\pm.042}$	$0.365_{\pm.022}$	$0.759_{\pm.015}$	$0.630_{\pm.029}$	$0.330_{\pm.014}$	$0.319_{\pm.001}$	$0.323_{\pm.001}$
	720	$0.377_{\pm.037}$	$0.490_{\pm.038}$	$0.406_{\pm.072}$	$0.382_{\pm.066}$	$0.711_{\pm.081}$	$0.822_{\pm.034}$	$0.622_{\pm.045}$	$0.578_{\pm.051}$	$0.410_{\pm.036}$	$0.793_{\pm.034}$	$0.707_{\pm.050}$	$0.373_{\pm.032}$	$0.362_{\pm.002}$	$0.358_{\pm.002}$
ETTm2	96	$0.225_{\pm.039}$	$0.221_{\pm.039}$	$0.210_{\pm.040}$	$0.216_{\pm.035}$	$0.212_{\pm.082}$	$0.279_{\pm.031}$	$0.224_{\pm.033}$	$0.146_{\pm.012}$	$0.158_{\pm.012}$	$0.525_{\pm.047}$	$0.413_{\pm.059}$	$0.144_{\pm.011}$	$0.140_{\pm.002}$	$0.137_{\pm.001}$
	192	$0.233_{\pm.026}$	$0.229_{\pm.031}$	$0.234_{\pm.038}$	$0.215_{\pm.022}$	$0.535_{\pm.029}$	$0.292_{\pm.041}$	$0.316_{\pm.040}$	$0.189_{\pm.012}$	$0.184_{\pm.012}$	$0.530_{\pm.060}$	$0.427_{\pm.033}$	$0.170_{\pm.009}$	$0.164_{\pm.001}$	$0.158_{\pm.002}$
	336	$0.267_{\pm.023}$	$0.245_{\pm.049}$	$0.276_{\pm.019}$	$0.234_{\pm.024}$	$0.407_{\pm.043}$	$0.309_{\pm.032}$	$0.397_{\pm.051}$	$0.248_{\pm.024}$	$0.205_{\pm.020}$	$0.566_{\pm.047}$	$0.580_{\pm.169}$	$0.187_{\pm.021}$	$0.182_{\pm.001}$	$0.178_{\pm.002}$
	720	$0.290_{\pm.033}$	$0.385_{\pm.042}$	$0.540_{\pm.052}$	$0.288_{\pm.034}$	$0.355_{\pm.048}$	$0.475_{\pm.029}$	$0.416_{\pm.065}$	$0.306_{\pm.040}$	$0.295_{\pm.034}$	$0.561_{\pm.044}$	$0.749_{\pm.385}$	$0.275_{\pm.035}$	$0.211_{\pm.002}$	$0.205_{\pm.003}$
ETTh1	96	$0.407_{\pm.052}$	$0.386_{\pm.092}$	$0.393_{\pm.142}$	$0.407_{\pm.022}$	$0.371_{\pm.034}$	$0.423_{\pm.047}$	$0.510_{\pm.029}$	$0.557_{\pm.022}$	$0.380_{\pm.024}$	$0.585_{\pm.058}$	$0.481_{\pm.037}$	$0.336_{\pm.041}$	$0.315_{\pm.000}$	$0.323_{\pm.002}$
	192	$0.396_{\pm.022}$	$0.388_{\pm.041}$	$0.406_{\pm.079}$	$0.405_{\pm.088}$	$0.430_{\pm.022}$	$0.451_{\pm.012}$	$0.596_{\pm.056}$	$0.625_{\pm.065}$	$0.410_{\pm.026}$	$0.680_{\pm.058}$	$0.531_{\pm.018}$	$0.372_{\pm.023}$	$0.350_{\pm.002}$	$0.352_{\pm.002}$
	336	$0.406_{\pm.028}$	$0.415_{\pm.022}$	$0.410_{\pm.063}$	$0.412_{\pm.024}$	$0.462_{\pm.049}$	$0.481_{\pm.041}$	$0.581_{\pm.035}$	$0.574_{\pm.026}$	$0.430_{\pm.022}$	$0.666_{\pm.047}$	$0.580_{\pm.064}$	$0.394_{\pm.022}$	$0.371_{\pm.001}$	$0.376_{\pm.002}$
	720	$0.412_{\pm.008}$	$0.449_{\pm.022}$	$0.468_{\pm.012}$	$0.428_{\pm.024}$	$0.496_{\pm.019}$	$0.455_{\pm.025}$	$0.657_{\pm.017}$	$0.657_{\pm.014}$	$0.455_{\pm.020}$	$0.672_{\pm.015}$	$0.643_{\pm.046}$	$0.396_{\pm.012}$	$0.405_{\pm.013}$	$0.391_{\pm.001}$
ETTh2	96	$0.249_{\pm.015}$	$0.234_{\pm.011}$	$0.243_{\pm.009}$	$0.247_{\pm.028}$	$0.292_{\pm.012}$	$0.345_{\pm.042}$	$0.421_{\pm.033}$	$0.214_{\pm.018}$	$0.210_{\pm.014}$	$0.448_{\pm.031}$	$0.548_{\pm.158}$	$0.189_{\pm.010}$	$0.176_{\pm.002}$	$0.179_{\pm.003}$
	192	$0.249_{\pm.032}$	$0.247_{\pm.040}$	$0.252_{\pm.022}$	$0.265_{\pm.091}$	$0.376_{\pm.112}$	$0.343_{\pm.044}$	$0.339_{\pm.033}$	$0.294_{\pm.027}$	$0.235_{\pm.020}$	$0.575_{\pm.089}$	$0.766_{\pm.223}$	$0.213_{\pm.021}$	$0.209_{\pm.004}$	$0.200_{\pm.002}$
	336	$0.274_{\pm.027}$	$0.297_{\pm.029}$	$0.291_{\pm.032}$	$0.314_{\pm.045}$	$0.454_{\pm.057}$	$0.333_{\pm.078}$	$0.427_{\pm.041}$	$0.353_{\pm.028}$	$0.278_{\pm.028}$	$0.606_{\pm.095}$	$0.942_{\pm.408}$	$0.263_{\pm.039}$	$0.228_{\pm.002}$	$0.218_{\pm.002}$
	720	$0.286_{\pm.042}$	$0.667_{\pm.012}$	$0.401_{\pm.022}$	$0.371_{\pm.021}$	$1.092_{\pm.019}$	$0.412_{\pm.020}$	$0.482_{\pm.022}$	$0.382_{\pm.030}$	$0.322_{\pm.025}$	$0.550_{\pm.018}$	$0.688_{\pm.161}$	$0.278_{\pm.020}$	$0.241_{\pm.003}$	$0.221_{\pm.002}$
Electricity	96	$0.146_{\pm.015}$	$0.134_{\pm.002}$	$0.137_{\pm.002}$	$0.168_{\pm.012}$	$0.108_{\pm.009}$	$0.114_{\pm.010}$	$0.441_{\pm.013}$	$0.203_{\pm.189}$	$0.104_{\pm.006}$	$0.119_{\pm.003}$	$0.118_{\pm.003}$	$0.093_{\pm.002}$	$0.086_{\pm.001}$	$0.089_{\pm.003}$
	192	$0.143_{\pm.023}$	$0.137_{\pm.022}$	$0.143_{\pm.112}$	$0.163_{\pm.032}$	$0.120_{\pm.033}$	$0.131_{\pm.008}$	$0.441_{\pm.005}$	$0.264_{\pm.129}$	$0.112_{\pm.010}$	$0.124_{\pm.005}$	$0.121_{\pm.003}$	$0.102_{\pm.010}$	$0.092_{\pm.000}$	$0.096_{\pm.008}$
	336	$0.151_{\pm.017}$	$0.136_{\pm.002}$	$0.139_{\pm.002}$	$0.168_{\pm.010}$	$0.122_{\pm.018}$	-	$0.571_{\pm.022}$	-	$0.118_{\pm.006}$	$0.126_{\pm.008}$	$0.123_{\pm.001}$	$0.107_{\pm.002}$	$0.103_{\pm.002}$	$0.102_{\pm.003}$
	720	$0.149_{\pm.025}$	$0.140_{\pm.009}$	$0.149_{\pm.012}$	$0.164_{\pm.024}$	$0.136_{\pm.098}$	-	$0.622_{\pm.142}$	-	$0.126_{\pm.014}$	$0.134_{\pm.004}$	$0.144_{\pm.017}$	$0.117_{\pm.019}$	$0.120_{\pm.002}$	$0.111_{\pm.001}$
Traffic	96	$0.377_{\pm.024}$	$0.332_{\pm.008}$	$0.332_{\pm.007}$	$0.228_{\pm.010}$	$0.274_{\pm.012}$	$0.265_{\pm.007}$	$0.342_{\pm.042}$	-	$0.238_{\pm.008}$	$0.234_{\pm.006}$	$0.231_{\pm.003}$	$0.230_{\pm.010}$	$0.282_{\pm.010}$	$0.223_{\pm.004}$
	192	$0.388_{\pm.011}$	$0.326_{\pm.009}$	$0.350_{\pm.010}$	$0.225_{\pm.012}$	-	-	$0.354_{\pm.012}$	-	$0.242_{\pm.006}$	$0.239_{\pm.004}$	$0.236_{\pm.002}$	$0.234_{\pm.003}$	$0.280_{\pm.004}$	$0.230_{\pm.006}$
	336	$0.416_{\pm.028}$	$0.335_{\pm.010}$	$0.405_{\pm.011}$	$0.242_{\pm.022}$	-	-	$0.392_{\pm.006}$	-	$0.250_{\pm.007}$	$0.246_{\pm.003}$	$0.248_{\pm.006}$	$0.242_{\pm.007}$	$0.277_{\pm.003}$	$0.235_{\pm.008}$
	720	$0.432_{\pm.032}$	$0.361_{\pm.030}$	$0.453_{\pm.022}$	$0.253_{\pm.012}$	-	-	$0.478_{\pm.006}$	-	$0.256_{\pm.010}$	$0.263_{\pm.001}$	$0.264_{\pm.006}$	$0.248_{\pm.010}$	$0.284_{\pm.007}$	$0.236_{\pm.009}$
Weather	96	$0.146_{\pm.019}$	$0.144_{\pm.017}$	$0.279_{\pm.027}$	$0.145_{\pm.016}$	$0.176_{\pm.011}$	$0.139_{\pm.010}$	$0.113_{\pm.022}$	$0.087_{\pm.012}$	$0.095_{\pm.010}$	$0.164_{\pm.023}$	$0.145_{\pm.017}$	$0.086_{\pm.011}$	$0.091_{\pm.005}$	$0.077_{\pm.009}$
	192	$0.148_{\pm.022}$	$0.145_{\pm.015}$	$0.264_{\pm.013}$	$0.144_{\pm.012}$	$0.166_{\pm.022}$	$0.160_{\pm.037}$	$0.144_{\pm.020}$	$0.086_{\pm.007}$	$0.092_{\pm.010}$	$0.158_{\pm.024}$	$0.147_{\pm.025}$	$0.083_{\pm.011}$	$0.093_{\pm.003}$	$0.080_{\pm.010}$
	336	$0.152_{\pm.032}$	$0.146_{\pm.011}$	$0.283_{\pm.021}$	$0.149_{\pm.023}$	$0.168_{\pm.014}$	$0.170_{\pm.027}$	$0.138_{\pm.033}$	$0.098_{\pm.002}$	$0.103_{\pm.008}$	$0.162_{\pm.006}$	$0.160_{\pm.012}$	$0.093_{\pm.010}$	$0.096_{\pm.003}$	$0.086_{\pm.005}$
	720	$0.162_{\pm.009}$	$0.147_{\pm.019}$	$0.317_{\pm.021}$	$0.152_{\pm.029}$	$0.149_{\pm.034}$	$0.148_{\pm.040}$	$0.141_{\pm.026}$	$0.102_{\pm.005}$	$0.106_{\pm.008}$	$0.136_{\pm.020}$	$0.135_{\pm.008}$	$0.099_{\pm.009}$	$0.102_{\pm.001}$	$0.086_{\pm.006}$
Exchange	96	$0.079_{\pm.005}$	$0.077_{\pm.001}$	$0.069_{\pm.007}$	$0.079_{\pm.002}$	$0.033_{\pm.003}$	$0.036_{\pm.009}$	$0.090_{\pm.010}$	$0.036_{\pm.005}$	$0.040_{\pm.004}$	$0.079_{\pm.002}$	$0.091_{\pm.009}$	$0.032_{\pm.002}$	$0.024_{\pm.000}$	$0.029_{\pm.002}$
	192	$0.081_{\pm.015}$	$0.078_{\pm.008}$	$0.060_{\pm.007}$	$0.081_{\pm.002}$	$0.044_{\pm.004}$	$0.058_{\pm.007}$	$0.106_{\pm.010}$	$0.058_{\pm.005}$	$0.049_{\pm.006}$	$0.100_{\pm.019}$	$0.087_{\pm.005}$	$0.040_{\pm.005}$	$0.035_{\pm.000}$	$0.037_{\pm.006}$
	336	$0.086_{\pm.003}$	$0.083_{\pm.005}$	$0.071_{\pm.005}$	$0.085_{\pm.010}$	$0.074_{\pm.017}$	$0.058_{\pm.009}$	$0.106_{\pm.010}$	$0.076_{\pm.006}$	$0.065_{\pm.006}$	$0.086_{\pm.008}$	$0.091_{\pm.002}$	$0.054_{\pm.001}$	$0.047_{\pm.001}$	$0.050_{\pm.002}$
	720	$0.116_{\pm.022}$	$0.113_{\pm.015}$	$0.097_{\pm.011}$	$0.126_{\pm.001}$	$0.182_{\pm.010}$	$0.191_{\pm.006}$	$0.142_{\pm.009}$	$0.173_{\pm.020}$	$0.094_{\pm.012}$	$0.113_{\pm.016}$	$0.103_{\pm.009}$	$0.084_{\pm.017}$	$0.080_{\pm.002}$	$0.077_{\pm.001}$
ILI	24	$0.303_{\pm.021}$	$0.265_{\pm.027}$	$0.271_{\pm.032}$	$0.382_{\pm.018}$	$0.124_{\pm.019}$	$0.118_{\pm.033}$	$0.242_{\pm.086}$	$0.263_{\pm.012}$	$0.135_{\pm.010}$	$0.296_{\pm.044}$	$0.283_{\pm.001}$	$0.116_{\pm.009}$	$0.116_{\pm.015}$	$0.101_{\pm.008}$
	36	$0.262_{\pm.013}$	$0.222_{\pm.047}$	$0.258_{\pm.058}$	$0.286_{\pm.037}$	$0.144_{\pm.011}$	$0.143_{\pm.089}$	$0.246_{\pm.117}$	$0.298_{\pm.011}$	$0.160_{\pm.010}$	$0.298_{\pm.048}$	$0.307_{\pm.007}$	$0.142_{\pm.008}$	$0.158_{\pm.010}$	$0.134_{\pm.009}$
	48	$0.334_{\pm.028}$	$0.262_{\pm.023}$	$0.225_{\pm.043}$	$0.291_{\pm.032}$	$0.159_{\pm.020}$	$0.160_{\pm.039}$	$0.275_{\pm.044}$	$0.301_{\pm.034}$	$0.173_{\pm.012}$	$0.320_{\pm.025}$	$0.314_{\pm.009}$	$0.152_{\pm.017}$	$0.168_{\pm.015}$	$0.140_{\pm.008}$
	60	$0.288_{\pm.031}$	$0.278_{\pm.017}$	$0.245_{\pm.017}$	$0.287_{\pm.023}$	$0.216_{\pm.014}$	$0.183_{\pm.019}$	$0.272_{\pm.020}$	$0.299_{\pm.013}$	$0.190_{\pm.012}$	$0.325_{\pm.068}$	$0.

Table 14: Additional comparison with diagnostic DeepAR, HeTVAE, and VHVM baselines. DeepAR provides an autoregressive recurrent likelihood control, HeTVAE is adapted from irregular imputation/interpolation to long-horizon forecasting by conditioning on the look-back window and querying the future horizon, and VHVM provides a volatility-specific recurrent heteroscedastic VAE control. VHVM entries are single-seed reproduced results under the same data split and horizon protocol. VolDy-VAE values are the corresponding means from Table 12 and Table 13. Lower is better; bold indicates the best value in each row group.

Dataset	Horizon	CRPS				NMAE			
		DeepAR	HeTVAE	VHVM	VolDy-VAE	DeepAR	HeTVAE	VHVM	VolDy-VAE
ETTm1	96	0.335	0.310	0.344	**0.220**	0.421	0.395	0.448	**0.286**
	192	0.502	0.357	0.348	**0.235**	0.638	0.462	0.439	**0.305**
	336	0.440	0.434	0.398	**0.249**	0.554	0.575	0.525	**0.323**
	720	0.454	0.466	0.489	**0.272**	0.586	0.636	0.639	**0.358**
ETTm2	96	0.416	0.215	0.273	**0.109**	0.509	0.292	0.345	**0.137**
	192	0.415	0.266	0.295	**0.126**	0.501	0.369	0.380	**0.158**
	336	0.490	0.278	0.334	**0.142**	0.586	0.378	0.456	**0.178**
	720	0.342	0.353	0.372	**0.164**	0.403	0.465	0.480	**0.205**
ETTh1	96	0.418	0.466	0.499	**0.243**	0.533	0.627	0.627	**0.323**
	192	0.443	0.460	0.476	**0.266**	0.548	0.616	0.635	**0.352**
	336	0.421	0.484	0.482	**0.284**	0.510	0.654	0.644	**0.376**
	720	0.480	0.589	0.559	**0.310**	0.591	0.755	0.765	**0.391**
ETTh2	96	0.300	0.316	0.417	**0.141**	0.380	0.428	0.550	**0.179**
	192	0.422	0.395	0.526	**0.160**	0.545	0.523	0.682	**0.200**
	336	0.409	0.500	0.552	**0.172**	0.533	0.631	0.713	**0.218**
	720	0.416	0.781	0.544	**0.174**	0.537	0.926	0.708	**0.221**
Electricity	96	0.101	0.256	0.096	**0.069**	0.130	0.366	0.125	**0.089**
	192	0.128	0.254	0.102	**0.075**	0.167	0.364	0.137	**0.096**
	336	0.117	0.255	0.110	**0.078**	0.154	0.365	0.148	**0.102**
	720	0.183	0.257	0.118	**0.080**	0.243	0.366	0.153	**0.111**
Traffic	96	0.260	0.221	0.215	**0.180**	0.323	0.254	0.274	**0.223**
	192	0.268	0.291	0.283	**0.187**	0.337	0.364	0.378	**0.230**
	336	0.524	0.361	0.347	**0.194**	0.637	0.464	0.463	**0.235**
	720	0.299	0.401	0.395	**0.192**	0.386	0.514	0.506	**0.236**
Weather	96	0.171	0.080	0.098	**0.070**	0.211	0.100	0.129	**0.077**
	192	0.139	0.128	0.098	**0.067**	0.161	0.163	0.123	**0.080**
	336	0.174	0.137	0.096	**0.071**	0.215	0.173	0.120	**0.086**
	720	0.137	0.090	0.199	**0.073**	0.161	0.112	0.248	**0.086**
Exchange	96	0.058	0.065	0.063	**0.022**	0.065	0.083	0.074	**0.029**
	192	0.101	0.068	0.092	**0.030**	0.114	0.092	0.106	**0.037**
	336	0.074	0.074	0.084	**0.041**	0.085	0.100	0.097	**0.050**
	720	0.130	0.108	0.087	**0.065**	0.148	0.126	0.103	**0.077**
ILI	24	0.208	0.300	0.253	**0.082**	0.238	0.321	0.273	**0.101**
	36	0.318	0.321	0.309	**0.110**	0.335	0.347	0.331	**0.134**
	48	0.346	0.302	0.293	**0.114**	0.361	0.323	0.308	**0.140**
	60	0.322	0.334	0.288	**0.130**	0.339	0.349	0.302	**0.155**

D.8 Architectural Ablation: Scale Head Design

One claim of this work is that modeling the temporal dynamics of volatility, rather than only estimating per-step variance, improves probabilistic forecasting. To isolate the contribution of the recurrent volatility dynamics module, we compare three scale head architectures while keeping all other components (encoder, latent dynamics, location head, NLL objective) identical: (i) **MLP** (**no-GRU independent-σ**): a feed-forward scale head that removes the GRU state and predicts each σ_t independently from $\mathbf{Z}_t$; (ii) **LSTM**: a Long Short-Term Memory unit replacing the GRU, serving as an alternative recurrent architecture with separate cell and hidden states; and (iii) **GRU** (ours): the proposed gated recurrent volatility dynamics module. This within-architecture comparison is the strict independent-σ control because it changes only the scale-state pathway. Autoregressive recurrent forecasters such as DeepAR use recurrent states for recursive value generation and likelihood-parameter emission, which is a different control covered empirically by the DeepAR diagnostic comparison in Table 14.

Table 15: Ablation on scale head architecture. MLP is the no-GRU independent-σ variant, while LSTM and GRU use recurrent scale states. We compare these scale heads on *ETTh1* and *ETTh2* ($L = 96$). RED: THE BEST.

Dataset	H	CRPS ↓			NMAE ↓		
		MLP	LSTM	GRU	MLP	LSTM	GRU
ETTh1	96	0.271	0.255	0.243	0.350	0.331	0.323
	192	0.296	0.280	0.266	0.381	0.368	0.352
	336	0.320	0.304	0.284	0.401	0.388	0.376
	720	0.328	0.314	0.310	0.406	0.393	0.391
ETTh2	96	0.162	0.151	0.141	0.206	0.186	0.179
	192	0.179	0.165	0.160	0.223	0.205	0.200
	336	0.195	0.179	0.172	0.243	0.225	0.218
	720	0.197	0.178	0.174	0.248	0.227	0.221

We note that the recurrent state is updated over latent patches, but the predicted scale is not restricted to be constant inside a patch. For each latent patch, the scale readout outputs a full vector that is reshaped to the same per-step resolution as the location output. Thus, the patch-level GRU provides memory across local windows, while within-patch variation is still represented by the decoder readout. The remaining trade-off is temporal resolution: if volatility changes faster than the selected patch length can represent, smaller or overlapping patches, or a multi-scale refinement head, would be appropriate extensions. The patch-length sensitivity analysis in Appendix Section D.9 evaluates this trade-off for $P \in \{16, 24, 32, 48\}$.

To directly evaluate uncertainty dynamics rather than only aggregate forecast scores, we further report four quantitative diagnostic metrics in Table 5: empirical 95% coverage, interval sharpness (average 95% interval width normalized by the target standard deviation; lower is sharper under comparable coverage), QICE, and scale persistence ρ_σ (lag-1 autocorrelation of the predicted scale; higher indicates more temporally coherent volatility estimates). These diagnostics are analyzed in Section 4.3; metric definitions are provided in Appendix Section D.2.

As shown in Table 15, the results reveal a clear hierarchy: GRU $>$ LSTM $>$ MLP across all settings. The performance gap between MLP and GRU widens with longer horizons (e.g., on *ETTh2* at $H = 720$, CRPS improves from 0.197 to 0.174, a 11.7% reduction), supporting the importance of temporal volatility memory for long-horizon uncertainty estimation. The MLP scale head, despite using the same NLL objective, produces independent per-step variance estimates that lack temporal coherence, leading to noisy and poorly calibrated confidence intervals. The diagnostic results in Table 5 make this failure mode explicit: MLP has lower 95% coverage while producing wider intervals and lower ρ_σ, whereas GRU maintains near-nominal coverage with sharper intervals and more persistent predicted scale dynamics.

The comparison between GRU and LSTM is also informative. While both recurrent architectures outperform MLP substantially, GRU achieves slightly better performance with fewer parameters (no separate cell state). One possible explanation is twofold: (i) the GRU's simpler gating structure may act as an implicit regularizer for the relatively low-dimensional scale estimation task; and (ii) the GRU's direct hidden state update (without the LSTM's cell state bottleneck) may allow more responsive adaptation to abrupt regime transitions, which aligns with the sharp volatility shifts observed in real-world data. This architectural choice echoes the main-text discussion of the location-scale volatility module: just as GARCH's parsimonious two-parameter recursion (α, β) suffices to capture volatility persistence, the GRU's minimal gating structure provides an efficient inductive bias for temporal volatility modeling.

D.9 Hyperparameter Sensitivity Analysis

We vary three hyperparameters to test the sensitivity of VolDy-VAE: Patch Length (P), Hidden Dimension (D), and the number of MLP encoder layers (N). Here, N refers only to the depth of the MLP-based encoder; the volatility scale head remains a single-layer GRU throughout all experiments. We perform experiments on the representative forecasting task ($L = 96 \rightarrow H = 192$). Table 16 details the performance on two distinct datasets: *ETTh1* (representative of datasets with strong periodicity) and *Weather* (representative of high-dimensional, volatile datasets).

Table 16: **Hyperparameter Sensitivity ($L = 96 \rightarrow H = 192$).** We report CRPS on *ETTh1* and *Weather*. The default setting is selected as a cross-dataset performance–efficiency trade-off: $P = 24, D = 256, N = 3$, where N is the number of MLP encoder layers.

Parameter	Patch Length (P)		Hidden Dim (D)		MLP Layers (N)	
Dataset	*Value*	*CRPS*	*Value*	*CRPS*	*Value*	*CRPS*
ETTh1	16	0.278	64	0.280	1	0.279
	24	0.266	128	0.279	2	0.278
	32	0.278	256	0.266	3	0.266
	48	0.294	512	0.278	4	0.274
Weather	16	0.069	64	0.084	1	0.074
	24	0.067	128	0.086	2	0.074
	32	0.077	256	0.067	3	0.067
	48	0.073	512	0.074	4	0.067

Analysis Across the settings in Table 16, CRPS varies modestly across patch length, hidden dimension, and encoder depth. This pattern suggests that the default configuration does not rely on narrow hyperparameter tuning. The default $P = 24, D = 256, N = 3$ achieves the lowest CRPS on *ETTh1* among the tested default-axis configurations and ties $N = 4$ on *Weather*. We therefore adopt it as a trade-off between performance and computational efficiency across datasets. This setting provides sufficient encoder capacity and receptive field coverage while maintaining a lightweight architecture, avoiding the marginal additional cost of deeper MLP encoders.

E Limitations and Future Work

Several properties of the current formulation define its applicability and motivate extensions. First, while the recurrent volatility module captures temporal coherence, its temporal resolution is tied to the selected patch length. Although the decoder outputs per-step scale values within each patch, extremely fast volatility changes at a finer-than-patch scale may still be smoothed by the latent representation. For domains where such sub-patch volatility is central, smaller or overlapping patches, multi-scale volatility heads, or explicit regime-switching mechanisms (e.g., Hidden Markov Models) could further improve adaptability. Second, VolDy-VAE uses a conditionally factorized Gaussian observation model whose scale dynamics are induced by a neural recurrent state over latent representations. This design is efficient and empirically effective for probabilistic forecasting. Explicitly specified stochastic-volatility or GARCH-like priors provide a complementary route for structural interpretability. Future work can combine VolDy with latent SV states, GARCH-inspired priors, or state-space volatility transitions when volatility inference itself is the scientific target. Third, our evaluation focuses on standard benchmarks with moderate dimensionality; very high-dimensional settings (e.g., thousands of correlated variables) call for scalable cross-variable structure. Finally, cross-variable volatility spillovers, which are important in domains such as financial contagion modeling, can be incorporated through multivariate volatility states or structured dependency priors.

F Additional Showcases

To provide additional qualitative evidence for VolDy-VAE's modeling capabilities, we present visualization results on four datasets: *ETTh2*, and *ETTm1*, *ETTm2*, *Weather*. These cases cover a wide spectrum of temporal dynamics, ranging from smooth periodic trends to highly volatile fluctuations and structural breaks.

Adaptive Responsiveness in Volatile Regimes Figure 6 illustrates performance on *ETTh2* and *ETTm1*, which contain complex volatility and abrupt regime shifts (e.g., the step-like drops in the

top row of *ETTh2*). Here, VolDy-VAE's predictive intervals expand around sudden transitions or volatility spikes and contract once the series stabilizes. These plots show scale adaptation; they do not isolate its effect on location gradients during training. Baseline models tend to generate more static, uniform bands that fail to capture the temporal structure of the uncertainty. This visual evidence supports the claim that the Location-Scale formulation makes the uncertainty estimate more context-aware.

Failure of Baselines in Stable Regimes As shown in Figure 7, specifically on the *Weather* and *ETTm2* datasets, the ground truth series exhibit high stability or smooth periodicity. In these scenarios, an ideal probabilistic model should output narrow confidence intervals (CIs) to reflect low aleatoric uncertainty. However, baseline methods, including the generative K^2VAE and the probabilistic-enhanced point forecasters (Koopa, PatchTST), produce wider and less adaptive uncertainty bands. On the visualized *ETTh1* window reported in the main text, their average normalized sharpness values are substantially larger than VolDy-VAE's while their empirical coverage is also higher, indicating measurable over-coverage rather than only a visual impression. In contrast, VolDy-VAE estimates lower volatility in these windows, producing sharper CIs around the ground truth.

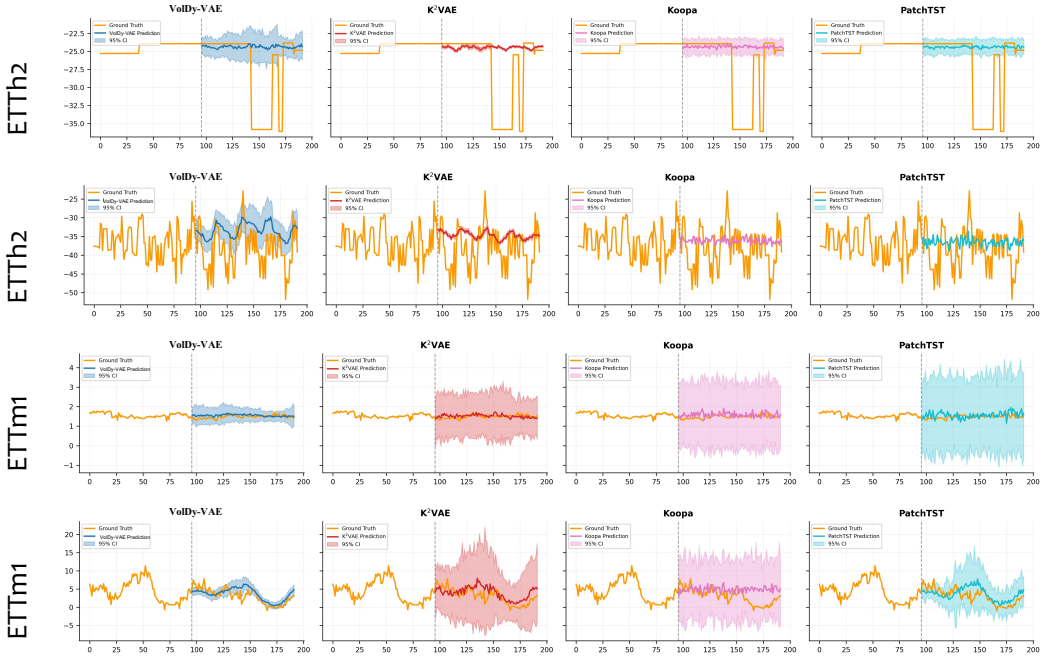

Figure 6: Visualization on Volatile Regimes (*ETTh2*, *ETTm1*). All subfigures share a unified Y-axis scale. The panels focus on periods with high volatility and structural breaks. Observation: In the top row of *ETTh2*, the data exhibits sudden jumps. VolDy-VAE produces uncertainty bands that expand near regime shifts and contract otherwise. In contrast, baselines produce more static bands that capture less of the dynamic risk pattern.



Figure 7: Visualization on Stable Regimes (*ETTm2*, *Weather*). All subfigures share a unified Y-axis scale. We compare VolDy-VAE against baselines on time series with smooth trends. Observation: Baselines (K^2VAE, Koopa, PatchTST) show uncertainty over-estimation, generating wide confidence intervals even for flat or smooth curves (e.g., *Weather*). VolDy-VAE (Blue) estimates lower aleatoric uncertainty in these stable windows, producing sharper intervals that better reflect the visual smoothness of the prediction target.