GETUP: GATED ENSEMBLE FOR TIME-SERIES UNCERTAINTY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Combining time-series point forecasts does not by itself provide reliable uncertainty estimates for the resulting ensemble. Error prediction has its own learning targets and may require different expert weights from point forecasting. We propose GETUP (Gated Ensemble for Time-Series Uncertainty Prediction), a simple approach that learns these weights separately. Its zero-shot implementation uses two independent gates based on the input spectrum: one combines point forecasts, while the other combines error estimates learned after freezing the point branch. Separating error magnitude from residual shape supports both Gaussian and non-Gaussian predictive distributions. We evaluate GETUP on the TFB and expanded ProbTS benchmark grids, each covering 36 dataset–horizon combinations, alongside in-domain experiments on six datasets and four forecast horizons. Benchmark comparisons show improvements over established forecasting baselines, while controlled experiments examine the contributions of predictor-specific error learning, input-dependent scales, and residual shape. The construction supports zero-shot forecasting, uncertainty estimation for frozen pretrained models, and training on target history. Across these settings, explicitly learning forecast errors provides a simple way to improve predictive uncertainty while preserving the point forecasts.

1 INTRODUCTION

Time-series forecasting requires accurate predictions together with uncertainty estimates that describe plausible outcomes (Zhang et al., 2024). Low point error alone does not ensure that prediction intervals attain their stated coverage (Kuleshov et al., 2018). For an ensemble, the uncertainty estimate must describe the error of the combined forecast. Yet the weights learned to predict future values are not explicitly trained to predict the magnitude of that error. We therefore study a separate ensemble of error predictors, with its own weights and a direct learning target: the residuals of a fixed point forecaster.

Input-dependent uncertainty can be learned jointly with a point predictor through a mean–variance network (Nix & Weigend, 1994a). This couples the optimisation of predictive means and scales, motivating training objectives that protect point accuracy (Seitzer et al., 2022; Stirn et al., 2023). A separate error model offers another route: DEUP learns a predictor's generalisation error from out-of-sample observations (Lahlou et al., 2023), while practical deep heteroskedastic regression fits variance heads to a frozen network's intermediate features (Jordahn et al., 2026). We build on this principle to learn and combine residual predictions in a time-series ensemble, keeping the point predictor fixed throughout uncertainty training.

We introduce GETUP (Gated Ensemble for Time-Series Uncertainty Prediction), a simple approach to learning uncertainty around an ensemble forecast. Its zero-shot implementation extends spectrum-gated linear ensembles (Nochumsohn et al., 2026) with an independently trained error branch. Two gates use the input history's frequency content to assign expert weights: one combines point forecasts, and the other combines predicted error magnitudes. We consider two ways to train the error branch. The first learns the absolute residual of the combined forecast. The second learns each expert's absolute residual and combines these estimates with disagreement between the experts. Both keep the point forecasts unchanged while fitting the uncertainty model.

The same residual-prediction principle supports three practical settings: transfer to unseen datasets after source training, uncertainty estimation for a frozen pretrained forecaster, and training on the

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

target dataset's history. The predicted error magnitude supplies a scale, while a separate residual distribution determines shape. This supports Gaussian and non-Gaussian forecasts: we use Gaussian marginals, empirical residual distributions fitted to historical errors, and mixtures centred on trajectories from a generative forecaster. The choice of residual distribution can thus change without retraining the point predictor.

We evaluate point accuracy, predictive distributions and calibration across datasets and forecast horizons. On the expanded ProbTS benchmark, the source-trained ensemble reduces pooled continuous ranked probability score (CRPS) by 7.1% relative to published AURORA results (Wu et al., 2026). On the TFB point-forecast benchmark (Qiu et al., 2024), its mean relative MSE is 2.07% higher than published AURORA. Separate controlled comparisons hold the point forecasts fixed to measure the contribution of uncertainty estimation directly.

With AURORA's deterministic forecasts unchanged, fitting an error head to its residuals reduces mean negative log-likelihood (NLL) from 1.349 to 1.060 and raises coverage of nominal 90% intervals from 73.8% to 81.7%, compared with borrowing another predictor's error bank. For the in-domain ensemble, input-dependent error estimates reduce pooled CRPS by 3.74% relative to historical constant scales, with the point forecasts and residual shape held fixed. These results demonstrate that learning forecast errors can improve predictive distributions and interval coverage while preserving point accuracy. Section 4 discusses the scope of these findings, and Appendix E provides the extended comparison with prior work.

2 METHOD

2.1 PROBABILISTIC FORECASTING THROUGH RESIDUAL PREDICTION

Given a history $x \in \mathbb{R}^{L\times C}$ of L observations from C channels, our goal is to predict a distribution for the next H observations $y \in \mathbb{R}^{H\times C}$. GETUP separates this task into a point forecast, a prediction of its error magnitude, and a residual distribution. For horizon step h and channel c, the marginal forecast takes the form

$$Y_{h,c} = \mu_{\theta,h,c}(x) + s_{\psi,h,c}(x)Z_{h,c}, \tag{1}$$

where μ_θ is the point predictor, $s_\psi > 0$ is the predictive scale, and Z specifies the residual shape. With zero-mean, unit-variance Z, μ_θ and s_ψ are the predictive mean and standard deviation. More generally, they are location and scale parameters of the chosen residual law. The default $Z_{h,c} \sim \mathcal{N}(0,1)$ gives a Gaussian marginal; Section 2.3 introduces non-Gaussian alternatives.

We first train or load the point predictor, then hold its parameters θ fixed while learning its errors. This follows the established use of a separate error model (Lahlou et al., 2023; Jordahn et al., 2026). An error model $a_\psi(x) > 0$ predicts the absolute residual $|y - \mu_\theta(x)|$ using

$$\mathcal{L}_{\text{error}}(\psi) = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{fit}}}\left[\frac{1}{HC}\sum_{h=1}^{H}\sum_{c=1}^{C}\left(a_{\psi,h,c}(x) - |y_{h,c} - \mu_{\theta,h,c}(x)|\right)^2\right]. \tag{2}$$

Here $\mathcal{D}_{\text{fit}}$ is the data used to fit the error model: a source corpus for zero-shot forecasting or target history for in-domain forecasting. The squared loss targets the conditional mean absolute error, $a^*(x) = \mathbb{E}[|Y - \mu_\theta(x)| \mid x]$, elementwise. The basic Gaussian construction uses the raw prediction as its scale, $s_\psi = a_\psi$; Appendix B.4 gives the conversion between this regression target and a Gaussian standard deviation. Freezing the point predictor makes the residual target fixed and allows the uncertainty model to be fitted without changing the point forecast.

2.2 SPECTRUM-GATED FORECASTS AND UNCERTAINTY

The zero-shot expert bank, GETUP-ZS, offers several candidate forecasts for the same history. A *gate* assigns a nonnegative, input-dependent weight to each expert, with weights summing to one. We use separate gates $G(x)$ and $S(x)$ for point prediction and error estimation. This allows a history to favour different experts for the two tasks, following the mixture-of-experts principle (Shazeer et al., 2017). Let $\mu_i(x)$ and $a_i(x) > 0$ denote the outputs of point expert i and error expert i, respectively, in banks of size N. For clarity, we omit horizon and channel indices below.

The point forecast and our default predictive scale are constructed together:

$$\bar{\mu}(x) = \sum_{i=1}^{N} G_i(x)\mu_i(x), \qquad s_{\mathrm{RU}}(x) = \sum_{i=1}^{N} S_i(x)a_i(x). \tag{3}$$

We call this scale *Residual Uncertainty* (RU). Each error expert learns the magnitude of the combined forecast's residual, $|y - \bar{\mu}(x)|$. The error gate is fitted against the same target through its weighted output s_{RU}, while the point experts and gate G remain frozen. Together, the two branches define the Gaussian forecast $\mathcal{N}(\bar{\mu}(x), s^2_{\mathrm{RU}}(x))$ shown in Figure 1.

From expert scores to weights. Each gate assigns one score to each expert: $u_{\mu,i}(x)$ measures its preference for point expert i, and $u_{s,i}(x)$ its preference for error expert i. At inference, the gates retain the k_μ and k_s highest-scoring experts, whose index sets are $\mathcal{I}_\mu(x)$ and $\mathcal{I}_s(x)$, and normalise their scores by softmax:

$$\begin{aligned} G_i(x) &= \frac{\mathbf{1}\{i \in \mathcal{I}_\mu(x)\}\exp(u_{\mu,i}(x)/\tau_\mu)}{\sum_{j\in\mathcal{I}_\mu(x)}\exp(u_{\mu,j}(x)/\tau_\mu)}, \\ S_i(x) &= \frac{\mathbf{1}\{i \in \mathcal{I}_s(x)\}\exp(u_{s,i}(x)/\tau_s)}{\sum_{j\in\mathcal{I}_s(x)}\exp(u_{s,j}(x)/\tau_s)}. \end{aligned} \tag{4}$$

Unselected experts receive zero weight. The temperatures $\tau_\mu, \tau_s > 0$ control concentration: lower values favour the largest scores more strongly. Appendix C.2 compares the two gates at inference. At $H = 96$, they rank different experts first on every evaluated window for four of six datasets; substituting the point gate's weights increases mean relative CRPS by 10.79%.

Using the spectrum to select experts. We compute these scores from the history's frequency content, following Super-Linear (Nochumsohn et al., 2026). For one channel, let $\widehat{x}_q$ be the Fourier coefficient of the mean-centred history at retained frequency $q \in \{1, \dots, K\}$. Its squared magnitude $P_q(x) = |\widehat{x}_q|^2$ is the periodogram value, which measures power at that frequency. Dividing by total power gives

$$\pi_q(x) = \frac{P_q(x)}{\sum_{r=1}^{K} P_r(x)}, \qquad \pi(x) = \big(\pi_1(x), \dots, \pi_K(x)\big)^\top. \tag{5}$$

For nonzero total power, the entries are nonnegative and sum to one; this is the ℓ_1-normalised periodogram. Each entry represents a frequency's share of the observed variation. It tells the gates whether the history is dominated by slow changes or faster oscillations, independently of its overall amplitude. We set $\pi(x) = \mathbf{0}$ for a constant history. Two learned linear maps then produce the scores:

$$u_\mu(x) = W_\mu \pi(x) + b_\mu, \qquad u_s(x) = W_s \pi(x) + b_s, \tag{6}$$

where $W_\mu, W_s \in \mathbb{R}^{N\times K}$ and $b_\mu, b_s \in \mathbb{R}^N$. The maps learn separate weights for value and error prediction from the same spectral features. The experts themselves receive the history x, including its amplitude; the normalised spectrum determines only their weights.

Adding expert disagreement. Our alternative, *Disagreement-Augmented Residual Uncertainty* (DAU), preserves the point-forecast rule and changes the error targets and scale aggregation. Each expert has a point head μ_i and an error head a_i. With the point branch frozen, head i learns its own expert's absolute residual, $|y - \mu_i(x)|$. We combine these estimates with the experts' deviations from the ensemble forecast:

$$s^2_{\mathrm{DAU}}(x) = \underbrace{\sum_{i=1}^{N} S_i(x)a_i^2(x)}_{\text{residual contribution}} + \underbrace{\sum_{i=1}^{N} S_i(x)\big(\mu_i(x) - \bar{\mu}(x)\big)^2}_{\text{disagreement contribution}}. \tag{7}$$

The first term aggregates the squares of the error estimates. The second increases the scale when the selected experts disagree with the returned point forecast, drawing on ensemble approaches that combine predictive spread and variation between predictions (Lakshminarayanan et al., 2017). Both terms use the error gate S, while G determines $\bar{\mu}$. Figure 1 shows both scale constructions, and Appendix A.1 compares the RU and DAU training targets. Their uncertainty interpretation is discussed in Section 4.

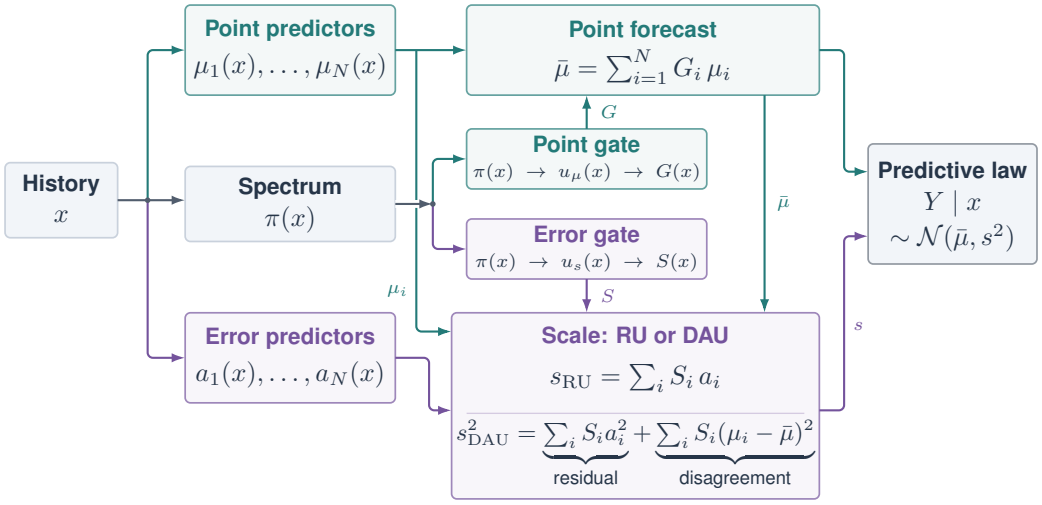

Figure 1: GETUP-ZS uses independent spectral gates for point and error predictions, with either RU or DAU supplying the predictive scale. RU trains error experts on $|y - \bar{\mu}|$; DAU trains paired error heads on $|y - \mu_i|$ and additionally uses μ_i and $\bar{\mu}$ in the scale block. Arguments x are omitted from the aggregation formulas.

The same bank can use several context lengths. Each context is a suffix ending at the forecast origin; context-specific views share learned weights and recompute normalisation and gating for that history. Expert architectures and bank sizes are specified in Appendix A.1; Appendix A.3 describes context training and forecast-head selection.

2.3 Other forecasters and residual distributions

Attaching uncertainty to an existing point predictor. GETUP-Point attaches an error head to a frozen pretrained point predictor. The head learns the predictor's residual magnitudes through Equation 2. We combine its scale with the chosen residual law, without updating the point predictor. The attachment experiment uses Gaussian marginals. The attachment experiment compares a head fitted to this predictor with an error bank borrowed from another predictor. Both use the same point forecasts, so the comparison isolates the effect of the scale model. Appendix A.4 specifies the backbone and fitting regime; Figure 5 in the appendix shows the two attachments.

A mixture around sampled trajectories. GETUP-Mix attaches uncertainty to a frozen generative forecaster by retaining its sampled trajectories as alternative centres. Given M trajectories $\mu_1(x), \ldots, \mu_M(x)$, a positive error head for each sample index predicts a component scale $s_j(x)$ from the history. The heads learn from the corresponding absolute residuals $|y - \mu_j(x)|$, giving

$$F_Y(\cdot \mid x) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{N}\big(\mu_j(x), s_j^2(x)\big). \tag{8}$$

This distribution combines variation across sampled forecasts with a learned spread around each forecast. This implementation uses absolute loss on the absolute-residual targets, whose population optimum is the conditional median absolute error. This differs from the squared loss in Equation 2; the training protocol is given in Appendix A.4. Appendix B defines mixture scoring, and Appendix D.1 gives the moment-matched Gaussian control.

Learning the residual shape. The in-domain version, GETUP-ID, fits a point ensemble and its uncertainty components on target history, including an empirical residual distribution. After fitting the point and scale models, we compute standardised residuals $z_{h,c} = (y_{h,c} - \mu_{\theta,h,c}(x))/s_{\psi,h,c}(x)$ on the residual-fitting data. A smoothed empirical CDF $\widehat{F}_c$ pools these residuals within channel c,

yielding the marginal predictive CDF

$$F_{h,c}(v \mid x) = \widehat{F}_c\left(\frac{v - \mu_{\theta,h,c}(x)}{s_{\psi,h,c}(x)}\right). \tag{9}$$

The history-dependent point and scale models determine location and spread; $\widehat{F}_c$ supplies the channel's residual shape. Because the fitted shape need not be centred or have unit variance, μ_θ remains the issued point forecast but need not equal the distribution's mean. We evaluate an offline version with a fixed CDF and an adaptive version that updates this component as observations become available. Their in-domain implementation is described in Appendix A.5, including the pooling of training and validation residuals used to fit the shape.

3 EXPERIMENTS

3.1 SETUP AND COMPARISONS

We evaluate whether our residual prediction GETUP can transfer to new datasets, clarify uncertainty for pretrained forecasters, and adapt to a dataset specifics. So, our test include different variants of GETUP. Specifically, the ZS suffix denotes *zero-shot* performance with training performed using a source corpus, and the ID suffix denotes *in-target-data*, meaning that the model is fitted using historical data from the target dataset. Point and Mix identify attachment constructions. We report adaptive GETUP-ID separately because it updates its residual distribution as observations become available during evaluation. The controlled attachment compares GETUP-POINT with GETUP-ZS+AURORA, which borrows the GETUP-ZS error bank, while keeping AURORA's point forecasts fixed.

Table 1 list these variants with their names, distinguishing the source of the point forecasts from the data used to fit their uncertainty. In addition to the considered options, we complete specific sensitivity studies with in-place defined acronyms. Specifically, the experiments consider how the historical-scale control test whether input-dependent error estimates add value, and the moment-matched mixture control isolates residual shape.

Table 1: Forecasting settings and uncertainty-fitting regimes.

Method	Point forecasts	Uncertainty fitting data	Problem addressed
GETUP-ZS	Trained on source corpus	Source corpus	Uncertainty estimation on unseen datasets without target-specific fitting
GETUP-POINT	Frozen AURORA deterministic head	Source corpus; target history when serving as the GETUP-MIX control	Predictive uncertainty for an existing deterministic forecaster
GETUP-MIX	Samples from frozen AURORA	Target history	Residual uncertainty around generated trajectories
GETUP-ID	Trained on target history	Target history	Dataset-specific uncertainty estimation when historical target data are available

We report point error (MSE and MAE), distribution quality scores (CRPS and NLL), and calibration and error-ranking diagnostics. Appendix B defines the metrics, aggregation rules, and scoring of sampled forecasts; Appendix C gives the full benchmark grids. Model sizes, training details, and context choices are collected in Appendix A.

Benchmark comparisons assess the complete forecasting systems. The TFB (Qiu et al., 2024) and expanded ProbTS (Zhang et al., 2024) grids each contain 36 dataset–horizon cells; the main in-domain comparison covers six datasets, four horizons and three seeds. Each table identifies its paired subset and fitting regime.

3.2 MAIN RESULTS

Figure 2 compares probabilistic accuracy within each fitting regime on the complete ProbTS subset. In the zero-shot setting, GETUP-ZS has the best mean CRPS rank, 1.30, followed by GETUP-POINT at 1.88. In the in-domain setting, GETUP-ID has a mean rank of 2.00, close to VolDy-VAE at 1.93. The full dataset–horizon scores, including cells outside this common subset, are reported in Table 16 in Appendix C.

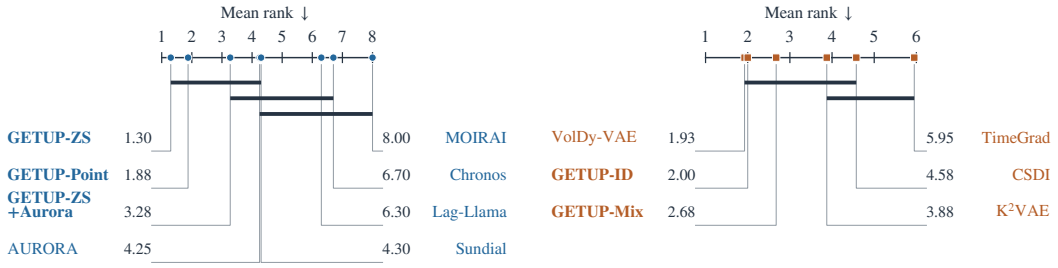

Figure 2: CRPS ranks in the zero-shot (left) and in-domain (right) settings. Each panel uses five datasets at four horizons, with scores available for every compared method. Ranks are computed within dataset–horizon pairs and then averaged. Thick bars connect methods not separated by the dataset-block reference described in Appendix B.2; they do not establish equivalence.

Table 15 summarises the broader ProbTS comparisons over all available pairs. GETUP-ZS with RU has lower pooled CRPS than each of the five source-trained baselines; the reduction relative to published AURORA is 7.1%, with lower scores in 30 of 36 settings. Against the target-trained VolDy-VAE, GETUP-ZS has 5.4% higher pooled CRPS over 36 settings, while GETUP-ID has 2.8% lower pooled CRPS over its 24 available settings. These aggregates use a broader set of results than the complete subset in Figure 2.

For point forecasting, GETUP-ZS with RU has 2.07% higher mean relative MSE than published AURORA across the 36 TFB settings. Table 10 reports this comparison and its paired bootstrap interval; Appendix C.1 gives the serving-rule and tokenisation ablations.

3.3 LEARNING THE SCALE OF A FIXED POINT PREDICTOR

Table 2 compares two ways to estimate uncertainty for the same AURORA point forecasts: GETUP-ZS+AURORA borrows the GETUP-ZS error bank, whereas GETUP-POINT fits an error head to AURORA's residuals. Both error models are trained on source data and evaluated without target-specific fitting on nine datasets at four forecast horizons each. The standalone GETUP-ZS provides a reference with its own point forecasts. All three use marginal Gaussian distributions in this comparison.

Table 2: Mean scores over 36 dataset–horizon settings. Bold marks the best value in a column and underlining the second best, by the direction each metric declares in its header: lower for the errors and the scores, closest to 0.90 for coverage and to 1 for c^*. Coverage is compared with 0.90; c^* is the NLL-optimal scale multiplier (Equation 13), with reference value 1. The two attachment rows share our own run of AURORA's deterministic head; Table 11 places that run beside the published AURORA point metrics cell by cell, which is how far the shared forecast can be checked against the system it reproduces.

	MSE (↓)	MAE (↓)	CRPS (↓)	NLL (↓)	cov_{90} (→ 0.90)	AUC (↑)	c^* (→ 1)
GETUP-ZS (own pair)	0.4368	0.4164	0.3184	1.3810	0.7255	<u>0.6912</u>	1.8342
GETUP-ZS+AURORA (borrowed σ)	<u>0.4427</u>	0.4055	0.3116	<u>1.3490</u>	<u>0.7375</u>	0.6957	<u>1.8085</u>
GETUP-POINT (refit σ)	0.4427	0.4055	<u>0.3146</u>	1.0597	0.8165	0.6828	1.4318

GETUP-POINT has the lowest NLL and interval coverage closest to the nominal 90%, with the same point accuracy as the borrowed-head control. GETUP-ZS+AURORA has the lowest CRPS and the highest error-ranking AUC.

3.4 IN-DOMAIN UNCERTAINTY

Table 3 compares GETUP-ID configurations on six datasets at four horizons, with three training seeds. Versions 2.3, 2.5 and 2.5sc share identical point forecasts and differ only in uncertainty estimation. The default GETUP-ID is the offline 2.5 model; the adaptive 2.5sc version also updates its residual distribution during evaluation. Appendix A.5 describes the configurations.

Table 3: In-domain results under the ratio-split protocol at $L = 96$. Bold marks the best value in a column and underlining the second best, with coverage judged by distance from 0.90; where three configurations share a point forecast their MSE and MAE tie exactly and no second is marked. Score columns average 72 dataset–horizon–seed settings. The final column reports mean per-setting relative CRPS improvement over v1, with win counts in parentheses.

Version	MSE (↓)	MAE (↓)	CRPS (↓)	NLL (↓)	cov_{90} (→ 0.90)	ΔCRPS vs v1 (↑)
GETUP-ID v1	0.3365	0.3751	0.2842	0.9293	0.8102	—
GETUP-ID 2.3	0.3069	0.3618	0.2676	0.5096	<u>0.9155</u>	+4.71% (72/72)
GETUP-ID *(2.5)*	0.3069	0.3618	<u>0.2658</u>	0.4918	0.9072	+5.27% (72/72)
GETUP-ID (adaptive) *(2.5sc)*	0.3069	0.3618	0.2644	<u>0.4931</u>	0.8838	+5.82% (72/72)

At fixed point forecasts, the offline 2.5 configuration improves both CRPS and NLL over 2.3 and brings coverage closer to 90%. It also improves CRPS over v1 in all 72 settings, by 5.27% on average; that comparison also changes the point forecasts. The adaptive version achieves the lowest CRPS, while offline 2.5 retains the lowest NLL and coverage closest to nominal. Seed-level variation is reported in Appendix C.3.

Irregular sampling. An earlier GETUP-ID configuration also improves on the best published baseline in UFO's regular and irregular settings (Kuleshov et al., 2026) on both ETT datasets, with NCRPS reductions of 3.6%–16.6%. On Weather, it has higher NCRPS than UFO in both settings. Appendix C.4 gives the sampling protocol, full comparison and channel-level analysis.

3.5 INPUT-DEPENDENT SCALES AND RESIDUAL SHAPE

To isolate scale prediction, we keep the point forecast fixed and replace the learned scale with a historical constant for each horizon step and channel. This control captures systematic differences across the forecast grid but cannot adapt to the input history. Table 4 compares two versions: one preserves the model's empirical residual shape, and the other uses a Gaussian law.

Table 4: Historical-scale controls for GETUP-ID on 24 common cells, with the point forecast fixed and scale constants fitted on validation. The first control preserves the empirical residual shape; the second uses a Gaussian law. Dataset rows average four horizons, and the final score row averages all cells. The last line counts cells in which the replacement has a lower score. Coverage is compared with 0.900. Bold marks the best of the three laws in a row for that metric and underlining the second best; on the coverage/PIT-KS line the marks apply to the three pairs.

	GETUP-ID, own law		+ historical, own shape		+ historical, Gaussian	
Dataset	CRPS	NLL	CRPS	NLL	CRPS	NLL
ETTh1	0.3698	0.9660	0.3810	<u>1.0287</u>	<u>0.3791</u>	1.0918
ETTh2	0.2472	0.4770	0.2535	0.5506	<u>0.2504</u>	<u>0.5427</u>
ETTm1	0.3235	0.7748	0.3357	<u>0.8662</u>	<u>0.3305</u>	0.8793
ETTm2	0.2016	0.2900	<u>0.2087</u>	<u>0.3488</u>	0.2089	0.3631
Exchange	0.2534	<u>0.5390</u>	<u>0.2537</u>	0.5350	0.2665	1.0641
Weather	0.2120	0.0455	<u>0.2371</u>	0.5148	0.2454	<u>0.3700</u>
All 24 cells	0.2679	0.5154	<u>0.2783</u>	<u>0.6407</u>	0.2801	0.7185
cov_{90} / PIT-KS	0.8965 / 0.0419		0.9573 / 0.0981		0.9029 / 0.0878	
historical better, of 24	—		3 CRPS, 4 NLL, 0 KS		1 CRPS, 1 NLL, 0 KS	

The learned distribution has lower pooled CRPS by 3.74% than the shape-preserving control and by 4.36% than the Gaussian control, with pooled NLL lower by approximately 0.125 and 0.203

nats. PIT-KS is lower in all 24 cells under both substitutions. Across five nominal interval levels—50, 68, 80, 90 and 95%—the mean absolute coverage deviation is 0.0308 for the learned law and 0.0904 for the shape-preserving control. The shape-preserving comparison isolates the benefit of the learned scale at fixed point predictions and residual distribution. Additional scale and point-predictor substitutions are reported in Appendix D.

The same control on sampled baselines. Applying the substitution to TARFVAE and VolDy-VAE gives the same qualitative picture and a sharper one for likelihood. On Exchange at $H = 96$, with VolDy-VAE scored under its own sampled law, replacing its scale leaves CRPS unchanged to three digits (0.1863 against 0.1856) while NLL falls from 2.4520 to 0.3504 and coverage moves from 0.8265 to 0.8571 against a nominal 0.900: a different calibration profile behind an unchanged CRPS. For the reported TARFVAE configurations the historical-scale Gaussian improves CRPS, NLL and distance from nominal coverage together. Scoring a sampler instead through a Gaussian fitted to its sample moments changes these likelihoods, and in the direction that flatters our own comparison—Appendix B.5 records how far that approximation can move a log score—so every cell-level grid in Appendix D.2 scores each model under the law its own artefact declares. Those expanded control grids fit constants on the first half of each saved test bundle and score the second half. They are separate from the validation-fitted control in Table 4.

Residual shape. The sampled-trajectory mixture is compared with a single-head forecast and, on five dataset–horizon cells, with a Gaussian sharing its mean and variance. The latter control changes shape alone; Appendix D.1 reports the raw and oracle-rescaled scores separately. Persistence-error diagnostics also show that residual shape varies across channels and horizons, including heavy tails and concentrations at zero (Appendix D.3). These diagnostics motivate flexible residual laws without assuming that every trained predictor has the same residual distribution.

3.6 Inference and training cost

Table 7 compares single-window CPU inference on ETTh2 at $H = 96$. The bank takes 4.556 ms of wall time and 4.675 ms of processor time per window, against 104.410 ms and 164.162 ms for AURORA with 100 samples: ratios of 22.9 and 35.1. The difference reflects the evaluated inference procedures rather than model size alone—the bank issues its Gaussian parameters directly, whereas the AURORA run generates the 100 samples its own protocol specifies. The ratio is therefore a statement about these two procedures as evaluated, not about the architectures at equal sample budgets.

Training cost separates the two regimes: an in-domain model is trained once per cell, whereas the bank's source pretraining is shared across every zero-shot task, and at $H = 720$ DAU takes 3307 s of wall time against RU's 7766 s. Appendix B.8 gives the latency table itself, the recorded training and inference costs, their loaders and their window counts.

4 Limitations

4.1 Experimental coverage and protocol

The controlled experiments test predictor-specific error fitting, scale prediction, and residual shape. They do not compare staged training with matched joint-likelihood, β-NLL, or faithful-regression alternatives. The gate ablation in Appendix C.2 substitutes weights at serving time on trained error experts, so it measures the effect of replacing their weights rather than what a jointly trained single-gate model would learn; it also records one dataset, ETTm1 at $H = 96$, on which a uniform average of the selected error experts beats the gate's own weights. The attachment uses one pretrained backbone; the moment-matched mixture control covers five datasets at one horizon. Target-fitted mixture results concern adaptation rather than zero-shot transfer. The benchmark grids in Appendix C identify missing cells and results obtained with earlier configurations.

The in-domain error models and stackers use residuals from the point models' training windows, without cross-fitting. These residuals can underestimate out-of-sample error. The scale anchor uses validation residuals, while shape fitting pools training and validation residuals. Freezing stabilises the target but does not make it out of sample. We also lack $K = 1$ and $M = 1$ controls, so the

contributions of the uncertainty and point ensembles are not isolated. The adaptive multiplier grid and gate choice were informed by earlier evaluation-fitted measurements; the offline arm avoids this adaptive selection, but the adaptive choices require fresh confirmation.

Scoring protocols impose further limits. A Gaussian fitted to sample moments can change a sampler's scores; sample-based density estimates depend on sample count and bandwidth. In the per-cell law grids the likelihood, coverage and PIT columns come from a model's own trajectories, and so does VolDy-VAE's CRPS on 40 of its 48 cells; TARFVAE's and the other eight are the Gaussian reading of two moments, and that substitution moves the score by at most 0.4% where both exist. An earlier -51% to $+68\%$ figure also changed the window subset and the normalisation, so it did not isolate the estimator. The expanded historical-scale grids fit on the first half of saved test predictions, whose windows can share target timestamps with the scoring half. A separate calibration period with a horizon-sized gap is needed. These grids are distinct from the validation-fitted control in Table 4. The $@c$ columns fit dilation on scoring labels and are oracle diagnostics. Paired-cell intervals describe variation across related dataset–horizon cells, not independent training runs; the AUC differences lack dedicated intervals. Post-hoc serving rules also require evaluation on new tasks. Reported costs depend on hardware and serving settings; GETUP-ID's single-window latency, its training cost, the attachment and the ten-sample mixture remain unmeasured.

4.2 Limits of the construction

An absolute-error regressor does not automatically output a calibrated standard deviation. The conversion in Appendix B.4 assumes conditionally centred Gaussian residuals and depends on the loss. The raw-scale experiments retain this mismatch, and source-trained attachments remain below nominal coverage despite improved likelihood. Positivity also does not prevent scale collapse. Appendix D.4 records a failed long-horizon head and its recovery after several training changes; that comparison does not identify which change resolved the failure.

RU and DAU describe how the scale is constructed, rather than identifying data (aleatoric) and model (epistemic) uncertainty (Kendall & Gal, 2017). Residuals contain observation variability and predictor error. DAU measures disagreement around the G-weighted forecast using weights S, so its rule is not generally the variance decomposition of an S-weighted mixture. Disagreement can indicate model uncertainty, but local residual targets may already contain expert bias; the two terms need not identify separate sources.

The bank predicts marginal distributions without learning joint dependence across forecast steps or channels. Mixture heads condition on history rather than on sampled trajectories, whose exchangeable indices do not define stable trajectory types. Finally, marginal calibration diagnostics such as PIT and ENCE do not establish conditional calibration.

5 Conclusion

We introduced GETUP, a simple approach to learning uncertainty around a frozen forecaster. Its zero-shot version, GETUP-ZS, uses separate spectrum-based gates for point forecasting and uncertainty estimation. The gates combine linear experts, allowing different weights for predicting future values and estimating their errors. We construct uncertainty either by predicting the error magnitude of the combined forecast or by combining individual experts' error estimates with their disagreement. Both constructions use regression on absolute residuals while keeping the point predictor frozen. Modelling predictive scale and residual shape separately supports both Gaussian and non-Gaussian distributions, including mixtures and empirical residual models.

Our experiments examine the contributions of error learning and residual shape through controlled comparisons. At fixed AURORA point forecasts, a head fitted to the predictor's residuals improves likelihood and interval coverage over a borrowed error bank. Controls for GETUP-ID show that input-dependent scales improve average CRPS and likelihood over historical constants while preserving the point forecasts and residual shape. The mixture experiment further demonstrates gains from retaining non-Gaussian shape when the first two moments are fixed. These findings support a simple, modular approach to uncertainty estimation, with each component evaluated through its effect on the predictive distribution. Extending these comparisons to more pretrained backbones and learning dependence across forecast steps are natural next directions.

USE OF AI TOOLS

Large language model assistance was used in preparing this submission, and the authors take full responsibility for its contents. Its use was confined to three kinds of work: editing of prose we had written; entering the recorded results into the tables and the LaTeX that typesets them, including their layout and the marking of best and second-best entries; and refactoring of code we had already written. No experimental result, number or table entry was produced by a language model, and no method, model or evaluation protocol was designed by one. Every reported figure comes from the runs described in Appendix B and is read from their recorded outputs by the scripts named beside it.

ETHICS STATEMENT

This work uses only public benchmark time-series collections—ETT, Weather, Electricity, Traffic, Exchange, Illness, Wind, Solar, PEMS08 and NYSE—together with a pretraining corpus assembled from public series. None contains personal or sensitive data about identifiable individuals, and no human subjects were involved. The method predicts uncertainty for forecasts and carries the usual caution that attends any calibration claim: a predictive interval that is well calibrated on these datasets is not thereby calibrated on a deployment whose distribution differs, and Section 4 states where our diagnostics stop. We report the failures we found in our own pipeline rather than only the successes, and we identify the cells we did not measure.

REPRODUCIBILITY STATEMENT

The construction is specified in Section 2 and Appendix A; the gate inputs, expert inventory and every hyperparameter we set are listed in Appendix A.2. Appendix B states, for each experiment, the data and cells, what is held fixed, what is fitted and on which split, and where its scores are read; Table 6 collects that in one place. The benchmark grids in Appendix C report every cell we measured and mark those we did not, so the coverage of each comparison can be read off rather than inferred. Source code is submitted as supplementary material, including the scripts that generate the tables from the recorded results and the checks that verify them; the per-cell figures behind four claims in the text are shipped as CSV under `results/`.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006. URL https://jmlr.org/papers/v7/demsar06a.html.

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

Tilmann Gneiting, Adrian E. Raftery, Anton H. Westveld, III, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005. doi: 10.1175/MWR2904.1.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://openreview.net/forum?id=wEc1mgAjU-.

Eric P. Grimit, Tilmann Gneiting, Veronica J. Berrocal, and Nicholas A. Johnson. The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification. *Quarterly Journal of the Royal Meteorological Society*, 132(621C):2925–2942, 2006. doi: 10.1256/qj.05.235.

Mikkel Jordahn, Jonas Vestergaard Jensen, James Harrison, Michael Riis Andersen, and Mikkel N. Schmidt. Practical deep heteroskedastic regression, 2026. URL https://arxiv.org/abs/2603.01750. arXiv:2603.01750.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.04977.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2307.11494.

Ilya Kuleshov, Alexander Marusov, and Alexey Zaytsev. U-Former ODE: Fast probabilistic forecasting of irregular time series, 2026. URL https://arxiv.org/abs/2602.11738. arXiv:2602.11738.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2796–2804, 2018. URL https://proceedings.mlr.press/v80/kuleshov18a.html.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=eGLdVRvvfQ.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors*, 22(15):5540, 2022. doi: 10.3390/s22155540.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping, 2023. URL https://arxiv.org/abs/2305.10721v1. arXiv:2305.10721v1.

Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, and Bin Yang. TSFM-Bench: A comprehensive and unified benchmark of foundation models for time series forecasting. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 5595–5606, 2025. doi: 10.1145/3711896.3737442.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-MMD: Multi-domain multimodal dataset for time series analysis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 77888–77933, 2024. doi: 10.52202/079017-2476.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models, 2025. URL https://arxiv.org/abs/2502.00816. arXiv:2502.00816.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2211.14730.

David A. Nix and Andreas S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, volume 1, pp. 55–60, 1994a. doi: 10.1109/ICNN.1994.374138.

David A. Nix and Andreas S. Weigend. Learning local error bars for nonlinear regression. In *Advances in Neural Information Processing Systems*, volume 7, pp. 489–496, 1994b. URL https://papers.nips.cc/paper/1994/hash/061412e4a03c02f9902576ec55ebbe77-Abstract.html.

Liran Nochumsohn, Raz Marshanski, Hedi Zisling, and Omri Azencot. Super-Linear: A lightweight pretrained mixture of linear experts for time series forecasting. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=av8niDGYMk.

Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2):884–900, 2023. doi: 10.1016/j.ijforecast.2022.03.001.

Pedro Pessoa, Paul Campitelli, Douglas P. Shepherd, S. Banu Ozkan, and Steve Pressé. Mamba time series forecasting with uncertainty quantification. *Machine Learning: Science and Technology*, 6(3):035012, 2025. doi: 10.1088/2632-2153/adec3b. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC12281171/.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17(9): 2363–2377, 2024. doi: 10.14778/3665844.3665863.

Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. Quantifying point-prediction uncertainty in neural networks via residual estimation with an I/O kernel. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=rkxNh1Stvr.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868, 2021a. URL https://proceedings.mlr.press/v139/rasul21a.html.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations (ICLR)*, 2021b. URL https://arxiv.org/abs/2002.06103.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards foundation models for probabilistic time series forecasting, 2024. URL https://arxiv.org/abs/2310.08278v3. arXiv:2310.08278v3.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2203.09168.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1701.06538.

Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. Faithful heteroscedastic regression with neural networks. In *International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5593–5613, 2023. URL https://proceedings.mlr.press/v206/stirn23a.html.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021. URL https://arxiv.org/abs/2107.03502.

Yijun Wang, Qiyuan Zhuang, Larysa Marchanka, and Xiu-Shen Wei. Beyond static uncertainty: Modeling temporal uncertainty dynamics for probabilistic time series forecasting, 2026. URL https://arxiv.org/abs/2603.24254. arXiv:2603.24254.

Jiawen Wei, Lan Jiang, Pengbo Wei, Ziwen Ye, Teng Song, Chen Chen, and Guangrui Ma. TARFVAE: Efficient one-step generative time series forecasting via TARFLOW based VAE, 2025. URL https://arxiv.org/abs/2511.22853. arXiv:2511.22853.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Kieran Wood, Stefan Zohren, and Stephen J. Roberts. DeRegiME: Deep regime mixtures for probabilistic forecasting under distribution shift, 2026. URL https://arxiv.org/abs/2605.19231. arXiv:2605.19231.

Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. K^2VAE: A Koopman-Kalman enhanced variational AutoEncoder for probabilistic time series forecasting. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67562–67583, 2025. URL https://proceedings.mlr.press/v267/wu25w.html.

Xingjian Wu, Jianxin Jin, Wanghui Qiu, Peng Chen, Yang Shu, Bin Yang, and Chenjuan Guo. Aurora: Towards universal generative multimodal time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.22295.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11559–11569, 2021. URL https://proceedings.mlr.press/v139/xu21h.html.

Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38707–38727, 2023. URL https://proceedings.mlr.press/v202/xu23r.html.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72112–72130, 2025. URL https://proceedings.mlr.press/v267/ye25i.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. ProbTS: Benchmarking point and distributional forecasting across diverse prediction horizons. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1523.

Vincent Zhihao Zheng and Lijun Sun. Multivariate probabilistic time series forecasting with correlated errors. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL https://arxiv.org/abs/2402.01000.

Appendix A describes the models and their training. Appendix B gives scoring and cost details; Appendix C collects the benchmark results. Appendix D examines predictive scale and residual shape, and Appendix E discusses related work.

A MODEL AND TRAINING DETAILS

Table 5 identifies the forecasting components and fitting regime of each system. The following subsections describe the expert banks, their context and horizon choices, the pretrained attachments, and the in-domain ensemble.

Table 5: Systems and training regimes. "Analytic" denotes an issued distribution that requires no predictive samples. Parameter counts describe the recorded bank or loaded forecasting components, at $H = 720$; LH denotes a linear error map with $L \times H$ weights. The in-domain count is read off the released weights by `tools/count_params.py` and depends on the horizon only: 3.79M at $H = 96$, 4.78M at 192, 6.26M at 336 and 10.21M at 720, the spread across the six datasets at a fixed horizon being under 0.02% because the experts are channel-independent with shared parameters. The borrowed variant retains the bank checkpoint. The deterministic attachments bypass the flow decoder. The GETUP-POINT row describes its source-trained use; the mixture experiment fits its GETUP-POINT control on target history.

System	Point / error model	Parameters (M)	Inference	Head fitting
GETUP-ZS RU	Separate banks	74.13	analytic	source corpus
GETUP-ZS DAU	Shared bank, paired heads	37.19	analytic	source corpus
GETUP-ZS+AURORA	AURORA point / borrowed bank	$10.90 + 74.13$	analytic	source corpus
GETUP-POINT	AURORA point / refitted head	$10.90 + LH$	analytic	source corpus
GETUP-MIX	AURORA samples / ten heads	$10.90 + 10\,LH$	10 samples	target history
GETUP-ID	Point ensemble / error ensemble	10.21	analytic	target history
AURORA	Flow model	210.86 (10.90 active)	100 samples	pretrained
TARFVAE	Generative model	1.80	200 samples	target history
VolDy-VAE	Generative model	5.13	100 samples	target history

A.1 EXPERT BANKS AND RESIDUAL SUPERVISION

The reported banks contain $N = 51$ experts: 37 frequency-specific experts, two simple forecasting rules, and 12 complementary experts. They process channels independently with shared parameters. The learned linear predictors follow RLinear (Li et al., 2023); instance normalisation and rescaling return forecasts to the series' original units (Kim et al., 2022). These choices specify the implementation used in the experiments; the gated construction in Section 2.2 does not require a particular expert architecture or bank size.

Routing uses the first half of a fixed-length Fourier transform of the mean-centred history, normalised as in Equation 5. During training, Gaussian noise is added to the expert scores before selection, following the spectral gate of Super-Linear (Nochumsohn et al., 2026). At inference, selection is deterministic as specified in Equation 4. The point bank and its gate are frozen before fitting the error branch.

RU uses separate point and error banks; DAU uses paired heads in one bank. At $H = 720$, their parameter counts are 74.13M and 37.19M, respectively. RU supplies the main self-contained bank results. The experiment identifiers V1 and V2 refer to RU and DAU. In figures and existing result tables, σ denotes the issued scale s; Figure 1 uses s for the scale of the selected variant.

The uncertainty variants differ in both their training targets and their aggregation rule. RU first fits the point bank and its gate, then freezes both. Its error experts are trained on $|y - \bar{\mu}(x)|$, and its error gate combines their outputs using the squared loss on that same ensemble residual. DAU freezes the point branch and fits each error head to its paired expert's residual $|y - \mu_i(x)|$. Its scale additionally includes the disagreement term in Equation 7. Figure 3 shows which fixed prediction supplies each residual target.

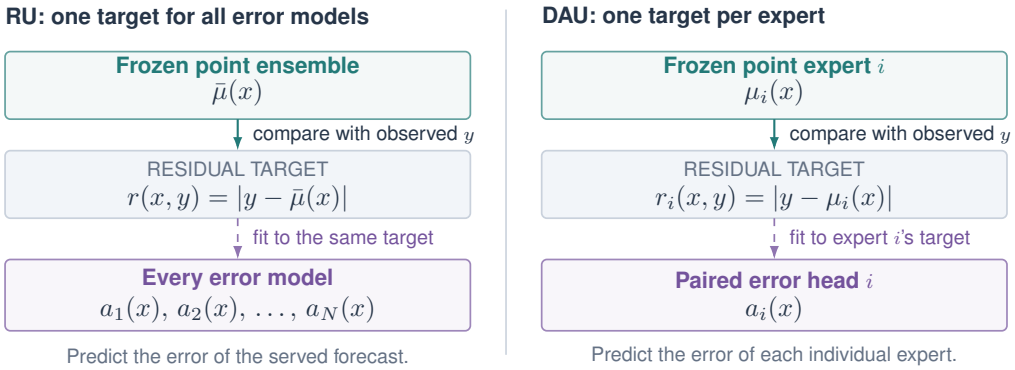

Figure 3: Residual supervision for the two uncertainty variants. Solid arrows construct targets from observed outcomes; dashed arrows indicate fitting. The point branch remains frozen throughout this stage.

A.2 Configuration as recorded in the released code

Every value below is the default in the released configuration file rather than a number typed into this paper, and each is named with the file that holds it, so that a reader can check the description against the code instead of trusting it. `experiments/h27_superlinear/sl_config.py` sets the bank: the expert list is 37 frequency-specific predictors together with a mean and a persistence rule, and 12 complementary experts are added, which is the $N = 51$ of Section 2.2; the layer type is RLinear; both gates retain $k_\mu = k_s = 12$ experts and both temperatures are $\tau_\mu = \tau_s = 1$, so no temperature was tuned; the training-time gate noise has standard deviation 0.1 and is absent at inference; instance normalisation divides by $\sqrt{\text{var} + 10^{-5}}$; and the scale floor is zero, so the issued scale is never clipped from below by a batch statistic. The spectral summary comes from a 5000-point transform of which the first half is retained, so each gate is one linear map from 2500 inputs to the expert scores, and the error gate is a second map of the same shape rather than a re-use of the first (`fission_moe.py`). Pretraining geometry is $L = 512$ and $H = 96$; the horizon-specific banks reported in Table 9 are trained at their own H.

The analyses in this paper are entry points in the same repository, each producing the table it is cited for: `experiments/h27_superlinear/serving.py` assembles a row under a named serving rule and refuses rules that leak evaluation information, which is where *smallest2* is defined; `experiments/h27_superlinear/gate_ablate.py` produces the gate ablation of Appendix C.2 and writes the two tracked CSV files it cites; `experiments/h27_superlinear/dau_terms.py` decomposes the two terms of Equation 7 on the served bank; `tools/probts_ranks.py` computes the mean and rank rows of Table 16 together with the Friedman statistic and the Nemenyi band, from the cells that table prints; `analysis/naive_error_gaussianity.py` produces the persistence-residual diagnostics of Appendix D.3, which involve no trained model; and `analysis/profile_run.py` is the profiler behind Table 8, recording wall time, processor time and peak resident memory per process so that the per-process figures stay comparable when runs share a machine while wall time does not.

A.3 Nested contexts and forecast-head selection

At forecast time t, a context of length L is the suffix $x^{(L)} = (x_{t-L+1}, \ldots, x_t)$. The nested-context implementation evaluates context-specific views of shared weights and recomputes normalisation and the periodogram for each history. Training uses $L \in \{2880, 2112, 576, 528, 360, 96\}$. Error targets are computed from the point forecast at the same context length as the error model, so the target matches the forecast used at inference. Figure 4 illustrates these context views.

Context length specifies the available observations. A separate serving rule selects among output heads trained for different forecast horizons. For a requested horizon H, *largest* always uses the longest available head, *smallest* uses the shortest head that covers H, and *all_covering* averages the

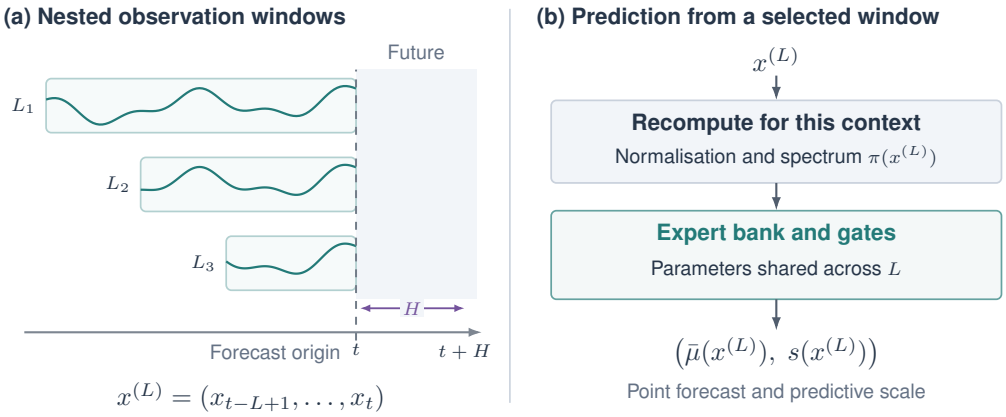

Figure 4: Nested contexts share a forecast origin and learned parameters. Each view recomputes its input statistics before producing an H-step forecast.

forecasts of all heads that cover H. Outputs are restricted to the requested horizon. The primary *smallest2* configuration applies the smallest rule to heads trained at 96 and 720 steps.

The post-hoc *largest_or_floor* rule uses the shortest head for requests below the shortest trained horizon and the longest head otherwise. The corresponding eight-bank ensemble is reported as *sweep8+floor*. Each rule depends on the requested horizon, not the target outcome. However, the post-hoc rule was chosen after inspecting results; we retain this distinction from the preregistered *smallest2* rule when reporting point accuracy.

A.4 Pretrained attachments

The attachment study uses AURORA's frozen deterministic `linear_head`, bypassing its flow decoder. GETUP-ZS+AURORA pairs this point forecast with the RU scale bank trained for GETUP-ZS. GETUP-POINT uses the identical point forecast and fits a new error head to AURORA's absolute residuals on the source corpus. This head is an $L \times H$ linear map followed by softplus. Source windows are normalised before fitting, and predicted scales are returned to the forecast's units at inference. The two attachments consequently have identical MSE and MAE.

Figure 5 shows the two source-trained attachments. The borrowed bank was trained against the bank forecast μ_{bank}, whereas the refitted head learns the residual of AURORA's fixed prediction μ_A. Table 2 compares their probabilistic predictions at this fixed point forecast.

The mixture experiment, GETUP-MIX, uses $M = 10$ trajectories from AURORA and one positive linear error head per sample index. Its heads minimise absolute loss on the absolute-residual targets, whose population optimum is the conditional median absolute error. Both GETUP-MIX and its single-head GETUP-POINT control are fitted on target history in this experiment. This target-fitted control is distinct from the source-trained attachment above.

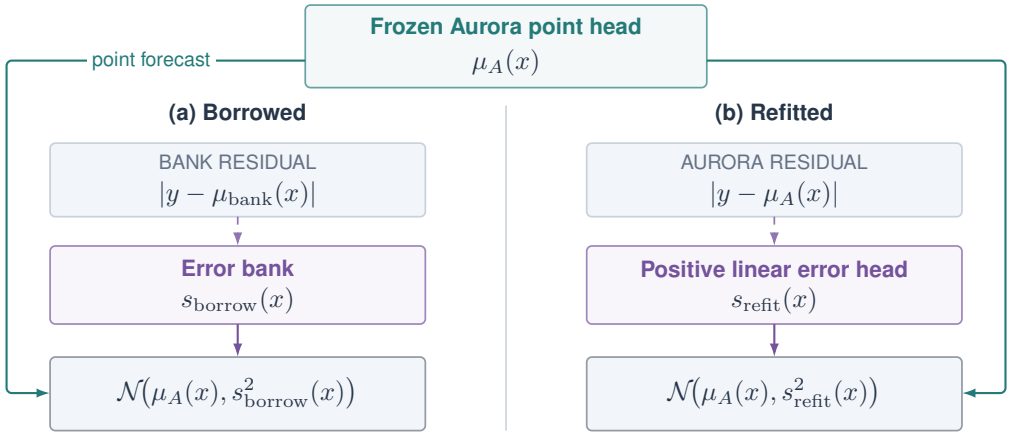

Figure 5: Source-trained attachments: GETUP-ZS+AURORA (a) and GETUP-POINT (b). Dashed arrows identify fitting targets; solid arrows show forecast assembly.

A.5 IN-DOMAIN ENSEMBLE AND EMPIRICAL RESIDUAL SHAPE

GETUP-ID, the in-domain training version, keeps an already fitted point ensemble of $M = 10$ members fixed while fitting uncertainty. Its uncertainty branch holds $K = 2$ ensembles of $P = 10$ value–error pairs each, so twenty pairs in total.

Two properties of this arm are easy to assume wrongly. First, its point ensemble is heterogeneous rather than a repetition of one architecture, and it is *identical* between the offline versions compared in Table 3, which is why their MSE and MAE agree: the versions differ in the error branch alone. The earlier version fills its pairs from a single head family, and the reported version draws them from three families in a fixed proportion—this is the change the 2.3-to-2.5 comparison isolates. The head list of any run is recorded in its checkpoint directory name together with M, K and P, so the composition behind a released bundle can be read off rather than inferred. Second, this arm carries *no* spectral gate: the routing of Section 2.2 belongs to the zero-shot bank, and here the ridge stack described below plays the part of combining error predictors. A gate appears only in the adaptive variant, and there as a test-time rule applied to the offline version's own weights rather than as a retrained model.

Each of the K ensembles is stacked by its own ridge regression onto the absolute error of the reported point forecast. The stack is elementwise over the forecast grid: one independent ridge per horizon step and channel, with penalty 1, the bias left unpenalised, and the prediction clamped at zero because a scale cannot be negative. The stackers are fitted on the training split from the same windows on which the point members were fitted, without cross-fitting.

Let σ_k be the nonnegative output of stacker k, and let $\bar{\mu}_k$ be the mean prediction of that uncertainty ensemble's value heads. Their average is $\bar{\mu}_{\text{v}} = K^{-1} \sum_k \bar{\mu}_k$, which is distinct from the issued point forecast μ_θ. The scale combines the stacked errors with variation between these value predictions:

$$s^2 = \underbrace{\tfrac{1}{K} \textstyle\sum_k \sigma_k^2}_{\text{within ensembles}} + \underbrace{\tfrac{1}{K} \textstyle\sum_k (\bar{\mu}_k - \bar{\mu}_{\text{v}})^2}_{\text{between ensembles}}, \tag{10}$$

At $K = 1$, the second term is zero. This aggregation does not change the external point ensemble's prediction.

The resulting spread is calibrated per channel. For channel c the predicted scales are rescaled to $s_c(x)\, v_c / \overline{s_c}$, where $\overline{s_c}$ is their mean over the calibration windows and v_c is a scale statistic of the point ensemble's *validation* residuals, by default their mean absolute value; the standard deviation and the alternative rescaling $\text{MAE}/\sqrt{\ln 2}$ are the other available choices. The latter is an implemented anchor rule, not in general the minimiser of aggregate CRPS. After this step, the per-channel mean scale on the calibration windows equals v_c by construction. The anchor uses validation observations, not the test outcomes being scored.

After fitting location and scale, the model estimates a smoothed empirical CDF $\widehat{F}_c$ of $(y - \mu_\theta)/s$ for each channel from standardised residuals, jittered by a Gaussian kernel at Silverman bandwidth and represented by at most 4000 anchor points. Validation residuals are pooled with training residuals in the proportion of their window counts. Long horizons can leave few validation windows: Exchange at $H = 720$ leaves 39 in the recorded configuration. Overlapping windows are not independent observations. The offline model holds this CDF fixed at test time (Figure 6). The adaptive variant, recorded as 2.5sc, updates the residual-shape component as observations become available, and in the offline evaluation path it also rescales the issued scale by the selected multiplier; in the benchmark-harness path the multiplier stretches the law about its median while retaining the supplied scale parameter. Both paths retain the point forecast; either can change the predictive distribution's variance. We distinguish them when reporting adaptive results. The offline predictive CDF is given by Equation 9.

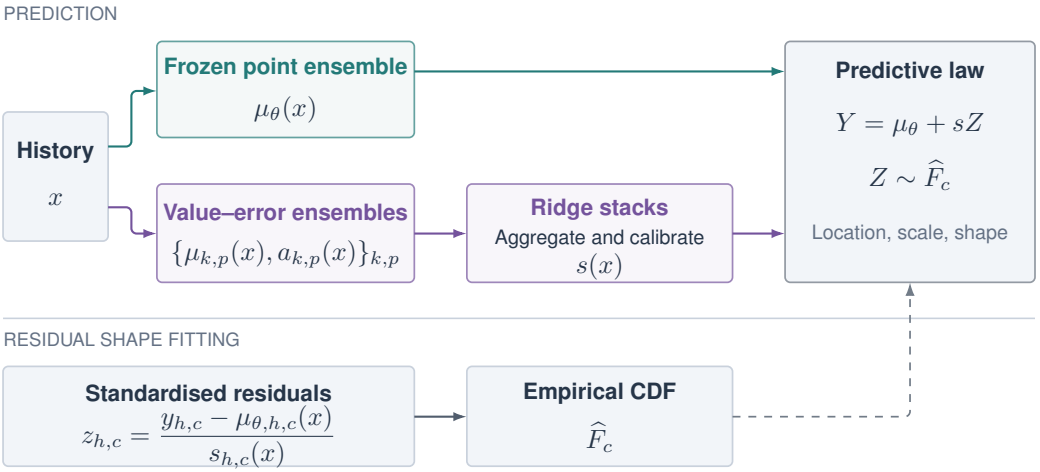

Figure 6: GETUP-ID forecast construction. Value–error ensembles supply the residual estimates and value predictions used in scale aggregation. The external point ensemble supplies the issued forecast. The residual CDF is fitted from training and validation residuals.

Earlier in-domain versions. Versions 2.3 and 2.5 use empirical residual shape with different offline uncertainty configurations. Version 2.4 adds an online gate that mixes the empirical law with a Gaussian, while 2.5sc adds a centred, scaled mixture gate to the offline 2.5 configuration described above. All four share the same point branch, which is why their MSE and MAE coincide in Table 3.

B Evaluation and Computational Cost

This section specifies the fitting regimes, comparison sets, scoring rules, and computational measurements used in Section 3. The benchmark grids in Appendix C and the uncertainty analyses in Appendix D state any experiment-specific differences.

B.1 Training Regimes and Comparison Sets

We organise the experiments by the data used to fit the error model. In the zero-shot setting, the point and error models are fixed before evaluation on the target dataset. In the in-domain setting, they use the target's historical training data. The mixture experiment uses a frozen pretrained point model but fits error heads on target history. The adaptive in-domain variant additionally updates its residual distribution during evaluation. Table 5 in Appendix A lists these regimes, their parameter counts and their inference requirements.

Comparison sets. The TFB (Qiu et al., 2024) point comparison contains 36 dataset–horizon cells: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Wind, Electricity and Traffic at $H \in \{96, 192, 336, 720\}$,

and NYSE at $H \in \{24, 36, 48, 60\}$. The controlled Gaussian pairing experiment uses the same 36 cells and identical windows within each cell; the direct comparison with the moments of AURORA's sampled forecasts also contains 36 paired cells. The expanded ProbTS (Zhang et al., 2024) grid contains 36 cells: the four ETT datasets, Weather, Electricity, Traffic and Exchange at $H \in \{96, 192, 336, 720\}$, and Illness at $H \in \{24, 36, 48, 60\}$. The bank is evaluated on all 36 cells; other systems use the available paired subset, identified by the denominator in each comparison. The in-domain comparison uses six datasets, four horizons and three seeds, giving 72 dataset–horizon–seed cells at $L = 96$ under the ratio-split protocol. Published baseline results retain their source protocol; our re-evaluations use saved forecasts and state the predictive representation being scored. Headline relative changes are ratios of grand means over paired cells; where a figure is instead an average of per-cell ratios, with equal weight on every cell, the caption says so. The two differ when cells have different scales. Missing cells are marked — and do not enter an aggregate. Appendix C gives the full grids; the following subsections give the remaining protocol detail.

The mixture evaluation comprises a 36-cell comparison with a single-head control and a separate five-cell experiment that also includes a moment-matched Gaussian. The latter isolates the effect of residual shape and uses five datasets at $H = 96$; its protocol is reported separately in Appendix D.1.

B.2 Rank comparisons on complete cells

Figure 2 uses the CRPS entries in Table 16. For each fitting regime, we retain a dataset–horizon cell only when every method in that panel has a numerical score. Both panels retain the same 20 cells: ETTm1, ETTm2, ETTh1, ETTh2 and Weather at $H \in \{96, 192, 336, 720\}$. We rank methods separately within each cell and average those ranks, assigning average ranks to ties in the printed scores. The table's dataset-average rows are not additional observations.

The connecting bars use a conservative dataset-block Nemenyi reference, following the rank-comparison framework of Demšar (2006). We average the four horizon-specific ranks within each dataset and use $\mathrm{CD} = q_{0.90;k,\infty}\sqrt{k(k+1)/(6D)}/\sqrt{2}$, where q is a studentized-range quantile, k is the number of methods and $D = 5$ is the number of dataset blocks. The resulting thresholds are 4.3066 for the eight source-trained methods and 3.0628 for the six target-fitted methods. The nominal comparison level is 0.10. The omnibus check permutes method labels jointly across each dataset's horizons, using $200\,000$ permutations and a plus-one probability estimate; both panels reject at this level before the connecting bars are formed.

This is an exploratory reference, rather than an exact clustered Nemenyi test. Using five blocks avoids counting the horizons as independent datasets, but the hourly and minute ETT resolutions are themselves related. Grouping the two resolutions of each ETT source into one block leaves three blocks and increases the reference thresholds to 5.5598 and 3.9540. Rounded scores, the small dataset collection and this remaining dependence limit the strength of pairwise conclusions. A connecting bar indicates non-rejection under the stated reference, not equivalent forecasting performance.

What each experiment measures, in one place. Table 6 states, for every experiment in the paper, the data it uses, what is held fixed, what is fitted and on which data, and where the scores are read. Any deviation from this table is stated in the caption of the table that reports it.

B.3 Proper scores

MSE and MAE evaluate the point forecast μ alone. CRPS and negative log-likelihood evaluate the predictive distribution through proper scoring rules (Gneiting & Raftery, 2007). For a distribution F and observation y,

$$\mathrm{CRPS}(F, y) = \mathbb{E}|X - y| - \tfrac{1}{2}\mathbb{E}|X - X'|, \qquad X, X' \overset{\text{iid}}{\sim} F, \tag{11}$$

which is strictly proper on distributions with finite first moments. For a Gaussian forecast with fixed mean, its limit as the scale approaches zero is the absolute prediction error. This makes its response to a collapsing scale different from that of the logarithmic score: NLL is

$$\mathrm{NLL}(y; \mu, s) = \tfrac{1}{2}\log(2\pi) + \log s + \frac{(y-\mu)^2}{2s^2}, \tag{12}$$

Table 6: Inventory of the experiments. "Cells" counts dataset–horizon combinations unless seeds are named. "Fixed" is what is identical across the compared rows, so that the difference between them is attributable. A single pretrained bank serves every zero-shot cell, so those rows carry no seed replication; where seeds exist, their number is given.

Experiment	Data and cells	Fixed	Fitted on	Reported in
Controlled attachment	TFB set, 36 cells	AURORA point head	source corpus	Table 2
Point accuracy	TFB set, 36 cells	serving rule *smallest2*	source corpus	Table 10
ProbTS CRPS grid	9 datasets $\times$ 4 horizons	benchmark windows	per column	Table 16
In-domain versions	$6 \times 4 \times 3$ seeds, L=96	point ensemble	target train split	Table 3
Historical scale, val. fit	6×4 cells, 24	GETUP-ID point forecast	target validation	Table 4
Per-cell law grids	8×4 declared, 2 contexts	point forecast of each row	first half of test bundle	Tables 18–22
Mixture, expanded	36 cells	AURORA trajectories	target history	Table 25
Mixture, shape control	5 datasets at H=96	first two moments	256 windows per cell	Table 24
Gate ablation	6 datasets H=96, 5 at H=720	trained error experts	nothing refitted	Table 14
Irregular sampling	ETTm1, ETTm2, Weather, 3 seeds	UFO's dropped days	target train split	Table 23
Latency	ETTh2 at H=96, batch 1	window set, thread tuning	—	Table 7
Recorded cost	ETTh2 at L=H=96	loader and harness	—	Table 8

which diverges as s approaches zero for a nonzero residual. Both are proper; neither alone describes every calibration property of a forecast. We report NLL differences in nats, since NLL can cross zero. The mixture CRPS has a closed form (Grimit et al., 2006), given below. The ProbTS results use the benchmark's normalised quantile-loss approximation to CRPS at 19 quantiles; we retain its metric name and report its values in separate tables.

B.4 CALIBRATION DIAGNOSTICS

Coverage measures the fraction of observations within a central predictive interval; we report cov_{90} and compare it with 0.900. PIT-KS is the Kolmogorov–Smirnov distance between probability-integral-transform values and the uniform distribution. For Gaussian forecasts we also report

$$c^* = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\frac{y_j - \mu_j}{s_j}\right)^2}, \tag{13}$$

the scale multiplier that minimises Gaussian NLL when means and relative scales are fixed; its second-moment reference is 1 when s is the predictive standard deviation. The error target in Equation 2 requires a separate conversion: for conditionally centred Gaussian residuals, $\mathbb{E}[|e| \mid x] = s_0\sqrt{2/\pi}$, so an exact mean-absolute-error head needs $s_0 = \sqrt{\pi/2}\,a$ with $\sqrt{\pi/2} = 1.2533$. This is a reference for the regression target, not a different definition of calibrated uncertainty. Error-ranking AUC uses the issued scale to classify whether an absolute error belongs to the largest decile within a cell; it measures ordering and is unchanged by a strictly increasing rescaling. Expected normalised calibration error (ENCE) compares predicted variance with squared error within $K = 100$ bins of predicted scale (Levi et al., 2022). The ENCE formula and oracle $@c$ diagnostics are given below.

B.5 PREDICTIVE REPRESENTATIONS

For forecasters evaluated through sampled trajectories, we distinguish a Gaussian fitted to the sample moments from scores estimated directly from the samples. A Gaussian approximation can substantially change the log score: on ETTm2 at $h = 96$, 0.013% of prediction elements account for 75.8% of VolDy-VAE's total Gaussian NLL, with a minimum sample standard deviation of 4×10^{-6}. The per-cell sample-law grids therefore estimate density from 100 predictive samples using a Gaussian kernel, with the same Silverman bandwidth rule and lower bound used for our empirical residual shape. Coverage and PIT are computed from sample quantiles and ranks. Appendix B.7 specifies these estimators; the tables identify which predictive representation is scored.

The substitution is partial, and the part that remains matters. In the per-cell law grids the likelihood, coverage and PIT columns use the sample estimate, whereas CRPS and ENCE are computed from the sample mean and spread, because both quantities are defined on moments in the implementation that writes those dumps. For CRPS this keeps each model's own arm comparable with its own control, which is what those grids test, and makes the column incomparable between a sampled

baseline and a model whose law is an estimated shape. The size of that incomparability is recorded: against the benchmark evaluator's own normalised score on the same runs the Gaussian-moment value differs by -51% to $+68\%$ across the 48 VolDy-VAE cells, the sign fixed by the dataset, with the law, the window subset and the evaluator's 19-quantile approximation all differing at once (`results/voldy_crps_check.csv`). Scoring those rows from the trajectories would require re-running the dumps, which the recorded wall times put at 53 hours of inference; the cross-model CRPS comparison is therefore the published one in Table 16.

B.6 Mixture scoring and oracle scale adjustment

Confidence intervals in the point summaries are the reported paired-cell bootstrap intervals; we distinguish averages over paired cells from ratios of pooled scores.

For the Gaussian mixture in Equation 8, CRPS has a closed form (Grimit et al., 2006). Define $A(m, s) = m\{2\Phi(m/s) - 1\} + 2s\varphi(m/s)$, where Φ and φ are the standard normal CDF and density. Then

$$\mathrm{CRPS}\left(\sum_i w_i \mathcal{N}(\mu_i, s_i^2), y\right) = \sum_i w_i A(y - \mu_i, s_i) - \frac{1}{2}\sum_i \sum_j w_i w_j A\left(\mu_i - \mu_j, \sqrt{s_i^2 + s_j^2}\right). \tag{14}$$

This removes numerical sampling error from evaluation of a realised mixture; the mixture itself remains random because its centres come from the sampler. The implementation checks that identical components recover the single-Gaussian formula.

ENCE compares predicted variance with squared error within bins of predicted scale (Levi et al., 2022):

$$\mathrm{ENCE} = \frac{1}{K}\sum_{b=1}^{K} \frac{|\mathrm{RMV}_b - \mathrm{RMSE}_b|}{\mathrm{RMV}_b}, \qquad \mathrm{RMV}_b = \sqrt{\overline{s^2}_b}, \quad \mathrm{RMSE}_b = \sqrt{\overline{(y-\mu)^2}_b}, \tag{15}$$

with $K = 100$ bins defined by ranks of the predicted scale, which avoids empty quantile bins when predicted scales coincide. For a mixture, s^2 is its total predictive variance. For the median-absolute-error target used in the mixture experiment, the Gaussian conversion analogous to $\sqrt{\pi/2}$ is $1/\Phi^{-1}(0.75) \approx 1.4826$.

The columns marked $@c$ evaluate each law after fitting a single dilation factor to its scoring observations. A dilation about the predictive mean multiplies both component deviations from that mean and component scales by c, chosen to minimise NLL: analytically by Equation 13 for a Gaussian and numerically for a mixture. These columns describe performance under an oracle global scale adjustment and are not estimates of deployable calibration.

B.7 Estimating scores from predictive samples

For sampled baselines, we estimate a density from the predictive samples rather than fitting a single Gaussian to their moments. Let $x_1, \ldots, x_S$ be the S samples at one forecast step and channel, $\hat{\sigma}$ their standard deviation, and IQR their interquartile range. We use the Gaussian kernel bandwidth

$$b = \max\left\{0.9 \min\left(\hat{\sigma}, \frac{\mathrm{IQR}}{1.34}\right) S^{-1/5},\ 10^{-3}\right\}, \tag{16}$$

and evaluate

$$\hat{p}(y) = \frac{1}{Sb}\sum_{j=1}^{S} \varphi\left(\frac{y - x_j}{b}\right). \tag{17}$$

Here φ is the standard normal density. This uses the same bandwidth rule and lower bound as our empirical residual-shape estimator. The bandwidth floor prevents a zero-width kernel when samples coincide; the density remains an estimate of the sampler's distribution.

PIT values use the observation's rank among the samples, and interval coverage uses sample quantiles. These estimates are computed during forecast export, while the trajectories are available, without storing every sample for every grid cell. Section 4 discusses the finite-sample and bandwidth dependence of these diagnostics.

B.8 RECORDED TRAINING AND INFERENCE COSTS

Table 7: Single-window CPU inference on ETTh2 at $H = 96$, batch size 1: 200 timed calls after 10 warm-up calls, excluding model construction and checkpoint loading. Every row is that model's best of the thread counts tried, and the count that achieved it is printed in the row label, so no row is penalised by a setting chosen for another. An "analytic" forecast requires no predictive samples; a sampled row reports the cost of the sample count its own published protocol uses, and drawing fewer samples would lower both its cost and its scores. The AURORA parameter count includes its unimodal forecasting core. Bold marks the lowest latency in a column and underlining the second lowest. Parameter counts are not ranked, and for the GETUP-ID rows they are not on one convention either: the latency harness counts a GETUP-ID bundle's point ensemble and reaches no further, so the v1 and 2.3 counts are point-ensemble only, while GETUP-ID's 3.79M is the whole released bundle—point ensemble plus both error ensembles—read off the stored weights by `tools/count_params.py`. The two are therefore not comparable with each other, and the larger figure is ours. The other rows count every parameter of a single module and need no such qualification. Parameter counts describe the systems rather than comparing them, and the outputs differ in kind.

Model	Params (M)	Output	wall p50 (ms)	wall p99 (ms)	CPU (ms)
GETUP-ZS (4 thr)	74.13	analytic	4.556	4.919	4.675
GETUP-ID v1 (1 thr)	4.50	analytic	<u>4.880</u>	<u>5.251</u>	<u>4.909</u>
GETUP-ID 2.3 (1 thr)	2.94	analytic	9.033	9.569	9.049
TARFVAE (8 thr)	1.80	200 samples	9.661	10.443	18.868
VolDy-VAE (1 thr)	5.13	100 samples	51.307	60.195	52.258
AURORA (4 thr)	10.90	100 samples	104.410	105.175	164.162
GETUP-ID	3.79	analytic	—	—	—
GETUP-ZS+AURORA / GETUP-POINT	End-to-end attachment latency unmeasured				

Table 8 records the task-specific runs and the pretrained-bank inference sweep. These configurations differ from the batch-one comparison in the main text. Table 9 reports source-pretraining wall times separately.

Parameter count describes a separate resource. The bank has 74.13M parameters against 10.90M in AURORA's unimodal forecasting core; of AURORA's 210.86M total, 197.85M belong to text and vision encoders and 2.11M to associated modality components, none of which execute in the unimodal run.

Recorded in-domain training costs are 1361 CPU-s for GETUP-ID-Small, 4078 for GETUP-ID-Base, 4416 for TARFVAE and 29007 for VolDy-VAE on ETTh2 at $L = H = 96$.

Table 8: Recorded training and inference costs. In-domain runs use ETTh2 at $L = H = 96$; the pretrained bank's inference sweep uses $L = 512$, batch size 8 and 8 threads. CPU time is user plus system time, and RAM is peak resident memory. Test loaders produce 3389 windows for GETUP-ID and 2785 for TARFVAE and the bank, so the per-window columns are comparable while the totals are not. A dash is never a zero and never "not applicable"; each one has a reason, and the reasons differ. The bank's training columns are empty because a pretrained model is not trained per cell: its source-pretraining wall times are Table 9, and dividing them over target tasks would be a choice of denominator rather than a measurement. AURORA's training row cannot be filled from this work at all, since the corpus and the training entry point are not part of its release; its inference row is measurable under the present harness and is not yet measured. The GETUP-ID-Small and GETUP-ID-Base rows are earlier in-domain configurations of two capacities, not the released GETUP-ID of the row below them: the thirteenfold gap between their inference cost and GETUP-ID's is a gap between different systems, and the macro that names them prints the same family label. GETUP-ID's inference row is measured with the same entry point as the rows above it, on 2785 windows of ETTh2 at $L = H = 96$ with batch size 8 and 8 threads; its training columns are empty because its bundles were trained by the grid driver, which records no separate per-cell CPU time, so the total cannot be formed either. The GETUP-POINT row is the recorded attachment sweep (`logs/cost_bagx.txt`); it serves at $L = 528$ rather than 96, and that context is part of its cost. Per-window figures for every model under a different harness—single windows at batch 1 rather than throughput at batch 8—are in Table 7, and the two columns are not interchangeable. The remaining dashes here, VolDy-VAE's and AURORA's throughput and GETUP-ID's training, are measurements this harness can make and we have not made.

	Training		Inference			
Model	CPU-s	Peak RAM (GB)	CPU-s	ms/window	Peak RAM (GB)	Total (CPU-h)
In-domain: trained on the task, one training per cell						
GETUP-ID-Small	1361	1.33	11.7	3.5	0.51	0.38
GETUP-ID-Base	4078	2.60	26.2	7.7	0.53	1.14
GETUP-ID (L=96)	—	—	281.9	101.2	0.63	—
TARFVAE	4416	0.67	182.2	65.4	0.76	1.28
VolDy-VAE	29007	0.54	—	—	—	8.06
Zero-shot: pretrained once, then serves every cell without retraining						
GETUP-ZS (bank)	—	—	34.4	12.34	0.94	—
GETUP-POINT (L=528)	—	—	9614.7	3452.3	1.28	2.67
AURORA	—	—	—	—	—	—

Table 9: Recorded source-pretraining wall time for 51-expert banks. Parameter counts refer to $H = 720$. V3 is an additional recorded bank variant, retained for completeness. The bank is pretrained on a corpus of 12 848 series.

Variant	H=96	H=192	H=336	H=720	Params at H=720
RU (two banks)	113 s	6734 s	7119 s	7766 s	74.13M
DAU (one bank)	146 s	2266 s	2495 s	3307 s	37.19M
V3	2995 s	7364 s	7448 s	7886 s	69.97M

C DETAILED BENCHMARK RESULTS

The results are grouped by evaluation regime: source-trained forecasts, in-domain forecasts, and irregular sampling. The tables retain the configuration names and protocol distinctions used in the main text. A dash denotes an unmeasured result. A dagger denotes a result withheld by the source paper because that dataset appears in the model's pretraining data. The point grid marks our pretraining datasets with an asterisk and excludes them from the zero-shot aggregate. Numbers are comparable within the stated protocol; the same dataset and horizon in another block can use different windows, fitting data or scores.

C.1 ZERO-SHOT POINT FORECASTS AND PRETRAINED ATTACHMENTS

Table 10 summarises the point-forecast comparisons and Table 11 gives their per-cell scores. Tables 12 and 13 report the Gaussian comparison with AURORA's sample moments.

In the fixed-point comparison of Table 2, GETUP-POINT has coverage closer to the nominal 90% than the borrowed-head control GETUP-ZS+AURORA in 32 of 36 dataset–horizon settings. Both methods use the same deterministic point forecasts, so this comparison isolates the error model.

Table 10: Mean relative MSE differences on 36 TFB cells. Positive values indicate higher MSE than the reference. The left block uses published AURORA results; the right block uses our deterministic-head run. Intervals are the reported 95% paired-cell bootstrap intervals. Bold marks the one GETUP-ZS entry with lower MSE than the AURORA run it is compared against; every other configuration is worse than that run on every subset. The GETUP-POINT row repeats our AURORA run because it attaches a scale to that run's point forecasts. The nested-horizon row is empty in the right block because that configuration was scored against the published AURORA numbers only and never re-scored against our own run.

	vs AURORA published			vs AURORA as we ran it		
Configuration	MSE %	95% CI	win	MSE %	non-NYSE	NYSE
AURORA (our run)	-1.59	$[-3.87, +0.50]$	$24/36$	0.00	0.00	0.00
GETUP-ZS *smallest2* (pre-registered)	$+2.07$	$[-0.20, +4.20]$	$12/36$	$+4.45$	$+0.76$	$+33.91$
GETUP-ZS *sweep8+floor* (post-hoc)	$+1.02$	$[-1.30, +3.20]$	$15/36$	$+3.40$	-0.41	$+33.91$
GETUP-ZS *smallest*	$+2.69$	$[+0.50, +4.70]$	$10/36$	$+5.06$	$+1.46$	$+33.91$
GETUP-ZS DAU *smallest2*	$+2.75$	$[+0.50, +4.90]$	$11/36$	$+5.16$	$+1.32$	$+35.92$
GETUP-ZS *largest*	$+6.57$	$[+1.40, +12.80]$	$12/36$	$+10.04$	$+0.40$	$+87.15$
GETUP-ZS nested horizons	$+6.28$	$[+2.10, +11.10]$	$10/36$	—	—	—
GETUP-POINT	-1.59	$[-3.87, +0.50]$	$24/36$	0.00	0.00	0.00

A single bank trained across nested horizons yields a 6.28% difference, so it does not improve on the two-head configuration. A separate tokenisation ablation fixes AURORA's patch length at 48 instead of twice the dominant period, which increases mean MSE by 12.69%, with a 93% increase on ETTm1 at $H = 336$; the context and tokenisation settings are therefore part of the evaluated configuration.

Against a Gaussian fitted to the mean and standard deviation of 100 AURORA samples on identical windows, GETUP-POINT has lower Gaussian NLL and coverage closer to 0.900 in all 36 paired cells, and lower Gaussian CRPS in 28 (Table 12). Mean NLL falls from 39.2151 to 1.0872, while mean absolute deviation from nominal 90% coverage falls from 0.4290 to 0.0992. This comparison changes the point forecast as well: one arm uses a sample mean and the other the deterministic head. It therefore measures the complete predictive distribution, whereas Table 2 holds point predictions fixed.

In a separate substitution experiment, combining the bank's point forecast with AURORA's sample spread gives Gaussian NLL of 11.39 on ETTh1, 15.59 on Weather and 7.37 on NYSE against 1.17, -0.41 and 2.02 for the borrowed-scale arm.

The two AURORA columns of Table 11 are the reproduction check for every comparison in this paper that holds its point forecast fixed. On the 32 cells outside NYSE they agree to within 0.041 MSE and 0.022 MAE; on the four NYSE cells our run is lower by 0.055 to 0.139 MSE. NYSE is therefore the

Table 11: Point-forecast grid under the TFB implementation of the TSFM-Bench setting. The first AURORA column is transcribed from the source paper; the AURORA/GETUP-POINT column is our own run of its deterministic head, and the attachment changes uncertainty only, so those point metrics coincide with GETUP-POINT's; GETUP-ZS+AURORA shares that point forecast and is not printed, its column being a copy of its neighbour by construction. An asterisk marks a dataset that is in our bank's pretraining corpus: a zero-shot figure there would not be zero-shot, and the two such datasets are therefore left empty in our columns rather than filled. Bold marks the lowest value in a row among the three systems compared and underlining the second lowest of those three, the published AURORA column excluded because it and the run beside it are the same system. The last row is the mean over the 36 rows in which every column carries a value.

		AURORA publ.		AURORA / GETUP-POINT		GETUP-ZS		GETUP-ZS V2	
Dataset	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.294	0.351	0.293	0.350	0.295	0.343	0.295	0.343
	192	0.331	0.374	0.330	0.373	0.332	0.367	0.331	0.366
	336	0.359	0.391	0.359	0.390	0.364	0.386	0.363	0.386
	720	0.398	0.414	0.399	0.414	0.423	0.418	0.422	0.418
ETTm2	96	0.179	0.270	0.181	0.271	0.181	0.273	0.181	0.272
	192	0.232	0.307	0.236	0.309	0.231	0.307	0.232	0.307
	336	0.275	0.337	0.277	0.338	0.277	0.338	0.280	0.339
	720	0.338	0.380	0.339	0.381	0.347	0.384	0.350	0.386
ETTh1	96	0.340	0.381	0.338	0.378	0.359	0.386	0.359	0.387
	192	0.377	0.405	0.373	0.402	0.399	0.410	0.399	0.410
	336	0.399	0.422	0.396	0.420	0.428	0.425	0.428	0.425
	720	0.428	0.450	0.426	0.449	0.447	0.453	0.448	0.454
ETTh2	96	0.259	0.325	0.258	0.324	0.289	0.345	0.289	0.346
	192	0.324	0.370	0.322	0.369	0.342	0.383	0.341	0.381
	336	0.360	0.401	0.358	0.401	0.371	0.408	0.373	0.408
	720	0.403	0.441	0.402	0.440	0.395	0.433	0.396	0.433
Weather	96	0.160	0.207	0.164	0.220	0.155	0.210	0.155	0.209
	192	0.202	0.247	0.205	0.256	0.197	0.250	0.199	0.251
	336	0.252	0.288	0.249	0.288	0.243	0.285	0.244	0.285
	720	0.307	0.327	0.304	0.325	0.306	0.329	0.307	0.327
Electricity	96	0.134	0.234	0.133	0.235	0.155	0.250	0.155	0.250
	192	0.161	0.258	0.161	0.259	0.164	0.259	0.164	0.259
	336	0.193	0.287	0.193	0.288	0.181	0.276	0.181	0.276
	720	0.224	0.320	0.265	0.342	0.225	0.314	0.226	0.314
Traffic	96	0.435	0.314	0.436	0.313	0.488	0.341	0.488	0.341
	192	0.465	0.328	0.466	0.327	0.479	0.334	0.478	0.334
	336	0.525	0.355	0.525	0.355	0.495	0.341	0.494	0.340
	720	0.670	0.411	0.666	0.411	0.535	0.361	0.534	0.361
Wind	96	0.951	0.664	0.942	0.662	0.943	0.653	0.945	0.655
	192	1.115	0.747	1.105	0.743	1.054	0.730	1.094	0.737
	336	1.229	0.799	1.223	0.796	1.198	0.795	1.261	0.811
	720	1.309	0.840	1.313	0.840	1.310	0.848	1.379	0.864
NYSE	24	0.267	0.373	0.212	0.339	0.293	0.415	0.298	0.418
	36	0.420	0.477	0.349	0.439	0.479	0.542	0.487	0.548
	48	0.605	0.574	0.492	0.522	0.662	0.646	0.673	0.654
	60	0.820	0.681	0.681	0.622	0.855	0.743	0.864	0.751
Solar*	96	0.185	0.272	—	—	—	—	—	—
	192	0.198	0.282	—	—	—	—	—	—
	336	0.211	0.294	—	—	—	—	—	—
	720	0.218	0.307	—	—	—	—	—	—
PEMS08*	96	0.463	0.513	—	—	—	—	—	—
	192	0.599	0.577	—	—	—	—	—	—
	336	0.560	0.546	—	—	—	—	—	—
	720	0.629	0.570	—	—	—	—	—	—
mean	36	0.4372	0.4097	0.4270	0.4053	0.4416	0.4161	0.4476	0.4179

one dataset on which a comparison against the published system, rather than against our reproduction of it, would read differently.

Table 12: Gaussian scoring on common windows and contexts. The AURORA arm uses the mean and standard deviation of 100 sampled trajectories; GETUP-POINT uses the deterministic point head and its source-trained error model. The table evaluates Gaussian moments, not the native sampled law. Bold marks the better of the two arms in each cell, by the rule the metric declares: lower CRPS, NLL and ENCE, coverage closest to 0.900. Applied by `tools/mark_pairs.py`.

		AURORA Gaussian moments				GETUP-POINT			
Dataset	H	CRPS	NLL	cov_{90}	ENCE	CRPS	NLL	cov_{90}	ENCE
ETTh1	96	0.299	11.059	0.491	3.108	0.281	1.041	0.825	0.526
	192	0.328	13.296	0.463	3.464	0.303	1.104	0.813	0.544
	336	0.321	12.207	0.509	3.305	0.302	1.046	0.838	0.510
	720	0.406	8.156	0.542	2.537	0.381	1.217	0.830	0.494
ETTh2	96	0.197	3.334	0.567	1.776	0.186	0.520	0.767	0.525
	192	0.240	7.217	0.471	2.770	0.222	1.047	0.694	0.840
	336	0.251	8.603	0.396	3.207	0.229	1.206	0.665	0.958
	720	0.193	6.585	0.475	2.687	0.177	0.559	0.775	0.583
ETTm1	96	0.299	217.857	0.163	17.535	0.249	0.607	0.885	0.270
	192	0.299	176.088	0.184	15.058	0.251	0.739	0.851	0.394
	336	0.332	185.975	0.163	15.629	0.276	1.112	0.803	0.568
	720	0.348	198.891	0.172	14.828	0.293	1.231	0.821	0.560
ETTm2	96	0.197	69.814	0.228	8.812	0.163	0.201	0.798	0.313
	192	0.194	77.816	0.245	9.226	0.162	0.221	0.794	0.345
	336	0.194	72.765	0.264	8.926	0.163	0.282	0.800	0.379
	720	0.201	65.590	0.293	8.078	0.170	0.269	0.833	0.297
Weather	96	0.100	11.546	0.550	3.345	0.110	−0.389	0.933	0.178
	192	0.127	13.317	0.534	3.511	0.128	−0.224	0.899	0.182
	336	0.182	16.103	0.523	4.078	0.180	0.300	0.865	0.398
	720	0.318	70.366	0.444	9.045	0.300	5.512	0.740	2.353
Electricity	96	0.279	7.634	0.582	2.950	0.280	0.764	0.920	0.123
	192	0.278	7.916	0.561	3.062	0.281	0.768	0.920	0.140
	336	0.274	6.646	0.531	2.792	0.273	0.751	0.922	0.202
	720	0.303	8.723	0.472	3.201	0.292	0.953	0.900	0.301
Wind	96	0.702	11.763	0.454	3.229	0.613	1.550	0.772	0.427
	192	0.751	16.283	0.418	3.587	0.684	1.995	0.692	0.735
	336	0.931	41.197	0.283	6.645	0.817	2.844	0.593	1.175
	720	0.953	45.188	0.249	7.281	0.817	3.195	0.585	1.297
Traffic	96	0.288	2.133	0.730	1.315	0.309	0.840	0.954	0.206
	192	0.319	3.016	0.700	1.624	0.335	0.925	0.945	0.155
	336	0.301	3.101	0.722	1.561	0.326	0.936	0.950	0.185
	720	0.334	2.930	0.689	1.524	0.350	0.992	0.938	0.111
NYSE	24	0.298	1.336	0.768	0.636	0.265	0.772	0.875	0.293
	36	0.374	2.112	0.716	0.862	0.334	1.136	0.783	0.475
	48	0.430	2.290	0.717	0.871	0.385	1.372	0.732	0.559
	60	0.492	2.895	0.683	1.071	0.443	1.746	0.685	0.743

Table 13: Mean scores and calibration diagnostics for the Gaussian comparison in Table 12. Bold marks the better of the two laws in each column, lower being better throughout, the coverage column already being a distance from 0.900. The final row counts cells favouring GETUP-POINT.

Law	cells	CRPS	NLL	$\lvert\mathrm{cov}_{90}-0.9\rvert$	ENCE
AURORA own sampled law	36	0.3426	39.2151	0.4290	5.0871
GETUP-POINT	36	0.3147	1.0872	0.0992	0.5096
GETUP-POINT ahead, of	36	28/36	36/36	36/36	36/36

C.2 DO THE TWO GATES DO DIFFERENT WORK?

The construction claims that weighting the error experts separately from the point experts is what the second gate is for. This is a serving-time ablation of that claim: the error experts are the trained ones in every arm, and only the weights that combine them change. It measures the effect of replacing the learned weights at inference; a jointly trained single-gate model could learn different error heads, so this experiment does not bound its performance.

Three arms sit beside the served model. *Point weights* keeps the experts the error gate selected and takes the weights from the point gate's logits restricted to that same set, so it isolates the weighting from the selection. *Uniform* weights the same set by $1/k$, which asks whether the gate's weights beat a plain average. *Point gate outright* lets the point gate both select and weight, so it reaches error heads its own gate never routed training data to; it is the loosest of the three.

Table 14: Serving-time gate ablation for the RU bank at $H = 96$ on the benchmark test split. The upper block is $H = 96$ with 512 windows per dataset (215 for NYSE, whose loader drops five rows with missing values); the lower block repeats the experiment at $H = 720$ with 256 windows on the same bank family, where NYSE is out of scope because its horizons stop at 60. Effective expert count is $1/\sum_i w_i^2$ over the served weights, out of $k = 12$ selected; "shared" is the share of windows on which both gates rank the same expert first. The last three columns give ΔCRPS against the served arm, so positive means worse than serving. Produced by `experiments/h27_superlinear/gate_ablate.py` with `--csv`; the tracked artefacts are `results/gate_ablate_H96.csv` and `results/gate_ablate_H720.csv`, which carry every column of this table together with NLL, coverage, AUC, c^* and ENCE for all four arms.

	eff. experts				ΔCRPS, %		
Dataset	point	error	shared	Jaccard	point w.	uniform	point gate
ETTh1	2.285	3.367	21.4%	0.293	+17.28	+8.71	+17.46
ETTh2	4.616	2.990	0.0%	0.259	+16.19	+12.95	+17.39
ETTm1	2.587	5.778	0.0%	0.224	+10.07	−1.05	+8.82
ETTm2	4.966	6.335	0.0%	0.154	+9.13	+3.92	+11.91
Weather	3.280	4.947	56.5%	0.314	+6.68	+7.94	+8.16
NYSE	3.540	5.968	0.0%	0.194	+5.39	+3.81	+8.64
Mean, $H = 96$					+10.79	+6.05	+12.06
$H = 720$, the same bank family at its 720-step checkpoint, 256 windows							
ETTh1	3.130	3.626	15.5%	0.380	+15.73	+8.78	+15.42
ETTh2	5.387	2.835	8.8%	0.345	+13.33	+11.33	+14.21
ETTm1	2.633	3.884	1.0%	0.178	+18.30	+6.77	+14.98
ETTm2	4.712	3.555	6.5%	0.214	+13.39	+10.25	+15.23
Weather	3.426	5.106	20.1%	0.278	+3.26	+5.96	+4.84
Mean, $H = 720$					+12.80	+8.62	+12.94

The two gates route to different experts, at both horizons. At $H = 96$ they never agree on their first-ranked expert on four of the six datasets, agreeing on 21.4% of windows on ETTh1 and 56.5% on Weather; at $H = 720$ the agreement runs from 1.0% to 20.1%. The Jaccard overlap of the two selected sets is 0.154 to 0.380 across both blocks. Neither gate of this two-bank variant is degenerate: the effective expert count stays between 2.285 and 6.335 of the twelve selected. The paired-head variant behaves differently—its error gate is numerically one-hot, with an effective count

of 1.000—so this statement is about the reported two-bank bank and not about the construction in general.

Taking the weights from the point gate costs 10.79% of CRPS on average at $H = 96$ and 12.80% at $H = 720$, 18.30% at worst, and coverage falls on every dataset of both blocks. A plain average of the same experts is cheaper than that but still worse than the gate on five of the six. The exception is ETTm1 at $H = 96$, where uniform weights are better on all four diagnostics at once: CRPS by 1.05%, NLL 3.15 against 6.34, c^* 2.75 against 3.73 and ranking AUC 0.703 against 0.650. That exception does not survive the horizon: at $H = 720$ the same dataset costs 6.77% under uniform weights. Over the eleven dataset–horizon pairs measured here the learned weights are ahead in ten.

Two cells deserve naming. On NYSE the served scale has a ranking AUC of 0.2542, that is it orders errors worse than chance, while the same experts under the point gate's weights reach 0.7249; its coverage is 0.4632 against a nominal 0.900. NYSE is also the column with the largest point-accuracy gap in Table 10. On Weather, the one dataset where the two gates agree most often, the point gate's weights are the cheaper substitution of the two, which is what agreement between the gates would predict.

C.3 In-domain forecasts and predictive distributions

Table 15 compares available ProbTS cells while separating the fitting regimes. Table 16 gives the dataset–horizon CRPS grid, and Table 17 adds point metrics for GETUP-ID. Tables 18–22 then report proper scores and calibration diagnostics under each model's predictive law.

Variation across training seeds. For the ratio-split comparison in Table 3, the observed spread of pooled CRPS across the three seeds is at most 0.0002 for v1 and at most 0.0001 for each later configuration. This comparison averages six datasets and four horizons within each seed.

What the per-cell law grids compare, and what they do not. The declared grid is eight datasets at four horizons and two context lengths, $L = 96$ and $L = 720$. Of the thirty-two cells per context, twenty-four are scored: Electricity and Traffic are absent for the sampled baselines, and the tables mark them —. Every cell uses the test windows of its dataset at unit stride, the count of which is a property of the dataset and horizon rather than a choice, and each model is scored under the law its own stored artefact declares—an estimated residual shape for GETUP-ID, and the kernel estimate of Appendix B.7 for the samplers. Entries are means over the seeds a model has in that cell; the sampled baselines are single-run, so a seed spread is shown only where more than one exists.

Two limits of these grids are not visible in the numbers. First, the rows differ in replication budget rather than in protocol. The competitors' checkpoints follow the calendar split of their own published protocol, and GETUP-ID is retrained on that same split for the four ETT datasets: those cells are read from a separate calendar-split tree covering all four datasets at both context lengths and all four horizons, so the test windows match on both sides. What differs is how many runs a cell rests on: every ETT cell carries two training seeds, while the Exchange and Weather cells carry two or three depending on the cell, so the spreads in this grid are not comparable from row to row. Second, a sampled baseline is scored from the checkpoint with the lowest recorded validation CRPS among all of its completed runs for that cell, which is the most favourable reading of the competitor that its stored artefacts allow; where a run's best epoch is its first, that is what the recorded validation curve says and not an artefact of selection.

Under the ProbTS harness. We additionally evaluate GETUP-ID through the ProbTS harness on six datasets at three seeds, using stage-one bundles trained under the split the benchmark itself applies. The benchmark's CRPS approximation is 0.238, 0.263, 0.280 and 0.295 on ETTh1; 0.138, 0.157, 0.174 and 0.183 on ETTh2; 0.209, 0.229, 0.242 and 0.267 on ETTm1; 0.108, 0.130, 0.147 and 0.167 on ETTm2; 0.073, 0.077, 0.080 and 0.084 on Weather; and 0.018, 0.024, 0.032 and 0.044 on Exchange, at horizons 96, 192, 336 and 720. On Exchange these values are the lowest in the grid at every horizon, averaging 0.029 against 0.040 for the next system. On Weather the average is 0.079 against 0.062 for the zero-shot bank: the in-domain model is beaten there by our source-trained bank. Electricity and Traffic are measured at $H = 96$ and Illness at all four of its horizons; the longer Electricity and Traffic horizons remain unmeasured for GETUP-ID under this harness. On Illness, whose training split is 676 points, GETUP-ID scores 0.171 to 0.202 against 0.123 to 0.177 for our

own zero-shot bank: at that sample size the in-domain arm does not reach the pretrained one, which is a statement about how much history the staged fit needs.

The following expanded grids report the saved evaluations at each dataset, context and horizon. The GETUP-ID arm uses its empirical residual law; the sampled baselines use the estimators in Appendix B.7, except where a caption specifies Gaussian scoring. Each entry is a mean over seeds, and a $\pm$ suffix gives the half-range across seeds; where only one seed exists no spread is shown. These grids differ from the three-seed main comparison: the GETUP-ID arm has two seeds in 56 of 62 cells and one in the remaining six; sampled baselines have one seed each. Their reported spreads therefore describe different replication budgets. — means not measured and *n/c* means the model's authors published no configuration for that cell.

Table 15: Relative improvement (%) of pooled CRPS, computed from the two grand means over paired cells, over each named baseline on available ProbTS cells; positive values favour the row model. Small entries give wins/paired cells. Column groups identify the baselines' training regimes. GETUP-MIX and GETUP-ID use target-fitted components; the other row models are source-trained. Bold marks the entries in which the row model wins, that is every positive value; nothing here is a ranking among columns, since the columns are different baselines.

	ZERO-SHOT					IN-DOMAIN			
	AURORA	Sundial	Chronos	MOIRAI	Lag-Llama	VolDy-VAE	K^2VAE	CSDI	TimeGrad
GETUP-ZS	+7.1 30/36	+8.8 25/32	+26.9 30/32	+43.9 30/32	+24.7 31/32	−5.4 13/36	+5.0 27/36	+34.6 30/30	+50.1 35/36
GETUP-ZS+AURORA	+1.0 21/28	+2.7 17/24	+23.5 24/24	+43.7 24/24	+20.8 24/24	−10.4 5/28	+0.8 13/28	+26.5 18/22	+46.6 24/28
GETUP-POINT	+1.8 20/28	+6.0 19/24	+25.2 24/24	+45.6 24/24	+22.6 24/24	−9.4 8/28	+1.6 16/28	+29.5 20/22	+47.0 22/28
GETUP-MIX	+0.9 19/28	+5.3 18/24	+24.5 24/24	+45.2 24/24	+21.9 24/24	−10.5 7/28	+0.6 14/28	+29.0 19/22	+46.5 22/28
GETUP-ID†	+11.9 20/24	+13.5 24/24	+32.0 24/24	+45.8 24/24	+26.5 24/24	+2.8 15/24	+14.3 24/24	+32.9 22/24	+58.0 24/24

In Table 16, *mean* is the mean CRPS over the complete cells and *rank* the mean of the per-cell ranks, where 1 is the lowest CRPS in the cell and ties share the average rank. Those two rows compare all fourteen systems and are produced by `tools/probts_ranks.py`. Figure 2 instead ranks methods within each fitting regime, and Appendix B.2 specifies the dataset-block statistical comparison at level 0.10.

Table 16: ProbTS CRPS grid: nine datasets and four horizons. Every baseline column is transcribed from its source paper under the benchmark's own protocol; the GETUP-ZS, GETUP-ZS+AURORA, GETUP-POINT, GETUP-MIX and GETUP-ID columns are measured here through the same harness, so the mean and rank rows below mix reported and measured values and are read as a ranking of published numbers, not of runs we control. The vertical division separates source-trained systems from systems with target-fitted components, including the error heads of GETUP-MIX. The three attachment columns—GETUP-ZS+AURORA, GETUP-POINT and GETUP-MIX—are filled on seven datasets and are absent on Exchange and Illness, because an attachment is scored on our own run of AURORA, and that run covers the TFB set, which holds neither dataset. The published AURORA column does carry Illness, so the backbone can be run there; what our sweep lacks is a context and patch for two new frequencies, the inherited rule of twice the dominant period asking weekly Illness for a context of 1040 against its 966 points. The GETUP-ID column now also carries Electricity and Traffic at $H = 96$ and Illness at all four horizons, each on three seeds, and Electricity at $H = 192$ on two, where a superscript gives the seed count: such a cell is printed rather than dashed but does not enter the dataset-average row. The remaining Electricity and Traffic horizons are unmeasured. Recorded per-cell runtimes on Electricity (321 channels) and Traffic (862 channels) were 8–11 and 13–36 hours, respectively. The last two rows are computed on the 20 of 36 cells in which *every* column has a value, so that they are comparable across columns; a mean over each column's own cells would reward a model for the cells it skipped. Bold marks the lowest CRPS in a cell whatever its regime, and underlining the lowest CRPS on the other side of the vertical division. Every row therefore carries one overall best and one best-in-the-other-regime, because a target-fitted model beating a source-trained one is not a finding and the marks should not present it as one.

		SOURCE-TRAINED								TARGET-FITTED					
Dataset	H	AURORA	Sundial	Chronos	MOIRAI	Lag-Llama	GETUP-ZS	GETUP-ZS +AURORA	GETUP-POINT	GETUP-MIX	VolDy-VAE	K^2VAE	CSDI	TimeGrad	GETUP-ID
ETTm1	96	0.261	0.253	0.360	0.464	0.354	0.226	0.260	0.244	0.245	**0.208**	0.232	0.236	0.522	0.209
	192	0.278	0.279	0.404	0.467	0.368	0.243	0.279	0.261	0.261	0.240	0.259	0.291	0.603	**0.229**
	336	0.292	0.291	0.425	0.524	0.387	0.255	0.291	0.269	0.270	0.251	0.262	0.322	0.601	**0.242**
	720	0.312	0.318	0.461	0.514	0.403	0.279	0.311	0.287	0.286	0.279	0.294	0.448	0.621	**0.267**
	avg	0.286	0.285	0.413	0.492	0.378	0.251	0.285	0.265	0.265	0.244	0.262	0.324	0.587	**0.237**
ETTm2	96	0.131	0.128	0.134	0.176	0.163	0.115	0.130	0.120	0.120	**0.107**	0.126	0.115	0.427	0.108
	192	0.149	0.150	0.163	0.197	0.181	0.132	0.148	0.136	0.135	**0.129**	0.148	0.147	0.424	0.130
	336	0.163	0.167	0.190	0.229	0.206	**0.144**	0.162	0.146	0.146	0.147	0.164	0.190	0.469	0.147
	720	0.182	0.189	0.223	0.321	0.227	0.164	0.181	0.161	**0.160**	0.166	0.221	0.239	0.470	0.167
	avg	0.156	0.159	0.178	0.231	0.194	0.139	0.155	0.141	0.140	**0.137**	0.165	0.173	0.447	0.138
ETTh1	96	0.277	0.260	0.293	0.469	0.297	0.263	0.272	0.268	0.275	0.252	0.264	0.437	0.455	**0.238**
	192	0.297	0.297	0.348	0.496	0.312	0.276	0.291	0.287	0.292	0.277	0.290	0.496	0.516	**0.263**
	336	0.314	0.314	0.384	0.503	0.326	0.285	0.306	0.299	0.304	0.300	0.308	0.454	0.512	**0.280**
	720	0.334	0.312	0.413	0.526	0.340	0.300	0.325	0.311	0.316	0.310	0.314	0.528	0.523	**0.295**
	avg	0.306	0.296	0.359	0.498	0.319	0.281	0.299	0.291	0.297	0.285	0.294	0.479	0.502	**0.269**
ETTh2	96	0.148	0.157	0.163	0.212	0.178	0.145	0.144	0.138	0.139	0.148	0.162	0.164	0.358	**0.138**
	192	0.166	0.183	0.195	0.229	0.193	0.159	0.161	0.153	**0.153**	0.162	0.186	0.226	0.457	0.157
	336	0.186	0.203	0.226	0.253	0.211	0.171	0.180	0.169	**0.168**	0.175	0.257	0.274	0.481	0.174
	720	0.208	0.203	0.255	0.279	0.222	0.182	0.201	0.187	0.186	**0.174**	0.280	0.302	0.445	0.183
	avg	0.177	0.186	0.210	0.243	0.201	0.164	0.171	0.162	**0.162**	0.165	0.221	0.241	0.435	0.163
Weather	96	0.064	0.078	0.136	0.161	0.085	**0.058**	0.064	0.063	0.065	0.070	0.080	0.068	0.130	0.073
	192	0.069	0.083	0.145	0.173	0.094	**0.061**	0.069	0.067	0.068	0.067	0.079	0.068	0.127	0.077
	336	0.072	0.090	0.136	0.149	0.098	**0.063**	0.071	0.069	0.070	0.071	0.082	0.083	0.130	0.080
	720	0.076	0.096	0.151	0.233	0.107	**0.067**	0.075	0.075	0.075	0.073	0.084	0.087	0.113	0.084
	avg	0.070	0.087	0.142	0.179	0.096	**0.062**	0.070	0.069	0.069	0.070	0.081	0.077	0.125	0.079
Electricity	96	**0.066**	0.068	†	0.231	†	0.071	0.066	0.092	0.094	0.069	0.073	0.153	0.096	0.066
	192	0.075	0.074	†	0.236	†	0.075	0.074	0.093	0.095	0.075	0.080	0.200	0.100	**0.072**[2]
	336	0.087	0.083	†	0.245	†	0.081	0.087	0.102	0.104	**0.078**	0.084	—	0.102	—
	720	0.111	0.098	†	0.274	†	0.097	0.110	0.114	0.116	**0.080**	0.087	—	0.108	—
	avg	0.085	0.081	—	0.246	—	0.081	0.084	0.100	0.102	**0.076**	0.081	—	0.101	—
Traffic	96	0.199	†	0.250	†	0.255	0.210	0.200	0.232	0.237	**0.180**	0.186	—	0.202	0.205
	192	0.207	†	0.249	†	0.295	0.208	0.208	0.239	0.244	**0.187**	0.188	—	0.208	—
	336	0.223	†	0.269	†	0.335	0.210	0.224	0.246	0.251	**0.194**	0.195	—	0.213	—
	720	0.250	†	0.310	†	0.434	0.218	0.255	0.272	0.277	**0.192**	0.200	—	0.220	—
	avg	0.220	—	0.270	—	0.330	0.211	0.222	0.247	0.252	**0.188**	0.192	—	0.211	—
Exchange	96	0.024	0.024	0.021	0.025	0.042	0.022	—	—	—	0.022	0.031	0.028	0.068	**0.018**
	192	0.032	0.034	0.032	0.034	0.047	0.031	—	—	—	0.030	0.032	0.045	0.087	**0.024**
	336	0.045	0.046	0.045	0.047	0.061	0.041	—	—	—	0.041	0.048	0.060	0.074	**0.032**
	720	0.075	0.076	0.078	0.073	0.078	0.067	—	—	—	0.065	0.069	0.143	0.099	**0.044**
	avg	0.044	0.045	0.044	0.045	0.057	0.040	—	—	—	0.040	0.045	0.069	0.082	**0.029**
ILI	24	0.110	0.108	0.120	0.150	0.135	0.123	—	—	—	**0.082**	0.087	0.250	0.275	0.171
	36	0.155	0.152	0.179	0.171	0.163	0.158	—	—	—	**0.110**	0.113	0.285	0.272	0.202
	48	0.168	0.164	0.186	0.151	0.171	0.167	—	—	—	**0.114**	0.124	0.285	0.295	0.202
	60	0.156	0.168	0.196	0.163	0.156	0.177	—	—	—	**0.130**	0.142	0.283	0.295	0.188
	avg	0.147	0.148	0.170	0.159	0.156	0.156	—	—	—	**0.109**	0.116	0.276	0.284	0.191
	mean	0.1990	0.2026	0.2603	0.3288	0.2376	0.1794	0.1961	0.1855	0.1867	0.1803	0.2046	0.2588	0.4192	**0.1771**
	rank	7.63	8.15	11.33	13.03	10.75	**2.50**	6.35	3.70	4.43	3.23	7.45	9.80	13.45	3.23

Table 17: ProbTS metrics for GETUP-ID over three seeds. CRPS evaluates the predictive distribution; ND, NRMSE, MSE and MASE evaluate the point prediction.

Dataset	H	CRPS	ND	NRMSE	MSE	MASE
seed(s)	123, 42, 456					
ETTh1	96	0.238	0.313	0.590	8.326	1.042
	192	0.263	0.344	0.662	10.072	1.126
	336	0.280	0.365	0.694	10.880	1.194
	720	0.295	0.387	0.687	10.191	2.318
ETTh2	96	0.138	0.178	0.261	16.656	1.613
	192	0.157	0.204	0.295	21.438	1.966
	336	0.174	0.227	0.327	26.121	2.308
	720	0.183	0.241	0.345	28.100	4.553
ETTm1	96	0.209	0.267	0.504	6.817	2.829
	192	0.229	0.294	0.559	7.865	3.232
	336	0.242	0.312	0.591	8.431	3.474
	720	0.267	0.343	0.657	9.990	3.826
ETTm2	96	0.108	0.137	0.206	11.125	5.747
	192	0.130	0.164	0.245	15.721	7.900
	336	0.147	0.186	0.276	19.231	9.001
	720	0.167	0.214	0.311	24.192	10.886

Table 18: CRPS per cell for GETUP-ID, TARFVAE and VolDy-VAE at $L = 96$ and $L = 720$. Each model is scored under the law it states: GETUP-ID against the estimated residual shape its bundles declare, VolDy-VAE against the empirical law of its 100 trajectories on 40 of its 48 cells, and TARFVAE—with VolDy-VAE's remaining eight cells—against the Gaussian carrying the mean and spread of its trajectories, which is all those dumps stored. Substituting one reading for the other moves a cell by at most 0.36% (`tools/samplecrps_delta.py`, per-cell figures in `results/samplecrps_vs_gaussian.csv`), so the column compares models to that tolerance; the CRPS comparison against these baselines under the benchmark's own metric is Table 16. A dagger marks a cell in which the historical-scale control of Section 3.5—the same point forecast with one constant scale per horizon step and channel—scores better than the model's own law on that metric, and is unrelated to the daggers in the zero-shot grids above, which record a dataset withheld by a source paper. Bold marks the best of the three systems within one context length and underlining the second best, by the direction the metric declares: lower for CRPS, NLL, PIT-KS and ENCE, closest to 0.90 for coverage. The last row is the median over the 24 rows complete in both context blocks; the median rather than the mean, because this column also holds cells of order 10.

Dataset	H	L=96			L=720		
		GETUP-ID	TARFVAE	VolDy-VAE	GETUP-ID	TARFVAE	VolDy-VAE
ETTh1	96	**0.3086** ±0.0007	0.3144	0.3247	**0.3168** ±0.0001	0.3348†	0.3718
	192	**0.3415** ±0.0001	0.3575†	0.3464	**0.3408** ±0.0005	0.4068†	0.3753
	336	0.3741 ±0.0000	0.3912†	**0.3724**	**0.3584** ±0.0010	0.4245†	0.3858
	720	**0.3804** ±0.0005	0.4044†	0.3926	**0.3762** ±0.0002†	0.4317†	0.4221†
ETTh2	96	**0.3039** ±0.0003	0.3232†	0.3227	**0.3004** ±0.0000†	0.3778†	0.4014†
	192	**0.3611** ±0.0000	0.3950†	0.3816	**0.3416** ±0.0018†	0.4398†	0.4282†
	336	**0.3934** ±0.0002	0.4587†	0.4177†	**0.3632** ±0.0002†	0.4574†	0.4285†
	720	**0.3946** ±0.0005	0.4320†	0.4115	**0.3802** ±0.0004	0.4667†	0.4343†
ETTm1	96	**0.2651** ±0.0001	0.2656	0.2720	**0.2635** ±0.0000	0.2778†	0.2894
	192	**0.2892** ±0.0001	0.2962	0.2958	**0.2875** ±0.0001	0.2970†	0.3019
	336	**0.3077** ±0.0002	0.3191	0.3130	**0.3115** ±0.0015	0.3248†	0.3257
	720	**0.3430** ±0.0002	0.3650†	0.3576	**0.3593** ±0.0029	0.3716†	0.3662
ETTm2	96	0.2207 ±0.0002	**0.2196**†	0.2255†	**0.2190** ±0.0010	0.2341†	0.2664†
	192	0.2596 ±0.0000	**0.2589**†	0.2605†	**0.2537** ±0.0004	0.2726†	0.3067†
	336	0.2965 ±0.0001	0.2995†	**0.2957**†	**0.2897** ±0.0008	0.3082†	0.3369
	720	**0.3662** ±0.0002†	0.3782†	0.3726†	**0.3492** ±0.0013	0.3707†	0.3925
Exchange	96	**0.1528** ±0.0011	0.1692†	0.1866†	**0.1553** ±0.0003	0.1834†	0.3104†
	192	**0.2119** ±0.0012	0.2561†	0.2540†	**0.2198** ±0.0019	0.2511†	0.4532†
	336	**0.2916** ±0.0006	0.3907†	0.3662†	**0.3107** ±0.0015†	0.3744†	0.7364†
	720	**0.4119** ±0.0006	0.5900†	0.6272	**0.4069** ±0.0035	0.8314†	1.0693†
Weather	96	0.1547 ±0.0002	**0.1443**	0.1541	**0.1388** ±0.0005	0.1481	0.1695
	192	0.1834 ±0.0001	**0.1764**	0.1898	**0.1661** ±0.0002	0.1754	0.1945
	336	**0.2112** ±0.0000	0.2124	0.2219†	**0.1900** ±0.0000	0.2073	0.2128†
	720	**0.2408** ±0.0001	0.2434	0.2532†	**0.2165** ±0.0002	0.2283	0.2515†
Electricity	96	0.1937 ±0.0001	0.1757	—	0.1654 ±0.0001	0.1819	—
	192	0.1955 ±0.0000	0.1890	—	0.1742 ±0.0000	0.2042†	—
	336	0.2053 ±0.0000	0.2020	—	0.1854 ±0.0001	0.2310†	—
	720	0.2293 ±0.0001	0.2273†	—	0.2104	0.2569†	—
Traffic	96	0.2638 ±0.0001	*n/c*	—	0.2281	*n/c*	—
	192	0.2684 ±0.0001	*n/c*	—	0.2341	*n/c*	—
	336	0.2740	*n/c*	—	0.2390	*n/c*	—
	720	—	*n/c*	—	—	*n/c*	—
mean	24	0.2943	0.3192	0.3173	0.2881	0.3415	0.3846

Table 19: Negative log-likelihood per cell, same grid, in nats; daggers as in Table 18. For the sampled baselines the density is estimated from their own samples as described in Appendix B.7. That estimator has a bandwidth floor, so a cell in which the sampled trajectories nearly coincide while the observation falls far from them produces a very large score. Forty of the cells exceed 50 nats and the largest reaches $1.7 \cdot 10^4$: thirty-one of those are TARFVAE and nine VolDy-VAE, and every one is that case rather than a failed run. Such a cell says the trajectory cloud is far narrower than the error it has to cover, which the coverage column reads as the same thing from the other side—TARFVAE covers 0.56 to 0.64 on Weather against a nominal 0.90. Bold marks the best of the three systems within one context length and underlining the second best; lower is better throughout this table. The last row is the median over the 24 rows complete in both context blocks, not the mean: a column holding cells of order 10^4 has a mean that describes those cells and not the column.

Dataset	H	L=96			L=720		
		GETUP-ID	TARFVAE	VolDy-VAE	GETUP-ID	TARFVAE	VolDy-VAE
ETTh1	96	**0.743** $\pm$0.005	2.511†	1.053†	**0.774** $\pm$0.003	4.026†	1.055†
	192	**0.874** $\pm$0.001	5.768†	1.231†	**0.851** $\pm$0.005	22.734†	1.017†
	336	**0.992** $\pm$0.005	10.326†	1.138†	**0.917** $\pm$0.006	18.650†	1.011
	720	**1.031** $\pm$0.000	5.834†	1.313†	**1.006** $\pm$0.001†	16.180†	3.245†
ETTh2	96	**0.673** $\pm$0.003	902.350†	3.327†	**0.799** $\pm$0.007	113.063†	22.414†
	192	**0.872** $\pm$0.002	934.584†	6.211†	**0.876** $\pm$0.014	88.352†	12.581†
	336	**0.999** $\pm$0.004	1213.997†	5.102†	**0.959** $\pm$0.001	63.634†	12.508†
	720	**1.013** $\pm$0.006	728.812†	2.838†	**0.997** $\pm$0.019	45.223†	11.067†
ETTm1	96	**0.535** $\pm$0.001	308.248†	1.030†	**0.527** $\pm$0.000	2.796†	0.908†
	192	**0.638** $\pm$0.001	22.810†	0.823†	**0.644** $\pm$0.003	2.476†	0.942†
	336	**0.723** $\pm$0.004	48.436†	0.890†	**0.761** $\pm$0.021	3.848†	0.934†
	720	**0.870** $\pm$0.004	224.994†	1.165†	**0.946** $\pm$0.014	3.664†	1.045†
ETTm2	96	**0.309** $\pm$0.010	1377.208†	34.990†	**0.351** $\pm$0.001	20.484†	2.036†
	192	**0.501** $\pm$0.004	1807.974†	1.965†	**0.496** $\pm$0.007	22.098†	4.718†
	336	**0.654** $\pm$0.001	1210.290†	1.910†	**0.690** $\pm$0.005	27.916†	2.796†
	720	**0.905** $\pm$0.002	2027.215†	5.602†	**0.866** $\pm$0.005	21.061†	3.888†
Exchange	96	**0.039** $\pm$0.009	32.831†	15.421†	**0.071** $\pm$0.027	11.286†	78.271†
	192	**0.358** $\pm$0.015	76.511†	3.902†	**0.388** $\pm$0.013	20.938†	159.290†
	336	**0.641** $\pm$0.003	82.626†	1.635†	**0.748** $\pm$0.007	6.399†	441.257†
	720	**1.064** $\pm$0.002	16.237†	1.542	**1.161** $\pm$0.018	9.858†	111.473†
Weather	96	**−0.324** $\pm$0.010	7375.760†	131.701†	**−0.358** $\pm$0.006	2650.404†	1732.990†
	192	**−0.118** $\pm$0.004	$1.18 \cdot 10^4$†	12.461†	**−0.168** $\pm$0.000	3111.111†	2074.540†
	336	**0.039** $\pm$0.001	$1.38 \cdot 10^4$†	1.711†	**−0.040** $\pm$0.001	3426.279†	1976.595†
	720	**0.172** $\pm$0.004	$1.74 \cdot 10^4$†	0.549†	**0.067** $\pm$0.002	2405.773†	1408.378†
Electricity	96	0.194 $\pm$0.001	473.943†	—	0.079 $\pm$0.002	107.835†	—
	192	0.224 $\pm$0.000	606.853†	—	0.140 $\pm$0.000	201.370†	—
	336	0.283 $\pm$0.001	640.258†	—	0.204 $\pm$0.003	351.463†	—
	720	0.404 $\pm$0.001	555.490†	—	0.338	587.920†	—
Traffic	96	0.332 $\pm$0.002	*n/c*	—	0.199	*n/c*	—
	192	0.358 $\pm$0.000	*n/c*	—	0.231	*n/c*	—
	336	0.389	*n/c*	—	0.267	*n/c*	—
	720	—	*n/c*	—	—	*n/c*	—
median	24	0.6635	518.5300	1.8105	0.7545	20.9995	7.8925

Table 20: Nominal 90% interval coverage per cell, same grid; compared with 0.900, where deviation in either direction matters equally. Daggers as in Table 18. Bold marks the best of the three systems within one context length and underlining the second best, by the direction the metric declares: lower for CRPS, NLL, PIT-KS and ENCE, closest to 0.90 for coverage. The last row is the mean over the 24 rows complete in both context blocks.

Dataset	H	L=96			L=720		
		GETUP-ID	TARFVAE	VolDy-VAE	GETUP-ID	TARFVAE	VolDy-VAE
ETTh1	96	**0.898** ±0.002	0.777†	0.826†	**0.910** ±0.002	0.652†	0.856†
	192	**0.900** ±0.000	0.660†	0.828†	**0.916** ±0.004	0.408†	0.855†
	336	**0.894** ±0.000	0.627†	0.844	**0.902** ±0.002	0.462†	0.868†
	720	**0.912** ±0.001	0.680†	0.788†	**0.910** ±0.002	0.527†	0.661†
ETTh2	96	**0.850** ±0.001	0.492†	0.769	**0.830** ±0.001	0.272†	0.506†
	192	**0.830** ±0.000	0.466†	0.740	**0.837** ±0.004	0.279†	0.521†
	336	**0.836** ±0.001	0.377†	0.697	**0.852** ±0.001	0.329†	0.487†
	720	**0.883** ±0.006	0.480†	0.786	**0.891** ±0.004	0.376†	0.535†
ETTm1	96	**0.913** ±0.000†	0.813†	0.873†	**0.878** ±0.001†	0.722†	0.853†
	192	**0.910** ±0.000†	0.776†	0.880†	**0.885** ±0.001	0.758†	0.856†
	336	0.914 ±0.001	0.751†	**0.890**	**0.878** ±0.005	0.696†	0.856†
	720	**0.921** ±0.000	0.715†	0.868	**0.870** ±0.004	0.728†	0.860
ETTm2	96	**0.870** ±0.000	0.727†	0.808†	**0.864** ±0.004	0.648†	0.782†
	192	**0.868** ±0.000	0.697†	0.829	**0.860** ±0.002	0.707†	0.722†
	336	**0.867** ±0.002	0.717†	0.824	**0.846** ±0.000	0.662†	0.767
	720	**0.842** ±0.001	0.669†	0.766	**0.825** ±0.005	0.648†	0.724
Exchange	96	**0.897** ±0.006	0.538†	0.815†	**0.874** ±0.013	0.441†	0.594†
	192	**0.893** ±0.007	0.429†	0.743†	**0.876** ±0.004	0.355†	0.303†
	336	**0.883** ±0.002	0.391†	0.782†	**0.843** ±0.007	0.607†	0.169†
	720	**0.859** ±0.003	0.524†	0.775	**0.763** ±0.005	0.466†	0.171†
Weather	96	0.953 ±0.001†	0.585†	**0.919**	**0.948** ±0.003†	0.563†	0.615†
	192	0.947 ±0.000†	0.644†	**0.921**	**0.951** ±0.000†	0.608†	0.581†
	336	0.943 ±0.000†	0.618†	**0.923**	**0.946** ±0.002	0.518†	0.567†
	720	0.943 ±0.000†	0.561†	**0.928**	**0.943** ±0.001	0.602†	0.572†
Electricity	96	0.897 ±0.000	0.738†	—	0.890 ±0.000†	0.685†	—
	192	0.899 ±0.000	0.705†	—	0.889 ±0.000†	0.581†	—
	336	0.898 ±0.000	0.673†	—	0.887 ±0.000†	0.452†	—
	720	0.897 ±0.000†	0.586†	—	0.887†	0.331†	—
Traffic	96	0.849 ±0.001†	*n/c*	—	0.830†	*n/c*	—
	192	0.842 ±0.000†	*n/c*	—	0.834†	*n/c*	—
	336	0.842†	*n/c*	—	0.837†	*n/c*	—
	720	—	*n/c*	—	—	*n/c*	—
mean	24	0.8928	0.6131	0.8259	0.8791	0.5431	0.6367

Table 21: PIT–uniform Kolmogorov–Smirnov distance per cell; daggers as in Table 18. This diagnostic compares the predictive CDF values at observed outcomes with a uniform distribution. Bold marks the best of the three systems within one context length and underlining the second best, by the direction the metric declares: lower for CRPS, NLL, PIT-KS and ENCE, closest to 0.90 for coverage. The last row is the mean over the 24 rows complete in both context blocks.

Dataset	H	L=96			L=720		
		GETUP-ID	TARFVAE	VolDy-VAE	GETUP-ID	TARFVAE	VolDy-VAE
ETTh1	96	**0.0222** ±0.0012	0.0691†	0.0583†	0.0406 ±0.0013†	0.1496†	**0.0330**
	192	**0.0127** ±0.0007	0.1468†	0.0528†	0.0452 ±0.0020†	0.2828†	**0.0362**†
	336	0.0367 ±0.0004	0.1458†	**0.0366**	**0.0477** ±0.0025†	0.2427†	0.0534
	720	0.0835 ±0.0002†	0.1319†	**0.0799**†	**0.1008** ±0.0009†	0.2620†	0.1662†
ETTh2	96	0.0890 ±0.0009†	0.2075†	**0.0708**	**0.0666** ±0.0025†	0.3276†	0.2166†
	192	0.0928 ±0.0022†	0.2336†	**0.0883**	**0.0573** ±0.0019	0.3174†	0.2317†
	336	**0.0975** ±0.0006	0.2660†	0.1131	**0.0566** ±0.0023	0.3035†	0.2158†
	720	**0.0687** ±0.0014	0.2551†	0.1006	**0.0297** ±0.0031	0.3495†	0.2212†
ETTm1	96	**0.0161** ±0.0001	0.0512	0.0268	**0.0233** ±0.0009	0.1014†	0.0456†
	192	**0.0218** ±0.0008	0.0687	0.0228	**0.0192** ±0.0006	0.0810†	0.0471
	336	0.0331 ±0.0003	0.0803†	**0.0303**	**0.0255** ±0.0031	0.1148†	0.0327†
	720	**0.0244** ±0.0000	0.1026†	0.0267	**0.0294** ±0.0035	0.1020†	0.0410†
ETTm2	96	0.0674 ±0.0016	0.0984†	**0.0553**†	**0.0412** ±0.0005	0.1310†	0.0630†
	192	0.0926 ±0.0023†	0.1100†	**0.0470**†	**0.0495** ±0.0037	0.1191†	0.1103†
	336	0.0979 ±0.0012†	0.1095†	**0.0476**†	**0.0613** ±0.0058	0.1224†	0.0768
	720	0.1242 ±0.0018†	0.1308†	**0.0744**†	**0.0719** ±0.0091	0.1395†	0.0935
Exchange	96	**0.0889** ±0.0043†	0.2655†	0.1199†	**0.0523** ±0.0091	0.2465†	0.4001†
	192	**0.1147** ±0.0058†	0.3623†	0.1971†	**0.0806** ±0.0011	0.3288†	0.5887†
	336	**0.1767** ±0.0049	0.4138†	0.2688	**0.1429** ±0.0069	0.3448†	0.7460†
	720	**0.2394** ±0.0006	0.4840†	0.3924	**0.1601** ±0.0112	0.5376†	0.7303†
Weather	96	**0.0876** ±0.0007	0.1691†	0.0942	**0.1047** ±0.0000	0.1933†	0.1857†
	192	**0.0902** ±0.0003	0.1479†	0.0915	**0.1045** ±0.0012	0.1620†	0.2036†
	336	**0.0874** ±0.0003	0.1507†	0.1067	**0.1034** ±0.0010	0.2084†	0.2044†
	720	**0.0919** ±0.0003	0.1785†	0.1202	**0.0969** ±0.0003	0.1504†	0.2009†
Electricity	96	0.0194 ±0.0001	0.0943†	—	0.0172 ±0.0004	0.1291†	—
	192	0.0226 ±0.0001	0.1231†	—	0.0287 ±0.0001	0.1950†	—
	336	0.0272 ±0.0000	0.1436†	—	0.0376 ±0.0000	0.2381†	—
	720	0.0482 ±0.0001	0.1872†	—	0.0629	0.3131†	—
Traffic	96	0.0348 ±0.0002	*n/c*	—	0.0557	*n/c*	—
	192	0.0413 ±0.0004	*n/c*	—	0.0536	*n/c*	—
	336	0.0480	*n/c*	—	0.0467	*n/c*	—
	720	—	*n/c*	—	—	*n/c*	—
mean	24	0.0816	0.1825	0.0968	0.0671	0.2216	0.2060

Table 22: Expected normalised calibration error per cell, $K = 100$ rank bins of the predicted scale; daggers as in Table 18. Bold marks the best of the three systems within one context length and underlining the second best, by the direction the metric declares: lower for CRPS, NLL, PIT-KS and ENCE, closest to 0.90 for coverage. The last row is the mean over the 24 rows complete in both context blocks.

Dataset	H	L=96			L=720		
		GETUP-ID	TARFVAE	VolDy-VAE	GETUP-ID	TARFVAE	VolDy-VAE
ETTh1	96	0.287 ±0.013†	0.550†	**0.258**†	0.242 ±0.005†	0.950†	**0.138**
	192	0.307 ±0.000†	1.002†	**0.275**†	0.282 ±0.009†	2.717†	**0.122**
	336	0.323 ±0.001†	1.275†	**0.199**	0.274 ±0.008†	2.450†	**0.081**
	720	**0.299** ±0.003†	1.110†	0.331†	**0.356** ±0.007†	2.065†	0.863†
ETTh2	96	**0.097** ±0.002	4.471†	0.658†	**0.312** ±0.010	5.432†	2.086†
	192	**0.202** ±0.004	3.759†	0.784	**0.298** ±0.010	4.945†	1.839†
	336	**0.136** ±0.019	4.649†	0.893	**0.345** ±0.023	4.129†	1.963†
	720	**0.265** ±0.023	2.958†	0.476	**0.319** ±0.031	3.293†	1.615†
ETTm1	96	0.173 ±0.011†	0.708†	**0.155**†	**0.112** ±0.001	0.762†	0.181†
	192	0.192 ±0.014†	0.908†	**0.083**	**0.159** ±0.013†	0.641†	0.167†
	336	0.216 ±0.007†	0.979†	**0.064**	0.201 ±0.018†	0.902†	**0.155**
	720	0.217 ±0.005†	1.203†	**0.140**	0.269 ±0.005†	0.797†	**0.142**
ETTm2	96	**0.124** ±0.005	5.481†	1.261†	**0.204** ±0.020	1.572†	0.569†
	192	**0.099** ±0.003	5.708†	0.424†	**0.223** ±0.001	1.257†	0.945†
	336	**0.255** ±0.049	4.204†	0.410	**0.164** ±0.003	1.772†	0.586
	720	**0.208** ±0.012	5.843†	0.708†	**0.234** ±0.000	1.696†	0.764
Exchange	96	**0.129** ±0.010	1.801†	0.425†	**0.142** ±0.012	2.022†	2.162†
	192	**0.149** ±0.057	2.790†	0.525†	**0.159** ±0.041	2.626†	4.078†
	336	**0.139** ±0.011	3.631†	0.329†	**0.246** ±0.035	1.147†	6.376†
	720	**0.240** ±0.012	1.725†	0.269	**0.322** ±0.035	1.706†	4.896†
Weather	96	**0.356** ±0.010†	22.450†	0.618†	**0.345** ±0.021†	19.757†	7.321†
	192	0.326 ±0.002†	25.744†	**0.319**†	**0.324** ±0.017†	24.448†	7.014†
	336	0.309 ±0.001†	41.364†	**0.160**	**0.302** ±0.000†	26.195†	7.040†
	720	0.320 ±0.002†	90.882†	**0.134**	**0.285** ±0.006†	29.919†	7.105†
Electricity	96	0.116 ±0.001	1.082†	—	0.144 ±0.002	1.230†	—
	192	0.094 ±0.001	1.291†	—	0.110 ±0.001	1.868†	—
	336	0.066 ±0.000	1.421†	—	0.075 ±0.001	2.812†	—
	720	0.076 ±0.001	1.942†	—	0.074	4.242†	—
Traffic	96	0.321 ±0.001†	*n/c*	—	0.319	*n/c*	—
	192	0.357 ±0.001†	*n/c*	—	0.333	*n/c*	—
	336	0.361†	*n/c*	—	0.321	*n/c*	—
	720	—	*n/c*	—	—	*n/c*	—
median	24	0.2165	3.2945	0.3300	0.2715	2.0435	1.2800

C.4 IRREGULAR SAMPLING

The UFO protocol drops 30% of whole calendar days and forward-fills the missing observations for uniform-grid models. The runs use $L = H = 720$, three seeds and the original data scale. NCRPS normalises each channel's accumulated CRPS by its accumulated absolute target magnitude before averaging over channels.

Table 23: NCRPS under UFO's regular and irregular protocols at $L = H = 720$; lower is better. Baselines are transcribed from Kuleshov et al. (2026). Our measured row is an earlier in-domain configuration, not the released GETUP-ID: this protocol drops 30% of calendar days and no run of GETUP-ID under it exists, so the row reports that earlier configuration and is not a claim about the system the rest of the paper reports; the GETUP-ID row is empty for the same reason. An asterisk on the row label repeats that where the number is read. The reported variation across three seeds is shown below it. Bold marks the best value in a column and underlining the second best; lower NCRPS is better.

	ETTm1		ETTm2		Weather	
Model	reg.	irreg.	reg.	irreg.	reg.	irreg.
DeepAR	0.584	0.619	1.286	3.168	0.720	0.625
PatchTST (Nie et al., 2023)	0.439	0.457	1.233	1.344	0.670	0.665
LaST	0.401	0.404	1.838	2.325	0.646	0.673
K^{2}VAE	0.274	0.310	0.595	0.854	0.564	0.631
TSDiff	0.549	0.460	1.516	1.432	<u>0.363</u>	0.442
Latent ODE	0.533	0.522	2.211	2.078	0.446	0.533
Neural CDE	<u>0.261</u>	<u>0.290</u>	0.447	0.477	0.381	<u>0.427</u>
TFM	0.604	0.578	2.297	2.821	0.912	0.749
SLiCE	0.539	0.529	1.959	2.053	1.240	1.273
NCDE++	0.299	0.326	0.434	0.460	0.420	0.469
UFO	0.271	0.294	<u>0.429</u>	<u>0.430</u>	**0.360**	**0.412**
GETUP-ID earlier cfg.*	**0.2515**	**0.2419**	**0.3997**	**0.4048**	0.5371	0.6532
($\pm$ over 3 seeds)	0.0006	0.0015	0.0024	0.0014	0.0035	0.0054
GETUP-ID	—	—	—	—	—	—

Weather's irregular NCRPS is 0.6532 against UFO's 0.412. The channel breakdown identifies rain variables as large contributors, with NCRPS 5.66 for `rain (mm)` and 1.89 for `raining (s)`; these sparse, intermittent channels are particularly sensitive to normalisation by total absolute target magnitude. Scoring only originally observed target steps, which comprise 70.2% of ETT steps and 77.0% of Weather steps, gives 0.2608 ± 0.0006 and 0.2531 ± 0.0010 on ETTm1, 0.4047 ± 0.0025 and 0.4114 ± 0.0023 on ETTm2, and 0.5206 ± 0.0033 and 0.5486 ± 0.0036 on Weather. The published baseline table uses all target steps, so the two cannot be combined into a matched improvement.

Additional benchmark scope. Other benchmark settings reported with AURORA remain outside this evaluation. Time-MMD contains nine domains at four horizons and includes multimodal inputs (Liu et al., 2024); the present bank processes time-series values only. TFB's short-term suite contains 8068 series in seven frequency groups and uses MASE and msMAPE (Qiu et al., 2024). EPF contains five electricity markets with a 168-step context, a 24-step horizon and exogenous covariates (Olivares et al., 2023). Evaluating these settings requires their own data and metric interfaces, including an explicit treatment of covariates for EPF.

D UNCERTAINTY SCALE AND RESIDUAL SHAPE

The following analyses examine three aspects of the predictive distribution: mixture shape at fixed moments, historical-scale and online-shape controls, and the residuals and intervals underlying the qualitative diagnostics.

D.1 MIXTURES AROUND SAMPLED TRAJECTORIES

GETUP-MIX instantiates Equation 8 with $M = 10$ trajectories from AURORA and one positive error head per sample index. The fitting protocol is given in Appendix A.4. In the expanded 36-cell grid (Table 25), the mixture has lower CRPS in 24 cells and lower NLL in 34 cells. Its nominal 90% coverage is closer to 0.900 in 22 cells, and ENCE is lower in 31. After oracle dilation, it has lower CRPS in all 36 cells. These comparisons change both predictive scale and shape.

The five-cell mixture-shape experiment uses ETTh1, ETTh2, ETTm1, ETTm2 and Weather at $H = 96$, with 256 fitting windows and 96 scoring windows per dataset. Its moment-matched control is distinct from the expanded 36-cell comparison in Table 25.

To isolate the contribution of mixture shape, the five-cell experiment also evaluates the Gaussian with the same first two moments,

$$\mu_Y = \frac{1}{M}\sum_j \mu_j, \qquad s_Y^2 = \frac{1}{M}\sum_j s_j^2 + \frac{1}{M}\sum_j (\mu_j - \mu_Y)^2, \tag{18}$$

which preserves the mixture's mean and variance while removing its non-Gaussian shape.

Table 24: Target-fitted mixture experiment on ETTh1, ETTh2, ETTm1, ETTm2 and Weather at $H = 96$. Entries are medians across cells. The $@c$ columns use an oracle NLL-optimal dilation fitted on the scoring observations. Mixture wins against the single-head control out of five: CRPS 2, NLL 4, coverage 1, NLL$@c$ 5, CRPS$@c$ 5, ENCE 4. Bold marks the best value in a column, with coverage judged by distance from 0.90 and c^* by distance from 1; where two laws tie to the reported precision both are marked. Underlining marks the second best in a column.

Law	CRPS	NLL	cov_{90}	c^*	NLL$@c$	CRPS$@c$	ENCE
GETUP-POINT	0.2295	1.0023	0.9243	1.1340	0.9850	0.3148	0.3437
GETUP-MIX	<u>0.2319</u>	<u>0.8154</u>	<u>0.9488</u>	<u>0.9389</u>	<u>0.7763</u>	0.2120	0.2848
Moment-matched Gaussian	0.2337	0.7562	0.9499	0.9997	0.7375	<u>0.2127</u>	0.2848

The mixture has lower NLL in four cells and lower ENCE in four, while the single-head control has lower raw CRPS in three. After each law receives its own oracle dilation, the mixture has lower CRPS and NLL in all five cells. Weather gives the largest likelihood difference: the control has NLL 11.52 and $c^* = 4.80$ against 0.8154 and 1.2059 for the mixture, yet its raw CRPS is about 0.7% lower. The moment-matched Gaussian has higher CRPS in all five cells and higher NLL in four, supporting a benefit from mixture shape on the measured cells.

Table 25: Target-fitted mixture comparison on 36 cells. Both arms use the same fitting and scoring windows within each cell; the @c columns use oracle dilation fitted on the scoring labels. Bold marks the better arm in each cell, lower being better except for coverage, which is judged by distance from 0.900. The moment-matched shape control is reported separately in Table 24.

		GETUP-POINT					GETUP-MIX				
Dataset	H	CRPS	NLL	cov_{90}	CRPS@c	ENCE	CRPS	NLL	cov_{90}	CRPS@c	ENCE
ETTh1	96	0.326	1.067	0.889	0.334	0.274	0.331	0.946	0.915	0.330	0.178
	192	0.349	1.146	0.873	0.359	0.250	0.356	1.013	0.903	0.356	0.155
	336	0.384	1.252	0.844	0.393	0.286	0.388	1.108	0.878	0.390	0.152
	720	0.442	1.625	0.785	0.452	0.472	0.443	1.293	0.830	0.445	0.254
ETTh2	96	0.227	0.629	0.954	0.213	0.344	0.232	0.638	0.969	0.212	0.376
	192	0.251	0.694	0.950	0.239	0.307	0.252	0.683	0.958	0.238	0.308
	336	0.272	0.755	0.915	0.269	0.207	0.275	0.748	0.933	0.269	0.218
	720	0.312	0.924	0.887	0.314	0.227	0.312	0.874	0.908	0.310	0.195
ETTm1	96	0.308	1.002	0.902	0.315	0.288	0.307	0.875	0.916	0.305	0.248
	192	0.330	1.534	0.882	0.366	0.391	0.328	1.049	0.904	0.333	0.298
	336	0.348	1.660	0.860	0.389	0.454	0.346	1.148	0.881	0.357	0.356
	720	0.376	1.504	0.844	0.399	0.389	0.375	1.179	0.864	0.383	0.298
ETTm2	96	0.230	0.696	0.981	0.186	0.512	0.212	0.580	0.981	0.184	0.463
	192	0.225	0.631	0.942	0.214	0.336	0.231	0.636	0.967	0.208	0.386
	336	0.238	0.652	0.941	0.229	0.303	0.241	0.650	0.963	0.225	0.346
	720	0.258	0.754	0.920	0.255	0.295	0.258	0.711	0.939	0.250	0.282
Weather	96	0.199	11.524	0.924	0.512	0.396	0.200	0.815	0.949	0.212	0.285
	192	0.227	4.554	0.900	0.368	0.391	0.226	1.466	0.927	0.265	0.235
	336	0.249	3.025	0.882	0.336	0.404	0.247	0.813	0.910	0.254	0.210
	720	0.278	15.517	0.854	0.640	0.487	0.276	1.252	0.884	0.297	0.269
Electricity	96	0.183	7.006	0.674	0.215	1.497	0.178	0.728	0.820	0.183	0.500
	192	0.207	21.187	0.671	0.326	1.635	0.201	1.168	0.805	0.209	0.624
	336	0.224	10.058	0.676	0.287	1.563	0.217	1.111	0.800	0.223	0.606
	720	0.261	8.540	0.685	0.328	1.591	0.253	1.686	0.796	0.266	0.675
Wind	96	0.510	14.350	0.752	0.938	1.108	0.498	2.465	0.805	0.550	0.528
	192	0.550	10.982	0.730	0.875	1.239	0.535	2.054	0.796	0.561	0.539
	336	0.605	10.894	0.699	0.893	1.416	0.591	2.614	0.759	0.630	0.708
	720	0.630	13.165	0.679	0.966	1.509	0.614	2.601	0.743	0.646	0.720
Traffic	96	0.247	$2.15 \cdot 10^7$	0.651	303.093	3.049	0.241	482.402	0.862	0.764	0.583
	192	0.253	$4.55 \cdot 10^7$	0.642	459.927	3.138	0.246	$1.36 \cdot 10^4$	0.856	0.296	0.604
	336	0.270	$3.92 \cdot 10^8$	0.655	1541.464	3.125	0.263	$2.50 \cdot 10^6$	0.853	1.580	0.611
	720	0.302	$1.03 \cdot 10^9$	0.664	2995.481	3.254	0.295	$1.40 \cdot 10^7$	0.842	4.953	0.664
NYSE	24	0.936	2.039	0.993	0.374	0.812	0.726	1.812	0.995	0.373	0.750
	36	0.860	1.968	0.992	0.433	0.734	0.713	1.798	0.997	0.416	0.691
	48	0.808	1.928	0.974	0.575	0.687	0.702	1.762	0.992	0.481	0.624
	60	0.783	2.070	0.950	0.760	0.714	0.688	1.725	0.974	0.590	0.526

D.2 HISTORICAL-SCALE AND ONLINE-SHAPE CONTROLS

Table 26 reports historical-scale controls fitted on split test bundles, separately from the validation-fitted main-text controls. Section 4 discusses their overlapping target timestamps. Table 27 tests online residual-shape updates at fixed point forecasts and supplied scale parameters; the adaptive law can change its variance through the shape update and dilation.

Additional scale and point controls. The effect varies by dataset. A separate scale ladder over 23 cells, which replaces the learned scale by one constant per horizon step and channel, gives improvements of 16.23–17.95% on Weather and 4.85–5.77% on ETTh1 for the learned scale, while on Exchange that ladder favours the constant by -4.11 to -0.50%; in Table 4, fitted on validation over 24 cells, the learned law is ahead on Exchange as well. Proper-score and coverage rankings can therefore differ within our own model as well as in the baseline comparisons. The point predictor remains essential: replacing it with a seasonal-naive forecast and keeping the same historical-scale construction raises CRPS by 21.0% to 31.6% above the strongest reference on ETTm1, ETTm2 and Weather at $H = 96$.

Table 26: Historical (h, c) scale control against each model's predictive law. Entries count cells in which the control improves the named metric, divided by the number compared; coverage is evaluated by distance from 0.900. The point prediction is retained in both arms; scale constants are fitted on the first half of the available test bundle and both arms are scored on the second. Bold marks a clean sweep, that is a group in which the count equals the number of cells.

Model	L	Dataset	cells	CRPS	NLL	cov_{90}	PIT-KS	ENCE
GETUP-ID	96	ETTh1	4	0/4	0/4	0/4	1/4	**4**/4
		ETTh2	4	0/4	0/4	0/4	2/4	0/4
		ETTm1	4	0/4	0/4	2/4	0/4	**4**/4
		ETTm2	4	1/4	0/4	0/4	3/4	0/4
		Electricity	4	0/4	0/4	1/4	0/4	0/4
		Exchange	4	0/4	0/4	0/4	2/4	0/4
		Traffic	3	0/3	0/3	**3**/3	0/3	**3**/3
		Weather	4	0/4	0/4	**4**/4	0/4	**4**/4
all, L=96			31	*1/31*	*0/31*	*10/31*	*8/31*	*15/31*
GETUP-ID	720	ETTh1	4	1/4	1/4	0/4	**4**/4	**4**/4
		ETTh2	4	3/4	0/4	0/4	1/4	0/4
		ETTm1	4	0/4	0/4	1/4	0/4	3/4
		ETTm2	4	0/4	0/4	0/4	0/4	0/4
		Electricity	4	0/4	0/4	**4**/4	0/4	0/4
		Exchange	4	1/4	0/4	0/4	0/4	0/4
		Traffic	3	0/3	0/3	**3**/3	0/3	0/3
		Weather	4	0/4	0/4	2/4	0/4	**4**/4
all, L=720			31	*5/31*	*1/31*	*10/31*	*5/31*	*11/31*
TARFVAE	96	ETTh1	4	3/4	**4**/4	**4**/4	**4**/4	**4**/4
		ETTh2	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		ETTm1	4	1/4	**4**/4	**4**/4	2/4	**4**/4
		ETTm2	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		Electricity	4	1/4	**4**/4	**4**/4	**4**/4	**4**/4
		Exchange	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		Weather	4	0/4	**4**/4	**4**/4	**4**/4	**4**/4
all, L=96			28	*17/28*	*28/28*	*28/28*	*26/28*	*28/28*
TARFVAE	720	ETTh1	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		ETTh2	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		ETTm1	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		ETTm2	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		Electricity	4	3/4	**4**/4	**4**/4	**4**/4	**4**/4
		Exchange	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		Weather	4	0/4	**4**/4	**4**/4	**4**/4	**4**/4
all, L=720			28	*23/28*	*28/28*	*28/28*	*28/28*	*28/28*
VolDy-VAE	96	ETTh1	4	0/4	**4**/4	3/4	3/4	3/4
		ETTh2	4	1/4	**4**/4	0/4	0/4	1/4
		ETTm1	4	0/4	**4**/4	2/4	0/4	1/4
		ETTm2	4	**4**/4	**4**/4	1/4	**4**/4	3/4
		Exchange	4	3/4	3/4	3/4	2/4	3/4
		Weather	4	2/4	**4**/4	0/4	0/4	2/4
all, L=96			24	*10/24*	*23/24*	*9/24*	*9/24*	*13/24*
VolDy-VAE	720	ETTh1	4	1/4	3/4	**4**/4	2/4	1/4
		ETTh2	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		ETTm1	4	0/4	**4**/4	3/4	3/4	2/4
		ETTm2	4	2/4	**4**/4	2/4	2/4	2/4
		Exchange	4	**4**/4	**4**/4	**4**/4	**4**/4	**4**/4
		Weather	4	2/4	**4**/4	**4**/4	**4**/4	**4**/4
all, L=720			24	*13/24*	*23/24*	*21/24*	*19/24*	*17/24*

Table 27: ProbTS CRPS with online residual-shape updates and the offline law. Bold marks the lower of the two CRPS values in each cell. The point branch is identical in both arms; the adaptive harness path stretches the law about its median while leaving the supplied scale parameter unchanged. Negative relative changes favour the online version; the final mean includes cells in which an update occurred.

Dataset	H	seeds	of them changed	GETUP-ID	GETUP-ID (adaptive)	Δ, %
ETTh1	96	3	3	0.2380	0.2365	−0.62
	192	3	3	0.2633	0.2619	−0.51
	336	3	2	0.2805	0.2795	−0.34
	720	3	0	0.2954	0.2954	+0.00
ETTh2	96	3	1	0.1375	0.1377	+0.11
	192	3	1	0.1572	0.1573	+0.05
	336	3	1	0.1740	0.1738	−0.07
	720	3	0	0.1834	0.1834	+0.00
ETTm1	96	3	1	0.2087	0.2083	−0.21
	192	3	1	0.2289	0.2285	−0.17
	336	3	1	0.2423	0.2416	−0.31
	720	3	1	0.2673	0.2661	−0.45
ETTm2	96	3	1	0.1081	0.1079	−0.13
	192	3	1	0.1298	0.1296	−0.12
	336	3	1	0.1466	0.1466	−0.03
	720	3	1	0.1667	0.1669	+0.14
mean over the 14 cells where it fired						−0.19

D.3 Residual diagnostics and predictive intervals

Residual shape. Persistence errors on the four ETT datasets have excess kurtosis between 4.53 and 8.20 when pooled, falling to 1.85–2.96 after standardising within each channel–horizon group; Anderson–Darling rejects normality in all 224 cells. Shape also changes with horizon—on ETTm1's LUFL channel excess kurtosis falls from 36.04 at $h = 1$ to 0.060 at $h = 720$—and several channels carry a concentration of exact zeros, with 78.5% and 79.1% of HULL residuals exactly zero at $h = 1$ in ETTh2 and ETTm2. These are properties of that predictor and that grouping, not a proof that a trained model's conditional residuals are non-Gaussian. The full summaries, normality statistics, and aligned Weather windows are shown below.

We examine the errors of a persistence forecast on the four ETT datasets. Let z be the series standardised with training statistics; the error at lead time h is $e_h = z_{t+h} - z_t$, with a common set of forecast origins across horizons. Four datasets, seven channels and $h \in \{1, 6, 24, 48, 96, 192, 336, 720\}$ give 224 channel–horizon cells.

Under the reported normal-reference thresholds, Anderson–Darling rejects normality in all 224 cells; Jarque–Bera and D'Agostino each do not reject in six cells, and Shapiro–Wilk does not reject in one. Across the HULL and LULL groups, 19 of 32 cells have a degenerate interquartile range, and they account for 21 of the 22 cells whose fitted Student-t degrees of freedom are below one: a sharply concentrated centre can affect a fitted tail parameter.

The mean interval half-width grows over the horizon from 0.768 to 1.406 for GETUP-ID and from 0.910 to 1.893 for VolDy-VAE, factors of 1.83 and 2.08; the historical control starts at 0.232 and grows by a factor of 5.41. Average absolute changes in scale between successive horizon steps are 0.0241, 0.0675 and 0.0064. Coverage over all 1306 windows is 0.939, 0.983 and 0.966 against the nominal 0.950 level. The two windows are the most dynamic window and the one with the largest discrepancy between VolDy-VAE's dispersion and its realised residuals under the stored selection rule, so they are illustrative rather than representative.

Table 28: Persistence-residual summaries on the four ETT datasets. Raw rows pool channels and horizons directly; standardised rows first centre and scale each channel–horizon group. AD and KS are the reported normality-test statistics and ν is the fitted Student-t degrees of freedom. The reference AD critical value is approximately 0.787.

Dataset	Pooling	n	skew	ex. kurt.	AD stat.	KS	ν
ETTh1	raw	121 016	−0.050	8.198	4842.1	0.1244	1.87
ETTh2	raw	121 016	−0.011	4.534	4811.4	0.1341	1.17
ETTm1	raw	604 856	−0.088	7.942	30597.3	0.1412	1.47
ETTm2	raw	604 856	0.029	5.404	27705.2	0.1448	0.91
ETTh1	standardised	121 016	−0.125	2.914	1385.2	0.0707	3.02
ETTh2	standardised	121 016	0.053	1.850	1780.9	0.0882	2.72
ETTm1	standardised	604 856	−0.081	2.956	3591.6	0.0532	4.46
ETTm2	standardised	604 856	0.059	2.264	11285.1	0.1052	2.33

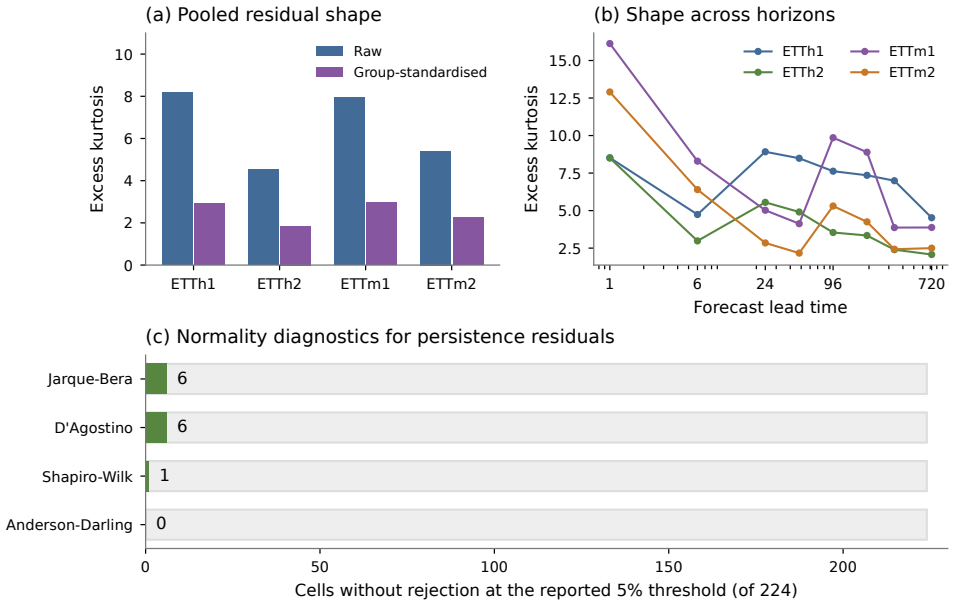

Figure 7: Persistence-forecast residuals on ETT. The panels compare raw and group-standardised excess kurtosis, show its change across lead times, and summarise the normality-test statistics.

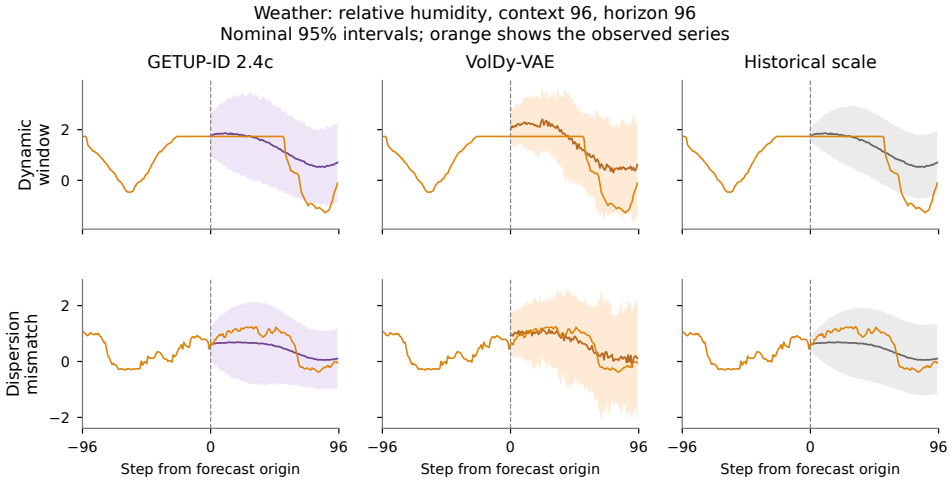

Figure 8: Nominal 95% intervals on two selected Weather windows at $L = H = 96$, on the same 1306 aligned scoring windows. Columns show GETUP-ID 2.4c, VolDy-VAE and the historical (h, c) scale control.

D.4 LONG-HORIZON SCALE COLLAPSE

Positivity alone does not prevent scale collapse. A recorded shared-bank $H = 720$ head has a softplus bias of -23.18, a validation 90% coverage of 0.0577 against nominal 0.900 and a CRPS of 230.16 under the empirical shape. Retraining at $H = 96$ raises coverage from 0.362 to 0.906; retraining at $H = 720$ raises it from 0.0577 to 0.8814 and lowers the empirical-shape CRPS from 230.16 to 0.5762, while the Gaussian validation CRPS rises from 0.4283 to 0.5860—consistent with a collapsed scale making Gaussian CRPS resemble absolute error. That retrained run differs from the original bank in three respects at once—the stage-three learning rate, the number of stage-three steps per epoch and the early-stopping patience—so the recovery cannot be attributed to any one of them and a separating run is still required.

E RELATED WORK

Mean–variance estimation and separate training. Neural estimation of input-dependent mean and variance predates current deep forecasting models. Nix & Weigend (1994a) use separate hidden units for the two outputs, initialise the mean predictor first, and subsequently optimise both branches through a likelihood. Their later error-bar construction includes a frozen-mean stage that fits variance using exchanged data splits before joint refinement (Nix & Weigend, 1994b). These procedures distinguish separate architectures, staged initialisation and permanent freezing, which have different training dependencies. In time-series forecasting, MambaProbTSF combines a Mamba mean network with a separate variance network, warm-starts the mean using squared error, and then updates both through Gaussian likelihood (Pessoa et al., 2025). GETUP keeps the point predictor fixed throughout residual learning.

Joint likelihood training can reduce the influence of poorly fitted observations through its variance-weighted mean gradients. β-NLL modifies these gradients with a detached variance weight (Seitzer et al., 2022). Faithful heteroscedastic regression rescales the mean gradient and blocks variance gradients from shared features, preserving the optimisation of the corresponding mean-only model (Stirn et al., 2023). Freezing provides another way to protect an existing point forecast. This property motivates our training choice; it is not a new guarantee of uncertainty quality or evidence that staged training dominates these alternatives.

Predicting the errors of a fixed model. DEUP learns a primary predictor's pointwise risk from out-of-sample errors. Its epistemic interpretation subtracts an aleatoric-risk estimate when available; its interactive setting uses newly acquired observations and features that describe the learner's changing knowledge (Lahlou et al., 2023). RIO fits a Gaussian process to a neural predictor's residuals, providing both an uncertainty estimate and a correction to its prediction (Qiu et al., 2020). Practical deep heteroskedastic regression (PDHR) is a particularly close post-hoc construction: it fits positive linear variance heads to intermediate features of a frozen network using held-out likelihood, and also considers an ensemble of these heads with a common mean (Jordahn et al., 2026). Its evaluation concerns molecular regression.

These methods establish error prediction and frozen-model uncertainty fitting. GETUP-ZS studies their application to a time-series expert bank: separate spectral gates aggregate forecasts and residual estimates, with RU targeting the combined forecast and DAU targeting individual experts. The distinction lies in this supervision and aggregation. The deterministic attachment experiment tests predictor-specific error fitting; it is not a comparison against every post-hoc method.

Ensemble uncertainty and calibration. Ensemble model output statistics (EMOS) fits a predictive distribution from ensemble forecasts and spread using CRPS (Gneiting et al., 2005). Deep ensembles combine learned component variances with disagreement between component means (Lakshminarayanan et al., 2017). DAU follows this aggregation pattern but uses absolute-error predictions and a scale gate that differs from the point gate. Its interpretation is therefore stated in Section 4, rather than identified with an exact aleatoric–epistemic variance decomposition.

Post-hoc regression recalibration adjusts a predictive CDF to align probability levels with empirical frequencies (Kuleshov et al., 2018). This differs from learning a history-dependent error magnitude, although the two can be combined. For time series, EnbPI constructs sequential intervals from

bootstrap-ensemble residuals (Xu & Xie, 2021), while SPCI models conditional residual quantiles and their temporal dependence (Xu & Xie, 2023). Our empirical residual law also uses historical errors, but produces marginal densities and quantiles through a location–scale model. We evaluate its calibration empirically without asserting a conformal coverage guarantee.

Linear experts and spectral routing. DLinear and RLinear show that simple linear predictors can be competitive on long-horizon benchmarks (Zeng et al., 2023; Li et al., 2023); reversible instance normalisation handles changes in level and scale (Kim et al., 2022). Super-Linear combines frequency-specialised linear experts and routes histories through normalised periodograms (Nochumsohn et al., 2026), providing the direct architectural basis of our point bank. We retain these components and fit an error branch with its own gate, within the broader mixture-of-experts framework (Shazeer et al., 2017). The architectural contribution concerns the routing of residual predictions, rather than introducing spectral routing or linear forecasting.

Forecast distributions and residual structure. DeepAR learns autoregressive conditional distributions (Salinas et al., 2020). Conditioned normalising flows model flexible multivariate densities (Rasul et al., 2021b); TimeGrad, TSDiff and CSDI use diffusion for forecasting or conditional reconstruction (Rasul et al., 2021a; Kollovieh et al., 2023; Tashiro et al., 2021). K^2VAE combines Koopman and Kalman constructions in a variational model (Wu et al., 2025). TARFVAE enriches the variational posterior with a flow and generates the full forecast horizon in one step (Wei et al., 2025). These methods have different inference costs; the number of generative steps alone does not determine the cost of a scored predictive distribution.

NsDiff uses pretrained mean and variance predictors to construct an uncertainty-dependent diffusion endpoint and noise schedule (Ye et al., 2025). Its scale supervision uses local variation in the future series, whereas our error targets are residuals of a specified predictor. VolDy-VAE models a recurrent volatility state along the forecast horizon (Wang et al., 2026). DeRegiME uses a regime-mixing kernel and Student-t observation model within a single sparse variational Gaussian process (Wood et al., 2026). Its gate changes kernel and noise structure; ours weights point forecasts and residual estimates. Zheng and Sun explicitly model contemporaneous and temporal error covariance and change the probabilistic training objective (Zheng & Sun, 2024). That joint dependence is outside the marginal location–scale construction evaluated here.

Pretrained forecasting and evaluation. Chronos, MOIRAI, Sundial and Lag-Llama learn predictive distributions from collections of time series (Ansari et al., 2024; Woo et al., 2024; Liu et al., 2025; Rasul et al., 2024). AURORA extends pretrained forecasting to multimodal inputs and uses flow matching to generate trajectories (Wu et al., 2026). Our experiments use its time-series inputs, deterministic head and sampled forecasts as distinct sources of predictions; target-fitted mixture heads are reported separately from source-trained attachments.

TFB and ProbTS motivate consistent evaluation across forecasting methods and horizons (Qiu et al., 2024; Zhang et al., 2024); TSFM-Bench focuses on pretrained forecasters (Li et al., 2025). The Monash archive provides series for pretraining (Godahewa et al., 2021). We combine proper scoring rules (Gneiting & Raftery, 2007) with regression calibration diagnostics (Levi et al., 2022). Fixed-point scale substitutions and moment-matched shape controls identify different sources of improvement; the metrics themselves are established evaluation tools.