Two-stage Odd Residual Flows for Mean-Preserving Probabilistic Time Series Forecasting

Kiran Madhusudhanan*, Christian Klötergens, Lars Schmidt-Thieme, Vijaya Krishna Yalavarthi

Institute of Computer Science
University of Hildesheim
Hildesheim, Germany

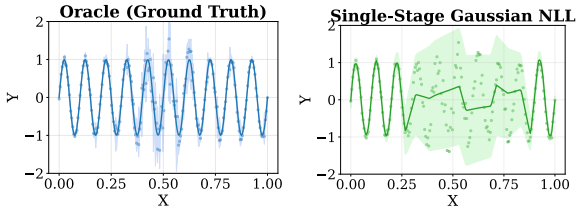
Abstract

Probabilistic forecasting plays an essential role in risk-sensitive decision-making, particularly in long-horizon settings. However, existing approaches often face a fundamental trade-off between distributional flexibility and accurate mean prediction. Traditional parametric methods, such as Mean Variance Estimation (MVE), can suffer from degraded point accuracy when trained under joint Negative Log-Likelihood (NLL) objectives, while modern-flexible generative models, including Normalizing Flows and Diffusion Models, typically rely on costly Monte Carlo sampling and may yield suboptimal mean estimates. To address this limitation, we propose Two-stage Odd Residual Flows (TORF), a framework that decouples mean forecasting from uncertainty estimation. In the first stage, a pre-trained deterministic model is used to produce an accurate mean prediction. In the second stage, a Restricted Normalizing Flow, with strictly odd functions learns flexible residual distributions around the point forecast, guaranteeing mean preservation from the first stage without sampling. Experiments show that TORF achieves state-of-the-art deterministic accuracy (NMAE) while providing strong density estimation performance (CRPS) on short and long-horizon forecasting.

1 Introduction

Time Series Forecasting (TSF) is central to decision-making in domains such as demand planning, energy management, and finance (Bandara et al. 2019; Dimoulkas et al. 2018; Luo et al. 2018), where quantifying uncertainty is key to risk-aware decisions. While deterministic mean prediction has advanced rapidly (Wang et al. 2026a), uncertainty quantification (UQ) (Zhang et al. 2024) still faces a trade-off between distributional flexibility and mean accuracy.

The de-facto method for probabilistic prediction is Mean-Variance Estimation (MVE) (Nix and Weigend 1994), which assumes a parametric target distribution (typically a heteroscedastic Gaussian) and optimizes its parameters via Negative Log-Likelihood (NLL) loss (Salinas et al. 2020). Although MVE methods provide an analytical, direct mean, their Gaussian distribution assumption is limiting in practice. Furthermore, training MVE models via joint NLL optimization triggers an inherent trade-off between mean and variance learning (Seitzer et al. 2022). For example, consider Figure 1a, where the target follows a simple sinusoidal function with non-Gaussian noise, $y = \sin(2\pi f x) +$ Student-t$(\nu(x), 0, \gamma(x))$. Direct NLL training Figure 1b fails to recover the true mean in high-noise regions because the loss function heavily prioritizes low-noise regimes (Seitzer et al. 2022), whereas a two-stage approach that first fits the mean and then the variance recovers both Figure 1c.



(a) Oracle; $y = \sin(2\pi f x) +$ Student-t$(\nu(x), 0, \gamma(x))$

(b) Single-stage CRPS: 0.2628; MAE: 0.2628

(c) Two-stage Gauss CRPS: 0.0986; MAE: 0.1383

(d) Two-stage Flow CRPS: 0.0967; MAE: 0.1383

Figure 1: Collapse of Gaussian negative log likelihood learning based single stage approaches on a simple sinusoidal signal with Student's t-distribution noise.

Flexible density modeling approaches based on Diffusion (Tashiro et al. 2021), Normalizing Flows (Rasul et al. 2021b), or VAEs (Wu et al. 2025) sidestep rigid distributional assumptions and can model complex densities. However, they inherit poor mean accuracy for two distinct reasons. First, like MVE, they optimize a distributional objective (e.g., the ELBO or a log-likelihood loss) rather than a mean-focused loss, so the predictive mean is only an implicit by-product of fitting the full density and is never directly supervised. Second, unlike MVE, they lack an analytical mean altogether: recovering it requires expensive Monte Carlo sampling, which introduces additional estimation error that worsens in high dimensions as samples become scarce relative to the space

*Corresponding author. Email: kiranmadhusud@ismll.de

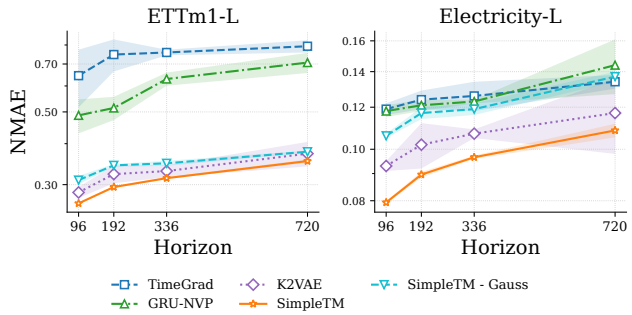


Figure 2: NMAE results on two time series datasets. Current probabilistic frameworks (TimeGrad, GRU-NVP, K^2VAE, SimpleTM-Gauss/MVE(SimpleTM)) exhibit higher NMAE than the point-prediction model (SimpleTM).

they must cover. These effects compound, and as a result state-of-the-art probabilistic models like K^2VAE (Wu et al. 2025) consistently trail deterministic point-predictors like SimpleTM (Chen et al. 2025) in mean accuracy (Figure 2).

To break this trade-off, we introduce Two-stage Odd Residual Flows (TORF), completely isolating mean prediction from uncertainty estimation through a decoupled framework: *Stage 1 (Deterministic)*: An off-the-shelf point-prediction model (e.g., SimpleTM) captures the conditional mean. Free from NLL-driven variance distortion, it retains a high point prediction accuracy. *Stage 2 (Probabilistic)*: A Restricted Normalizing Flow (RNF) (Kobayashi and Aotani 2023) models the residual density. By restricting the flow architecture to strictly odd functions, we capture highly flexible, symmetric error distributions while mathematically guaranteeing that the exact analytical mean from Stage 1 is preserved without sampling. As shown in Figure 1d the two-stage approach retains the point-prediction accuracy of its Stage-1 baseline and builds the predictive uncertainty around that fixed mean, circumventing the joint NLL trade-off entirely.

1. We show that a SimpleTM based two-stage Gaussian baseline (MVE-2S) already matches or beats K^2VAE on 11/12 distributional-accuracy comparisons (Table 6).
2. We introduce TORF, which extends this decoupled framework with an odd-constrained residual flow (ROSS), modeling flexible, non-Gaussian residuals while provably preserving the exact Stage-1 mean, without sampling.
3. We design a lightweight CNN architecture that efficiently maps time-series contexts to ROSS's parameters.
4. TORF with SimpleTM (Chen et al. 2025) as 1st stage model sets a new state of the art, winning 7/9 CRPS and 9/9 NMAE on long-horizon forecasting and 6/8 CRPS and 5/8 NMAE on short-horizon forecasting, with TORF adding a consistent gain over K^2VAE and MVE-2S.

2 Related works

Deterministic forecasting has progressed from classical statistical models like ARIMA (Box, Jenkins, and Reinsel 1994) to modern deep architectures like Transformers. These include Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), FEDformer (Zhou et al. 2022), Yformer (Madhusudhanan et al. 2022), and Pyraformer (Liu et al. 2022) to name a few. Among them iTransformer (Liu et al. 2024) and PatchTST (Nie et al. 2023) remain among the strongest performers. Recently, SimpleTM (Chen et al. 2025) achieved state-of-the-art point accuracy combining wavelet preprocessing with a geometric-product attention. Owing to its strong performance, we adopt SimpleTM as the Stage 1 model in our experiments. TORF is agnostic to this choice.

	Analytic Mean	Analytic Median	Flexible Density	Exact NLL
K^2VAE	✗	✗	✓	✗
TimeGrad	✗	✗	✓	✗
MVE	✓	✓	✗	✓
GRU-NVP	✗	✓	✓	✓
TORF (Ours)	✓	✓	✓	✓

Table 1: Comparison of Probabilistic Forecasting Models.

Probabilistic forecasting aims to recover the full predictive distribution rather than a single point estimate. Autoregressive models such as DeepAR (Salinas et al. 2020) learn Gaussian parameters via an RNN. Modern diffusion-based methods treat forecasting as an iterative denoising process, like TimeGrad (Rasul et al. 2021a), CSDI (Tashiro et al. 2021), and TSDiff (Kollovieh et al. 2023). Conditional normalizing flows such as GRU-NVP (Rasul et al. 2021b) and TSFlow (Kollovieh et al. 2025) instead model complex joint densities without a fixed parametric form. K^2VAE (Wu et al. 2025), a VAE-based method for long-horizon probabilistic forecasting, combines a Koopman latent dynamical system with a KalmanNet-based correction. Despite this diversity, all of these methods either commit to a fixed distributional family or fall back on Monte Carlo sampling to recover a predictive mean. Table 1 differentiates these approaches from TORF, which decouples mean estimation from density estimation entirely, obtaining both an analytical mean and a flexible density, inspired by Kobayashi and Aotani (2023)'s restriction of the flow architecture to be odd.

3 Preliminaries

Probabilistic Time Series Forecasting A *forecasting instance* is a tuple $(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^{L\times C} \times \mathbb{R}^{H\times C}$, where $\mathbf{X} = (\mathbf{z}_{t-L+1}, \ldots, \mathbf{z}_t)$ is the look-back window of length L and $\mathbf{Y} = (\mathbf{z}_{t+1}, \ldots, \mathbf{z}_{t+H})$ is the target of horizon H over C channels. Probabilistic forecasting aims to estimate the full conditional distribution $p_\theta(\mathbf{Y} \mid \mathbf{X})$. Given N i.i.d. instances drawn from an unknown joint distribution P, the standard training objective is the NLL:

$$\min_\theta \ \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim P}[-\log p_\theta(\mathbf{Y} \mid \mathbf{X})]\,. \tag{1}$$

A de-facto method is assuming p to be Gaussian and train for Gaussian NLL.

Pitfalls of NLL Training Seitzer et al. (2022) show that jointly training a Gaussian NLL head creates a mean-variance conflict. The gradient of the NLL with respect to the

mean $\boldsymbol{\mu}$ is scaled element-wise by $1/\boldsymbol{\sigma}^2$, so high-variance regions contribute less to mean learning (Figure 1). This effect worsens in high-dimensional, long-horizon settings, where density heads must model complex distributions, degrading point accuracy relative to a stand-alone mean predictor.

Faithful Heteroscedastic Regression Stirn et al. (2023) eliminate this conflict by *fully decoupling* the two objectives. A first-stage mean predictor f_{θ_1} is trained under an MSE objective and frozen, then a second-stage network models only the residual distribution $p_{\theta_2}(\mathbf{Y} - f_{\theta_1}(\mathbf{X}) \mid \mathbf{X})$. Because the mean network is fixed, the joint model's mean is guaranteed to be *at least as accurate* as the stand-alone predictor, a property they call *faithfulness*. However, they restrict the residual distribution to Gaussian, limiting expressiveness for heavy-tailed or multi-modal targets. TORF keeps this faithfulness guarantee while replacing the Gaussian residual with a Normalizing Flow, adding flexible, non-Gaussian density estimation without sacrificing mean accuracy.

Normalizing Flows A Normalizing Flow (NF) (Papamakarios et al. 2021) defines a bijective, differentiable map $f : \mathbb{R}^K \to \mathbb{R}^K$ from a tractable base distribution $p_{\mathbf{u}}(\mathbf{u})$, typically $\mathcal{N}(\mathbf{0}, \mathbf{I})$, to a complex target $p_{\mathbf{v}}(\mathbf{v})$ via the change-of-variables formula:

$$p_{\mathbf{v}}(\mathbf{v};\theta) = p_{\mathbf{u}}\big(f^{-1}(\mathbf{v};\theta)\big)\left|\det \frac{\partial f^{-1}(\mathbf{v};\theta)}{\partial \mathbf{v}}\right|. \quad (2)$$

For conditional density estimation, f is additionally conditioned on the context $\mathbf{c}$, written $f(\mathbf{u};\mathbf{c},\theta)$. Tractability requires an efficient inverse and a Jacobian log-determinant computable in closed form.

Rational Quadratic Neural Spline Flows (RQ-NSF). Durkan et al. (2019) proposed monotonic piecewise rational-quadratic splines as the bijective transformation within the flow, yielding highly non-linear yet analytically invertible maps. Each dimension of the input is transformed independently by its own univariate spline, making the Jacobian diagonal and its log-determinant a sum of scalar log-derivatives. The spline is supported on a bounded interval $[-B, B]$, defaulting to the identity outside, and defined by N_b bins parameterized by widths $\mathbf{w}_b \in \mathbb{R}^{N_b}_{>0}$, heights $\mathbf{h}_b \in \mathbb{R}^{N_b}_{>0}$, and interior knot derivatives $\mathbf{d}_b \in \mathbb{R}^{N_b-1}_{>0}$, giving $3N_b - 1$ parameters per dimension. For a conditional distribution, these are predicted by a neural network conditioned on $\mathbf{X}$:

$$\boldsymbol{\phi} = (\mathbf{w}_b,\ \mathbf{h}_b,\ \mathbf{d}_b) = \mathrm{NN}(\mathbf{X};\theta_s), \qquad \boldsymbol{\phi} \in \mathbb{R}^{3N_b-1}_{>0}. \quad (3)$$

Odd Flows for Analytically Tractable Mean For an unconstrained NF the predictive mean $\mathbb{E}[\mathbf{v}] = \int \mathbf{v}\, p_{\mathbf{v}}(\mathbf{v})\, d\mathbf{v}$ is generally intractable. Kobayashi and Aotani (2023) showed that restricting f to the class of *odd functions*,

$$f_{\mathrm{odd}}(-\mathbf{u}) = -f_{\mathrm{odd}}(\mathbf{u}), \quad \forall\, \mathbf{u} \in \mathbb{R}^K, \quad (4)$$

guarantees $\mathbb{E}[f_{\mathrm{odd}}(\mathbf{u})] = 0$ whenever $p_{\mathbf{u}}$ is symmetric about the origin, making the mean analytically tractable without Monte Carlo sampling. For spline flows, odd symmetry is enforced by requiring $f(0) = 0$ and mirroring bin parameters exactly across positive and negative domains.

4 Methodology

Two-Stage Residual Modeling

As established in §3, jointly optimizing mean and density under the NLL degrades mean accuracy. TORF resolves this by fully decoupling the two objectives into a sequential two-stage framework.

Stage 1: Mean Estimation. Any point-predictor f_{θ_1} trained independently under an MSE objective to produce the conditional mean estimate $\hat{\mathbf{Y}} = f_{\theta_1}(\mathbf{X})$. TORF is model-agnostic at this stage i.e., any point-predictor can serve as f_{θ_1}.

Stage 2: Residual Density Estimation. Let $\boldsymbol{\varepsilon} = \mathbf{Y} - f_{\theta_1}(\mathbf{X})$ denote the residual. Since $\mathbf{Y} \mapsto \boldsymbol{\varepsilon}$ is a translation, its Jacobian is the identity and $p_{\varepsilon}(\boldsymbol{\varepsilon} \mid \mathbf{X}) = p_Y(f_{\theta_1}(\mathbf{X}) + \boldsymbol{\varepsilon} \mid \mathbf{X})$. With θ_1 frozen, a normalizing flow p_{θ_2} is trained to model $p(\boldsymbol{\varepsilon} \mid \mathbf{X})$ directly, constrained by construction to satisfy $\mathbb{E}_{p_{\theta_2}}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$, and the full predictive distribution is recovered as:

$$\hat{p}(\mathbf{Y} \mid \mathbf{X}) = p_{\theta_2}(\mathbf{Y} - f_{\theta_1}(\mathbf{X}) \mid \mathbf{X}). \quad (5)$$

A sample $\hat{Y} \sim \hat{p}(\mathbf{Y} \mid \mathbf{X})$ can be read as $f_{\theta_1}(x) + \hat{\varepsilon}$ where $\hat{\varepsilon} \sim p_{\theta_2}(\varepsilon \mid \mathbf{X})$. The design of p_{θ_2} is described in the following subsections. We make two assumptions on $p(\boldsymbol{\varepsilon} \mid \mathbf{X})$:

1. **Conditional Independence.** $p(\boldsymbol{\varepsilon} \mid \mathbf{X}) = \prod_{c,h} p(\varepsilon_{h,c} \mid \mathbf{X})$
2. **Symmetry.** $p(\boldsymbol{\varepsilon} \mid \mathbf{X}) = p(-\boldsymbol{\varepsilon} \mid \mathbf{X})$

These assumptions trade a small amount of distributional generality for the exact mean-preservation guarantee at the core of our method, and they are far milder in practice than they may appear: even under both, TORF outperforms state-of-the-art probabilistic forecasters on CRPS, their primary evaluation metric. In the ablation study, we introduce a mixture-of-TORF extension that captures asymmetric distributions. In future work, we discuss extending TORF to multivariate distributions.

Context Embedding

The flow components are conditioned on $\mathbf{X}$ through a shared context embedding $\mathbf{F} \in \mathbb{R}^{H \times C}$. There are two natural choices for constructing $\mathbf{F}$: (i) introduce a separate encoder trained from scratch alongside the second stage, or (ii) reuse the encoder learned by f_{θ_1}. When f_{θ_1} has differentiable parameters, as is the case for neural network-based predictors, option (ii) is preferable, as the encoder has already learned high-quality temporal representations under MSE, providing a strong initialization at no additional parameter cost. When f_{θ_1} is non-differentiable (e.g., gradient boosting), option (i) is necessary, and a separate neural encoder such as SimpleTM can be used instead.

In our experiments, f_{θ_1} is SimpleTM, so we adopt option (ii). All f_{θ_1} layers are frozen and the final linear prediction head is duplicated and kept trainable, allowing $\mathbf{F}$ to adapt to the needs of the flow without disturbing the learned encoder:

$$\mathbf{F} = \mathbf{W} \cdot \mathrm{sg}[\mathrm{enc}(\mathbf{X};\, \theta_1)] + \mathbf{b}, \quad (6)$$

Here, $\mathrm{enc}(\mathbf{X};\, \theta_1)$ denotes the output of frozen encoder layers of f_{θ_1}. $\mathbf{W}$ and $\mathbf{b}$ are the parameters of the duplicated,

Algorithm 1: TORF Forward Pass

Input: history $\mathbf{X}$, target $\mathbf{Y}$, pretrained f_{θ_1}
Parameter: flow layers K, $\theta_2 = \{\mathbf{W}, \mathbf{b}, \boldsymbol{\omega}, \boldsymbol{\Theta_s}\}$
Output: training loss $\mathcal{L}$

$\hat{\mathbf{Y}} \leftarrow f_{\theta_1}(\mathbf{X})$
$\mathbf{F} \leftarrow \mathbf{W} \cdot \mathrm{sg}[\mathrm{enc}(\mathbf{X};\, \theta_1)] + \mathbf{b}$
$\mathbf{U}^{(0)} \leftarrow \mathbf{Y} - \hat{\mathbf{Y}}$
$\mathbf{S} \leftarrow \mathrm{ScaleNet}(\mathbf{F};\, \boldsymbol{\omega})$
$\mathrm{ldj} \leftarrow 0$
for $k = 0$ **to** $K - 1$ **do**
  $\boldsymbol{\Phi}^{(k)} \leftarrow \mathrm{SplineNet}^{(k)}(\mathbf{F};\, \boldsymbol{\theta}_s^{(k)})$
  $\mathbf{U}^{(k+\frac{1}{2})},\, \mathrm{ldj}_{\mathrm{s}}^{(k)} \leftarrow \mathrm{LSL}(\mathbf{U}^{(k)};\, \mathbf{S})$
  $\mathbf{U}^{(k+1)},\, \mathrm{ldj}_{\mathrm{r}}^{(k)} \leftarrow \mathrm{ROSS}(\mathbf{U}^{(k+\frac{1}{2})};\, \boldsymbol{\Phi}^{(k)})$
  $\mathrm{ldj} \leftarrow \mathrm{ldj} + \mathrm{ldj}_{\mathrm{s}}^{(k)} + \mathrm{ldj}_{\mathrm{r}}^{(k)}$
end for
$\mathbf{U}^{(K+1)},\, \mathrm{ldj}_{\mathrm{s}}^{(K)} \leftarrow \mathrm{LSL}(\mathbf{U}^{(K)};\, \mathbf{S})$
$\mathrm{ldj} \leftarrow \mathrm{ldj} + \mathrm{ldj}_{\mathrm{s}}^{(K)}$
return $\mathcal{L} = -\log \mathcal{N}\big(\mathbf{U}^{(K+1)};\, \mathbf{0}, \mathbf{I}\big) - \mathrm{ldj}$

unfrozen final layer of f_{θ_1}, which are initialized from the pre-trained weights and fine-tuned during the second stage. $\mathrm{sg}\,[\cdot]$ denotes the stop-gradient operator, which blocks gradients from flowing into $\mathrm{enc}(\mathbf{X};\, \theta_1)$. All subsequent modules take $\mathbf{F}$ as their conditioning input.

Residual Odd Splines (ROSS)

An unconstrained p_{θ_2} can introduce a location shift in $p(\boldsymbol{\varepsilon} \mid \mathbf{X})$, decoupling the predictive mean of the model from $f_{\theta_1}(\mathbf{X})$. We prevent this by restricting the spline to the class of odd functions (§3), analytically enforcing $\mathbb{E}_{p_{\theta_2}}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$.

Let S be a monotonic rational-quadratic spline defined over $[0, B]$, defaulting to the identity outside. Rather than mirroring bin parameters across both domains, we process only the magnitude of the input through S and restore the sign $(\mathrm{sgn}(\cdot))$ afterward. The forward and inverse transformations for each scalar element $u_{h,c}$ are:

$$v_{h,c} = \mathrm{sgn}(u_{h,c}) \cdot S\big(|u_{h,c}|;\, \boldsymbol{\phi}_{h,c}\big)\,, \tag{7}$$

$$u_{h,c} = \mathrm{sgn}(v_{h,c}) \cdot S^{-1}\big(|v_{h,c}|;\, \boldsymbol{\phi}_{h,c}\big)\,, \tag{8}$$

where $\boldsymbol{\phi}_{h,c} = (\mathbf{w}_b, \mathbf{h}_b, \mathbf{d}_b)$ are the per-element spline parameters defined in §3. Since the spline maps $[0, B]$ onto itself, $S(0) = 0$ holds by construction, and the convention $\mathrm{sgn}(0) = 1$ then ensures continuity at the origin; S^{-1} is available analytically since S is monotone. This focuses the entire capacity of S on the non-negative half-axis, with odd symmetry satisfied exactly by construction. The element-wise application yields a diagonal Jacobian whose forward-direction log-determinant is $\mathrm{ldj}_{\mathrm{r}} = \sum_{h,c} \log S'(|u_{h,c}|;\, \boldsymbol{\phi}_{h,c})$. Combining (7)–(8) and the log-determinant:

$$\mathbf{V},\, \mathrm{ldj}_{\mathrm{r}} = \mathrm{ROSS}(\mathbf{U};\, \boldsymbol{\Phi}). \tag{9}$$

SplineNet: Parameterizing ROSS. Recent works have adopted spline-based flows for probabilistic regression (Madhusudhanan et al. 2025) and forecasting (Yalavarthi et al. 2026), parameterizing $\boldsymbol{\phi}$ via MLPs. However, in the multivariate sequential setting, predicting $\boldsymbol{\Phi} \in \mathbb{R}_{>0}^{H \times C \times (3N_b - 1)}$ naively with an MLP incurs $O(H^2C^2)$ parameter cost, tractable for univariate targets but prohibitive for long-horizon forecasting. Since $\mathbf{F}$ already encodes long-range temporal structure from f_{θ_1}, local refinements over the time axis are sufficient to produce spline parameters at each time step. We further assume that residual distributions vary smoothly across time, making local convolution a reasonable architectural choice. We therefore use a lightweight 1D convolutional network operating over the time axis of $\mathbf{F}$:

$$\boldsymbol{\Phi} = \mathrm{Conv1D}(\mathrm{ReLU}(\mathrm{Conv1D}(\mathbf{F}))) \in \mathbb{R}^{H \times C \times (3N_b - 1)}, \tag{10}$$

where the first Conv1D maps $\mathbf{F}$ to a hidden representation via C_{hid} filters. $\boldsymbol{\Phi}$ is sliced along the last dimension to yield $\mathbf{w}_b$, $\mathbf{h}_b$, and $\mathbf{d}_b$, each passed through a softplus to ensure positivity. This reduces parameter cost to $O(k \cdot C_{\mathrm{hid}} \cdot N_b)$ for kernel size $k \ll H$, independent of the forecast horizon.

Linear Scaling Layer (LSL)

ROSS is defined over $[0, B]$, defaulting to the identity outside. Residuals $\boldsymbol{\varepsilon}$ are not guaranteed to lie within this range, reducing the effective capacity of the spline. We therefore prepend a conditional scaling layer that adapts the input range to the spline support:

$$v_{h,c} = u_{h,c} \cdot s_{h,c}, \qquad u_{h,c} = v_{h,c} / s_{h,c}, \tag{11}$$

where $s_{h,c} > 0$ is a positive scale conditioned on $\mathbf{X}$, predicted independently per channel by a network whose parameters are shared across channels (see ScaleNet below). Yalavarthi et al. (2025) use both scaling and translation as element-wise flow components; however, a translation term shifts the residual distribution, breaking odd symmetry and decoupling the predictive mean of $p(\mathbf{Y} \mid \mathbf{X})$ from $f_{\theta_1}(\mathbf{X})$. LSL therefore restricts to scaling only, which is an odd function and preserves $\mathbb{E}_{p_{\theta_2}}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$. Its forward-direction log-determinant is:

$$\mathbf{V},\, \mathrm{ldj}_{\mathrm{s}} = \mathrm{LSL}(\mathbf{U};\, \mathbf{S}), \qquad \mathrm{ldj}_{\mathrm{s}} = \textstyle\sum_{hc} \log s_{h,c}. \tag{12}$$

ScaleNet: Parameterizing LSL. The scale matrix $\mathbf{S} \in \mathbb{R}_{>0}^{H \times C}$ is predicted from $\mathbf{F}$ via a bounded MLP applied independently at each channel:

$$\mathbf{s}_{:,c} = \exp\left(a \cdot \tanh\left(\frac{\mathrm{MLP}(\mathbf{F}_{:,c};\, \boldsymbol{\omega})}{a}\right)\right), \qquad \mathbf{s}_{:,c} \in \mathbb{R}_{>0}^{H}, \tag{13}$$

where $\mathbf{F}_{:,c} \in \mathbb{R}^H$ is the context of channel c, and $a > 0$ constrains each $\log s_{h,c} \in [-a, a]$ keeping the scale bounded. The parameters $\boldsymbol{\omega}$ are shared across all C channels.

TORF Architecture

TORF composes LSL and ROSS into a K-block normalizing flow that transforms the residual $\boldsymbol{\varepsilon}$ into a standard Gaussian. Each block consists of a LSL layer followed by a ROSS layer, with a trailing LSL after the final block. The full architecture is illustrated in Alg. 1 and, in Figure 3 (Appendix D).

For better regularization and parameter efficiency, the LSL parameters $\mathbf{s}$ are shared across all $K + 1$ LSL layers, while

Model	Metric	ETTm1-L	ETTm2-L	ETTh1-L	ETTh2-L	Electricity-L	Traffic-L	Weather-L	Exchange-L	ILI-L
PatchTST	CRPS	0.304±0.029	0.229±0.036	0.323±0.020	0.304±0.018	0.127±0.015	0.214±0.001	0.142±0.005	0.097±0.007	0.233±0.019
	NMAE	0.382±0.066	0.288±0.034	0.428±0.024	0.371±0.021	0.164±0.024	0.253±0.012	0.152±0.029	0.126±0.001	0.287±0.023
iTrans.	CRPS	0.455±0.021	0.311±0.024	0.350±0.019	0.542±0.015	0.109±0.044	0.284±0.004	0.133±0.004	0.087±0.023	0.222±0.020
	NMAE	0.490±0.038	0.385±0.042	0.449±0.022	0.667±0.012	0.140±0.009	0.361±0.030	0.147±0.019	0.113±0.015	0.278±0.017
GRU NVP	CRPS	0.546±0.036	0.561±0.273	0.502±0.039	0.539±0.090	0.114±0.013	0.211±0.004	0.110±0.004	0.079±0.009	0.307±0.005
	NMAE	0.707±0.050	0.749±0.385	0.643±0.046	0.688±0.161	0.144±0.017	0.264±0.006	0.135±0.008	0.103±0.009	0.333±0.005
TimeGrad	CRPS	0.621±0.037	0.470±0.054	0.523±0.027	0.445±0.016	0.108±0.003	0.220±0.002	0.113±0.011	0.099±0.015	0.295±0.083
	NMAE	0.793±0.034	0.561±0.044	0.672±0.015	0.550±0.018	0.134±0.004	0.263±0.001	0.136±0.020	0.113±0.016	0.325±0.068
CSDI	CRPS	0.448±0.038	0.239±0.035	0.528±0.012	0.302±0.040	—	—	0.087±0.003	0.143±0.020	0.283±0.012
	NMAE	0.578±0.051	0.306±0.040	0.657±0.014	0.382±0.030	—	—	0.102±0.005	0.173±0.020	0.299±0.013
K^2VAE	CRPS	*0.294*±0.026	*0.221*±0.023	*0.314*±0.011	*0.280*±0.014	**0.057**±0.005	**0.200**±0.001	*0.084*±0.003	*0.069*±0.005	*0.142*±0.008
	NMAE	*0.373*±0.032	*0.275*±0.035	*0.396*±0.012	*0.278*±0.020	*0.117*±0.019	*0.248*±0.010	*0.099*±0.009	*0.084*±0.017	*0.167*±0.007
TORF	CRPS	**0.279**±0.001	**0.166**±0.000	**0.286**±0.001	**0.181**±0.000	*0.081*±0.000	*0.203*±0.001	**0.079**±0.001	**0.059**±0.000	**0.131**±0.000
	NMAE	**0.354**±0.003	**0.206**±0.001	**0.367**±0.001	**0.229**±0.020	**0.108**±0.003	**0.242**±0.004	**0.096**±0.001	**0.080**±0.000	**0.158**±0.007
Imp (%)	CRPS	+5.1	+24.9	+8.9	+35.4	−42.1	−1.5	+6.0	+14.5	+7.7
	NMAE	+5.1	+25.1	+7.3	+17.6	+7.7	+2.4	+3.0	+4.8	+5.4

Table 2: Comparison on long-term probabilistic forecasting (horizon 720) across nine real-world datasets. Lower CRPS/NMAE is better. Means and standard errors are from 5 independent runs. **Bold** = best, *italic* = second best. Full results for all four horizons $(96, 192, 336, 720)$ are in Appendix Tables 11 and 12. '-' indicates results required excessive time and memory consumption.

each block k employs its own spline network $\mathrm{SplineNet}^{(k)}$. Since both components (LSL, ROSS) are strictly odd functions, their composition is also odd, analytically enforcing $\mathbb{E}_{p_{\theta_2}}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$ throughout. When $K = 0$, no ROSS layer is applied and TORF reduces to a conditional Gaussian model, exactly recovering the MVE-2S baseline. An equivalence proof is presented in the Appendix G for reference. Increasing K progressively introduces non-linear flexibility through the ROSS layers.

The training loss is the NLL computed via the change-of-variables formula with the accumulated log-determinant of the forward map

$$\mathcal{L} = -\log \mathcal{N}\Big(\mathbf{U}^{(K+1)};\, \mathbf{0}, \mathbf{I}\Big) - \sum_{k=0}^{K-1} \Big(\mathrm{ldj}_{\mathrm{s}}^{(k)} + \mathrm{ldj}_{\mathrm{r}}^{(k)}\Big) - \mathrm{ldj}_{\mathrm{s}}^{(K)} \tag{14}$$

5 Experiments

Following the protocol and datasets from K^2VAE (Wu et al. 2025), we conduct experiments on 8 short-term and 9 long-term forecasting datasets. For short-term tasks, we use ETTh1-S, ETTh2-S, ETTm1-S, ETTm2-S, Solar-S, Electricity-S, Traffic-S and Exchange-S, with context lengths L and horizons H set to 24 (or 30 for Exchange). For long-term forecasting, we utilize the ETT series, Electricity-L, Traffic-L, Exchange-L, Weather-L, and ILI-L. The forecasting horizons H range from $\{96, 192, 336, 720\}$ in general except for ILI-L with H range from $\{24, 36, 48, 60\}$, while the context length is fixed at $L = 96$ ($L = 36$ for ILI) to ensure a fair comparison across all models. Dataset descriptions are provided in Appendix section E.

We compare TORF against 12 baselines, 4 point forecasters and 8 generative probabilistic models. The point forecasters, each fitted with an MVE head to produce a Gaussian predictive distribution, are FITS (Xu, Zeng, and Xu 2024), PatchTST (Nie et al. 2023), iTransformer (Liu et al. 2024), and Koopa (Liu et al. 2023). The generative baselines are TSDiff (Kollovieh et al. 2023), D3VAE (Li et al. 2022), GRU-NVP, GRU-MAF, and Trans-MAF (Rasul et al. 2021b), TimeGrad (Rasul et al. 2021a), CSDI (Tashiro et al. 2021), and K^2VAE (Wu et al. 2025). The main table reports the 6 strongest; full results for all 12 are in Appendix J.

TORF uses SimpleTM for Stage-1. We tune hyperparameters (Appendix N) with Optuna (Akiba et al. 2019) over 20 trials per configuration for both main and ablation results, and report mean and standard deviation over 5 Stage-2 runs with frozen Stage-1. Code is provided as supplementary material.

We evaluate probabilistic accuracy with CRPS (Continuous Ranked Probability Score) and point accuracy with NMAE (Normalized Mean Absolute Error). Since CRPS only captures univariate marginals, we also report Energy Score in the ablation study to assess multivariate prediction quality. Metric details are described in Appendix F.

Results

Table 2 and Table 3 report the full probabilistic forecasting results. In the long-term setting, TORF wins on CRPS for 7 of 9 datasets and on NMAE for all 9 datasets. The largest gains reach $+35.4\%$ on CRPS (ETTh2-L) and $+25.1\%$ on NMAE (ETTm2-L). In the short-term setting, TORF wins on CRPS for 6 of 8 datasets and on NMAE for 5 of 8 datasets. The largest gains reach $+30.3\%$ on CRPS (ETTm2-S) and $+28.1\%$ on NMAE (ETTm2-S). NMAE gains come from the stronger mean predictor, whereas CRPS gains come from the flexible odd flow conditioned on that mean.

The long-term wins are profound. This is likely because SimpleTM, TORF's Stage-1 model, was originally designed for long-term forecasting. We investigate the two settings where TORF fails to achieve the best results. On Traffic-L, the CRPS gap is close to a tie with a 1.5% difference from K^2VAE. This loss is plausibly within evaluation noise rather than a systematic weakness. The Electricity-L result reported for K^2VAE deserves closer scrutiny. K^2VAE reports a CRPS

Model	Metric	Exchange-S	Solar-S	Electricity-S	Traffic-S	ETTh1-S	ETTh2-S	ETTm1-S	ETTm2-S
PatchTST	CRPS	0.052±0.016	0.491±0.008	0.063±0.003	0.278±0.018	0.314±0.022	0.207±0.006	0.234±0.011	0.212±0.018
	NMAE	0.069±0.013	0.663±0.010	0.085±0.006	0.363±0.023	0.407±0.030	0.260±0.009	0.271±0.009	0.257±0.011
iTrans.	CRPS	0.059±0.018	0.504±0.012	0.066±0.004	0.244±0.011	0.317±0.020	0.219±0.008	0.254±0.012	0.201±0.018
	NMAE	0.081±0.022	0.695±0.017	0.087±0.006	0.319±0.019	0.408±0.028	0.276±0.017	0.291±0.017	0.242±0.009
GRU NVP	CRPS	0.019±0.006	0.530±0.008	0.062±0.003	0.168±0.008	0.398±0.034	0.309±0.023	0.455±0.029	0.276±0.014
	NMAE	0.024±0.007	0.670±0.011	0.081±0.006	0.209±0.013	0.477±0.040	0.375±0.024	0.584±0.047	0.349±0.028
TimeGrad	CRPS	*0.009*±0.001	0.465±0.016	0.057±0.002	0.130±0.005	0.273±0.007	0.184±0.006	0.186±0.003	0.148±0.004
	NMAE	*0.012*±0.002	0.609±0.015	0.073±0.004	**0.155**±0.007	0.356±0.013	0.224±0.014	0.246±0.007	0.189±0.006
CSDI	CRPS	*0.009*±0.001	*0.392*±0.006	**0.051**±0.001	0.147±0.014	0.262±0.012	0.133±0.006	0.140±0.012	0.144±0.018
	NMAE	0.013±0.001	*0.533*±0.007	**0.066**±0.001	0.175±0.013	0.339±0.009	0.161±0.013	0.169±0.021	0.181±0.024
K^2VAE	CRPS	*0.009*±0.001	**0.367**±0.006	*0.053*±0.002	*0.129*±0.004	*0.256*±0.008	*0.128*±0.006	*0.135*±0.008	*0.122*±0.008
	NMAE	**0.009**±0.001	**0.480**±0.008	*0.068*±0.002	*0.157*±0.007	*0.312*±0.008	*0.140*±0.007	*0.152*±0.007	*0.146*±0.009
TORF	CRPS	**0.008**±0.000	0.456±0.002	*0.053*±0.000	**0.126**±0.000	**0.231**±0.001	**0.106**±0.000	**0.108**±0.000	**0.085**±0.000
	NMAE	**0.009**±0.001	0.583±0.025	0.070±0.001	0.159±0.000	**0.286**±0.001	**0.132**±0.001	**0.139**±0.001	**0.105**±0.003
Imp (%)	CRPS	+11.1	−24.3	−3.9	+2.3	+9.8	+17.2	+20.0	+30.3
	NMAE	+0.0	−21.5	−6.1	−2.6	+8.3	+5.7	+8.6	+28.1

Table 3: Comparison on short-term probabilistic forecasting across eight real-world datasets. Lower CRPS/NMAE is better. Means and standard errors are from 5 independent runs. **Bold** = best, *Italic* = second best.

Variant	2-Stage	Odd	Spline	LSL	CNN-ROSS	Mixture
TORF-1S	✗	–	–	–	–	–
TORF	✓	✓	✓	✓	✓	✗
w/o LSL	✓	✓	✓	✗	✓	✗
w/o CNN-ROSS	✓	✓	✓	✓	✗	✗
w RealNVP	✓	✗	✗	✓	✓	✗
w OddRealNVP	✓	✓	✗	✓	✓	✗
w Spline	✓	✗	✓	✓	✓	✗
w Mixture	✓	✓	✓	✓	✓	✓

Table 4: Ablation variants relative to TORF.

	ETTm1	Exchange		Solar
Method	336	336	30	24
TORF-1S	0.5122	0.1390	0.1430	0.6078
TORF	**0.2457**	**0.0382**	**0.0077**	**0.4558**
w/o LSL	0.2508	0.0399	0.0081	0.4670
w/o CNN-ROSS	*0.2483*	*0.0383*	*0.0078*	**0.4558**

Table 5: Architectural Analysis in CRPS.

of 0.057 ± 0.005 at the 720-step horizon, but both, our rerun of the K^2VAE code and a concurrent work, report this value as approximately 0.087 (Wang et al. 2026b). This also contradicts K^2VAE's own per-horizon trend, where CRPS is lower at 720 than at 192 horizon. Corrected, TORF's Electricity-L CRPS of 0.081 ± 0.000 would surpass K^2VAE's.

The losses in the short-term setting concentrate on Solar-S and Electricity-S. On Solar-S, TORF trails K^2VAE by a reported 24.3% on CRPS and 21.5% on NMAE. On Electricity-S the gaps are smaller, at 3.9% and 6.1%. Rerunning K^2VAE on Solar-S ourselves gave a CRPS of 0.413 ± 0.017 and an NMAE of 0.531 ± 0.018, both only about 10% worse than TORF, not 24.3%. We attribute this remaining gap to the Stage-1 point forecaster rather than the flow. TORF's flow is provably mean-preserving, so its distributional accuracy is bounded by the frozen Stage-1 model, and on both datasets TORF's NMAE, inherited from SimpleTM, also lags K^2VAE's. Because the flow is agnostic to the choice of backbone, a stronger, dataset-appropriate Stage-1 model, such as a Koopman-style predictor, could close this gap while keeping TORF's flow for uncertainty quantification. With SimpleTM as its mean predictor, TORF is state-of-the-art on every dataset except Solar-S and Electricity-S, across both horizons.

Ablation Study

To isolate each design choice's contribution, we ablate multiple factors as summarized in Table 4, and include two Gaussian MVE baselines on the same SimpleTM backbone so that any gap is attributable to the decomposition, and not backbone strength.

In Table 4, *TORF-1S* removes the two-stage decomposition entirely. Among the two-stage variants, LSL is dropped in *w/o LSL*; *w/o CNN-ROSS* is SplineNet's CNN backbone, replaced by an MLP; Odd restricts the coupling transform to be odd (*TORF, w OddRealNVP*) versus unrestricted (*w Spline, w RealNVP*); Spline uses a rational-quadratic spline (*TORF, w Spline*) in place of an affine RealNVP-style map; and *w Mixture* extends TORF to a mixture of odd splines for asymmetric distribution modelling with analytical-mean. We also compare with two Gaussian MVE baselines: *MVE-1S*, trained jointly for μ and σ via NLL, and *MVE-2S*, a two-stage variant mirroring TORF that predicts $\hat{\mathbf{Y}}$ from a pre-trained Stage 1 and σ in Stage 2. Appendix H and Appendix I detail the construction of TORF w OddRealNVP and TORF w Mixture, respectively.

Architectural Analysis Table 5 shows that TORF-1S underperforms the two-stage variants. Removing LSL in TORF (w/o LSL) substantially weakens TORF across all horizons and datasets, as LSL allows TORF to learn a simple Gaussian distribution using only the LSL layer when needed. Adding an MLP in place of Conv1D (w/o CNN-ROSS) does not

		EnergyScore						CRPS					
		ETTm1		Exchange			Solar	ETTm1		Exchange			Solar
Type	Methods	96	336	96	336	30	24	96	336	96	336	30	24
	MVE-1S	0.6020	0.3832	0.0064	*0.0065*	*0.0047*	59.2827	0.2430	0.3301	0.0210	0.0399	0.0077	0.4774
	K^2VAE	0.4990	0.3079	0.0099	0.0082	0.0066	49.0566	0.2359	0.2698	0.0322	0.0519	0.0106	**0.4129**
2-stage	MVE-2S	0.4697	0.2997	0.0063	**0.0064**	0.0048	48.6663	0.2117	0.2491	0.0204	0.0391	**0.0074**	0.4705
	TORF	0.4651	*0.2933*	**0.0061**	**0.0064**	*0.0047*	48.2329	0.2088	0.2457	**0.0198**	**0.0382**	0.0077	*0.4558*
	w RealNVP	0.4674	0.2956	0.0063	0.0066	**0.0046**	48.4335	0.2101	0.2517	0.0210	0.0412	0.0077	0.4774
	w Spline	*0.4587*	0.2938	0.0065	*0.0065*	*0.0047*	48.2329	*0.2051*	0.2452	0.0213	0.0394	0.0078	*0.4558*
	w OddRealNVP	0.4656	0.2946	*0.0062*	**0.0064**	**0.0046**	*48.1577*	0.2086	*0.2450*	*0.0203*	0.0397	**0.0074**	0.4588
	w Mixture	**0.4559**	**0.2930**	*0.0062*	**0.0064**	0.0048	**48.0786**	**0.2038**	**0.2436**	0.0208	*0.0390*	*0.0076*	0.4653

Table 6: Energy Score (joint, multivariate density) and CRPS (univariate marginal density) results.

	ETTm1-336		Exchange-336	
Method	Param	Mem	Param	Mem
K^2VAE	1,593	0.018	1,611	0.019
TORF	22,183	0.512	722	0.106
w/o CNN-ROSS	OOM	OOM	66,881	0.354
w Mixture	65,006	1.168	2,116	0.213

Table 7: Trainable parameters (thousands) and peak memory (GB) across methods and datasets. OOM = out of memory.

		ETTm1	Exchange		Solar
Type	Methods	336	336	30	24
	MVE-1S	0.349	0.056	0.011	0.599
	K^2VAE	0.339	0.055	0.011	**0.531**
2-stage	MVE-2S	**0.314**	**0.051**	**0.010**	*0.583*
	Odd-TORF	**0.314**	**0.051**	**0.010**	*0.583*
	w RealNVP	*0.315*	*0.052*	**0.010**	0.630
	w Spline	0.319	*0.052*	**0.010**	0.629

Table 8: Point prediction results (NMAE, lower is better).

hurt accuracy and achieves the second-best performance, but comes with a real computational cost (Table 7). It runs out of memory even at horizon 336 on ETTm1, making it impractical for longer horizons or more channels, while the CNN-based SplineNet uses a fraction of the memory.

Probabilistic and Point-Forecast Analysis Extending the ablation to a shorter 96-horizon in Table 6, TORF and its variants are state-of-the-art on 11 of 12 Energy-Score/CRPS columns, losing only Solar-24 CRPS to K^2VAE, consistent with the known Stage-1 SimpleTM weakness on Solar. The two-stage decomposition drives most of this gain. MVE-2S, a plain two-stage Gaussian head, already beats K^2VAE on 11/12 columns on distributional metrics as well as on 3/4 on NMAE (Table 8). MVE-2S also beats the jointly-trained MVE-1S across the board on both distributional metrics, since joint NLL training fails to learn the best achievable mean. On Exchange specifically, MVE-1S alone already beats K^2VAE, plausibly since the residual is close to Gaussian and K^2VAE's extra complexity fails to exploit that simplicity. Beyond marginal accuracy, TORF also beats K^2VAE on the multivariate metric. TORF has better Energy Score in Solar-24 even though it loses on the univariate CRPS. Among design choices, spline outperforms the affine RealNVP transform, although the odd-restricted affine variant alone (w/OddRealNVP) still wins 3/12 columns. TORF-Mixture wins on more columns overall (6/12 vs. 4/12) but its edge is mostly insignificant, trades-off its analytical median and costs substantially more compute, so we recommend it only where the infrastructure allows, or if the residual is genuinely asymmetric. In Table 8, all odd-constrained variants (Odd-TORF) are provably mean-preserving and inherit an identical Stage-1 mean, whereas the non-odd w RealNVP and w Spline can change the mean, visibly degrading NMAE (up to 0.629-0.630 vs. 0.583 on Solar-24), exactly the failure mode the odd constraint rules out. TORF's Stage-1 model-agnosticism is further tested with iTransformer as a backbone in Appendix M.

6 Conclusion

We presented TORF, a two-stage probabilistic framework for TSF that pairs a frozen, model-agnostic point forecaster with a constrained odd flow whose predictive mean exactly matches the Stage 1 forecast, while a lightweight Conv1D SplineNet keeps it efficient enough to scale to long horizons. Across 17 real-world tasks, TORF achieves state-of-the-art results, with gains of up to $+35.4\%$ CRPS and $+28.1\%$ NMAE over the strongest baseline.

Limitations & Future Work. TORF assumes independent, symmetric residuals, which is sufficient here, but limiting when residuals are asymmetric or cross-dimensionally dependent. The Mixture extension relaxes symmetry at a higher parameter cost for only marginal gains, and joint time/channel dependence remains unaddressed. Future work includes component-dependent mixture weights, coupling- or MAF-based components, and mean-preserving diffusion or flow-matching variants.

References

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Bandara, K.; Shi, P.; Bergmeir, C.; Hewamalage, H.; Tran, Q.; and Seaman, B. 2019. Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *International conference on neural information processing*, 462–474. Springer.

Box, G. E. P.; Jenkins, G. M.; and Reinsel, G. C. 1994. *Time Series Analysis: Forecasting and Control*. Prentice Hall, 3rd edition.

Chen, H.; Luong, V.; Mukherjee, L.; and Singh, V. 2025. SimpleTM: A Simple Baseline for Multivariate Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.

Dimoulkas, I.; Herre, L.; Khastieva, D.; Nycander, E.; Amelin, M.; and Mazidi, P. 2018. A Hybrid Model Based on Symbolic Regression and Neural Networks for Electricity Load Forecasting. In *2018 15th International Conference on the European Energy Market (EEM)*, 1–5.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. In *International Conference on Learning Representations*.

Durkan, C.; Bekasov, A.; Murray, I.; and Papamakarios, G. 2019. Neural spline flows. *Advances in neural information processing systems*, 32.

Kobayashi, T.; and Aotani, T. 2023. Design of restricted normalizing flow towards arbitrary stochastic policy with computational efficiency. *Advanced Robotics*, 37: 719 – 736.

Kollovieh, M.; Ansari, A. F.; Bohlke-Schneider, M.; Zschiegner, J.; Wang, H.; and Wang, Y. B. 2023. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 36: 28341–28364.

Kollovieh, M.; Lienen, M.; Lüdke, D.; Schwinn, L.; and Günnemann, S. 2025. Flow Matching with Gaussian Process Priors for Probabilistic Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.

Li, Y.; Lu, X.; Wang, Y.; and Dou, D. 2022. Generative Time Series Forecasting with Diffusion, Denoise, and Disentanglement. In *Advances in Neural Information Processing Systems*, volume 35, 23009–23022.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2022. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*.

Liu, Y.; Li, C.; Wang, J.; and Long, M. 2023. Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors. In *Advances in Neural Information Processing Systems*, volume 36.

Luo, R.; Zhang, W.; Xu, X.; and Wang, J. 2018. A neural stochastic volatility model. In *proceedings of the AAAI conference on artificial intelligence*, volume 32.

Madhusudhanan, K.; Burchert, J.; Duong-Trung, N.; Born, S.; and Schmidt-Thieme, L. 2022. U-net inspired transformer architecture for far horizon time series forecasting. In *Joint European conference on machine learning and knowledge discovery in databases*, 36–52. Springer.

Madhusudhanan, K.; Yalavarthi, V. K.; Sonntag, J.; Stubbemann, M.; and Schmidt-Thieme, L. 2025. TabResFlow: A Normalizing Spline Flow Model for Probabilistic Univariate Tabular Regression. In *ECAI 2025 - 28th European Conference on Artificial Intelligence*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*.

Nix, D. A.; and Weigend, A. S. 1994. Estimating the mean and variance of the target probability distribution. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, 1: 55–60 vol.1.

Papamakarios, G.; Nalisnick, E.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57): 1–64.

Rasul, K.; Seward, C.; Schuster, I.; and Vollgraf, R. 2021a. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, 8857–8868. PMLR.

Rasul, K.; Sheikh, A.-S.; Schuster, I.; Bergmann, U. M.; and Vollgraf, R. 2021b. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *International Conference on Learning Representations*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191.

Seitzer, M.; Tavakoli, A.; Antic, D.; and Martius, G. 2022. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *International Conference on Learning Representations*.

Stirn, A.; Wessels, H.; Schertzer, M.; Pereira, L.; Sanjana, N. E.; and Knowles, D. A. 2023. Faithful Heteroscedastic Regression with Neural Networks. In Ruiz, F. J. R.; Dy, J. G.; and van de Meent, J., eds., *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, 5593–5613. PMLR.

Székely, G. J.; and Rizzo, M. L. 2013. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8): 1249 – 1272.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34: 24804–24816.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Wang, C.; Long, M.; and Wang, J. 2026a. Deep Time Series Models: A Comprehensive Survey and Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–20.

Wang, Y.; Zhuang, Q.; Marchanka, L.; and Wei, X.-S. 2026b. Beyond Static Uncertainty: Modeling Temporal Uncertainty Dynamics for Probabilistic Time Series Forecasting. *arXiv preprint arXiv:2603.24254*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, 22419–22430.

Wu, X.; Qiu, X.; Gao, H.; Hu, J.; Yang, B.; and Guo, C. 2025. K^2VAE: A Koopman-Kalman Enhanced Variational AutoEncoder for Probabilistic Time Series Forecasting. In *Forty-second International Conference on Machine Learning*.

Xu, Z.; Zeng, A.; and Xu, Q. 2024. FITS: Modeling Time Series with $10k$ Parameters. In *International Conference on Learning Representations*.

Yalavarthi, V. K.; Scholz, R.; Born, S.; and Schmidt-Thieme, L. 2025. Probabilistic Forecasting of Irregularly Sampled Time Series with Missing Values via Conditional Normalizing Flows. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20): 21877–21885.

Yalavarthi, V. K.; Scholz, R.; Klötergens, C.; Madhusudhanan, K.; Born, S.; and Schmidt-Thieme, L. 2026. Marginalization Consistent Mixture of Separable Flows for Probabilistic Irregular Time Series Forecasting. In *Fourteenth International Conference on Learning Representations*.

Zhang, J.; Wen, X.; Zhang, Z.; Zheng, S.; Li, J.; and Bian, J. 2024. ProbTS: Benchmarking Point and Distributional Forecasting across Diverse Prediction Horizons. *Advances in Neural Information Processing Systems 37*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.

A Additional Related Works

Deterministic forecasting has progressed from classical statistical models ARIMA (Box, Jenkins, and Reinsel 1994) to modern deep architectures like Transformers. These include Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), FEDformer (Zhou et al. 2022), Yformer (Madhusudhanan et al. 2022), and Pyraformer (Liu et al. 2022). iTransformer (Liu et al. 2024) and PatchTST (Nie et al. 2023) remain among the strongest performers. iTransformer attends across variables. PatchTST attends over channel-independent patches. Recently, SimpleTM (Chen et al. 2025) achieved state-of-the-art point accuracy. SimpleTM is a lightweight mixer-style predictor. It combines wavelet preprocessing with a geometric-product attention variant. Owing to its strong performance, we adopt SimpleTM as the Stage 1 model in our experiments. TORF is agnostic to this choice.

Probabilistic Time Series Forecasting Probabilistic forecasting aims to recover the full predictive distribution rather than a single point estimate. Autoregressive models such as DeepAR (Salinas et al. 2020) learn Gaussian parameters via an RNN. Diffusion models treat forecasting as an iterative denoising process. These include TimeGrad (Rasul et al. 2021a), CSDI (Tashiro et al. 2021), and TSDiff (Kollovieh et al. 2023). Conditional normalizing flows (Rasul et al. 2021b; Kollovieh et al. 2025) instead model complex joint densities without a fixed parametric form. K^2VAE (Wu et al. 2025) is a VAE based method developed for probabilistic LTSF. It combines a Koopman-linearized latent dynamical system with a KalmanNet-based correction. This correction fixes error accumulation over long horizons. Despite this diversity, all of the above methods either commit to a fixed distributional family or fall back on Monte Carlo sampling to recover a predictive mean. TORF instead decouples mean estimation from density estimation entirely. It obtains both an exact analytical mean and a flexible, sampling-free residual density. Table 1 compares popular density-estimation methods and highlights the need for a flexible density-modeling approach with an analytical mean.

Pitfalls of NLL Training and Two-Stage Fixes Nix and Weigend (1994) introduced the original Mean-Variance Estimation (MVE) formulation. It trains a single network to jointly predict a distribution's mean and variance under NLL. This joint training is known to trigger a mean-variance conflict. Seitzer et al. (2022) show that the NLL gradient is scaled by the inverse predicted variance. As a result, high-uncertainty regions contribute less to mean learning (Section 3). Several works address this problem through explicit decoupling. The most direct approach is Stirn et al. (2023), who formalize *faithfulness*. They freeze an independently trained mean network before fitting a residual density. This guarantees mean accuracy at least as good as the standalone predictor. TORF builds on this decoupling principle. We replace the frozen, Gaussian-only residual head with a flexible, sampling-free odd normalizing flow. This extends the faithfulness guarantee to arbitrary symmetric residual densities in the high-dimensional, long-horizon time series setting.

Odd Transformations for Analytically Tractable Moments Normalizing flows have been applied to flexible probabilistic prediction in irregularly-sampled and tabular domains (Yalavarthi et al. 2026; Madhusudhanan et al. 2025). They remain comparatively underexplored for LTSF. One reason is that their unconstrained transformations make the predictive mean accessible only through Monte Carlo sampling. Our restriction on the flow architecture is directly inspired by Kobayashi and Aotani (2023). In a reinforcement learning context, they design a Restricted Normalizing Flow (RNF) for stochastic robot control policies. They constrain the invertible transformation to be an odd function of a symmetric base variable. This makes the policy's mean action analytically computable. TORF repurposes this odd-function restriction for the probabilistic LTSF task. By construction, the residual flow's exact, sampling-free mean and median are zero. This preserves the Stage 1 point prediction exactly through Stage 2 density estimation.

B Theory

Lemma 1 (Analytical mean and median of a single odd flow)**.** *Let $\mathbf{u} \sim p_{\mathbf{u}}$ be a base random variable whose density is symmetric about the origin, i.e. $p_{\mathbf{u}}(\mathbf{u}) = p_{\mathbf{u}}(-\mathbf{u})$ for all $\mathbf{u} \in \mathbb{R}^K$, and let $f : \mathbb{R}^K \to \mathbb{R}^K$ be an odd, invertible transformation, $f(-\mathbf{u}) = -f(\mathbf{u})$ (Eq. (4)). Let $\mathbf{v} = f(\mathbf{u})$ with induced density $p_{\mathbf{v}}$. Then, provided $\mathbb{E}[\,\|\mathbf{v}\|\,] < \infty$,*

$$\mathbb{E}[\mathbf{v}] = \mathbf{0}, \text{ and } \operatorname{median}(v_j) = 0 \text{ for every coordinate } j.$$

Proof. **Symmetry of $p_{\mathbf{v}}$.** Since f is an odd bijection, f^{-1} is also odd: $f^{-1}(-\mathbf{v}) = -f^{-1}(\mathbf{v})$. Its Jacobian satisfies $\left|\det \partial_{\mathbf{v}} f^{-1}(-\mathbf{v})\right| = \left|\det \partial_{\mathbf{v}} f^{-1}(\mathbf{v})\right|$, because differentiating the odd map f^{-1} yields an even Jacobian. By the change-of-variables formula (Eq. (2)),

$$\begin{aligned} p_{\mathbf{v}}(-\mathbf{v}) &= p_{\mathbf{u}}\big(f^{-1}(-\mathbf{v})\big)\,\left|\det \partial_{\mathbf{v}} f^{-1}(-\mathbf{v})\right| \\ &= p_{\mathbf{u}}\big(-f^{-1}(\mathbf{v})\big)\,\left|\det \partial_{\mathbf{v}} f^{-1}(\mathbf{v})\right| \\ &= p_{\mathbf{u}}\big(f^{-1}(\mathbf{v})\big)\,\left|\det \partial_{\mathbf{v}} f^{-1}(\mathbf{v})\right| = p_{\mathbf{v}}(\mathbf{v}), \end{aligned}$$

using $p_{\mathbf{u}}(-\cdot) = p_{\mathbf{u}}(\cdot)$. Thus $p_{\mathbf{v}}$ is symmetric about the origin.

Zero mean. With $\mathbf{w} := -\mathbf{v}$,

$$\begin{aligned} \mathbb{E}[\mathbf{v}] &= \int \mathbf{v}\, p_{\mathbf{v}}(\mathbf{v})\, d\mathbf{v} = \int (-\mathbf{w})\, p_{\mathbf{v}}(-\mathbf{w})\, d\mathbf{w} \\ &= -\int \mathbf{w}\, p_{\mathbf{v}}(\mathbf{w})\, d\mathbf{w} = -\mathbb{E}[\mathbf{v}], \end{aligned}$$

so $\mathbb{E}[\mathbf{v}] = \mathbf{0}$; absolute integrability ensures the integral is well defined.

Zero median. For any coordinate j, symmetry $p_{\mathbf{v}}(\mathbf{v}) = p_{\mathbf{v}}(-\mathbf{v})$ implies $\Pr(v_j \le 0) = \Pr(v_j \ge 0)$. Since these probabilities sum to 1, each is $\frac{1}{2}$, so 0 is the median of v_j. □

Remark 1. *In TORF, both LSL (Eq. (11), scaling) and ROSS (Eq. (7), spline) are odd, so their composition is odd; applied to the base $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (symmetric about $\mathbf{0}$) in the generative*

direction, this gives $\mathbb{E}_{p_{\theta_2}}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$ *and coordinate-wise zero median, so* $f_{\theta_1}(\mathbf{X})$ *is preserved as both the analytical mean and median of* $\hat{p}(\mathbf{Y} \mid \mathbf{X})$.

Lemma 2 (Analytical mean of TORF-Mixture)**.** *Let each component* $i = 1, \ldots, M$ *have conditional mean* $\mathbb{E}[\varepsilon_{h,c}^{(i)} \mid \mathbf{F}] = \beta_{h,c}^{(i)}$*, with mixture weights* $\pi_{h,c}^{(i)} \geq 0$*,* $\sum_{i=1}^{M} \pi_{h,c}^{(i)} = 1$*. If*

$$\sum_{i=1}^{M} \pi_{h,c}^{(i)} \beta_{h,c}^{(i)} = 0, \tag{15}$$

then $\mathbb{E}[\varepsilon_{h,c} \mid \mathbf{F}] = 0$.

Proof. Each component has mean $\beta_{h,c}^{(i)}$ (odd transform of a symmetric base, shifted by $\beta_{h,c}^{(i)}$; Lemma 1). By the Law of total expectation rule,

$$\mathbb{E}[\varepsilon_{h,c} \mid \mathbf{F}] = \sum_{i=1}^{M} \pi_{h,c}^{(i)} \beta_{h,c}^{(i)} = 0, \tag{16}$$

□

Remark 2 (Mean, not median)**.** *Because the constraint fixes only the weighted average of the* $\beta_{h,c}^{(i)}$ *rather than each individually, the components are generally centered at distinct nonzero locations, so the mixture is asymmetric. Consequently the coordinate-wise median of* $\varepsilon_{h,c}$ *need not be* 0*: unlike the single-flow case (Lemma 1), TORF-Mixture guarantees mean preservation but not median preservation.*

C Synthetic Data Experiment

To evaluate the robustness of our model under non-Gaussian, heteroscedastic conditions, we define a synthetic data generating process where the observed variable y_i is a function of a latent signal $\mu(x_i)$ and input-dependent heavy-tailed noise $\epsilon(x_i)$:

$$y_i = \mu(x_i) + \epsilon(x_i), \quad x_i \in [0, 1] \tag{17}$$

The deterministic component $\mu(x)$ is a high-frequency sinusoidal signal:

$$\mu(x) = \sin(2\pi f x), \quad \text{with } f = 10.0 \tag{18}$$

The stochastic component $\epsilon(x)$ follows a Student's t-distribution characterized by input-dependent scale $\gamma(x)$ and degrees of freedom $\nu(x)$:

$$\epsilon(x) \sim \text{Student-t}(\nu(x), 0, \gamma(x)) \tag{19}$$

To simulate realistic "noise traps" where the uncertainty structure changes across the input space, we define $\gamma(x)$ and $\nu(x)$ using Gaussian kernels centered at $x = 0.5$:

- **Scale Parameter:** $\gamma(x) = 0.05 + 0.3 \exp\left(-\frac{(x-0.5)^2}{0.02}\right)$
- **Degrees of Freedom:** $\nu(x) = 3.0 + 7.0 \exp\left(-\frac{(x-0.5)^2}{0.02}\right)$

Under this formulation, the noise exhibits heavier tails (lower ν) away from the center, while the magnitude of the residuals (γ) peaks at the center. Importantly, since the Student's t-distribution is symmetric about zero, the conditional expectation remains $\mathbb{E}[y|x] = \mu(x)$, allowing us to strictly evaluate the decoupling performance of the TORF framework.

D TORF Architecture

Figure 3 shows the full TORF computation graph, forward (training, black) and inverse (sampling, red dashed) passes overlaid on the same nodes.

Training: forward pass. The frozen Stage-1 encoder $\text{enc}(\mathbf{X}; \theta_1)$ feeds two heads: the frozen Linear Head, which reproduces the Stage-1 point forecast $\hat{\mathbf{Y}}$, and the duplicated, trainable Linear Head′, which produces the context $\mathbf{F}$ that conditions every downstream network. The residual $\mathbf{U}^{(0)} = \mathbf{Y} - \hat{\mathbf{Y}}$ then enters the K-block loop (dashed box, $\times K$): $\mathbf{F}$ is mapped once, by the shared ScaleNet, into the scale $\mathbf{s}$ (Equation (13)) reused by every LSL in the loop, and once per block, by SplineNet$^{(k)}$, into the spline parameters $\mathbf{\Phi}^{(k)}$ (Equation (10)) that condition that block's ROSS. Each block applies $\text{LSL}(\cdot; \mathbf{s})$ to obtain $\mathbf{U}^{(k+\frac{1}{2})}$ and then $\text{ROSS}(\cdot; \mathbf{\Phi}^{(k)})$ to obtain $\mathbf{U}^{(k+1)}$, and a trailing LSL after block K produces $\mathbf{U}^{(K+1)}$, with the log-determinant ldj accumulated at every LSL and ROSS along the way (Algorithm 1). Training pushes $\mathbf{U}^{(K+1)}$ toward $\mathcal{N}(\mathbf{0}, \mathbf{I})$ through the loss $\mathcal{L} = -\log \mathcal{N}(\mathbf{U}^{(K+1)}; \mathbf{0}, \mathbf{I}) - \text{ldj}$ (Equation (14)); since gradients never reach the frozen path (snowflake nodes), $\hat{\mathbf{Y}}$ and the Stage-1 mean forecast it encodes are entirely untouched by this Stage-2 objective.

Inference: inverse pass. At inference, the same flow runs in reverse (red, dashed arrows) to produce the predictive distribution $\hat{p}(\mathbf{Y} \mid \mathbf{X})$ by sampling rather than by density evaluation, so no query point $\mathbf{Y}$ is needed. A draw $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is pushed back through the trailing LSL, then through each block's ROSS and LSL applied in their inverse direction, in reverse block order, re-using the same $\mathbf{s}$ and $\mathbf{\Phi}^{(k)}$ already computed from $\mathbf{F}$, until it reaches $\mathbf{U}^{(0)}$. Because every LSL and ROSS is odd and $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is symmetric, $\mathbf{U}^{(0)}$ is a draw from a symmetric, possibly multimodal residual density, centered exactly at the origin, rather than from any fixed parametric family. Adding back the frozen Stage-1 forecast, $\hat{\mathbf{Y}} + \mathbf{U}^{(0)}$, translates this density into the predictive distribution $\hat{p}(\mathbf{Y} \mid \mathbf{X})$, centered at $\hat{\mathbf{Y}}$ by construction: samples from TORF still average back to the exact Stage-1 mean (Lemma 1).

E Data Description

TORF is evaluated on the dataset collection[1] and split, assembled by K^2VAE (Wu et al. 2025), which in turn builds on ProbTS (Zhang et al. 2024), a general-purpose benchmark

[1]Datasets available at https://drive.google.com/drive/folders/1l0c4H57xYKKQQ5Tm7kd4C8M2nCepky-y

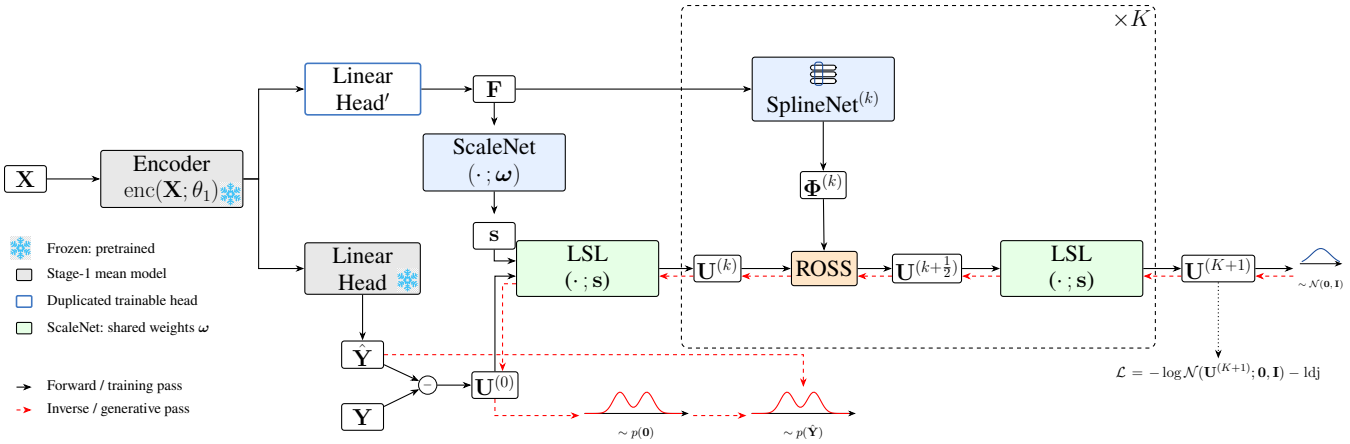


Figure 3: TORF architecture. Forward pass (black): the frozen encoder and head give $\hat{\mathbf{Y}}$; a trainable duplicate head gives context $\mathbf{F}$, which conditions the shared ScaleNet and per-block SplineNet$^{(k)}$. After K many layers TORF transforms the residual distribution into a standard normal distribution $\mathbf{U}^{(K+1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Both LSL and ROSS being odd preserves $\hat{\mathbf{Y}}$ as the analytical mean. Inverse pass (red, dashed): sampling back through the flow yields the multimodal residual $p(\mathbf{0})$, which shifts to $p(\hat{\mathbf{Y}})$ once $\hat{\mathbf{Y}}$ is added.

for probabilistic forecasters spanning short and long horizons. The short-horizon split contains eight series ETTh1-S, ETTh2-S, ETTm1-S, ETTm2-S, Solar-S, Electricity-S, Traffic-S, and Exchange-S for which the context window and the forecast horizon coincide: $L = H = 24$ for every series except Exchange-S, where $L = H = 30$. The long-horizon split spans nine series, the four ETT variants, Electricity-L, Traffic-L, Exchange-L, Weather-L, and ILI-L evaluated at $H \in \{96, 192, 336, 720\}$, with ILI-L instead swept over $H \in \{24, 36, 48, 60\}$; the context window is held fixed within each split at $L = 96$ ($L = 36$ for ILI-L) so that every model sees the same amount of history.

A dataset and its counterpart from the other split can share a name yet be different data: Electricity-S and Electricity-L, for instance, are separate extracts with their own channel counts and lengths rather than the same recording sliced at two horizons. Table 15 lists, for every series, the number of channels, the value range, the sampling frequency, the total number of recorded timesteps, and a brief description.

F Evaluation Metrics

Following the ProbTS protocol (Zhang et al. 2024), we report point-forecast accuracy with NMAE and marginal probabilistic accuracy with CRPS. We additionally report the Energy Score, which scores the predictive distribution jointly over the full $H \times C$ target grid, to check whether TORF's multivariate structure is well calibrated, unlike NMAE and CRPS, both of which score one (h, c) entry at a time.

- **Normalized Mean Absolute Error (NMAE)** measures the accuracy of the Stage-1 point forecast $\hat{\mathbf{Y}} = f_{\theta_1}(\mathbf{X})$ against the ground truth $\mathbf{Y}$, summed over the H horizon steps and C channels and normalized to be comparable across datasets of different scale:

$$\mathrm{NMAE} = \frac{\sum_{c=1}^{C} \sum_{h=1}^{H} \left| Y_{h,c} - \hat{Y}_{h,c} \right|}{\sum_{c=1}^{C} \sum_{h=1}^{H} \left| Y_{h,c} \right|}. \tag{20}$$

- **Continuous Ranked Probability Score (CRPS)** measures how well a single scalar predictive distribution matches a single scalar observation y, by comparing its predictive CDF F against y:

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} \big(F(z) - \mathbb{I}\{y \le z\} \big)^2 \, dz. \tag{21}$$

  F here is this scalar predictive CDF, unrelated to the context embedding $\mathbf{F}$ of Equation (6). We evaluate (21) at every entry $Y_{h,c}$ of the target, using F's own marginal of $\hat{p}(\mathbf{Y} \mid \mathbf{X})$ at that entry, and report the mean over (h, c).
- **Energy Score (ES)** extends CRPS to the full joint predictive distribution (Székely and Rizzo 2013): rather than scoring each (h, c) entry against its own marginal, it flattens the entire $H \times C$ target grid into one HC-dimensional vector $\mathbf{Y}$ and scores it jointly against samples drawn from $\hat{p}(\mathbf{Y} \mid \mathbf{X})$, rewarding a predictive distribution that captures dependence across time steps and channels together, not just per-entry accuracy:

$$\mathrm{ES}\big(\hat{p}(\mathbf{Y} \mid \mathbf{X}), \mathbf{Y}\big) = \mathbb{E}\big\|\mathbf{Y}^{(1)} - \mathbf{Y}\big\|_2 - \tfrac{1}{2}\,\mathbb{E}\big\|\mathbf{Y}^{(1)} - \mathbf{Y}^{(2)}\big\|_2, \tag{22}$$

  where $\| \cdot \|_2$ is the Euclidean norm over all HC flattened entries and $\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)} \overset{\text{i.i.d.}}{\sim} \hat{p}(\mathbf{Y} \mid \mathbf{X})$ are two independent samples from TORF's predictive distribution.

Both CRPS and the Energy Score are proper scoring rules: each is minimized exactly when the predictive distribution matches the true data-generating distribution, so a lower score reflects better calibration, not merely a narrower interval. Neither has a closed form for TORF's predictive distribution, so we estimate both from samples. Drawing M

i.i.d. samples $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(M)} \sim \hat{p}(\mathbf{Y} \mid \mathbf{X})$ per test window via the inverse flow, we evaluate (21) by replacing F with the empirical CDF $\hat{F}(z) = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\{Y^{(i)} \le z\}$ at each (h, c) entry, and (22) with the standard empirical estimator over the same M samples,

$$\widehat{\mathrm{ES}} = \frac{1}{M}\sum_{i=1}^{M} \left\|\mathbf{Y}^{(i)} - \mathbf{Y}\right\|_2 - \frac{1}{2M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \left\|\mathbf{Y}^{(i)} - \mathbf{Y}^{(j)}\right\|_2, \tag{23}$$

computed per test window and averaged over the evaluation set. We use $M = 100$ samples throughout. Because the raw (23) grows with the dimensionality HC of the flattened target, the ES values we report (Table 6) additionally divide this per-window average by the horizon length H. This does not fully normalize the score, since $\widehat{\mathrm{ES}}$ scales roughly with $\sqrt{HC}$ rather than H, but it removes the dominant horizon-driven growth and keeps values across the different horizons compared in that table on a roughly similar scale.

G Conditional Scaling Achieves an Exact Gaussian Transformation

This appendix proves the $K{=}0$ special case of the K-block construction (Section 4, Algorithm 1): with no ROSS blocks, the only transform applied is the trailing LSL, mapping the residual $\mathbf{U}^{(0)} = \boldsymbol{\varepsilon}$ directly to $\mathbf{U}^{(1)}$. We show this reduces TORF exactly to a heteroscedastic Gaussian model, recovering the MVE-2S baseline.

Recall from Equation (11) that LSL maps $u_{h,c} \mapsto v_{h,c} = u_{h,c}\, s_{h,c}$ (forward) and $v_{h,c} \mapsto u_{h,c} = v_{h,c}/s_{h,c}$ (inverse), with the scale $s_{h,c} > 0$ predicted by ScaleNet from the context $\mathbf{F}_{:,c}$ (Equation (13)). With $K{=}0$, $u_{h,c} = \varepsilon_{h,c}$ is the raw residual and $v_{h,c} = U^{(1)}_{h,c}$ is assumed standard normal, so the induced distribution of the residual is $\varepsilon_{h,c} = v_{h,c}/s_{h,c}$ with $v_{h,c} \sim \mathcal{N}(0,1)$, i.e. $\varepsilon_{h,c} \sim \mathcal{N}(0, \sigma^2_{h,c})$ with $\sigma_{h,c} := 1/s_{h,c}$ the implied per-entry standard deviation. The forward log-determinant is exactly Equation (12), restated here for reference,

$$\mathrm{ldj}_{\text{scale}} = \sum_{h,c} \log s_{h,c}, \tag{24}$$

since the map is diagonal and factorizes over all $(h, c) \in [H] \times [C]$ entries.

Equivalence to a direct Gaussian NLL. If the flow terminates here, i.e. $\mathbf{U}^{(1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ exactly, the resulting model is mathematically identical to directly parameterizing $\varepsilon_{h,c} \mid \mathbf{F}_{:,c} \sim \mathcal{N}(0, \sigma^2_{h,c})$ and minimizing its negative log-likelihood. Writing out the per-entry change-of-variables loss,

$$\begin{aligned}
\mathcal{L}_{\text{scale}} &= -\log \mathcal{N}(v_{h,c}; 0, 1) - \mathrm{ldj}_{\text{scale}} \\
&= \frac{1}{2}\log(2\pi) + \frac{1}{2}\left(u_{h,c}\, s_{h,c}\right)^2 - \log s_{h,c} \\
&= \frac{1}{2}\log\left(2\pi\, \sigma^2_{h,c}\right) + \frac{\varepsilon^2_{h,c}}{2\,\sigma^2_{h,c}},
\end{aligned} \tag{25}$$

which is exactly the negative log-density of $\mathcal{N}(0, \sigma^2_{h,c})$ evaluated at $\varepsilon_{h,c}$, which is the same objective realized by a direct second stage heteroscedastic Gaussian model (MVE-2S) trained with `GaussianNLLLoss` and a zero mean. This equivalence holds regardless of the specific link function used to map ScaleNet's raw output to $s_{h,c}$ (equivalently $\sigma_{h,c}$), as long as the same function is reproducible by both parameterizations; in our implementation we use the bounded, numerically stable form of Equation (13). **The practical implication is that conditional affine scaling, by itself, can only ever transform the residual into a distribution that is exactly Gaussian** (given a perfectly trained ScaleNet): it can stretch or compress the density along each axis as a function of $\mathbf{F}_{:,c}$, but it has no mechanism to alter the *shape* of the distribution beyond second-order (variance) effects. Any genuinely non-Gaussian structure in the residual like heavy tails, multi-modality, or sharply peaked unimodal mass near zero is invisible to a purely affine transform and is consequently absorbed into the likelihood as irreducible model misspecification.

H Odd RealNVP Construction

This appendix details the construction of *w OddRealNVP* variant of TORF: ROSS is replaced by a RealNVP-style affine coupling layer (Dinh, Sohl-Dickstein, and Bengio 2017), while LSL is left unchanged, so the ablation isolates the coupling transform alone.

Restoring oddness in a RealNVP coupling. At each time step h, the coupling layer masks the C channels with $\mathbf{m} \in \{0,1\}^C$ and applies $v_{h,c} = s_{h,c} \cdot u_{h,c} + t_{h,c}$ to the unmasked channels, with $s_{h,c}, t_{h,c}$ predicted from the masked channels $\mathbf{u}^{\mathbf{m}}_h := \mathbf{m} \odot \mathbf{U}_{h,:}$ and context $\mathbf{F}_{h,:}$ with the help of an MLP network r. Left unconstrained, this need not satisfy the odd condition. Substituting $\mathbf{U}_{h,:} \mapsto -\mathbf{U}_{h,:}$ shows the map is odd (with $\mathbf{F}$ fixed) precisely when $s_{h,c}$ is an *even* and $t_{h,c}$ an *odd* function of $\mathbf{u}^{\mathbf{m}}_h$:

$$s_{h,c}(-\mathbf{u}^{\mathbf{m}}_h) = s_{h,c}(\mathbf{u}^{\mathbf{m}}_h), \tag{26}$$

$$t_{h,c}(-\mathbf{u}^{\mathbf{m}}_h) = -t_{h,c}(\mathbf{u}^{\mathbf{m}}_h), \tag{27}$$

the same even/odd requirement Kobayashi and Aotani (2023) give for restricting a RealNVP coupling to be odd. Any raw, unconstrained network output $r(\cdot)$ is split into such a pair by evaluating it on both $+$ and $-$ its input:

$$s_{h,c} = \tfrac{1}{2}\left[r_s(\mathbf{u}^{\mathbf{m}}_h) + r_s(-\mathbf{u}^{\mathbf{m}}_h)\right], \tag{28}$$

$$t_{h,c} = \tfrac{1}{2}\left[r_t(\mathbf{u}^{\mathbf{m}}_h) - r_t(-\mathbf{u}^{\mathbf{m}}_h)\right], \tag{29}$$

exactly what the `EvenNetwork`/`OddNetwork` pair computes: each runs the same backbone once on $+\mathbf{u}^{\mathbf{m}}_h$ and once on $-\mathbf{u}^{\mathbf{m}}_h$, recombining with a $+$ (scale) or $-$ (translation) sign.

Listing 1: Odd-making step of OddRealNVP.

```
xc_pos = r(x)
xc_neg = r(-x)
t = 0.5 * (xc_pos - xc_neg)   # odd
s = 0.5 * (xc_pos + xc_neg)   # even
```

Model	Metric	ETTm1-L	ETTm2-L	ETTh1-L	ETTh2-L	Electricity-L	Traffic-L	Weather-L	Exchange-L	ILI-L
FITS	CRPS	0.305±0.024	0.449±0.034	0.348±0.025	0.314±0.022	0.115±0.024	0.374±0.004	0.267±0.003	0.074±0.011	0.211±0.011
	NMAE	0.406±0.072	0.540±0.052	0.468±0.012	0.401±0.022	0.149±0.012	0.453±0.022	0.317±0.021	0.097±0.011	0.245±0.017
PatchTST	CRPS	0.304±0.029	0.229±0.036	0.323±0.020	0.304±0.018	0.127±0.015	0.214±0.001	0.142±0.005	0.097±0.007	0.233±0.019
	NMAE	0.382±0.066	0.288±0.034	0.428±0.024	0.371±0.021	0.164±0.024	0.253±0.012	0.152±0.029	0.126±0.001	0.287±0.023
iTrans.	CRPS	0.455±0.021	0.311±0.024	0.350±0.019	0.542±0.015	0.109±0.044	0.284±0.004	0.133±0.004	0.087±0.023	0.222±0.020
	NMAE	0.490±0.038	0.385±0.042	0.449±0.022	0.667±0.012	0.140±0.009	0.361±0.030	0.147±0.019	0.113±0.015	0.278±0.017
Koopa	CRPS	0.295±0.027	0.233±0.025	0.318±0.009	0.293±0.026	0.113±0.018	0.358±0.022	0.140±0.007	0.091±0.012	0.228±0.022
	NMAE	0.377±0.037	0.290±0.033	0.412±0.008	0.286±0.042	0.149±0.025	0.432±0.032	0.162±0.009	0.116±0.022	0.288±0.031
TSDiff	CRPS	0.478±0.027	0.344±0.046	0.516±0.027	0.406±0.056	0.478±0.005	0.391±0.002	0.152±0.003	0.082±0.010	0.263±0.022
	NMAE	0.622±0.045	0.416±0.065	0.657±0.017	0.482±0.022	0.622±0.142	0.478±0.006	0.141±0.026	0.142±0.009	0.272±0.020
GRU NVP	CRPS	0.546±0.036	0.561±0.273	0.502±0.039	0.539±0.090	0.114±0.013	0.211±0.004	0.110±0.004	0.079±0.009	0.307±0.005
	NMAE	0.707±0.050	0.749±0.385	0.643±0.046	0.688±0.161	0.144±0.017	0.264±0.006	0.135±0.008	0.103±0.009	0.333±0.005
GRU MAF	CRPS	0.536±0.033	0.272±0.029	0.393±0.043	0.990±0.023	0.106±0.007	—	0.122±0.006	0.160±0.019	0.172±0.034
	NMAE	0.711±0.081	0.355±0.048	0.496±0.019	1.092±0.019	0.136±0.098	—	0.149±0.034	0.182±0.010	0.216±0.014
Trans MAF	CRPS	0.688±0.043	0.355±0.043	0.363±0.053	0.327±0.033	—	—	0.113±0.004	0.148±0.017	0.155±0.018
	NMAE	0.822±0.034	0.475±0.029	0.455±0.025	0.412±0.020	—	—	0.148±0.040	0.191±0.006	0.183±0.019
TimeGrad	CRPS	0.621±0.037	0.470±0.054	0.523±0.027	0.445±0.016	0.108±0.003	0.220±0.002	0.113±0.011	0.099±0.015	0.295±0.083
	NMAE	0.793±0.034	0.561±0.044	0.672±0.015	0.550±0.018	0.134±0.004	0.263±0.001	0.136±0.020	0.113±0.016	0.325±0.068
CSDI	CRPS	0.448±0.038	0.239±0.035	0.528±0.012	0.302±0.040	—	—	0.087±0.003	0.143±0.020	0.283±0.012
	NMAE	0.578±0.051	0.306±0.040	0.657±0.014	0.382±0.030	—	—	0.102±0.005	0.173±0.020	0.299±0.013
K^2VAE	CRPS	*0.294*±0.026	*0.221*±0.023	*0.314*±0.011	*0.280*±0.014	**0.057**±0.005	**0.200**±0.001	*0.084*±0.003	*0.069*±0.005	*0.142*±0.008
	NMAE	*0.373*±0.032	*0.275*±0.035	*0.396*±0.012	*0.278*±0.020	*0.117*±0.019	*0.248*±0.010	*0.099*±0.009	*0.084*±0.017	*0.167*±0.007
TORF	CRPS	**0.279**±0.001	**0.166**±0.000	**0.286**±0.001	**0.181**±0.000	*0.081*±0.000	*0.203*±0.001	**0.079**±0.001	**0.059**±0.000	**0.131**±0.000
	NMAE	**0.354**±0.003	**0.206**±0.001	**0.367**±0.001	**0.229**±0.020	**0.108**±0.003	**0.242**±0.004	**0.096**±0.001	**0.080**±0.000	**0.158**±0.007
Imp (%)	CRPS	+5.1	+24.9	+8.9	+35.4	−42.1	−1.5	+6.0	+14.5	+7.7
	NMAE	+5.1	+25.1	+7.3	+17.6	+7.7	+2.4	+3.0	+4.8	+5.4

Due to the excessive time and memory consumption, some results are unavailable and denoted as -.

Table 9: Comparison on long-term probabilistic forecasting (forecasting horizon L=720) scenarios across nine real-world datasets. Lower CRPS or NMAE values indicate better predictions. The means and standard errors are based on 5 independent runs of retraining and evaluation. **Bold**: the best, *underlined italic*: the 2nd best. The full results of all four horizons 96, 192, 336, 720 are listed in Section K.

I Mixture of Odd Flows Construction

Single-component TORF is restricted to *symmetric* residual densities (Section 4, Assumption 2), so it cannot represent skewed residuals. TORF-Mixture lifts this restriction by mixing M independently parameterized copies of the LSL→ROSS flow, while keeping the exact zero-mean guarantee of the single-component model.

Shifted components. Let $T^{(i)}(\cdot\,;\mathbf{F})$, $i = 1, \ldots, M$, be M independent LSL→ROSS flows, each with its own parameters and each individually odd. Instead of applying $T^{(i)}$ directly, component i first shifts $u_{h,c}$ by its own learned location $\beta^{(i)}_{h,c}(\mathbf{F})$:

$$v^{(i)}_{h,c},\ \ell^{(i)}_{h,c} = T^{(i)}\Big(u_{h,c} - \beta^{(i)}_{h,c}(\,\mathbf{F})\Big)\,, \tag{30}$$

with $\ell^{(i)}_{h,c}$ the log-determinant of $T^{(i)}$ at (h, c). Since $T^{(i)}$ is odd and $\mathcal{N}(0, 1)$ is symmetric, component i's density over $u_{h,c}$ is symmetric about $\beta^{(i)}_{h,c}$ rather than about 0.

Weights and the centering constraint. Two lightweight MLPs on $\mathbf{F}$, applied per time step and shared across channels as ScaleNet is (Section 4), output the raw weight logits and shifts $\beta^{(i)}_{h,c}$. Weights are a softmax floored at $\delta = \texttt{min\_weight}$ so no component's weight vanishes during training:

$$\pi^{(i)}_{h,c} = (1 - M\delta)\cdot \mathrm{softmax}_i(\cdot) + \delta, \qquad \textstyle\sum_{i=1}^{M} \pi^{(i)}_{h,c} = 1. \tag{31}$$

The raw shifts are then re-centered against these weights,

$$\beta^{(i)}_{h,c} \;\leftarrow\; \beta^{(i)}_{h,c} - \textstyle\sum_{j=1}^{M} \pi^{(j)}_{h,c}\,\beta^{(j)}_{h,c}, \tag{32}$$

which forces $\sum_{i=1}^{M} \pi^{(i)}_{h,c}\beta^{(i)}_{h,c} = 0$ exactly, for every (h, c) and $\mathbf{F}$, by construction rather than by training. The shift head is zero-initialized, so training starts from M coincident components equivalent to single-component TORF, and only splits them apart when doing so improves likelihood.

Each $T^{(i)}$ acts elementwise, the mixture factorizes exactly per residual entry and, under Conditional Independence, training minimizes $\mathcal{L} = -\sum_{h,c} \log p(u_{h,c} \mid \mathbf{F})$ as before. Sampling draws a component $z_{h,c} \sim \mathrm{Categorical}(\pi^{(1)}_{h,c}, \ldots, \pi^{(M)}_{h,c})$.

Mean preservation despite asymmetry. Component i has mean $\mathbb{E}[u_{h,c} \mid z_{h,c} = i, \mathbf{F}] = \beta^{(i)}_{h,c}$, so the mixture mean is

$$\mathbb{E}[u_{h,c} \mid \mathbf{F}] = \sum_{i=1}^{M} \pi^{(i)}_{h,c}\,\beta^{(i)}_{h,c} = 0 \tag{33}$$

by the centering constraint (32) alone – for any weights, any M, and any component shapes, so $\mathbb{E}_{p_{\theta_2}}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$ carries over from single-component TORF unchanged.

Model	Metric	Exchange-S	Solar-S	Electricity-S	Traffic-S	ETTh1-S	ETTh2-S	ETTm1-S	ETTm2-S
FITS	CRPS	$0.012_{\pm 0.002}$	$0.516_{\pm 0.011}$	$0.068_{\pm 0.003}$	$0.298_{\pm 0.022}$	$0.320_{\pm 0.017}$	$0.212_{\pm 0.012}$	$0.193_{\pm 0.005}$	$0.199_{\pm 0.003}$
	NMAE	$0.017_{\pm 0.003}$	$0.701_{\pm 0.014}$	$0.092_{\pm 0.004}$	$0.392_{\pm 0.028}$	$0.423_{\pm 0.033}$	$0.278_{\pm 0.009}$	$0.249_{\pm 0.007}$	$0.260_{\pm 0.011}$
PatchTST	CRPS	$0.052_{\pm 0.016}$	$0.491_{\pm 0.008}$	$0.063_{\pm 0.003}$	$0.278_{\pm 0.018}$	$0.314_{\pm 0.022}$	$0.207_{\pm 0.006}$	$0.234_{\pm 0.011}$	$0.212_{\pm 0.018}$
	NMAE	$0.069_{\pm 0.013}$	$0.663_{\pm 0.010}$	$0.085_{\pm 0.006}$	$0.363_{\pm 0.023}$	$0.407_{\pm 0.030}$	$0.260_{\pm 0.009}$	$0.271_{\pm 0.009}$	$0.257_{\pm 0.011}$
iTrans.	CRPS	$0.059_{\pm 0.018}$	$0.504_{\pm 0.012}$	$0.066_{\pm 0.004}$	$0.244_{\pm 0.011}$	$0.317_{\pm 0.020}$	$0.219_{\pm 0.008}$	$0.254_{\pm 0.012}$	$0.201_{\pm 0.018}$
	NMAE	$0.081_{\pm 0.022}$	$0.695_{\pm 0.017}$	$0.087_{\pm 0.006}$	$0.319_{\pm 0.019}$	$0.408_{\pm 0.028}$	$0.276_{\pm 0.017}$	$0.291_{\pm 0.017}$	$0.242_{\pm 0.009}$
Koopa	CRPS	$0.012_{\pm 0.001}$	$0.545_{\pm 0.016}$	$0.085_{\pm 0.014}$	$0.253_{\pm 0.018}$	$0.326_{\pm 0.013}$	$0.211_{\pm 0.019}$	$0.288_{\pm 0.022}$	$0.220_{\pm 0.015}$
	NMAE	$0.015_{\pm 0.002}$	$0.742_{\pm 0.022}$	$0.112_{\pm 0.019}$	$0.330_{\pm 0.019}$	$0.423_{\pm 0.017}$	$0.266_{\pm 0.022}$	$0.329_{\pm 0.026}$	$0.278_{\pm 0.022}$
TSDiff	CRPS	$0.077_{\pm 0.019}$	$0.568_{\pm 0.015}$	$0.111_{\pm 0.013}$	$0.189_{\pm 0.009}$	$0.304_{\pm 0.016}$	$0.204_{\pm 0.006}$	$0.209_{\pm 0.013}$	$0.124_{\pm 0.008}$
	NMAE	$0.096_{\pm 0.024}$	$0.635_{\pm 0.012}$	$0.115_{\pm 0.018}$	$0.206_{\pm 0.011}$	$0.400_{\pm 0.025}$	$0.272_{\pm 0.015}$	$0.276_{\pm 0.008}$	$0.162_{\pm 0.008}$
D^3VAE	CRPS	$0.011_{\pm 0.002}$	$0.769_{\pm 0.029}$	$0.071_{\pm 0.009}$	$0.143_{\pm 0.008}$	$0.324_{\pm 0.019}$	$0.216_{\pm 0.015}$	$0.198_{\pm 0.015}$	$0.303_{\pm 0.024}$
	NMAE	*0.012*$_{\pm 0.002}$	$0.998_{\pm 0.049}$	$0.092_{\pm 0.013}$	$0.178_{\pm 0.013}$	$0.410_{\pm 0.016}$	$0.267_{\pm 0.018}$	$0.250_{\pm 0.018}$	$0.378_{\pm 0.031}$
GRU NVP	CRPS	$0.019_{\pm 0.006}$	$0.530_{\pm 0.008}$	$0.062_{\pm 0.003}$	$0.168_{\pm 0.008}$	$0.398_{\pm 0.034}$	$0.309_{\pm 0.023}$	$0.455_{\pm 0.029}$	$0.276_{\pm 0.014}$
	NMAE	$0.024_{\pm 0.007}$	$0.670_{\pm 0.011}$	$0.081_{\pm 0.006}$	$0.209_{\pm 0.013}$	$0.477_{\pm 0.040}$	$0.375_{\pm 0.024}$	$0.584_{\pm 0.047}$	$0.349_{\pm 0.028}$
GRU MAF	CRPS	$0.012_{\pm 0.003}$	$0.486_{\pm 0.007}$	$0.056_{\pm 0.002}$	$0.144_{\pm 0.022}$	$0.258_{\pm 0.013}$	$0.160_{\pm 0.008}$	$0.151_{\pm 0.009}$	$0.146_{\pm 0.011}$
	NMAE	$0.016_{\pm 0.002}$	$0.603_{\pm 0.009}$	$0.073_{\pm 0.004}$	$0.182_{\pm 0.029}$	$0.326_{\pm 0.016}$	$0.208_{\pm 0.003}$	$0.198_{\pm 0.004}$	$0.193_{\pm 0.008}$
Trans MAF	CRPS	$0.012_{\pm 0.001}$	$0.442_{\pm 0.011}$	$0.054_{\pm 0.002}$	$0.133_{\pm 0.004}$	$0.309_{\pm 0.009}$	$0.200_{\pm 0.012}$	$0.139_{\pm 0.005}$	$0.180_{\pm 0.010}$
	NMAE	$0.016_{\pm 0.001}$	$0.577_{\pm 0.014}$	$0.071_{\pm 0.003}$	$0.160_{\pm 0.006}$	$0.400_{\pm 0.011}$	$0.256_{\pm 0.009}$	$0.162_{\pm 0.006}$	$0.224_{\pm 0.009}$
TimeGrad	CRPS	*0.009*$_{\pm 0.001}$	$0.465_{\pm 0.016}$	$0.057_{\pm 0.002}$	$0.130_{\pm 0.005}$	$0.273_{\pm 0.007}$	$0.184_{\pm 0.006}$	$0.186_{\pm 0.003}$	$0.148_{\pm 0.004}$
	NMAE	*0.012*$_{\pm 0.002}$	$0.609_{\pm 0.015}$	$0.073_{\pm 0.004}$	**0.155**$_{\pm 0.007}$	$0.356_{\pm 0.013}$	$0.224_{\pm 0.014}$	$0.246_{\pm 0.007}$	$0.189_{\pm 0.006}$
CSDI	CRPS	*0.009*$_{\pm 0.001}$	*0.392*$_{\pm 0.006}$	**0.051**$_{\pm 0.001}$	$0.147_{\pm 0.014}$	$0.262_{\pm 0.012}$	$0.133_{\pm 0.006}$	$0.140_{\pm 0.012}$	$0.144_{\pm 0.018}$
	NMAE	$0.013_{\pm 0.001}$	*0.533*$_{\pm 0.007}$	**0.066**$_{\pm 0.001}$	$0.175_{\pm 0.013}$	$0.339_{\pm 0.009}$	$0.161_{\pm 0.013}$	$0.169_{\pm 0.021}$	$0.181_{\pm 0.024}$
K^2VAE	CRPS	*0.009*$_{\pm 0.001}$	**0.367**$_{\pm 0.006}$	*0.053*$_{\pm 0.002}$	*0.129*$_{\pm 0.004}$	*0.256*$_{\pm 0.008}$	*0.128*$_{\pm 0.006}$	*0.135*$_{\pm 0.008}$	*0.122*$_{\pm 0.008}$
	NMAE	**0.009**$_{\pm 0.001}$	**0.480**$_{\pm 0.008}$	*0.068*$_{\pm 0.002}$	*0.157*$_{\pm 0.007}$	*0.312*$_{\pm 0.008}$	*0.140*$_{\pm 0.007}$	*0.152*$_{\pm 0.007}$	*0.146*$_{\pm 0.009}$
TORF	CRPS	**0.008**$_{\pm 0.000}$	$0.456_{\pm 0.002}$	*0.053*$_{\pm 0.000}$	**0.126**$_{\pm 0.000}$	**0.231**$_{\pm 0.001}$	**0.106**$_{\pm 0.000}$	**0.108**$_{\pm 0.000}$	**0.085**$_{\pm 0.000}$
	NMAE	**0.009**$_{\pm 0.001}$	$0.583_{\pm 0.025}$	$0.070_{\pm 0.001}$	$0.159_{\pm 0.000}$	**0.286**$_{\pm 0.001}$	**0.132**$_{\pm 0.001}$	**0.139**$_{\pm 0.001}$	**0.105**$_{\pm 0.003}$
Imp (%)	CRPS	$+11.1$	-24.3	-3.9	$+2.3$	$+9.8$	$+17.2$	$+20.0$	$+30.3$
	NMAE	$+0.0$	-21.5	-6.1	-2.6	$+8.3$	$+5.7$	$+8.6$	$+28.1$

Table 10: Comparison on short-term probabilistic forecasting scenarios across eight real-world datasets. Lower CRPS or NMAE values indicate better predictions. The means and standard errors are based on 5 independent runs of retraining and evaluation. **Bold**: the best, *underlined italic*: the 2nd best.

This same construction is also what lets the mixture be asymmetric. Components symmetric about a *common* point would keep the mixture symmetric about that point regardless of weights. Here each component is symmetric about its own $\beta_{h,c}^{(i)}$, and only their weighted *average* is pinned to zero, so the resulting mixture can be asymmetric, as in classical two-piece and split-normal distributions, here with M learned, flow-shaped components. **One consequence is that the guarantee is now on the *mean*, not the median once the mixture is genuinely asymmetric, its median need no longer coincide with $f_{\theta_1}(\mathbf{X})$.** Setting $M = 1$ collapses TORF-Mixture back to TORF exactly, with $\pi_{h,c}^{(1)} = 1$ and $\beta_{h,c}^{(1)} = 0$. TORF-Mixture is a strict generalization, not a separate model family.

J All Baselines - Main results

Table 9 and Table 10 report the full probabilistic forecasting results on all the 12 baselines. The best result in each column is shown in **bold** and the second-best is shown in *italics*. We highlight the following results below.

In the long-term setting, TORF wins on CRPS for 7 of 9 datasets and on NMAE for all 9 datasets. The largest gains reach $+35.4\%$ on CRPS (ETTh2-L) and $+25.1\%$ on NMAE (ETTm2-L). In the short-term setting, TORF wins on CRPS for 6 of 8 datasets and on NMAE for 5 of 8 datasets. The largest gains reach $+30.3\%$ on CRPS (ETTm2-S) and $+28.1\%$ on NMAE (ETTm2-S). This reflects the advantage of the two-stage design. TORF inherits an accurate, analytically exact mean from Stage 1. It then models a flexible residual distribution around that same mean, without resorting to expensive sampling.

Looking beyond K^2VAE, TORF also dominates every other baseline in Tables 9 and 10. In the long-term setting, TORF beats every non-K^2VAE baseline on both CRPS and NMAE for all 9 datasets, a clean 9/9 sweep. No single alternative is a consistent runner-up across datasets, either: Koopa is the closest competitor on three of the four ETT variants (ETTm1-L, ETTh1-L, ETTh2-L), PatchTST on ETTm2-L, GRU MAF on Electricity-L, GRU NVP on Traffic-L etc. Each general-purpose flow, diffusion, or point-forecaster-turned-probabilistic baseline is only competitive on a narrow subset of dataset types, whereas TORF is uniformly strong across all of them.

In the short-term setting, against whichever baseline is actually strongest in each column (not always K^2VAE), TORF still wins 6 of 8 columns on CRPS and 5 of 8 on NMAE, losing only on Solar-S (CRPS and NMAE), Electricity-S (CRPS and NMAE), and Traffic-S NMAE alone by a 2.6% margin (TORF still wins Traffic-S CRPS by 2.3%). Notably,

Dataset	Horizon	Koopa	iTransformer	FITS	PatchTST	GRU MAF	Trans MAF	TSDiff	CSDI	TimeGrad	GRU NVP	K^2VAE	TORF
ETTm1-L	96	0.285±0.018	0.301±0.033	0.267±0.023	0.261±0.051	0.295±0.055	0.313±0.045	0.344±0.050	0.236±0.006	0.522±0.105	0.383±0.053	*0.232*±0.010	**0.2088**±0.0011
	192	0.289±0.024	0.314±0.023	0.261±0.022	0.275±0.030	0.389±0.033	0.424±0.029	0.345±0.035	0.291±0.025	0.603±0.092	0.396±0.030	*0.259*±0.013	**0.2310**±0.0003
	336	0.286±0.035	0.311±0.029	0.275±0.030	0.285±0.028	0.429±0.021	0.481±0.019	0.462±0.043	0.322±0.033	0.601±0.028	0.486±0.032	*0.262*±0.030	**0.2457**±0.0005
	720	0.295±0.027	0.455±0.021	0.305±0.024	0.304±0.029	0.536±0.033	0.688±0.043	0.478±0.027	0.448±0.038	0.621±0.037	0.546±0.036	*0.294*±0.026	**0.2796**±0.0007
ETTm2-L	96	0.178±0.023	0.181±0.031	0.162±0.053	0.142±0.034	0.177±0.024	0.227±0.013	0.175±0.019	*0.115*±0.009	0.427±0.042	0.319±0.044	0.126±0.007	**0.1091**±0.0001
	192	0.185±0.014	0.190±0.010	0.185±0.053	0.172±0.023	0.411±0.026	0.253±0.037	0.255±0.029	*0.147*±0.008	0.424±0.061	0.326±0.025	0.148±0.009	**0.1279**±0.0001
	336	0.198±0.015	0.206±0.055	0.218±0.053	0.195±0.042	0.377±0.023	0.253±0.013	0.328±0.047	0.190±0.018	0.469±0.049	0.449±0.145	*0.164*±0.010	**0.1441**±0.0001
	720	0.233±0.025	0.311±0.024	0.449±0.034	0.229±0.036	0.272±0.029	0.355±0.043	0.344±0.046	0.239±0.035	0.470±0.054	0.561±0.273	*0.221*±0.023	**0.1656**±0.0002
ETTh1-L	96	0.307±0.033	0.292±0.032	0.294±0.023	0.312±0.036	0.293±0.037	0.333±0.045	0.395±0.052	0.437±0.018	0.455±0.046	0.379±0.030	*0.264*±0.020	**0.2426**±0.0005
	192	0.301±0.014	0.298±0.020	0.304±0.028	0.313±0.034	0.348±0.075	0.351±0.063	0.467±0.044	0.496±0.051	0.516±0.038	0.425±0.019	*0.290*±0.016	**0.2606**±0.0006
	336	0.312±0.019	0.327±0.043	0.318±0.023	0.319±0.035	0.377±0.026	0.371±0.031	0.450±0.027	0.454±0.025	0.512±0.026	0.458±0.054	*0.308*±0.021	**0.2752**±0.0010
	720	0.318±0.009	0.350±0.019	0.348±0.025	0.323±0.020	0.393±0.043	0.363±0.053	0.516±0.027	0.528±0.012	0.523±0.027	0.502±0.039	*0.314*±0.011	**0.2860**±0.0006
ETTh2-L	96	0.199±0.012	0.185±0.013	0.187±0.011	0.197±0.021	0.239±0.019	0.263±0.020	0.336±0.021	0.164±0.013	0.358±0.026	0.432±0.141	*0.162*±0.009	**0.1377**±0.0004
	192	0.198±0.022	0.199±0.019	0.195±0.022	0.204±0.055	0.313±0.034	0.273±0.024	0.265±0.043	0.226±0.018	0.457±0.081	0.625±0.170	*0.186*±0.018	**0.1606**±0.0001
	336	0.262±0.019	0.271±0.033	*0.246*±0.044	0.277±0.054	0.376±0.034	0.265±0.042	0.350±0.031	0.274±0.022	0.481±0.078	0.793±0.319	0.257±0.023	**0.1766**±0.0007
	720	0.293±0.026	0.542±0.015	0.314±0.022	0.304±0.018	0.990±0.023	0.327±0.033	0.406±0.056	0.302±0.040	0.445±0.016	0.539±0.090	*0.280*±0.014	**0.1809**±0.0002
Electricity-L	96	0.110±0.004	0.102±0.004	0.105±0.006	0.126±0.005	0.083±0.009	0.088±0.014	0.344±0.006	0.153±0.137	0.096±0.002	0.094±0.003	*0.073*±0.002	**0.0619**±0.0001
	192	0.109±0.011	0.104±0.014	0.112±0.104	0.123±0.032	0.093±0.024	0.097±0.009	0.345±0.006	0.200±0.094	0.100±0.004	0.097±0.002	*0.080*±0.004	**0.0687**±0.0001
	336	0.121±0.011	0.104±0.010	0.111±0.014	0.131±0.024	0.095±0.001	-	0.462±0.054	-	0.102±0.007	0.099±0.001	**0.054**±0.001	*0.0742*±0.0001
	720	0.113±0.018	0.109±0.044	0.115±0.024	0.127±0.015	0.106±0.007	-	0.478±0.005	-	0.108±0.003	0.114±0.013	**0.057**±0.005	*0.0807*±0.0001
Traffic-L	96	0.297±0.019	0.256±0.004	0.258±0.004	0.194±0.002	0.215±0.003	0.208±0.004	0.294±0.003	-	0.202±0.004	*0.187*±0.002	**0.086**±0.001	0.1925±0.0002
	192	0.308±0.009	0.250±0.002	0.275±0.003	0.198±0.004	-	-	0.306±0.004	-	0.208±0.003	*0.192*±0.001	**0.088**±0.002	0.1961±0.0003
	336	0.334±0.017	0.261±0.001	0.327±0.001	0.204±0.002	-	-	0.317±0.006	-	0.213±0.003	0.201±0.004	*0.195*±0.003	**0.1936**±0.0003
	720	0.358±0.022	0.284±0.004	0.374±0.004	0.214±0.001	-	-	0.391±0.002	-	0.220±0.002	0.211±0.004	**0.200**±0.001	*0.2027*±0.0001
Weather-L	96	0.132±0.008	0.131±0.011	0.210±0.013	0.131±0.007	0.139±0.008	0.105±0.011	0.104±0.020	**0.068**±0.008	0.130±0.017	0.116±0.013	0.080±0.007	*0.0693*±0.0010
	192	0.133±0.017	0.132±0.018	0.205±0.019	0.131±0.014	0.143±0.020	0.142±0.022	0.134±0.012	**0.068**±0.006	0.127±0.019	0.122±0.021	0.079±0.009	*0.0752*±0.0012
	336	0.136±0.021	0.132±0.010	0.221±0.005	0.137±0.008	0.129±0.012	0.133±0.014	0.137±0.010	0.083±0.002	0.130±0.006	0.128±0.011	*0.082*±0.010	**0.0781**±0.0017
	720	0.140±0.007	0.133±0.004	0.267±0.003	0.142±0.005	0.122±0.006	0.113±0.004	0.152±0.003	0.087±0.003	0.113±0.011	0.110±0.004	*0.084*±0.003	**0.0792**±0.0007
Exchange-L	96	0.063±0.006	0.061±0.003	0.048±0.004	0.063±0.006	0.026±0.010	0.028±0.002	0.079±0.007	*0.028*±0.003	0.068±0.003	0.071±0.006	0.031±0.002	**0.0198**±0.0001
	192	0.065±0.020	0.062±0.010	0.049±0.011	0.067±0.008	0.034±0.009	0.046±0.017	0.093±0.011	0.045±0.003	0.087±0.013	0.068±0.004	*0.032*±0.010	**0.0265**±0.0001
	336	0.072±0.008	0.067±0.008	0.052±0.013	0.071±0.017	0.058±0.023	*0.045*±0.010	0.081±0.007	0.060±0.004	0.074±0.009	0.072±0.002	0.048±0.004	**0.0382**±0.0001
	720	0.091±0.012	0.087±0.023	0.074±0.011	0.097±0.007	0.160±0.019	0.148±0.017	0.082±0.010	0.143±0.020	0.099±0.015	0.079±0.009	*0.069*±0.005	**0.0591**±0.0002
ILI-L	24	0.245±0.018	0.212±0.013	0.233±0.015	0.312±0.014	0.097±0.010	0.092±0.019	0.228±0.024	0.250±0.013	0.275±0.047	0.257±0.003	*0.087*±0.003	**0.0813**±0.0007
	36	0.214±0.008	0.182±0.016	0.217±0.023	0.241±0.021	0.117±0.017	0.115±0.011	0.235±0.010	0.285±0.010	0.272±0.057	0.281±0.004	*0.113*±0.005	**0.1122**±0.0009
	48	0.271±0.021	0.213±0.012	0.185±0.026	0.242±0.018	0.128±0.019	0.133±0.022	0.265±0.039	0.285±0.036	0.295±0.033	0.288±0.008	*0.124*±0.010	**0.1073**±0.0008
	60	0.228±0.022	0.222±0.020	0.211±0.011	0.233±0.019	0.172±0.034	0.155±0.018	0.263±0.022	0.283±0.012	0.295±0.083	0.307±0.005	*0.142*±0.008	**0.1313**±0.0003

Due to the excessive time and memory consumption, some results are unavailable in our implementation and denoted as -.

Table 11: Results of CRPS (mean_{std}) on long-term forecasting scenarios, each containing five independent runs with different seeds. The context length is set to 36 for the ILI-L dataset and 96 for the others. Lower CRPS values indicate better predictions. The means and standard errors are based on 5 independent runs of retraining and evaluation. **Bold**: the best, *italics*: the 2nd best.

Dataset	Horizon	Koopa	iTransformer	FITS	PatchTST	GRU MAF	Trans MAF	TSDiff	CSDI	TimeGrad	GRU NVP	K^2VAE	TORF (SimpleTM)
ETTm1-L	96	0.362±0.022	0.369±0.029	0.349±0.032	0.329±0.100	0.402±0.087	0.456±0.042	0.441±0.021	0.308±0.005	0.645±0.129	0.488±0.058	*0.284*±0.011	**0.2630**±0.0010
	192	0.365±0.032	0.384±0.041	0.341±0.032	0.338±0.022	0.476±0.046	0.553±0.012	0.441±0.019	0.377±0.026	0.748±0.084	0.514±0.042	*0.323*±0.020	**0.2950**±0.0020
	336	0.364±0.026	0.380±0.020	0.356±0.022	0.344±0.013	0.522±0.019	0.590±0.047	0.571±0.033	0.419±0.042	0.759±0.015	0.630±0.029	*0.330*±0.014	**0.3140**±0.0020
	720	0.377±0.037	0.490±0.038	0.406±0.072	0.382±0.066	0.711±0.081	0.822±0.034	0.622±0.045	0.578±0.051	0.793±0.034	0.707±0.050	*0.373*±0.032	**0.3540**±0.0040
ETTm2-L	96	0.225±0.039	0.221±0.039	0.210±0.040	0.216±0.035	0.212±0.082	0.279±0.031	0.224±0.033	0.146±0.012	0.525±0.047	0.413±0.059	*0.144*±0.011	**0.1350**±0.0010
	192	0.233±0.026	0.229±0.031	0.234±0.038	0.215±0.022	0.535±0.029	0.292±0.041	0.316±0.040	0.189±0.012	0.530±0.060	0.427±0.033	*0.170*±0.009	**0.1600**±0.0010
	336	0.267±0.023	0.245±0.049	0.276±0.019	0.234±0.024	0.407±0.043	0.309±0.032	0.397±0.051	0.248±0.024	0.566±0.047	0.580±0.169	*0.187*±0.021	**0.1800**±0.0010
	720	0.290±0.033	0.385±0.042	0.540±0.052	0.288±0.034	0.355±0.048	0.475±0.029	0.416±0.065	0.306±0.040	0.561±0.044	0.749±0.385	*0.275*±0.035	**0.2060**±0.0010
ETTh1-L	96	0.407±0.052	0.386±0.092	0.393±0.142	0.407±0.022	0.371±0.034	0.423±0.047	0.510±0.029	0.557±0.022	0.585±0.058	0.481±0.037	*0.336*±0.041	**0.3150**±0.0020
	192	0.396±0.022	0.388±0.041	0.406±0.079	0.405±0.088	0.430±0.022	0.451±0.012	0.596±0.056	0.625±0.065	0.680±0.058	0.531±0.018	*0.372*±0.023	**0.3420**±0.0020
	336	0.406±0.028	0.415±0.022	0.410±0.063	0.412±0.024	0.462±0.049	0.481±0.041	0.581±0.035	0.574±0.026	0.666±0.047	0.580±0.064	*0.394*±0.022	**0.3610**±0.0020
	720	0.412±0.008	0.449±0.022	0.468±0.012	0.428±0.024	0.496±0.019	0.455±0.025	0.657±0.017	0.657±0.014	0.672±0.015	0.643±0.046	*0.396*±0.012	**0.3670**±0.0020
ETTh2-L	96	0.249±0.015	0.234±0.011	0.243±0.009	0.247±0.028	0.292±0.012	0.345±0.042	0.421±0.033	0.214±0.018	0.448±0.031	0.548±0.158	*0.189*±0.010	**0.1750**±0.0040
	192	0.249±0.032	0.247±0.040	0.252±0.022	0.265±0.091	0.376±0.112	0.343±0.044	0.339±0.033	0.294±0.027	0.575±0.089	0.766±0.223	*0.213*±0.021	**0.2030**±0.0010
	336	0.274±0.027	0.297±0.029	0.291±0.032	0.314±0.045	0.454±0.057	0.333±0.078	0.427±0.041	0.353±0.028	0.606±0.095	0.942±0.408	*0.263*±0.039	**0.2250**±0.0010
	720	0.286±0.042	0.667±0.012	0.401±0.022	0.371±0.021	1.092±0.019	0.412±0.020	0.482±0.022	0.382±0.030	0.550±0.018	0.688±0.161	*0.278*±0.020	**0.2290**±0.0030
Electricity-L	96	0.146±0.015	0.134±0.002	0.137±0.002	0.168±0.012	0.108±0.009	0.114±0.010	0.441±0.013	0.203±0.189	0.119±0.003	0.118±0.003	*0.093*±0.002	**0.0794**±0.0005
	192	0.143±0.023	0.137±0.022	0.143±0.112	0.163±0.032	0.120±0.033	0.131±0.008	0.441±0.005	0.264±0.129	0.124±0.005	0.121±0.003	*0.102*±0.010	**0.0896**±0.0011
	336	0.151±0.017	0.136±0.002	0.139±0.002	0.168±0.010	0.122±0.018	-	0.571±0.022	-	0.126±0.008	0.123±0.001	*0.107*±0.002	**0.0966**±0.0007
	720	0.149±0.025	0.140±0.009	0.149±0.012	0.164±0.024	0.136±0.098	-	0.622±0.142	-	0.134±0.004	0.144±0.017	*0.117*±0.019	**0.1085**±0.0032
Traffic-L	96	0.377±0.024	0.332±0.008	0.332±0.007	**0.228**±0.010	0.274±0.012	0.265±0.007	0.342±0.042	-	0.234±0.006	0.231±0.003	*0.230*±0.010	0.2450±0.0010
	192	0.388±0.011	0.326±0.009	0.350±0.010	**0.225**±0.012	-	-	0.354±0.012	-	0.239±0.004	0.236±0.002	*0.234*±0.003	0.2470±0.0020
	336	0.416±0.028	0.335±0.010	0.405±0.011	*0.242*±0.022	-	-	0.392±0.006	-	0.246±0.003	0.248±0.006	**0.242**±0.007	0.2450±0.0030
	720	0.432±0.032	0.361±0.030	0.453±0.022	0.253±0.012	-	-	0.478±0.006	-	0.263±0.001	0.264±0.006	**0.248**±0.010	*0.2520*±0.0020
Weather-L	96	0.146±0.019	0.144±0.017	0.279±0.027	0.145±0.016	0.176±0.011	0.139±0.010	0.113±0.022	0.087±0.012	0.164±0.023	0.145±0.017	*0.086*±0.011	**0.0850**±0.0010
	192	0.148±0.022	0.145±0.015	0.264±0.013	0.144±0.012	0.166±0.022	0.160±0.037	0.144±0.020	*0.086*±0.007	0.158±0.024	0.147±0.025	**0.083**±0.011	0.0880±0.0010
	336	0.152±0.032	0.146±0.011	0.283±0.021	0.149±0.023	0.168±0.014	0.170±0.027	0.138±0.033	0.098±0.002	0.162±0.006	0.160±0.012	*0.093*±0.010	**0.0910**±0.0010
	720	0.162±0.009	0.147±0.019	0.317±0.021	0.152±0.029	0.149±0.034	0.148±0.040	0.141±0.026	0.102±0.005	0.136±0.020	0.135±0.008	*0.099*±0.009	**0.0960**±0.0010
Exchange-L	96	0.079±0.005	0.077±0.001	0.069±0.007	0.079±0.002	0.033±0.003	0.036±0.009	0.090±0.010	0.036±0.005	0.079±0.002	0.091±0.009	*0.032*±0.002	**0.0250**±0.0010
	192	0.081±0.015	0.078±0.008	0.069±0.007	0.081±0.002	0.044±0.004	0.058±0.007	0.106±0.010	0.058±0.005	0.100±0.019	0.087±0.005	*0.040*±0.005	**0.0360**±0.0020
	336	0.086±0.003	0.083±0.005	0.071±0.005	0.085±0.010	0.074±0.017	0.058±0.009	0.106±0.010	0.076±0.006	0.086±0.008	0.091±0.002	*0.054*±0.001	**0.0520**±0.0010
	720	0.116±0.022	0.113±0.015	0.097±0.011	0.126±0.001	0.182±0.010	0.191±0.006	0.142±0.009	0.173±0.020	0.113±0.016	0.103±0.009	*0.084*±0.017	**0.0800**±0.0000
ILI-L	24	0.303±0.021	0.265±0.027	0.271±0.032	0.382±0.018	0.124±0.019	0.118±0.033	0.242±0.086	0.263±0.012	0.296±0.044	0.283±0.001	*0.116*±0.011	**0.1119**±0.0094
	36	0.262±0.013	0.222±0.047	0.258±0.058	0.286±0.037	0.144±0.011	0.143±0.089	0.246±0.117	0.298±0.011	0.298±0.048	0.307±0.007	*0.142*±0.008	**0.1140**±0.0104
	48	0.334±0.028	0.262±0.023	0.225±0.043	0.291±0.032	0.159±0.020	0.160±0.039	0.275±0.044	0.301±0.034	0.320±0.025	0.314±0.009	*0.152*±0.017	**0.1457**±0.0078
	60	0.288±0.031	0.278±0.017	0.245±0.017	0.287±0.023	0.216±0.014	0.183±0.019	0.272±0.020	0.299±0.013	0.325±0.068	0.333±0.005	*0.167*±0.007	**0.1584**±0.0071

Due to the excessive time and memory consumption, some results are unavailable in our implementation and denoted as -.

Table 12: Results of NMAE (mean_{std}) on long-term forecasting scenarios, each containing five independent runs with different seeds. The context length is set to 36 for the ILI-L dataset and 96 for the others. Lower NMAE values indicate better predictions. The means and standard errors are based on 5 independent runs of retraining and evaluation. **Bold**: the best, *italics*: the 2nd best.

		EnergyScore						CRPS					
		ETTm1		Exchange			Solar	ETTm1		Exchange			Solar
Type	Methods	96	336	96	336	30	24	96	336	96	336	30	24
	MVE-1S	0.6020±0.0144	0.3832±0.0078	0.0064±0.0001	*0.0065*±0.0000	*0.0047*±0.0001	59.2827±3.4358	0.2430±0.0050	0.3301±0.0066	0.0210±0.0010	0.0399±0.0006	0.0077±0.0002	0.4774±0.0156
	K2VAE	0.4990±0.0103	0.3079±0.0014	0.0099±0.0009	0.0082±0.0009	0.0066±0.0002	49.0566±0.9311	0.2359±0.0072	0.2698±0.0011	0.0322±0.0027	0.0519±0.0043	0.0106±0.0004	**0.4129**±0.0167
2-stage	MVE-2S	0.4697±0.0021	0.2997±0.0015	0.0063±0.0001	**0.0064**±0.0000	0.0048±0.0000	48.6663±0.2327	0.2117±0.0009	0.2491±0.0008	0.0204±0.0000	0.0391±0.0001	**0.0074**±0.0001	0.4705±0.0035
	TORF	0.4651±0.0010	*0.2933*±0.0002	**0.0061**±0.0000	**0.0064**±0.0000	*0.0047*±0.0000	48.2329±0.0590	0.2088±0.0011	0.2457±0.0005	**0.0198**±0.0001	**0.0382**±0.0001	0.0077±0.0000	*0.4558*±0.0020
	w RealNVP	0.4674±0.0008	0.2956±0.0008	0.0063±0.0002	0.0066±0.0000	**0.0046**±0.0002	48.4335±0.1224	0.2101±0.0004	0.2517±0.0036	0.0210±0.0007	0.0412±0.0001	0.0077±0.0002	0.4774±0.0156
	w Spline	*0.4587*±0.0028	0.2938±0.0006	0.0065±0.0002	*0.0065*±0.0001	*0.0047*±0.0000	48.2329±0.0590	*0.2051*±0.0009	0.2452±0.0010	0.0213±0.0007	0.0394±0.0007	0.0078±0.0001	*0.4558*±0.0020
	w OddRealNVP	0.4656±0.0018	0.2946±0.0003	*0.0062*±0.0000	**0.0064**±0.0000	**0.0046**±0.0001	*48.1577*±0.1690	0.2086±0.0004	*0.2450*±0.0009	*0.0203*±0.0001	0.0397±0.0000	**0.0074**±0.0001	0.4588±0.0029
	w Mixture	**0.4559**±0.0007	**0.2930**±0.0006	*0.0062*±0.0001	**0.0064**±0.0001	0.0048±0.0001	**48.0786**±0.0729	**0.2038**±0.0005	**0.2436**±0.0004	0.0208±0.0002	*0.0390*±0.0000	*0.0076*±0.0001	0.4653±0.0038

Table 13: Energy Score and CRPS results. **Bold** = best (lowest); *italic* = second best.

	ETTm1	Exchange		Solar
Method	336	336	30	24
TORF-1S	0.5122±0.0079	0.1390±0.0032	0.1430±0.0005	0.6078±0.0269
TORF	**0.2457**±0.0005	**0.0382**±0.0001	**0.0077**±0.0000	**0.4558**±0.0020
w/o LSL	0.2508±0.0015	0.0399±0.0018	0.0081±0.0003	0.4670±0.0031
w/o CNN-ROSS	*0.2483*±0.0004	*0.0383*±0.0000	*0.0078*±0.0001	**0.4558**±0.0020

Table 14: CRPS results with standard deviation over 5 runs.

on Electricity-S the strongest baseline overall is CSDI (0.051 CRPS, 0.066 NMAE), not K^2VAE (0.053, 0.068, the latter exactly tied with TORF on CRPS); the reported improvement percentages already reflect this, since they are computed against whichever baseline is strongest per column. CSDI, a diffusion-based method, is the strongest baseline specifically on Electricity-S, but is not competitive elsewhere in either table, underscoring the same pattern as the long-term setting. Individual alternative methods win narrow, dataset-specific pockets, while TORF is competitive across the full spread of short- and long-horizon, univariate and highly multivariate datasets.

K All Horizon Results

We extend the analysis to include Table 11 and Table 12, where we compare the performance of TORF across all the datasets, horizons and baselines for long-horizon forecasting.

Across all four horizons, TORF's advantage is stable rather than an artifact of the $H{=}720$ headline comparison: on the four ETT variants, Exchange-L, and ILI-L, TORF is the best or second-best CRPS at every horizon, and the outright best NMAE at every horizon without exception and on several of these, its margin over K^2VAE actually *widens* with horizon rather than shrinking (e.g. ETTh2-L). This reflects a structural advantage of TORF's two-stage design: SimpleTM is trained purely to produce the best point forecast as Stage 1, whereas the probabilistic baselines' point predictions are only a byproduct of a density-estimation objective they were never trained to optimize.

The two datasets where TORF actually struggles show clear, horizon-dependent patterns. On **Traffic-L**, TORF trails both K^2VAE and GRU NVP on CRPS at the shorter horizons ($H{=}96, 192$; NMAE ranks it 3rd there too), but its ranking improves steadily with horizon: it overtakes K^2VAE outright at $H{=}336$ and is a close 2nd by $H{=}720$. On **Weather-L**, CSDI, and not K^2VAE is the CRPS leader at $H{=}96, 192$, but TORF overtakes it by $H{=}336, 720$. NMAE follows almost the same pattern, with a single-horizon dip to 3rd at $H{=}192$.

Electricity-L runs the other way on CRPS: TORF wins clearly at $H{=}96, 192$ but trails K^2VAE at $H{=}336, 720$, even though TORF's NMAE stays best across all four Electricity-L horizons.

L Results with Standard Deviations

Probabilistic Analysis We extend the ablation with a shorter horizon ($H{=}96$) for ETTm1 and Exchange, in addition to the existing horizon-336 settings, comparing TORF and its variants against K^2VAE (Wu et al. 2025) and the Gaussian MVE-1S/MVE-2S baselines described earlier. Table 13 reports Energy Score, which measures joint distribution quality, and CRPS, which measures per-channel marginal quality, together with standard deviations across the 5 independent runs.

TORF and its variants are state-of-the-art on 11 of the 12 columns. The one exception is Solar-24 CRPS, where K^2VAE remains best, consistent with the Stage-1 SimpleTM limitation on Solar-S already noted in Table 10.

Among the 1-stage baselines, MVE-1S beats K^2VAE on Exchange on both Energy Score and CRPS, even though their NMAE is comparable (Table 16), so the gap cannot come from point-forecast accuracy; it suggests Exchange's residual is close to Gaussian, a shape K^2VAE's more complex structure fails to exploit. MVE-2S already beats K^2VAE on 11 of the 12 columns, losing only on the same Solar-24 CRPS column every method loses on. Comparing MVE-1S against MVE-2S shows that training the mean and the distributional head sequentially, rather than jointly, avoids the optimization pitfalls of combined NLL training, confirming the two-stage approach is the better training strategy even before any flow is introduced.

Among the TORF variants, all are state-of-the-art on 11 of 12 tasks, losing only the same Solar-24 CRPS column. TORF and TORF-Mixture are the strongest, winning 4/12 and 6/12 columns respectively. K^2VAE wins the univariate Solar-24 CRPS, but every TORF variant still beats it on the corresponding multivariate Solar-24 Energy Score, evidence of better joint distributional modeling even where the marginal residual is harder to fit.

TORF-Mixture wins the most columns overall, but its margin over TORF is mostly insignificant except on Solar-24 and ETTm1-96, and, as Table 7 shows, it costs substantially more compute. We therefore propose plain TORF as the default, with TORF-Mixture as the better choice where the infrastructure allows, since it can model asymmetric residual distributions that a single odd spline cannot.

Horizon	Dataset	#var.	Range	Freq.	Timesteps	Description
Long-term	ETTh1-L / ETTh2-L	7	$\mathbb{R}^+$	H	17,420	Electricity transformer temperature, recorded hourly
	ETTm1-L / ETTm2-L	7	$\mathbb{R}^+$	15min	69,680	Electricity transformer temperature, recorded every 15 minutes
	Electricity-L	321	$\mathbb{R}^+$	H	26,304	Household electricity consumption (kWh)
	Traffic-L	862	$(0, 1)$	H	17,544	Road occupancy rates
	Exchange-L	8	$\mathbb{R}^+$	Business day	7,588	Daily exchange rates for 8 countries
	ILI-L	7	$(0, 1)$	W	966	Fraction of patients presenting influenza-like illness
	Weather-L	21	$\mathbb{R}^+$	10min	52,696	Local climatological measurements
Short-term	ETTh1-S / ETTh2-S	7	$\mathbb{R}^+$	H	17,420	Electricity transformer temperature, recorded hourly
	ETTm1-S / ETTm2-S	7	$\mathbb{R}^+$	15min	69,680	Electricity transformer temperature, recorded every 15 minutes
	Exchange-S	8	$\mathbb{R}^+$	Business day	6,071	Daily exchange rates for 8 countries
	Solar-S	137	$\mathbb{R}^+$	H	7,009	Solar power production records
	Electricity-S	370	$\mathbb{R}^+$	H	5,833	Household electricity consumption
	Traffic-S	963	$(0, 1)$	H	4,001	Road occupancy rates

Table 15: Statistical information of the datasets.

		ETTm1	Exchange		Solar
Type	Methods	336	336	30	24
1-stage	TORF-1S	0.512±0.008	0.169±0.003	0.188±0.000	0.834±0.008
	MVE-1S	0.349±0.006	0.056±0.001	0.011±0.000	0.599±0.011
	K2VAE	0.339±0.002	0.055±0.002	0.011±0.000	**0.531**±0.018
2-stage	MVE-2S	**0.314**±0.002	**0.051**±0.000	**0.010**±0.000	*0.583*±0.025
	Odd TORF	**0.314**±0.002	**0.051**±0.000	**0.010**±0.000	*0.583*±0.025
	w RealNVP	*0.315*±0.001	*0.052*±0.001	**0.010**±0.000	0.630±0.001
	w Spline	0.319±0.001	0.052±0.000	**0.010**±0.000	0.629±0.002

Table 16: NMAE results with standard deviation over 5 runs.

		NMAE		CRPS	
Dataset	Horizon	iTrans	SimpleTM	iTrans	SimpleTM
ETTm1	96	0.2762±0.0048	**0.263**±0.001	0.2209±0.0011	**0.2088**±0.0011
	336	0.3211±0.0042	**0.314**±0.002	0.2581±0.0003	**0.2457**±0.0005
Exchange	96	**0.0241**±0.0012	0.025±0.001	**0.0186**±0.0001	0.0198±0.0001
	336	**0.0449**±0.0007	0.052±0.001	**0.03532**±0.0001	0.0382±0.0001

Table 17: First stage method comparison across datasets and horizons (NMAE and CRPS, mean with std in scriptsize). Best per column in **bold**.

Among the non-odd, non-mixture alternatives, TORF-w/Spline wins more columns than TORF-w/RealNVP. TORF-w/OddRealNVP is a compelling baseline on its own, winning outright on 3/12 columns using only an affine transform, reaffirming the odd residual flow as a strong, resource-efficient choice for distribution modeling.

M Backbone Transfer Analysis

We test whether TORF's flow hyperparameters need to be re-tuned whenever the Stage-1 point forecaster changes. We take the flow configuration (including the number of ROSS blocks K) tuned for TORF with SimpleTM as Stage-1, and reuse it unchanged with iTransformer substituted as Stage-1, without any additional hyperparameter search for the flow.

Dataset	H	LR	Weight decay	C_{hid}	K	N_b	k	k_f
ETTh1-S	24	3.44×10^{-4}	2.42×10^{-4}	32	8	16	4	8
ETTh2-S	24	5.28×10^{-5}	4.86×10^{-4}	512	8	16	4	8
ETTm1-S	24	4.13×10^{-4}	8.19×10^{-3}	256	8	32	7	4
ETTm2-S	24	7.91×10^{-4}	4.85×10^{-3}	256	8	32	7	4
Electricity-S	24	3.18×10^{-4}	8.97×10^{-3}	512	8	16	7	4
Traffic-S	24	3.87×10^{-4}	9.48×10^{-3}	256	8	32	7	4
Solar-S	24	6.59×10^{-4}	8.39×10^{-6}	32	2	32	7	4
Exchange-S	30	9.69×10^{-4}	7.97×10^{-4}	128	0	32	2	16

Table 18: Best hyperparameters selected by Optuna for each short-term dataset (horizon H).

Table 17 reports both point-forecast accuracy (NMAE) and probabilistic accuracy (CRPS) for both backbones on ETTm1 and Exchange.

The pattern is consistent across all four dataset-horizon combinations. Whichever backbone is the better point forecaster is also the one that yields the better CRPS once plugged into the unchanged flow. On ETTm1, SimpleTM has the lower NMAE at both horizons and also the lower CRPS. On Exchange the ranking flips. iTransformer is the better point forecaster and correspondingly achieves the better CRPS. Simply swapping in whichever backbone is the stronger point forecaster for the dataset at hand, without re-tuning TORF's flow hyperparameters, is enough to improve probabilistic performance.

This indicates that TORF's flow hyperparameters transfer across different pretrained Stage-1 models without requiring their own expensive re-tuning. Probabilistic performance tracks the quality of whichever point forecaster is plugged in, rather than being tied to the specific backbone the flow happened to be tuned on.

Dataset	H	LR	Weight decay	C_{hid}	K	N_b	k	k_f
ETTh1-L	96	2.16×10^{-4}	3.61×10^{-6}	64	2	16	25	4
	192	1.08×10^{-4}	1.20×10^{-5}	64	2	16	13	16
	336	1.24×10^{-4}	6.52×10^{-5}	128	1	16	43	8
	720	7.93×10^{-4}	8.59×10^{-3}	256	0	32	46	16
ETTh2-L	96	4.28×10^{-5}	2.49×10^{-3}	256	8	32	25	4
	192	9.29×10^{-4}	1.16×10^{-4}	32	1	32	7	32
	336	3.70×10^{-4}	1.97×10^{-5}	64	1	16	43	8
	720	3.16×10^{-6}	1.67×10^{-4}	512	8	4	46	16
ETTm1-L	96	7.15×10^{-4}	7.98×10^{-3}	256	8	32	25	4
	192	7.15×10^{-4}	7.98×10^{-3}	256	8	32	49	4
	336	3.09×10^{-5}	1.07×10^{-6}	512	8	32	85	4
	720	1.24×10^{-4}	6.52×10^{-5}	128	1	16	91	8
ETTm2-L	96	2.65×10^{-4}	2.47×10^{-3}	32	8	32	13	8
	192	6.36×10^{-5}	2.00×10^{-3}	256	8	16	25	8
	336	6.66×10^{-5}	1.37×10^{-6}	256	8	16	22	16
	720	1.59×10^{-5}	8.64×10^{-6}	128	8	16	91	8
Electricity-L	96	3.15×10^{-4}	1.42×10^{-6}	128	8	16	13	8
	192	1.35×10^{-4}	7.99×10^{-6}	32	8	4	25	8
	336	3.35×10^{-5}	1.24×10^{-4}	128	8	8	85	4
	720	1.18×10^{-4}	1.48×10^{-6}	128	8	4	91	8
Traffic-L	96	3.07×10^{-4}	1.30×10^{-6}	512	8	32	25	4
	192	9.69×10^{-4}	1.67×10^{-4}	32	4	32	7	32
	336	3.46×10^{-4}	1.04×10^{-6}	128	1	16	43	8
	720	3.46×10^{-4}	1.04×10^{-6}	128	1	16	91	8
Exchange-L	96	1.58×10^{-5}	2.79×10^{-6}	512	4	8	4	32
	192	9.95×10^{-4}	8.19×10^{-3}	256	0	32	13	16
	336	3.53×10^{-5}	1.89×10^{-6}	512	4	4	11	32
	720	4.89×10^{-4}	2.95×10^{-5}	64	2	32	181	4
Weather-L	96	9.60×10^{-4}	2.37×10^{-4}	256	8	16	13	8
	192	4.89×10^{-4}	2.95×10^{-5}	64	2	32	49	4
	336	3.82×10^{-4}	8.19×10^{-3}	256	8	32	85	4
	720	2.35×10^{-4}	8.39×10^{-6}	32	2	32	91	8
ILI-L	24	9.79×10^{-4}	2.82×10^{-4}	512	0	4	1	32
	36	9.69×10^{-4}	4.01×10^{-4}	256	4	16	2	32
	48	9.95×10^{-4}	4.58×10^{-4}	256	0	32	7	8
	60	3.77×10^{-4}	1.97×10^{-6}	32	8	32	8	8

Table 19: Best hyperparameters selected by Optuna for each long-term dataset, per forecasting horizon H.

N Hyperparameter selection and Implementation details

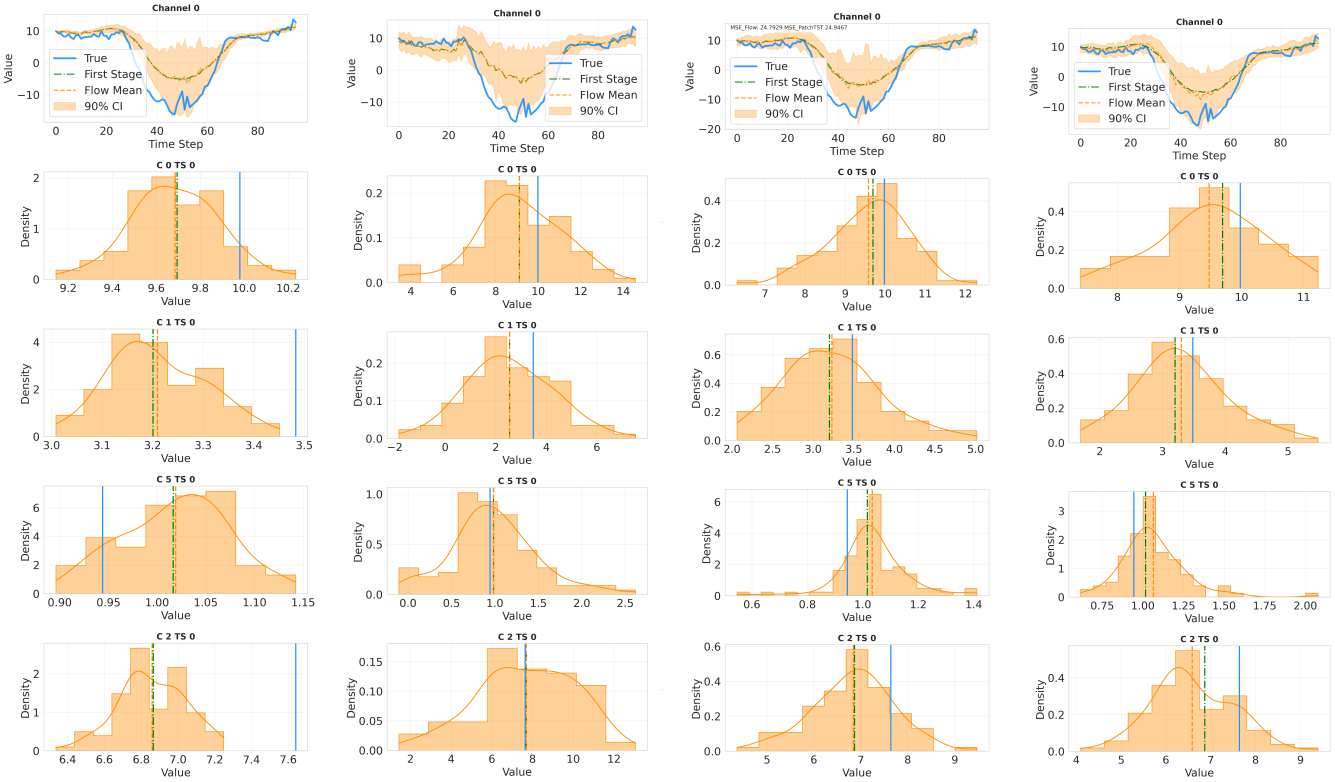
For each dataset–horizon pair, we tune the learning rate, weight decay, SplineNet-Conv1D hidden width C_{hid}, number of ROSS blocks K, number of spline bins N_b, and kernel size k via Optuna described in the main text (20 trials per configuration). We use AdamW optimizer with weight decay for training. Tables 18 and 19 report the resulting best configuration for every short-term and long-term dataset–horizon pair, respectively. Each selected configuration is then retrained and evaluated with five seeds ($\{0, 1, 2, 3, 4\}$), and we report the mean and standard deviation across these runs throughout the paper.

Search space. Each trial draws the learning rate and weight decay log-uniformly from $[10^{-6}, 10^{-3}]$ and $[10^{-6}, 10^{-2}]$ respectively, and the remaining flow hyperparameters from fixed categorical grids: $C_{\text{hid}} \in \{32, 64, 128, 256, 512\}$, $K \in \{0, 1, 2, 4, 8\}$, and $N_b \in \{4, 8, 16, 32\}$. SplineNet's Conv1D kernel size is not tuned directly; instead Optuna selects a kernel factor $k_f \in \{4, 8, 16, 32\}$, and the kernel size is derived as $k = \lfloor H/k_f \rfloor + 1$ for horizon H, so it automatically scales with the forecast length. Tables 18 and 19 report both the derived k and the underlying tuned k_f for every selected configuration.

Compute infrastructure. All experiments ran on an internal Linux SLURM cluster using Python 3.10 and PyTorch 2.13, spanning several GPU generations, from NVIDIA GTX 1080 Ti (11 GB) up to NVIDIA A40 (48 GB). Jobs were assigned by memory requirement rather than uniformly at random: short-horizon runs and datasets with few channels ran on the smaller-VRAM 1080 Ti/2080 Ti/A4000 nodes, while the two highest-channel-count datasets like Electricity (321 channels long-term, 370 short-term) and Traffic (862 long-term, 963 short-term) and every H=720 run, whose per-batch tensors are the largest in the benchmark, were scheduled on the higher-VRAM 3090/4090/A40 nodes (24–48 GB) to avoid out-of-memory failures. For the memory requirement and inference time analysis (ablation), all the experiments were run on the same NVIDIA RTX 2070 SUPER with 8 GB of memory for comparability.

O Visual Analysis

Figure 4 compares all four methods on the exact same test window and channel, so the differences in predictive shape are directly attributable to the model rather than to the underlying data. We use n=100 samples to generate each predictive distribution shown, consistent with the convention in prior work (Wu et al. 2025); with this few samples, the histograms are noisy estimates of the true underlying density rather than exact densities, so fine-grained differences in shape should be read qualitatively. MVE-2S's residual histograms are the narrowest of the four, consistent with it being an exactly Gaussian model (Section G): on C1 TS0 and C2 TS0 in particular, the true value falls almost entirely outside its 90% mass, a visual symptom of the overconfidence that a purely affine, unimodal transform produces once the true residual is not Gaussian. K^2VAE sits at the opposite extreme: its densities are visibly flatter and wider, and on C1 TS0 its mass extends into negative values even though every true value of that channel is positive, suggesting its predictive distribution is not well constrained by the data's actual support. TORF and TORF-Mixture both produce densities that are noticeably tighter around the true value than either baseline while still crossing it, and their Stage-1 (green) and predictive-mean (orange) lines stay visually coincident in every panel, the mean-preservation guarantee of Lemma 1 holding exactly, even though the shape around that mean adapts per channel. The one visible difference between the two TORF variants is on C2 TS0, where TORF-Mixture's density is noticeably more skewed than the single-component TORF's, illustrating in a single test window the asymmetric residual modeling that Section I motivates analytically.



(a) MVE-2S (Gaussian) (b) K^2VAE (c) TORF (d) TORF-Mixture

Figure 4: Qualitative comparison of predictive distributions for four methods (MVE-2Stage, K^2VAE, TORF, TORF-Mixture) on the same ETTm1 (H=96) test window and channel. In each panel, the top plot shows the forecast horizon (true trajectory in blue, Stage-1 point forecast in green, predictive mean in dashed orange, 90% CI shaded); the four plots above show the predictive density at fixed (channel, time step) pairs C0/C1/C5/C2 at TS 0, with the true value (blue), Stage-1 forecast (green), and predictive mean from the flow (dashed orange) marked as vertical lines.