FORECASTING AS A VERIFIABLE TASK: WHAT RL AND TEST-TIME COMPUTE CAN (AND CANNOT) BUY

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilistic forecasting is verifiable: once the future arrives, a proper scoring rule grades every sampled trajectory. Time-series foundation models are therefore natural targets for the reinforcement-learning (RL) post-training and test-time-compute recipes that transformed language models. We ask what these recipes buy when the verifier grades a *distribution* rather than a single answer. We decompose the excess CRPS of a sampled forecaster into Monte-Carlo, optimisation and information gaps, derive the per-sample reward under which policy gradient is unbiased for CRPS (it is necessarily pairwise), and analyse GRPO under this reward. GRPO's group-relative advantages are provably biased for pairwise rewards: the group-mean baseline lowers the effective dispersion coefficient from $\frac{1}{2}$ to $\frac{G-2}{2(G-1)}$, and per-group standard-deviation scaling adds an $O(1)$ shrinkage that no group size removes. The bias predicts the under-dispersion of every uncorrected run across two model families and three model sizes (coverage $0.63 \to 0.55$ for Chronos-T5-small); a two-line correction removes it. Corrected RL matches a supervised fine-tuning control at equal budget, with no significant GIFT-Eval gain, while preserving calibration at half the KL divergence. Test-time compute fails for a different reason: its only verifier, a self-backtest, has fidelity $\rho \approx 0.1$, so every backtest-driven intervention hurts. The bias and its correction apply to any pairwise or set-level reward. Code is available at `https://anonymous.4open.science/r/verifiable-forecasting-72ED`.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become the standard second stage of training for language models: sample several candidate answers, grade each with a deterministic verifier, and push probability mass towards the candidates that pass (Shao et al., 2024; DeepSeek-AI et al., 2025). Test-time compute—sampling more candidates and selecting among them with the same verifier—is its inference-time counterpart (Snell et al., 2024; Beirami et al., 2024). Probabilistic time-series forecasting looks like an ideal target for both recipes. The task is verifiable by construction: once the future is observed, a proper scoring rule such as the continuous ranked probability score (CRPS; Matheson & Winkler, 1976) grades every sampled trajectory. The dominant open time-series foundation models (TSFMs) are tokenised autoregressive language models over quantised values (Ansari et al., 2024), so their samples are policy trajectories, and the benchmarks are large and standardised (Aksu et al., 2024). Yet neither the reward design nor its interaction with group-relative training has been examined carefully.

This paper does so, and the answer is more subtle than "the recipe transfers". Three properties distinguish forecasting from question answering. First, the scored object is a *distribution*, not a trajectory: CRPS rewards accuracy *and* calibrated spread, so a reward attached to a single sample must encode how that sample contributes to the spread of the set. Second, the sample count N is itself a test-time-compute knob whose effect on the score has a closed form. Third, no external verifier exists at test time; the only proxy is the model's own recent history (a "self-backtest").

We make three contributions. **(i) The Monte-Carlo gap.** We decompose the excess CRPS of a sampled forecaster into a Monte-Carlo, an optimisation and an information gap and derive the first

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.

in closed form (Prop. 1), accurate to $< 0.1\%$ on real data. The formula explains why Chronos' default top-k sampler, an implicit sharpener, helps at $N \leq 4$ and hurts at $N \geq 8$: at the default $N = 20$ it costs 2.9% in weighted quantile loss (WQL) on GIFT-Eval. **(ii) Credit assignment and the GRPO bias.** We derive the per-sample reward under which policy gradient is unbiased for CRPS (Prop. 2) and prove that GRPO's group-relative advantages are *not* valid control variates for it (Cor. 1): the group mean lowers the effective dispersion coefficient from $\frac{1}{2}$ to $\frac{G-2}{2(G-1)}$, and the per-group standard deviation adds an $O(1)$ shrinkage that persists as $G \to \infty$. Toy fixed points, unit tests and training runs agree quantitatively. Every uncorrected run shrinks the model's Gini mean difference by roughly a third to a half at the default group size, regardless of corpus, horizon, seed, KL weight, model size or model family, and a two-line correction removes the effect. The corrected reward coincides with the fair set-marginal credit of concurrent work (App. A). **(iii) A controlled study.** On GIFT-Eval we run a reward ablation, an estimator ablation, an SFT control at matched budget, a 2×2 corpus$\times$horizon design, three model sizes, a second model family and three quantile-output baselines (App. D), plus a self-backtest test-time-compute study whose verifier fidelity ($\rho \approx 0.1$) explains why every backtest-driven intervention is harmful. Once the estimator is unbiased, the picture is sober: RL on a proper scoring rule yields a small, calibrated gain on T5-small, comparable to SFT at matched budget (neither significant across configurations), that does not carry over to larger models or across corpora. RL does what SFT does on the same data, with calibration preserved.

2 Background and setup

Sampled forecasters as policies. Chronos-T5 (Ansari et al., 2024) scales a context window by its mean absolute value, quantises values into 4094 bins and generates the next H tokens autoregressively. A trajectory $x_{1:H}$ is a sample from the policy $\pi_\theta(\cdot \mid c)$, and a forecast distribution is the empirical measure of N trajectories. Unless stated otherwise we use *exact* sampling (temperature 1, no top-k, special tokens suppressed), so that the sampling distribution equals the model distribution and teacher-forced log-probabilities are exact.

Scoring. For a predictive distribution F and outcome y, $\mathrm{CRPS}(F, y) = \mathbb{E}_F|X - y| - \frac{1}{2}\mathbb{E}_F|X - X'|$. The second term is the Gini mean difference $\mathrm{GMD}(F)$. With N samples we use the unbiased ("fair") estimator, which divides the pairwise sum by $N(N-1)$. GIFT-Eval (Aksu et al., 2024) reports MASE and the weighted quantile loss (WQL) over nine quantiles, normalised per dataset by the seasonal-naive error. We follow the official protocol (97 dataset/frequency/term configurations) and, for paired comparisons, a 512-window subset per configuration drawn with a fixed seed.

3 Theory

3.1 Excess CRPS decomposes into three gaps

Let $F^\star$ be the true conditional law, F_θ the model law and F_N its N-sample empirical law. Because CRPS is the Cramér distance (Bellemare et al., 2017), $\mathbb{E}_y\mathrm{CRPS}(F, y) - \mathbb{E}_y\mathrm{CRPS}(F^\star, y) = \frac{1}{2}\int (F - F^\star)^2$ is a squared distance, and the excess CRPS of F_N splits into a *Monte-Carlo gap* (F_N vs. F_θ), an *optimisation gap* (F_θ vs. the best model in the class) and an *information gap* (the best model vs. $F^\star$). Test-time computation can attack only the first; post-training attacks the second.

Proposition 1 (Monte-Carlo gap)**.** *For i.i.d. samples,* $\mathbb{E}\big[\mathrm{CRPS}(F_N, y)\big] = \mathrm{CRPS}(F_\theta, y) + \dfrac{\mathrm{GMD}(F_\theta)}{2N}$.

The gap is proportional to the model's own dispersion and decays as $1/N$, so averaging more samples never buys more than $\mathrm{GMD}/2$. On 256 real windows the prediction matches the empirical curve to within 0.1% for $N \in \{1, \ldots, 256\}$, for both model sizes and both samplers (Table 1, Figure 1a). Top-k truncation lowers GMD (by 28% for Chronos-T5-small) but worsens F_θ itself (by 4.9% in the $N \to \infty$ limit), so it wins for $N \leq 4$ and loses for $N \geq 8$; at the default $N = 20$ the default sampler is 2.9% worse in WQL on GIFT-Eval.

3.2 PER-SAMPLE CREDIT ASSIGNMENT FOR A PROPER REWARD

Maximising $J(\theta) = -\mathbb{E}_y \mathrm{CRPS}(\pi_\theta, y)$ by policy gradient (Williams, 1992) requires a reward for *each* trajectory, although CRPS is a property of the *set*.

Proposition 2 (Policy gradient of CRPS). $\nabla_\theta J = -\mathbb{E}_{x\sim\pi_\theta}\big[(|x-y| - \mathbb{E}_{x'\sim\pi_\theta}|x-x'|)\nabla_\theta \log \pi_\theta(x)\big]$.

The dispersion coefficient is 1, not $\frac{1}{2}$, because each sample enters the pairwise term twice. With a group of G rollouts the leave-one-out reward

$$r_i = -\Big(|x_i - y| - \tfrac{c}{G-1}\textstyle\sum_{j\neq i}|x_i - x_j|\Big), \qquad c = 1, \tag{1}$$

gives an unbiased estimate $\frac{1}{G}\sum_i r_i \nabla \log \pi(x_i)$ of ∇J. Two natural alternatives are improper. The naive reward $-|x_i - y|$ ($c = 0$) drives the policy to a point mass at the conditional median, and the "fair split" $c = \frac{1}{2}$ optimises $\mathbb{E}|X - y| - \frac{1}{4}\mathrm{GMD}$, whose optimum is systematically under-dispersed.

3.3 GRPO'S NORMALISATION IS BIASED FOR PAIRWISE REWARDS

GRPO (Shao et al., 2024) forms advantages $a_i = (r_i - \bar{r})/s$ from the group mean $\bar{r}$ and standard deviation s. For rewards that depend on one trajectory only, $\bar{r}$ and s are valid control variates. For equation 1 they are not: r_j contains $|x_j - x_i|$, so both $\bar{r}$ and s are functions of x_i.

Corollary 1 (Group-mean baseline). *With* $a_i = r_i - \bar{r}$, $\mathbb{E}\big[\sum_i a_i \nabla \log \pi(x_i)\big] = -(G-1)\,\nabla\Big[\mathbb{E}|X - y| - c\,\frac{G-2}{2(G-1)}\,\mathrm{GMD}\Big]$.

Hence the "proper" reward with $c = 1$ silently optimises a scoring rule with dispersion coefficient $\frac{G-2}{2(G-1)}$: 0.43 at $G = 8$ and 0.48 at $G = 32$ instead of $\frac{1}{2}$. For a Gaussian policy against $y \sim \mathcal{N}(0, 1)$ the fixed point is $\sigma^\star = 0.76$ at $G = 8$ (Table 4). The correction is $c = \frac{G-1}{G-2}$, or a baseline computed from samples independent of x_i.

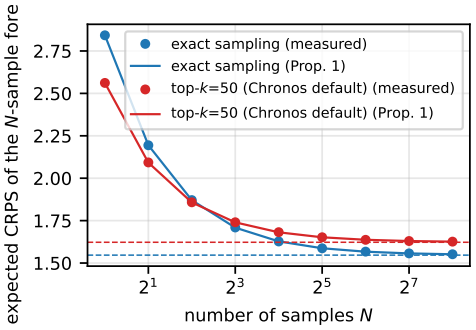
Dividing by the per-group standard deviation ("std") adds a second, larger bias with no closed form. For a calibrated policy, y falls inside the sample range in 94% of groups at $G = 32$; in these groups the advantage is scaled by $\mathbb{E}[1/s] \approx 2.3$, whereas "surprise" groups, the only ones whose gradient expands the distribution, are scaled by ≈ 1.1. The resulting fixed point is $\sigma^\star = 0.50$ at $G = 8$, 0.615 at $G = 32$ and still 0.64 at $G = 128$: the bias is $O(1)$ in G. The coefficient correction alone does not remove it (0.71 at $G = 8$); a scalar normaliser over the whole batch, whose correlation with a single sample is $O(1/(BG))$, does. The analysis reaches beyond GRPO. With the fair CRPS estimator, the set-marginal credit $r_k = S_{-k} - S_G$ of concurrent work (Park et al., 2026) *is* the corrected pairwise reward, std shrinkage included (App. A), and an RLOO baseline with subgroup-recomputed rewards implements the correction implicitly (Table 4).

4 EXPERIMENTAL DESIGN

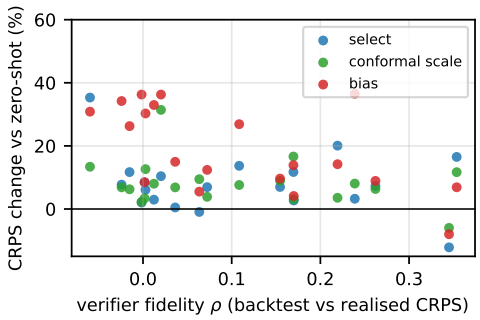
Models and data. Chronos-T5 small (20M) and base (200M) as policies; Chronos-Bolt small/base and Chronos-2 as quantile-output baselines; seasonal naive as the normaliser. Post-training uses either the univariate GiftEval-Pretrain corpus (61 datasets, 31.6k series, 162M points, median length 5990; disjoint from the GIFT-Eval test sets; the *pre-training corpus* below) or the GIFT-Eval training split (49 datasets, 8.5k series, 15.9M points, median length 729; every window strictly precedes the test windows of the same series; the *GIFT training split*).

Training. GRPO with $B = 16$ windows $\times$ $G = 8$ rollouts per step, $H = 64$, context 64–512, exact sampling, PPO-clip 0.2 (Schulman et al., 2017), the k_3 KL estimator to the frozen reference with $\beta = 0.01$, AdamW 2×10^{-5}, 2000 steps, dropout disabled so that rollouts and teacher-forced log-probabilities share one distribution (KL is exactly 0 at step 0). The SFT control uses the same sampler, budget and optimiser with the native token cross-entropy.

Evaluation. (i) GIFT-Eval official protocol; (ii) paired 512-window subsets with geometric-mean ratios to the zero-shot model (win rates here count configurations); (iii) robust in-corpus validation



(a) Prop. 1 on 256 GiftEval-Pretrain windows (T5-small, $H = 32$): measured N-sample CRPS vs. the closed form; dashed: $\mathrm{CRPS}(F_\theta, y)$. The sharpened sampler wins at small N and loses beyond the crossover.



(b) Self-backtest test-time compute: CRPS change vs. the zero-shot model for the three backtest interventions of §5.5, against verifier fidelity ρ (20 configurations; Table 12). Only one configuration benefits.

Table 1: Proposition 1 on real data: expected fair CRPS of an N-sample forecast (256 windows of GiftEval-Pretrain, $H = 32$, pool of $M = 512$ samples), empirical vs. $\mathrm{CRPS}(F_\theta, y) + \mathrm{GMD}/(2N)$, for exact sampling and Chronos' default top-k=50 sampler. Values are in units of each window's seasonal-naive context MAE.

Model	sampler	$N=1$	$N=2$	$N=4$	$N=8$	$N=16$	$N=64$	$N=256$	$N\to\infty$	GMD
T5-small	exact	2.842	2.194	1.871	1.710	1.627	1.567	1.552	1.547	2.592
	predicted	(2.842)	(2.195)	(1.871)	(1.709)	(1.628)	(1.567)	(1.552)		
T5-small	top-k=50	2.562	2.093	1.858	1.740	1.682	1.637	1.626	1.622	1.879
	predicted	(2.562)	(2.092)	(1.857)	(1.740)	(1.681)	(1.637)	(1.626)		
T5-base	exact	2.889	2.220	1.887	1.719	1.635	1.572	1.556	1.551	2.677
	predicted	(2.889)	(2.220)	(1.885)	(1.718)	(1.634)	(1.571)	(1.556)		
T5-base	top-k=50	2.587	2.117	1.881	1.765	1.706	1.662	1.651	1.647	1.881
	predicted	(2.587)	(2.117)	(1.882)	(1.765)	(1.706)	(1.662)	(1.651)		

on held-out windows (per-window median ratio, win rate = share of windows below one, Wilcoxon), in-sample at the series level and hence only a training diagnostic (a series-level held-out check in App. D.3 confirms its ordering); (iv) SynthOracle, synthetic families with known conditional laws that make the three gaps exactly computable; (v) coverage of the nominal 80% interval ("cov80"), whose calibrated value at $N = 20$ with interpolated quantiles is 0.73.

5 Results

5.1 The Monte-Carlo gap and the sampler

Table 1 checks Prop. 1 on real windows: the measured N-sample CRPS matches $\mathrm{CRPS}(F_\theta, y) + \mathrm{GMD}/(2N)$ to within Monte-Carlo error at every N, and the fair estimator is flat in N, so the closed form accounts for the entire Monte-Carlo gap. Two consequences follow. First, the gap is a property of the *scorer*, not of the model, and shrinks as $1/N$: RL cannot remove it, and a comparison between sampled and quantile-output models at small N mostly measures it. Second, it explains Chronos' default sampler. Top-k truncation sharpens the distribution. At $N \le 4$ the reduced GMD lowers the $\mathrm{GMD}/(2N)$ penalty by more than it raises $\mathrm{CRPS}(F_\theta, y)$, so the sharpened sampler wins; at $N \ge 8$ the ordering flips, and at the default $N = 20$ exact sampling is 2.9% better in WQL and 5.4% in CRPS across the 55 short-term configurations (Table 2; top-k wins on only 38% of configurations and loses 0.09 of coverage). Raising N from 20 to 64 lowers WQL by a further 2.3% while the fair CRPS is unchanged, and top-k stays 3.9% behind, as the closed form predicts for a purely Monte-Carlo effect. Part of the benchmark gap between sampled and quantile-output models at $N = 20$ is therefore scorer noise rather than model quality. This free improvement is as large as most reported post-training gains on this benchmark, so we use exact sampling throughout.

Table 2: Monte-Carlo gap on the benchmark: zero-shot Chronos-T5-small on the 55 short-term GIFT-Eval configurations (512-window subsets), WQL and fair CRPS as geometric-mean ratios to exact sampling at $N = 20$, and empirical coverage of the nominal 80% interval. "top-k": Chronos' default sampler ($k = 50$, special tokens not suppressed), an implicit sharpener.

N	exact sampling			top-k (Chronos default)		
	WQL	CRPS	cov80	WQL	CRPS	cov80
1	1.709	1.936[‡]	–	1.531	1.734[‡]	–
2	1.373	0.997	0.22	1.279	1.054	0.20
4	1.139	0.998	0.43	1.128	1.049	0.37
8	1.054	1.001	0.56	1.070	1.054	0.48
20	1.000	1.000	0.62	1.029	1.053	0.54
64	0.977	1.005	0.66	1.015	1.058	0.57

[‡]At $N = 1$ every sample-based score reduces to the one-sample absolute error $|x - y|$ (WQL, fair and naive CRPS coincide; Prop. 1 with $N = 1$), and interval coverage is undefined.

Table 3: Reward ablation (Chronos-T5-small, GiftEval-Pretrain, $H = 64$, 2000 steps, GRPO default normalisation). Theory predicts proper > half > naive ≈ MSE and increasing under-dispersion.

Run	pretrain-corpus validation (vs. zero-shot)				GIFT-Eval short (vs. zero-shot, paired)					
	med. ratio	win	cov80	GMD ratio	WQL	CRPS	win	Δcov80	GMD ratio	n
proper ($c = 1$)	0.988*	0.54	0.553	0.59	1.000	1.013	0.45	-0.078	0.73	55
half ($c = \frac{1}{2}$)	1.033	0.43	0.388	0.35	1.039**	1.078	0.20	-0.226	0.48	55
naive ($c = 0$)	1.075**	0.34	0.287	0.24	1.090**	1.150	0.11	-0.324	0.35	55
MSE	1.095**	0.31	0.283	0.26	1.115**	1.172	0.05	-0.322	0.37	55

Median ratio: per-window fair-CRPS ratio to the zero-shot model on 256 held-out windows of the named corpus; zero-shot cov80 = 0.63 (calibrated value at N=20: 0.73). GIFT columns: geometric-mean ratios over configurations, every row paired with the zero-shot model of its own size. Stars: Wilcoxon $^{*}p < 0.05$, $^{**}p < 0.01$ (per window for the validation columns; per configuration, on log WQL ratios, for the GIFT WQL column).

5.2 Reward ablation: theory predicts the ordering

Table 3 holds the estimator fixed (GRPO defaults) and varies only the reward. The ordering predicted by Cor. 1 and its improper counterparts appears in every column: the proper reward is the only one that does not destroy calibration, the half-coefficient reward under-disperses more strongly, and the naive per-sample $|x_i - y|$ reward and the MSE reward collapse the group towards a point forecast (cov80 falls from 0.63 to below 0.35, and the GMD to a fraction of the zero-shot value). Because CRPS is proper, the collapsed models are also worse in WQL: sharpening past the calibrated point is penalised. The proper reward itself, however, is no improvement on GIFT-Eval short-term (1.000 WQL ratio, win rate 0.45), and it too loses 0.08 of coverage. This is the first sign that the estimator, not the reward, is the binding constraint.

5.3 The estimator bias in practice

Table 4 isolates the estimator with a Gaussian policy and the trainer's own reward and advantage code. With the GRPO defaults ($c = 1$, per-group std) the fixed point is $\sigma^\star = 0.50$ at $G = 8$ and still 0.64 at $G = 128$. Removing the std normalisation gives 0.762, exactly the value $\sigma^{\star 2} = 2c_{\text{eff}}^2/(1 - 2c_{\text{eff}}^2)$ predicted by Cor. 1, and the corrected coefficient with a mean-only normaliser recovers $\sigma^\star = 1$ to within sampling error (a batch-level scalar normaliser is constant within a group and shares this fixed point).

Table 5 shows the same fingerprint on Chronos-T5. Every uncorrected run—two seeds, $\beta = 0$, both corpora, both horizon schedules, and the larger T5-base and T5-large—multiplies the median Gini mean difference by 0.51–0.62 and lowers coverage by 0.07–0.14 (Figure 2), consistent with the toy value 0.50 given the KL anchor and the finite step budget. Model size does not rescue the uncorrected recipe: T5-base and T5-large (710M) are at 1.004 and 1.000 WQL on GIFT-Eval short-term

Table 4: Fixed points of the GRPO estimator for a Gaussian policy $\mathcal{N}(\mu, \sigma)$ trained against $y \sim \mathcal{N}(0, 1)$ (optimum $\sigma^\star = 1$), using the reward and advantage code of our trainer verbatim (3 seeds, sd ≤ 0.006). "std": per-group standard-deviation normalisation (GRPO default); "mean": group-mean baseline only; theory: $\sigma^{\star 2} = 2c_{\text{eff}}^2/(1 - 2c_{\text{eff}}^2)$ with $c_{\text{eff}} = c\,\frac{G-2}{2(G-1)}$.

reward	G	std	mean	theory (mean)
proper	8	0.497	0.762	0.762
proper	32	0.615	0.939	0.938
proper	128	0.641	0.987	0.984
half	8	0.186	0.318	0.318
naive	8	0.005	0.042	0.000
proper, $c = (G-1)/(G-2)$	8	0.712	1.000	1.000
proper, $c = (G-1)/(G-2)$	32	0.658	1.000	1.000
proper, no baseline	8	–	0.997	1.000
proper, leave-one-out baseline	8	0.712	1.000	1.000

Table 5: Estimator ablation for the proper reward (Chronos-T5, GiftEval-Pretrain, $H = 64$, 2000 steps). "std"/"mean"/"batch": advantage normalisation; c: dispersion coefficient in equation 1; multi-seed rows give min–max across seeds.

Run	pretrain-corpus validation (vs. zero-shot)				GIFT-Eval short (vs. zero-shot, paired)					
	med. ratio	win	cov80	GMD ratio	WQL	CRPS	win	Δcov80	GMD ratio	n
std, $c = 1$ (uncorrected, 2 seeds)	0.980–0.988*	0.54–0.56	0.55–0.56	0.59–0.62	0.999–1.000	1.012–1.013	0.44–0.45	-0.08	0.73	55
std, $c = 1$, $\beta = 0$	0.988**	0.54	0.544	0.57	0.997	1.014	0.38	-0.096	0.70	55
std, $c = 1$, $G = 32$, $B = 4$	0.976**	0.63	0.610	0.73	0.984	0.992	0.60	-0.032	0.84	55
mean, $c = 1$	0.993*	0.54	0.599	0.76	0.992	0.998	0.49	-0.036	0.85	55
batch, $c = 1$ (normaliser only)	0.991**	0.54	0.592	0.75	0.995	1.003	0.47	-0.038	0.84	55
std, $c = 7/6$ (3 seeds)	0.976–0.979**	0.60–0.63	0.63–0.64	0.80–0.86	0.987–0.991	0.989–0.996	0.55–0.65	-0.02–0.00	0.92–0.97	55
mean, $c = 7/6$ (corrected)	0.993**	0.54	0.650	0.98	1.003	1.005	0.47	-0.005	1.00	55
batch, $c = 7/6$ (corrected, 3 seeds)	0.987–0.990**	0.55–0.59	0.64–0.66	0.92–0.96	0.993–1.002	0.995–1.002	0.45	-0.01–$+0.01$	0.97–1.03	55
T5-base: std, $c = 1$	0.969**	0.59	0.546	0.54	1.004	1.017	0.51	-0.091	0.68	55
T5-base: batch, $c = 7/6$ (corrected, 2 seeds)	0.969–0.981**	0.61	0.65	0.86–0.88	1.008–1.013	1.008–1.013	0.49–0.51	-0.011	0.96–0.99	55
T5-large: std, $c = 1$	0.980**	0.57	0.550	0.51	1.000	1.014	0.60	-0.092	0.68	55
T5-large: batch, $c = 7/6$ (corrected, 2 seeds)	0.980–0.985**	0.57–0.60	0.66–0.67	0.84–0.86	0.997–0.999	0.997–1.001	0.47–0.53	-0.02–$+0.01$	0.97–1.02	55

Conventions as in Table 3 (Wilcoxon $^{*}p < 0.05$, $^{**}p < 0.01$; on multi-seed rows the weakest seed); every row is paired with the zero-shot model of its own size.

(Δcov80 -0.091 and -0.092), the same null with the same fingerprint. The two mechanistic controls move as predicted: $G = 32$ (c_{eff} $0.43 \to 0.48$) shrinks less, as does mean-only normalisation.

The coefficient-corrected recipe ($c = 7/6$ with std normalisation, i.e. the coefficient half of the fix alone) restores dispersion. On held-out windows its median Gini ratio is 0.80 instead of 0.59 and its coverage 0.64 instead of 0.55; its median CRPS ratio is 0.978 (win rate 0.60, $p < 0.01$) against 0.988 uncorrected, and its mean validation CRPS is unchanged (4.297 vs. 4.289; zero-shot 4.448), so the correction costs none of the mean improvement. On GIFT-Eval it is the only RL run whose short-term WQL moves below one at unchanged calibration: 0.987 for seed 0 at Δcov80 -0.002, and 0.991/0.991 for seeds 1–2, against 1.000 at Δcov80 -0.078 for the same-seed uncorrected run. These effects are small and not significant (n.s.) across the 55 configurations ($p = 0.28, 0.21, 0.053$ by seed; Table 5). The mean- and batch-normalised corrected runs also keep calibration (GMD ratio 1.00) but learn more slowly (WQL 1.003 and 0.993–1.002), consistent with their smaller effective step in the toy analysis. A batch normaliser with $c = 1$ isolates the two halves of the fix: the coefficient bias alone leaves the held-out GMD ratio at 0.75 (the toy fixed point for $c_{\text{eff}} = 3/7$ is 0.762) and coverage at 0.59, between the uncorrected (0.54–0.56) and corrected (0.63–0.66) runs.

The coefficient-corrected run is thus comparable to the SFT control at matched budget (0.982/0.988 over two seeds, also n.s.; Table 6), but unlike SFT it leaves the coverage where the zero-shot model had it. Larger models behave alike in-corpus but not on the benchmark. The T5-base run with the same corrected estimator (batch, $c = 7/6$) has median ratio 0.981, win rate 0.61, coverage $0.63 \to 0.65$ and GMD ratio 0.88, against 0.54 for the uncorrected T5-base run (a second seed is alike; Table 5), yet is at 1.008 WQL on GIFT-Eval short-term (win rate 0.51, Δcov80 -0.011; second seed 1.013). T5-large repeats the pattern (two seeds: in-corpus 0.980/0.985 at GMD ratio 0.86/0.84; GIFT-Eval 0.997/0.999 at unchanged coverage). The correction therefore restores calibration at all three model sizes, whereas the transfer of the in-corpus gain to GIFT-Eval is partial for T5-small (std 0.987, batch 0.993) and absent for the larger models. The full pattern—fingerprint, restoration

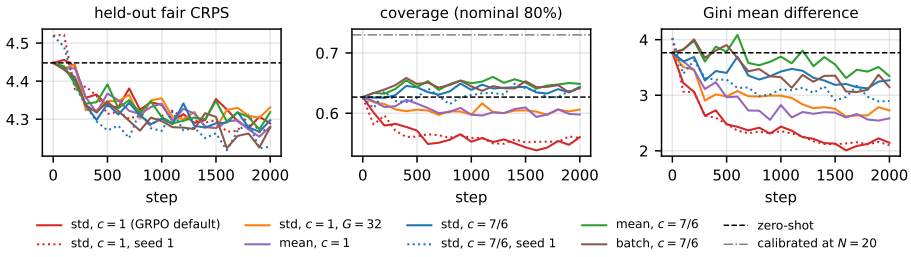


Figure 2: Training diagnostics on 256 held-out GiftEval-Pretrain windows ($N = 20$, evaluated every 100 steps) for the estimator ablation of Table 5. Every uncorrected run (red/orange/purple) drives the Gini mean difference and the coverage far below the zero-shot value (black dashed); the corrected coefficient (blue/green/brown) keeps both at the zero-shot level while lowering CRPS more.

Table 6: Transfer: SFT control at matched budget and the 2×2 corpus $\times$ horizon design (uncorrected estimator, $c = 1$, std; last row: corrected estimator, batch, $c = 7/6$). "GIFT": the GIFT-Eval training split; "H mix": $H \in \{8, \ldots, 64\}$. Both held-out validation corpora are reported so that every run is also judged on its own training corpus.

Run	pretrain-corpus validation (vs. zero-shot)				gift-corpus validation		GIFT-Eval short (vs. zero-shot, paired)					
	med. ratio	win	cov80	GMD ratio	med. ratio	win	WQL	CRPS	win	Δcov80	GMD ratio	n
SFT, pretrain, H=64	0.975**	0.59	0.690	1.16	1.004	0.48	0.982	0.979	0.58	+0.029	1.12	55
seed 2	0.985**	0.59	0.684	1.08	1.009	0.45	0.988	0.985	0.49	+0.028	1.11	55
SFT, GIFT, H mix	1.011	0.45	0.662	1.62	1.003	0.50	0.998	0.997	0.47	+0.026	1.55	55
GRPO, pretrain, H=64	0.988*	0.54	0.553	0.59	–	–	1.000	1.013	0.45	-0.078	0.73	55
GRPO, pretrain, H mix	0.992*	0.52	0.515	0.57	1.009	0.47	0.996	1.016	0.42	-0.120	0.64	55
GRPO, GIFT, H=64	1.004	0.49	0.516	0.59	0.988	0.54	0.999	1.017	0.49	-0.108	0.66	55
GRPO, GIFT, H mix	1.013	0.45	0.493	0.56	1.004	0.50	1.003	1.027	0.45	-0.136	0.60	55
GRPO corrected, GIFT, H mix	0.997	0.54	0.636	0.98	0.989*	0.56	0.992	0.992	0.55	+0.005	0.99	55

Conventions as in Table 3 (Wilcoxon $^*p < 0.05$, $^{**}p < 0.01$); every row is paired with the zero-shot model of its own size.

and the short-term transfer null—replicates on the masked-encoder Moirai-1.1-R family at all three sizes (App. D.5), where we also note a Moirai-specific, seed-stable medium/long-horizon gain.

5.4 Does RL transfer? SFT control, the 2×2 design and the budget

Table 6 asks whether the gains that RL does produce come from the RL objective or from having seen the data. The SFT control—the same windows, sampler, optimiser, budget and dropout treatment, but the native token cross-entropy—improves GIFT-Eval short-term WQL by 1.8% (win rate 0.58; second seed 1.2%) and *raises* coverage by 0.03, whereas every uncorrected GRPO cell of the 2×2 design is at 1.00 with reduced coverage. Neither the GIFT training split nor mixed horizons $\{8, \ldots, 64\}$ changes this. The most informative cell is GRPO on the GIFT training split with mixed horizons, on held-out windows of the *same* corpus: the mean validation CRPS falls by 8%, but the per-window median ratio is 1.004 with a win rate of 0.50 ($p = 0.96$). The mean improvement comes from a few large-scale windows, and the typical window is unchanged. Training-log improvements of the uncorrected recipe are therefore not evidence of a distributional improvement, and the "RL as a data-efficient adapter" hypothesis is not supported at this budget.

With the corrected estimator the same cell behaves differently. GRPO on the GIFT training split now improves the held-out windows of its own corpus (median ratio 0.989, win rate 0.56, $p = 0.03$) with unchanged coverage, transfers a smaller, non-significant gain to the pre-training corpus (0.997, $p = 0.06$), and improves GIFT-Eval short-term WQL by 0.8% (win rate 0.55, Δcov80 $+0.005$). It is the first RL cell of the design whose in-corpus and out-of-corpus effects have the same sign as the SFT control. The gains are in-distribution: no run improves a corpus it was not trained on by a significant margin (at most 0.5% at the median, all n.s.).

Tripling SFT to 6000 steps (Table 7) leaves its in-corpus median ratio at 0.975 (win rate $0.59 \rightarrow 0.61$): SFT is not budget-limited at 2000 steps. The corrected GRPO run is still improving at 6000 steps. Its in-corpus median ratio moves from 0.990 to 0.978 and its win rate from 0.55 to 0.64, with

Table 7: Training budget: the corrected recipe (batch, $c = 7/6$) and the SFT control at 2000 and 6000 steps (Chronos-T5-small, GiftEval-Pretrain, $H = 64$).

Run	pretrain-corpus validation (vs. zero-shot)				GIFT-Eval short (vs. zero-shot, paired)					
	med. ratio	win	cov80	GMD ratio	WQL	CRPS	win	Δcov80	GMD ratio	n
SFT, 2000 steps	0.975^{**}	0.59	0.690	1.16	0.982	0.979	0.58	+0.029	1.12	55
SFT, 6000 steps	0.975^{**}	0.61	0.684	1.08	1.000	0.997	0.47	+0.028	1.13	55
GRPO corrected, 2000 steps	0.990^{**}	0.55	0.643	0.92	0.993	0.995	0.45	-0.001	1.00	55
GRPO corrected, 6000 steps	0.978^{**}	0.64	0.654	0.93	1.003	1.004	0.44	+0.001	1.00	55

Conventions as in Table 3 (Wilcoxon $^{*}p < 0.05$, $^{**}p < 0.01$); every row is paired with the zero-shot model of its own size.

coverage (0.65) and GMD ratio (0.93) unchanged: it matches SFT's median ratio at a higher win rate and without SFT's over-dispersion, while on the held-out GIFT-split corpus it remains at 1.003 (n.s.). On GIFT-Eval short-term the longer runs are *worse*: SFT goes from 0.982 to 1.000 WQL (win rate $0.58 \to 0.47$) and the corrected GRPO run from 0.993 to 1.003 ($0.45 \to 0.44$). The additional in-corpus gain is thus bought by specialising to the pre-training corpus, and the 2000-step budget is already past the GIFT-Eval optimum for both recipes. On the validation windows, $\mathrm{KL}(\pi_{\text{post}} \,\|\, \pi_0)$ is 0.012/0.013 nats per token for the corrected RL runs (std/batch) against 0.025 for the SFT control at the same budget and gain (0.040 at 6000 steps): on-policy RL reaches the same gain at half the divergence, matching its KL-minimal bias (Shenfeld et al., 2025).

5.5 Test-time compute: self-backtest verifiers have low fidelity

The self-backtest is the only verifier available at test time; Table 12 (App. D) measures its worth. Across 15 GIFT-Eval configurations and 5 synthetic families, every backtest-driven intervention is worse than the zero-shot model: conformal rescaling by 7% (GIFT geometric mean), bias correction by 17%, selection among 24 sampling strategies by 9% and softmax weighting by 12%; the synthetic families repeat the pattern. The last column shows why. The Spearman correlation between the backtest CRPS and the realised CRPS of a strategy is 0.13 on average (Figure 1b), so selection is close to random (13% worse); only one configuration (covid deaths, $\rho = 0.35$) benefits, and the other configuration with $\rho > 0.3$ is harmed by all three interventions. The oracle column is 26% better, but this reflects the optimistic bias of choosing among sampling-noise realisations rather than an attainable ceiling: the zero-shot default (full context, $T = 1$) is already the best strategy on average, so there is nothing for a verifier to find. Averaging more backtest tails or learning the verifier does not change this (App. D.6). Test-time compute in the LLM sense requires a verifier whose fidelity is far above what a series' own past provides, as the theory of imperfect verifiers predicts (Setlur et al., 2025; Stroebl et al., 2024; Dorner et al., 2025).

5.6 Where does post-training change the forecast? Stratified analysis

Table 13 (App. D) stratifies the per-window CRPS change of the uncorrected and corrected proper-reward runs and of the SFT control by term, domain and series features, on the 27,853 windows (93 configurations) evaluated for all three. Three patterns stand out. (i) No model's effect is predicted by any feature: every within-configuration rank correlation in Table 14 is below 0.03 in magnitude, so the hypothesis that RL helps most on high-entropy or non-stationary series has no support. (ii) Nothing improves beyond the training horizon. The uncorrected run is neutral on short-term windows (-0.1%) and worse on medium- and long-term ones ($+2.3\%$, $+3.0\%$), where its reduced dispersion is penalised; the corrected run improves short-term windows by 2.4% and is neutral to slightly worse beyond (-0.8%, n.s.; $+1.3\%$); the SFT control is at -3.5%, -0.9% and $+0.6\%$. (iii) The strata that separate winners from losers are the *same* for all three models: strongly seasonal, low-trend and energy windows get worse, windows with a single change-point, a strong trend, or from healthcare and transport get better, and the size of the adverse strata is ordered uncorrected $>$ corrected $>$ SFT. The stratification thus supports the estimator ablation's conclusion: the recipe's effect is a global change in dispersion (a uniform penalty where the zero-shot model was already calibrated) plus the corpus effect that SFT also produces, not a learned improvement on identifiable series types.

6 Related work

Time-series foundation models. Chronos (Ansari et al., 2024) quantises scaled values and trains a T5 language model on them; Moirai (Woo et al., 2024) outputs a per-step parametric mixture density; Chronos-Bolt and Chronos-2 (Ansari et al., 2025) instead output quantiles directly, as does TimesFM (Das et al., 2024). Only a sampling distribution defines a trajectory-level policy that RL can act on, which is why we post-train Chronos (and Moirai as a replication family, App. D.5); the quantile models serve as strong baselines. GIFT-Eval (Aksu et al., 2024) is the standard benchmark and provides the seasonal-naive normalisation we adopt.

Proper scoring rules and CRPS estimation. CRPS is a strictly proper scoring rule and equals the Cramér distance (Gneiting & Raftery, 2007); the "fair" finite-sample estimator (Fricker et al., 2013; Ferro, 2014; Zamo & Naveau, 2018) removes the $O(1/N)$ ensemble-size bias (Ferro et al., 2008) that Prop. 1 quantifies. The energy score generalises CRPS to trajectories (Gneiting & Raftery, 2007); we score per horizon step and average, which keeps every step proper.

RL with verifiable rewards and GRPO. GRPO (Shao et al., 2024) replaces a learned critic (Schulman et al., 2016) by the leave-one-out group baseline of RLOO (Kool et al., 2019; Ahmadian et al., 2024). The known pathologies of its normalisation concern per-trajectory rewards: length and difficulty biases (Liu et al., 2025), and an overconfidence under stochastic outcomes that removing the per-group std fixes (Bereket & Leskovec, 2025). Our Cor. 1 is different in kind: with a pairwise (set-level) reward even the group-mean baseline is biased, and the coefficient must change too. Where the score is differentiable it can be trained on directly: AIFS-CRPS (Lang et al., 2024) trains an ensemble weather model on an almost-fair CRPS whose coefficient plays the role of our c, and Black et al. (2024) shape distributional rewards for diffusion models. For token-AR forecasters only the policy-gradient route remains, and concurrent work takes it: GRPO on a TSFM with accuracy rewards plus dispersion-restoring shaping (Li et al., 2026), on LLM-revised forecasts (Liu et al., 2026), and with a set-marginal CRPS reward for LLM regression (Park et al., 2026), which App. A shows is the pairwise reward in disguise. None analyses the estimator, whose bias every such reward inherits. Whether RL adds capability or reweights the base model's support is debated (Yue et al., 2025; Chu et al., 2025); our SFT control speaks to the same question for forecasting.

Test-time compute for forecasting. Conformal prediction and backtest-based recalibration are classical (Vovk et al., 2005; Stankeviciute et al., 2021); sampling-strategy selection by a verifier is the LLM analogue (Cobbe et al., 2021; Snell et al., 2024). We measure the self-backtest's fidelity directly and find it too low to support selection.

7 Conclusion

Forecasting is verifiable, but the verifier grades a distribution, and that changes both halves of the RLVR recipe. On the training side, the only CRPS-consistent per-sample reward is pairwise, and a pairwise reward is incompatible with GRPO's group-relative advantages: the group mean lowers the dispersion coefficient, and the per-group standard deviation adds a shrinkage that survives any group size. The resulting signature—halved dispersion, coverage 0.10 below the zero-shot model, no benchmark gain—appeared in every uncorrected run across two model families and three model sizes, and disappears with a two-line change of coefficient and normaliser. Once corrected, RL yields what a matched SFT run yields on the same data, namely in-corpus gains that do not transfer across corpora and move the benchmark, within noise, only for T5-small; unlike SFT it keeps the dispersion calibrated, at half the KL divergence. At inference, more samples always help by exactly $\mathrm{GMD}/(2N)$, sharpening helps only below a crossover the formula locates, and a single-window self-backtest ($\rho \approx 0.1$) is too weak a verifier to buy anything. The estimator bias is not specific to forecasting: any pairwise or set-level reward (diversity, coverage, energy scores) interacts with group-relative advantages in the same way, and the same correction applies.

Reproducibility Statement

All code and result files are released at `https://anonymous.4open.science/r/verifiable-forecasting-72ED`. The repository contains the full implementation (the GRPO trainer with every estimator variant, the pairwise reward and its ablations, the sampling policies for both model families, and the test-time-compute study), the unit tests and toy fixed-point scripts behind the theoretical claims, and the exact launch scripts—with all hyperparameters and seeds—for every training and evaluation run in the paper. The per-configuration result files behind every table and figure are included: `scripts/paper_tables.py` and `scripts/paper_figs.py` regenerate all tables and figures from the shipped files without any GPU work. All datasets and checkpoints are public (GIFT-Eval and Hugging Face); the repository's README documents the pinned third-party commits and the corpus-building steps. Complete proofs of all propositions are in Appendix A.

AI Use Statement

In this work, we have not used generative AI tools for any of the tasks with required disclosure: the research questions, hypotheses, theoretical results and their proofs, the methods and their implementation, the experimental design, the synthetic data generators, data processing and the interpretation of results are the authors' own; translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools (large language models) for two tasks with recommended disclosure: (i) editing the manuscript to improve readability (grammar, wording and concision of author-written text), and (ii) reviewing author-written code and experiment outputs for potential errors. We have reviewed all AI-assisted work: every suggested wording change was checked by the authors to ensure it preserves the intended technical meaning, and every issue flagged in code review was independently verified by the authors before any fix, with the corrected code retested through the unit tests and scripts released with the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of ACL*, 2024.

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

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.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Kuken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The Cramér distance as a solution to biased Wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Michael Bereket and Jure Leskovec. Uncalibrated reasoning: GRPO induces overconfidence for stochastic outcomes. *arXiv preprint arXiv:2508.11800*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Florian E. Dorner, Yatong Chen, André F. Cruz, and Fanny Yang. ROC-n-reroll: How verifier imperfection affects test-time scaling. *arXiv preprint arXiv:2507.12399*, 2025.

Christopher A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014.

Christopher A. T. Ferro, David S. Richardson, and Andreas P. Weigel. On the effect of ensemble size on the discrete and continuous ranked probability scores. *Meteorological Applications*, 15 (1):19–24, 2008.

Thomas E. Fricker, Christopher A. T. Ferro, and David B. Stephenson. Three recommendations for evaluating climate predictions. *Meteorological Applications*, 20(2):246–255, 2013.

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

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Workshop on Deep RL Meets Structured Prediction*, 2019.

Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D. Dueben, et al. AIFS-CRPS: Ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *arXiv preprint arXiv:2412.15832*, 2024.

Siyang Li, Yize Chen, Zijie Zhu, Yuxin Pan, Yan Guo, Ming Huang, and Hui Xiong. TimeRFT: Stimulating generalizable time series forecasting for TSFMs via reinforcement finetuning. *arXiv preprint arXiv:2605.00015*, 2026.

Haoxin Liu, Yichen Zhou, Rajat Sen, B. Aditya Prakash, and Abhimanyu Das. Rethinking post-training recipes for multimodal time-series forecasting. *arXiv preprint arXiv:2605.29401*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976.

Jungsoo Park, Hyungjoo Chae, Ethan Mendes, Jay DeYoung, Varsha Kishore, Wei Xu, and Alan Ritter. Distribution-aware reward: Reinforcement learning over predictive distributions for LLM regression. *arXiv preprint arXiv:2605.20740*, 2026.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. Scaling test-time compute without verification or RL is suboptimal. In *International Conference on Machine Learning*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Kamile Stankeviciute, Ahmed M. Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Benedikt Stroebl, Sayash Kapoor, and Arvind Narayanan. The limits of inference scaling through resampling. *arXiv preprint arXiv:2411.17501*, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

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*, 2024.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Michaël Zamo and Philippe Naveau. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Mathematical Geosciences*, 50(2):209–234, 2018.

A PROOFS

Proposition 1. Let $x_1, \dots, x_N \sim F_\theta$ be i.i.d. and F_N their empirical law, so that $\mathrm{CRPS}(F_N, y) = \frac{1}{N}\sum_i |x_i - y| - \frac{1}{2N^2}\sum_{i,j} |x_i - x_j|$. Taking expectations, the first term is $\mathbb{E}|X - y|$ and the second is $\frac{1}{2N^2}\, N(N-1)\,\mathbb{E}|X - X'|$, because the N diagonal terms vanish. Hence $\mathbb{E}\,\mathrm{CRPS}(F_N, y) = \mathbb{E}|X - y| - \frac{1}{2}(1 - \frac{1}{N})\mathrm{GMD} = \mathrm{CRPS}(F_\theta, y) + \mathrm{GMD}/(2N)$. □

Proposition 2. $\nabla\mathbb{E}_x|x-y| = \mathbb{E}[|x-y|\nabla \log \pi(x)]$ by the score-function identity. For the pairwise term, $\nabla\mathbb{E}_{x,x'}|x - x'| = \mathbb{E}\big[|x - x'|(\nabla \log \pi(x) + \nabla \log \pi(x'))\big] = 2\,\mathbb{E}_x\big[\mathbb{E}_{x'}|x - x'|\,\nabla \log \pi(x)\big]$ by symmetry. Multiplying by $\frac{1}{2}$ gives the stated coefficient 1. □

Corollary 1. Write $D_i = \frac{1}{G-1}\sum_{j\neq i} |x_i - x_j|$, $r_i = -(|x_i - y| - cD_i)$ and $\bar{r} = \frac{1}{G}\sum_j r_j$. For any function f of the group, $\mathbb{E}[f \sum_i \nabla \log \pi(x_i)] = \nabla\mathbb{E}[f]$, since $\sum_i \nabla \log \pi(x_i)$ is the score of the product density. Applied to $f = \bar{r}$ this gives $\mathbb{E}[\sum_i \bar{r}\nabla \log \pi(x_i)] = \nabla\mathbb{E}[\bar{r}] = -\nabla\big(\mathbb{E}|X - y| - c\,\mathrm{GMD}\big)$, because $\mathbb{E}[D_i] = \mathrm{GMD}$ and *all* of its dependence on θ is differentiated. For the reward

term, the argument of Prop. 2 gives $\mathbb{E}[r_i \nabla \log \pi(x_i)] = -\nabla\big(\mathbb{E}|X-y| - \frac{c}{2}\mathrm{GMD}\big)$ for each i: only the law of x_i is differentiated, which yields half of $\nabla\mathrm{GMD}$. Summing over i and subtracting,

$$\mathbb{E}\Big[\sum_i (r_i - \bar{r})\nabla \log \pi(x_i)\Big] = -G\nabla\big(\mathbb{E}|X-y| - \tfrac{c}{2}\mathrm{GMD}\big) + \nabla\big(\mathbb{E}|X-y| - c\,\mathrm{GMD}\big)$$
$$= -(G-1)\nabla\Big[\mathbb{E}|X-y| - c\,\tfrac{G-2}{2(G-1)}\mathrm{GMD}\Big].$$

The effective coefficient equals $\frac{1}{2}$ iff $c = \frac{G-1}{G-2}$. For a Gaussian policy with $y \sim \mathcal{N}(0,1)$, $\mathbb{E}|X-y| = \sqrt{2(1+\sigma^2)/\pi}$ and $\mathrm{GMD} = 2\sigma/\sqrt{\pi}$, and the stationary point of $\mathbb{E}|X-y| - c_{\text{eff}}\mathrm{GMD}$ in σ is $\sigma^{\star 2} = 2c_{\text{eff}}^2/(1-2c_{\text{eff}}^2)$, which gives 0.762 at $G = 8$ ($c = 1$). □

Per-group standard deviation. With $a_i = (r_i - \bar{r})/s$ the expected gradient is $\mathbb{E}[1/s]\,\mathbb{E}[g] + \mathrm{Cov}(1/s, g)$, where $g = \sum_i (r_i - \bar{r})\nabla \log \pi(x_i)$. The covariance term has no closed form. Numerically (Table 4), groups whose realised y lies inside the sample range have a small s and a contracting g, whereas groups with y outside have a large s and an expanding g, so $1/s$ down-weights precisely the expanding groups. At the calibrated optimum the expected $\partial/\partial \log\sigma$ is -1.77 per group at $G = 8$ and -6.4 at $G = 32$ (the unbiased value is 0), so the relative bias does not vanish with G.

Set-marginal rewards. A natural alternative credit scores each sample by its marginal effect on the set-level score, $r_k = S_{-k} - S_G$, as Park et al. (2026) do for LLM regression. Writing $a_k = |x_k - y|$, $A = \sum_i a_i$, $P_k = \sum_{j \neq k} |x_k - x_j|$ and $P = \sum_k P_k$, the fair estimator $S_G = A/G - P/(2G(G-1))$ gives

$$r_k = -\tfrac{1}{G-1}\Big(a_k - \tfrac{G-1}{G-2}\,\tfrac{P_k}{G-1}\Big) + \tfrac{1}{G(G-1)}\Big(A - \tfrac{P}{G-2}\Big),$$

i.e. exactly the *corrected* pairwise reward equation 1 with $c = (G-1)/(G-2)$, up to a positive scale and a group constant (verified to 10^{-12} against the trainer's reward code). The standard estimator ($P/(2G^2)$ in place of $P/(2G(G-1))$) gives the *uncorrected* $c = 1$ reward instead. Group-relative advantages remove the constant, so the fair set-marginal reward inherits the corrected coefficient automatically; but the per-group std then produces advantages *identical* to those of the corrected pairwise reward, so the $O(1)$ shrinkage of Table 4 applies verbatim. Numerically (protocol of Table 4, $G = 8$): fair set-marginal $\sigma^\star = 0.712$ (std) vs. 1.000 (mean; theory 1); standard set-marginal 0.497 (std) vs. 0.762 (mean; theory 0.762).

B Implementation details

Policy and tokenisation. Chronos-T5 scales each context by its mean absolute value and quantises it into 4094 uniform bins on $[-15, 15]$ plus PAD/EOS. We generate with temperature 1, no top-k/top-p truncation and the two special tokens masked out of the softmax ("exact sampling"). The Chronos default instead uses top-$k = 50$ without masking, so PAD/EOS tokens occasionally decode to the lowest bin. Log-probabilities for the policy-gradient loss are recomputed by teacher forcing on the sampled tokens. Dropout is disabled in both the rollout and the teacher-forced pass, so that the two coincide and the KL to the frozen reference is exactly 0 at initialisation.

Reward and advantages. For a window with context c and target $y_{1:H}$, each of the G rollouts $x^{(i)}_{1:H}$ receives the per-step reward of equation 1, averaged over h and divided by the seasonal-naive MAE of the context (the scale MASE uses), with missing targets masked. Advantages are $r_i - \bar{r}$ divided by the per-group standard deviation ("std"), by nothing ("mean"), or by the standard deviation over the whole batch of $B \times G$ rollouts ("batch"). The loss is the PPO-clipped surrogate ($\epsilon = 0.2$) with a single optimisation step per batch, plus $\beta\, k_3$ KL. Windows are sampled uniformly over series and positions, with context length uniform in $[64, 512]$ and horizon $H = 64$ (or uniform over $\{8, 12, 16, 24, 32, 48, 64\}$ for the mixed-horizon runs).

Evaluation. GIFT-Eval configurations are evaluated with the official window definitions. The paired subsets take a fixed random subset (seed 0) of 512 test windows per configuration (all windows when fewer), identical for all models. WQL uses the nine quantiles $0.1, \ldots, 0.9$ interpolated

from the N samples, CRPS is the fair estimator, and MASE uses the median. All are normalised per configuration by the seasonal-naive value from the official leaderboard and aggregated by geometric mean. The in-corpus validation uses 256 windows drawn with a fixed seed from series of the training corpus, at positions never used for training; because the series themselves were seen, we treat it as a training diagnostic only. SynthOracle draws 200 series from each of five generative families (AR with seasonality, GP mixture, regime-switching AR, GARCH, level shift) whose conditional predictive law is available in closed form or by exact simulation, and reports the fair CRPS of the model minus that of the oracle.

Test-time compute. The backtest forecasts the last $k = H$ observed steps from the preceding context. The scale correction rescales the samples about their median by the split-conformal factor that makes the backtest's 80% interval cover its nominal fraction (nonconformity $|e|$/half-width, factor clipped to $[0.5, 4]$). The bias correction shifts them by the backtest residual mean shrunk towards zero by two standard errors (no shift when the mean is within two standard errors). Selection evaluates 24 strategies (context $\in \{64, 128, 256, 512\} \times$ temperature $\in \{0.8, 1.0, 1.2\} \times \{$zero-shot, RL$\}$) on the backtest and either selects the best or mixes their samples with weights $\propto \exp(R_{\mathrm{bt}}/\tau)$, $\tau = 0.05$; the oracle selects with the realised future. ρ is the Spearman correlation across the 24 strategies between backtest and realised CRPS, averaged over windows.

Compute. All runs use a single 40 GB GPU. A 2000-step GRPO run (32k windows, 256k rollouts) took 1.5–3 h of training time for T5-small, about 4 h for T5-base and 3 h for T5-large; a Moirai run takes 7–15 min on a shared GPU (one forward pass per step, no autoregressive rollout), excluding periodic evaluation. The paired GIFT-Eval subset takes 2–3 h per model at $N = 20$.

C Limitations

Our post-training budget (2000 steps, 32k windows) is small relative to pre-training. The estimator bias, its correction and the benchmark-transfer null all replicate on a second, architecturally disjoint family (Moirai's mixture-density head, three model sizes; App. D.5), so the mechanism is not an artefact of token quantisation; every effect size, however, remains family- and budget-specific. Window-level bootstrap intervals pool windows and ignore clustering by configuration. A configuration-level cluster bootstrap (App. D) removes the significance of every stratum-level effect, so the configuration-level Wilcoxon tests reported in the tables are the operative reading, and by that reading the post-training gains on GIFT-Eval are not significant.

D Additional results

D.1 SynthOracle: the three gaps on synthetic families

On families with known conditional laws, the zero-shot Chronos models are *over*-dispersed relative to the oracle (dispersion ratio 1.32–1.46), and their Cramér gap is dominated by the level-shift and GP-mixture families. The uncorrected GRPO run brings the dispersion ratio to 1.05, i.e. to the calibrated value on average, yet its Cramér gap is *larger* (0.315 vs. 0.282) and its coverage lower (0.68 vs. 0.75). The shrinkage is a global rescaling that is not conditioned on the series: it under-disperses exactly the windows where the base model was calibrated and leaves the over-dispersed ones over-dispersed. The normaliser-only run (batch, $c = 1$) shows the intermediate signature the theory predicts: dispersion 1.15 and coverage 0.70, partway from the uncorrected 1.05/0.68 towards the zero-shot 1.37/0.75.

The SFT control moves in the opposite direction. Its mean Cramér gap is unchanged (0.283; no family moves by more than 0.02), but its dispersion ratio rises from 1.37 to 1.57 (second seed 1.53; 1.60 after 6000 steps, at an unchanged gap of 0.279), and to 3.3 when trained on the GIFT training split with mixed horizons, an extreme that is mostly an EOS artefact (masked: 1.46; App. D.4). Cross-entropy fine-tuning at this budget widens an already over-dispersed model.

Neither recipe reduces the information gap: on families with a known conditional law the post-trained models are at best as far from the oracle as the zero-shot model, and differ from it only in how their spread is scaled. The corrected runs make this explicit. The std run keeps the zero-

Table 8: SynthOracle: five synthetic families with known conditional laws (200 series each, $H = 32$, $N = 64$ samples). Cramér gap = model fair CRPS − oracle CRPS (optimisation + information gap, exact up to MC noise); dispersion ratio = model GMD / oracle GMD (1 = calibrated spread); coverage of the nominal 80% interval (oracle ≈ 0.80).

Model	ar_seasonal	gp_mix	regime_ar	garch	level_shift	mean gap	disp. ratio	cov80
T5-tiny (zero-shot)	0.161	0.556	0.114	0.065	0.903	0.360	1.358	0.744
T5-mini (zero-shot)	0.115	0.475	0.157	0.067	0.780	0.319	1.324	0.735
T5-small (zero-shot)	0.108	0.436	0.088	0.054	0.724	0.282	1.370	0.752
T5-base (zero-shot)	0.130	0.397	0.129	0.063	0.685	0.281	1.456	0.765
T5-small + SFT	0.101	0.418	0.096	0.064	0.733	0.283	1.567	0.789
seed 2	0.107	0.403	0.100	0.050	0.738	0.280	1.532	0.791
6000 steps	0.120	0.429	0.097	0.083	0.668	0.279	1.601	0.791
T5-small + SFT (GIFT split, H mix)	0.127	0.443	0.142	0.185	0.941	0.368	3.287	0.774
T5-small + GRPO proper (std, c=1)	0.107	0.508	0.135	0.073	0.754	0.315	1.050	0.679
seed 1	0.115	0.501	0.125	0.057	0.669	0.294	1.063	0.681
H mix	0.129	0.524	0.150	0.060	0.722	0.317	0.949	0.636
T5-small + GRPO half	0.160	0.616	0.209	0.185	0.892	0.412	0.704	0.507
T5-small + GRPO naive	0.213	0.731	0.288	0.219	0.995	0.489	0.522	0.399
T5-small + GRPO MSE	0.227	0.759	0.312	0.252	1.038	0.518	0.528	0.394
T5-small + GRPO proper (std, c=1, G=32)	0.110	0.472	0.105	0.055	0.749	0.298	1.205	0.723
T5-small + GRPO proper (mean, c=1)	0.119	0.530	0.103	0.077	0.739	0.314	1.214	0.718
T5-small + GRPO proper (batch, c=1)	0.104	0.526	0.123	0.081	0.740	0.315	1.154	0.702
T5-small + GRPO proper (std, c=7/6)	0.104	0.456	0.096	0.029	0.778	0.293	1.344	0.753
seed 1	0.115	0.446	0.104	0.047	0.707	0.284	1.294	0.736
seed 2	0.113	0.451	0.101	0.054	0.732	0.290	1.289	0.741
T5-small + GRPO proper (mean, c=7/6)	0.104	0.548	0.111	0.087	0.792	0.328	1.381	0.754
T5-small + GRPO proper (batch, c=7/6)	0.110	0.504	0.099	0.073	0.758	0.309	1.369	0.756
seed 1	0.108	0.486	0.103	0.048	0.708	0.291	1.387	0.749
seed 2	0.107	0.453	0.099	0.069	0.806	0.307	1.444	0.771
T5-base + GRPO proper (std, c=1)	0.145	0.487	0.177	0.097	0.652	0.311	1.055	0.673
T5-base + GRPO proper (batch, c=7/6)	0.137	0.514	0.155	0.101	0.679	0.317	1.386	0.749

shot dispersion ratio (1.34; seeds 1–2: 1.29, 1.29) and coverage (0.75; 0.74, 0.74), and its mean Cramér gap (0.293; 0.284, 0.290) is essentially the zero-shot value (0.282), against 0.315 for the uncorrected run. The mean- and batch-normalised corrected runs also keep the zero-shot dispersion (1.38, 1.37) and coverage (0.75, 0.76), but their gaps (0.328, 0.309; batch seeds 1–2: 0.291, 0.307) lie above the zero-shot value, with about half of the increase coming from the GP-mixture family (0.548 and 0.504 vs. 0.436). No corrected run moves closer to the oracle on any family by more than the Monte-Carlo standard error, and the largest movement is again a loss.

The larger models repeat both signatures. The uncorrected T5-base run carries the small-model fingerprint (dispersion 1.06, coverage 0.67, gap 0.322 against 0.281), and the corrected T5-base run keeps dispersion (1.43 vs. 1.46) and coverage (0.75 vs. 0.77) at the zero-shot values while its mean gap rises from 0.281 to 0.315, again mainly through the GP-mixture family (0.498 vs. 0.397). For T5-large the uncorrected run lowers the dispersion ratio on every family (-18% to -35%; garch $1.27 \to 0.83$, level-shift $2.36 \to 1.64$) with coverage dropping throughout, whereas the two corrected seeds stay within 20% of the zero-shot dispersion on every family and within 0.04 of its coverage.

D.2 GIFT-Eval on the paired subsets

Table 9 reports every model's seasonal-naive-normalised WQL and MASE on the paired 512-window subsets, aggregated by geometric mean within the short-, medium- and long-term configurations and overall. The subsets are drawn with a fixed seed from the official windows, so rows are comparable with each other but not with the official leaderboard, which scores all windows. The zero-shot ordering is the expected one (Chronos-2, then Bolt-small ≈ Bolt-base, then T5-base, then T5-small; exact sampling beats the default top-k sampler by 2.9% short-term WQL). No post-trained T5 model reaches Bolt-small: the best post-trained T5-small rows (SFT 0.585; GRPO with the corrected coefficient, 0.588) close a quarter and a sixth of the 0.044 short-term gap to Bolt-small, and the corrected T5-base run (0.581) is within noise of its zero-shot model (0.576). On the medium- and long-term configurations the corrected T5-base run is clearly *worse* than its zero-shot model (WQL 0.745/0.770 vs. 0.689/0.682): training on $H = 64$ windows does not transfer to the much longer horizons of those configurations.

Table 9: GIFT-Eval on paired 512-window subsets of all 97 configurations (identical windows for every model). Geometric mean over configurations of WQL and MASE normalised by the seasonal-naive baseline (leaderboard convention; lower is better) and mean 80% coverage. Sample-based models use $N = 20$ samples.

Model	short		medium		long		all		cov80
	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	
Chronos-T5-small (exact sampling)	0.596	0.743	0.701	0.945	0.705	0.972	0.640	0.829	0.450
Chronos-T5-small (default top-k=50)	0.613	0.742	0.745	0.975	0.745	1.027	0.667	0.844	0.387
Chronos-T5-base (exact)	0.576	0.725	0.689	0.952	0.682	0.940	0.621	0.814	0.457
Chronos-Bolt-small	0.552	0.703	0.594	0.845	0.592	0.905	0.569	0.773	0.680
Chronos-Bolt-base	0.557	0.696	0.575	0.820	0.578	0.869	0.565	0.757	0.688
Chronos-2	0.515	0.638	0.469	0.661	0.483	0.694	0.498	0.655	0.753
T5-small + SFT (pretrain corpus)	0.585	0.741	0.685	0.947	0.705	0.988	0.630	0.832	0.488
T5-small + GRPO proper (c=1, std)	0.596	0.741	0.706	0.929	0.721	0.968	0.644	0.824	0.383
T5-small + GRPO proper (c fixed, std)	0.588	0.736	0.704	0.945	0.719	0.968	0.639	0.824	0.451
T5-small + GRPO proper (corrected, mean)	0.597	0.753	0.734	0.995	0.746	1.027	0.656	0.855	0.457
T5-small + GRPO proper (corrected, batch)	0.591	0.744	0.728	0.970	0.716	0.994	0.645	0.839	0.461
T5-base + GRPO proper (c=1, std)	0.578	0.714	0.730	0.944	0.726	0.978	0.639	0.812	0.381
T5-base + GRPO proper (corrected, batch)	0.581	0.732	0.741	0.995	0.766	1.038	0.650	0.844	0.457

D.3 Series-level held-out validation

The in-corpus validation of Section 4 draws held-out *windows* from series the training sampler could also touch. Replaying the training sampler (deterministic given the seed; `scripts/replay_train_series.py`) shows that all seed-0 runs saw the identical 18,711 of the corpus' 31,585 series, so we re-evaluate all of them on 256 windows drawn only from the 12,874 series no run ever sampled. The ordering and the fingerprints survive. The coefficient-corrected run keeps a per-window median CRPS ratio of 0.987 (win rate 0.57, Wilcoxon $p = 0.001$) at unchanged coverage (0.64 vs. zero-shot 0.64), the batch-corrected run 0.994 ($p = 0.02$, coverage 0.65), and the SFT control 0.989 ($p = 0.045$) with its familiar over-dispersion (coverage 0.68, GMD ratio 1.08). The uncorrected run's apparent gain vanishes entirely (0.999, $p = 0.44$) while its shrinkage persists (coverage 0.55, median GMD ratio 0.65): the shrinkage is a global property of the policy, whereas the gain was partly series-specific. Compared with the in-sample validation the median ratios shift by about $+0.01$ for every model, i.e. roughly one point of the apparent in-corpus gain is series-level memorisation, and the rest transfers to unseen series of the same corpus.

D.4 The EOS confound, removed

At $H < 64$ the mixed-horizon SFT control learns, alongside the values, *where to stop*: the cross-entropy trains the placement of Chronos' end-of-sequence token, an objective no RL run faces because rollouts have fixed length. We retrain the run identically with the EOS token masked from the loss (`--use_eos false` in `scripts/train_sft.py`). Every over-dispersion fingerprint collapses to the zero-shot level. The mean dispersion ratio on the synthetic families drops from 3.3 to 1.46 (garch 7.9 $\rightarrow$ 1.4, level-shift 3.7 $\rightarrow$ 2.4, GP-mixture 2.1 $\rightarrow$ 1.4), inside the zero-shot range of App. D.1, and on the GIFT-split validation windows the median GMD ratio against zero-shot falls from 1.74 to 0.96. The scores move slightly in its favour: the in-corpus median CRPS ratio goes from 1.003 ($p = 0.31$) to 0.992 ($p = 0.15$), and on the 97 GIFT-Eval configurations the geometric-mean WQL against zero-shot goes from 1.008 to 0.992 (fair CRPS 1.001 $\rightarrow$ 0.988), with per-configuration dispersion at $0.56\times$ its EOS-trained value and cov80 -0.01. Masking the token is free or mildly helpful. The confound explains the extra dispersion of the mixed-horizon SFT control, not its null transfer.

D.5 A second model family: Moirai

The analysis of Section 3 is estimator-level; whether its predicted effects appear in a different architecture is an empirical question. We repeat the protocol on Moirai-1.1-R (Woo et al., 2024) at all three released sizes (small 14M, base 91M, large 311M). Moirai is a masked-encoder model whose output is a per-step mixture density (Student-t, Normal, negative binomial, log-normal): a continuous policy with no tokeniser, no EOS and no autoregressive loop, in which sampling draws from the predictive mixture, $\log \pi_\theta$ is the mixture log-density of the drawn value, and each GRPO step

Table 10: Moirai-1.1-R: estimator conditions at three model sizes on the pretrain-corpus validation windows (vs. same-size zero-shot): median paired CRPS ratio, win rate, mean coverage, median per-window GMD ratio, full-run median k_3 KL per step. Small: three seeds at lr $= 2 \times 10^{-6}$ (plus two KL-matched runs at 5×10^{-7}); base/large: two seeds at 5×10^{-7}; ranges span seeds. Zero-shot cov80: small 0.730, base 0.734, large 0.708.

Size	Estimator	med. ratio	win	cov80	GMD ratio	KL
small	std, $c = 1$ (uncorrected)	0.972–0.984	0.55–0.59	0.61–0.62	0.63–0.64	0.11–0.12
small	std, $c = 7/6$	0.984–0.987	0.60–0.64	0.68–0.69	0.86–0.88	0.05–0.06
small	batch, $c = 7/6$ (corrected)	0.989–0.992	0.58–0.62	0.71–0.72	0.92–0.97	0.04–0.05
small	std, $c = 1$, lr 5×10^{-7}	0.985	0.60	0.63	0.73	0.030
small	batch, $c = 7/6$, lr 5×10^{-7}	0.991	0.62	0.71	0.94	0.007
base	std, $c = 1$ (uncorrected)	0.973–0.986	0.59–0.63	0.64–0.65	0.63–0.64	0.11
base	std, $c = 7/6$	0.976–0.987	0.61–0.67	0.73	0.90	0.05–0.06
base	batch, $c = 7/6$ (corrected)	0.991–0.992	0.57–0.61	0.74–0.75	0.99–1.01	0.05
large	std, $c = 1$ (uncorrected)	0.957–0.962	0.63–0.67	0.64	0.61–0.64	0.16
large	std, $c = 7/6$	0.955–0.956	0.66–0.73	0.72	0.86–0.87	0.11
large	batch, $c = 7/6$ (corrected)	0.969–0.972	0.68–0.69	0.73–0.74	0.96	0.09

Every row is significant (log-ratio Wilcoxon $p < 10^{-3}$; $n = 256$ windows shared across all policies of a size); the base/large probes at lr 2×10^{-6} (one seed) show the same ordering at higher KL.

Table 11: Moirai-1.1-R on GIFT-Eval, paired per configuration against the same-size zero-shot model ($N = 20$ samples, up to 512 windows per configuration; two seeds per condition, ranges span seeds). Left: the 55 short-term configurations (the aperture of the Chronos transfer claims); right: all 97. gWQL: geometric-mean WQL ratio; p: configuration-level Wilcoxon on log WQL ratios (weakest seed); Δcov80: mean coverage change; GMD ratio: geometric mean of per-configuration ratios. Zero-shot cov80 (all 97): small 0.755, base 0.736, large 0.741.

		short (55 configs)				all (97 configs)	
Size	Estimator	gWQL	p	Δcov80	GMD ratio	gWQL	Δcov80
small	std, $c = 1$ (uncorrected)	0.986–0.988	0.08	-0.124–-0.121	0.67–0.68	0.931–0.937	-0.141–-0.136
small	std, $c = 7/6$	0.988–1.000	0.35	-0.063–-0.044	0.89	0.960–0.966	-0.076–-0.059
small	batch, $c = 7/6$ (corrected)	0.996–1.000	0.34	-0.037–-0.026	0.92–0.95	0.961–0.968	-0.053–-0.041
base	std, $c = 1$ (uncorrected)	0.997–1.010	0.49	-0.103–-0.098	0.62–0.66	0.937–0.943	-0.125–-0.120
base	std, $c = 7/6$	0.992–1.009	0.82	-0.022–-0.013	0.87	0.934–0.946	-0.037–-0.031
base	batch, $c = 7/6$ (corrected)	1.011–1.016	0.16	-0.005–$+0.004$	0.94–0.95	0.956–0.967	-0.019–-0.012
large	std, $c = 1$ (uncorrected)	0.989–0.997	0.78	-0.113–-0.092	0.66–0.71	0.900–0.908	-0.131–-0.115
large	std, $c = 7/6$	0.987–0.999	0.62	-0.035–-0.015	0.89–0.93	0.902–0.915	-0.050–-0.033
large	batch, $c = 7/6$ (corrected)	0.994–1.007	0.99	-0.015–-0.008	0.98–0.99	0.915–0.933	-0.029–-0.021

Our Moirai wrapper uses a fixed patch size (32) and a 512-step context, not the official auto-patch inference pipeline, so absolute scores are not comparable to published Moirai GIFT-Eval numbers; the paired ratios are unaffected.

is a single forward pass. One stabilisation is needed. Samples landing in a near-degenerate mixture component have log-densities whose parameter gradients are two orders of magnitude above typical, so we saturate per-step log-densities at 3 nats, which zeroes those gradients for policy and reference alike. Zero-shot, Moirai-small matches Chronos-small on the corpus validation windows (median fair CRPS 0.765 vs. 0.800, paired ratio 0.99, $p = 0.83$), and Moirai is the better-calibrated family there (coverage 0.71–0.73 against Chronos' 0.63–0.65). Training reuses the corpus, sampler, reward, B, G, H, steps and β. Because the gradient scale of a continuous density differs from that of a token policy, the learning rate cannot be reused; we sweep it (5×10^{-7} to 10^{-5}) and anchor the comparison in divergence. At 5×10^{-7} the batch-corrected run's full-run median k_3 KL is 0.007 per step, matching the 0.008–0.009 of the corrected Chronos-small runs (same statistic, same aperture).

Table 10 shows that the Chronos findings replicate at every size. Uncorrected GRPO posts the largest apparent score gains (median ratio 0.957–0.985, all $p < 2 \times 10^{-4}$) and pays for them with the familiar fingerprint: coverage falls by 0.07–0.12 and the median GMD ratio drops to 0.60–0.64, the same shrinkage on a mixture-density head, dose-dependent in the learning rate. The corrected

estimator (batch, $c = 7/6$) keeps coverage within 0.02 of zero-shot and the GMD ratio at 0.92–1.01 at every size, while its gains remain significant (large: 0.969–0.972, $p \leq 10^{-11}$). The coefficient-only correction (std, $c = 7/6$) sits between the two; at the large size it is the strongest scorer in the grid (0.955–0.956, win rate up to 0.73) at a mild dispersion cost (GMD ratio 0.86). Within the corrected family the normaliser thus trades score against calibration, as in the Chronos runs.

Table 11 extends the replication to benchmark transfer. On the 55 short-term GIFT-Eval configurations, the aperture of the Chronos transfer claims, the transfer null replicates at every size. No condition posts a consistent WQL change (geometric-mean ratio 0.986–1.016 across two seeds per condition, weakest-seed $p \geq 0.08$; the two single-seed $p < 0.05$ both point towards improvement), the uncorrected runs still pay the calibration price (coverage -0.09 to -0.12, GMD ratio 0.62–0.71), and the batch-corrected runs keep calibration (coverage within 0.04 of zero-shot, GMD ratio 0.92–0.99). One phenomenon is new: on the full 97 configurations every Moirai condition improves WQL by 3–10%, which no Chronos condition does (Chronos all-97 ratios sit at 1.00–1.05). The gain is concentrated in the 42 medium/long-horizon configurations (geometric-mean WQL ratio 0.80–0.93 there, growing with model size, vs. 0.99–1.02 on short) and is seed-stable. The uncorrected runs buy the largest of these gains with up to a 0.14 coverage drop, so for them it is partly re-sharpening; the batch-corrected runs' 3–9% comes at coverage within 0.06. We record this as an observation rather than a claim: the medium/long configurations have fewer, mutually correlated windows, and the effect has no Chronos counterpart to triangulate against.

Two caveats remain. The small model uses three seeds, base and large two (GIFT-Eval: two per condition). And at a fixed learning rate the realised KL grows with model size (uncorrected, 2×10^{-6}: $0.11 \to 0.18 \to 0.27$), so part of the size trend in the score column reflects a larger realised divergence rather than scale itself.

D.6 Self-backtest test-time compute: per-configuration results

Table 12 lists the per-configuration results behind Section 5.5: every backtest-driven intervention, the random and oracle references, and the verifier fidelity ρ of each configuration.

Table 12: Self-backtest test-time compute (Chronos-T5-small, up to 200 windows per configuration, $N = 20$): fair CRPS / seasonal error. scale: split-conformal rescaling from a $k{=}H$ backtest; bias: backtest bias correction; select/softmax: backtest-based selection / softmax weighting among 24 sampling strategies (context $\times$ temperature $\times$ {base, RL}); random: uniformly random strategy; oracle: strategy selected with the realised future (optimistically biased). ρ: Spearman correlation between backtest and realised CRPS across strategies.

Config	base	scale	bias	scale+bias	select	softmax	random	oracle	ρ
bizitobs_l2c/H/short	0.629	0.713	0.823	0.859	0.851	0.829	0.797	0.381	-0.06
covid_deaths/D/short	36.364	34.189	33.457	32.224	31.946	32.344	34.784	26.012	0.35
electricity/D/short	1.067	1.108	1.199	1.200	1.141	1.200	1.152	0.794	0.07
electricity/H/short	0.701	0.757	0.957	0.954	0.724	0.755	0.818	0.538	0.24
ett1/H/short	0.630	0.652	0.684	0.691	0.684	0.688	0.686	0.475	0.00
hierarchical_sales/D/short	0.573	0.640	0.613	0.662	0.668	0.675	0.652	0.499	0.35
hospital/M/short	0.551	0.603	0.581	0.620	0.546	0.562	0.551	0.442	0.06
jena_weather/H/short	0.372	0.400	0.472	0.471	0.423	0.443	0.543	0.265	0.11
kdd_cup_2018/H/short	0.656	0.697	0.828	0.814	0.732	0.735	0.718	0.494	-0.02
loop_seattle/H/short	0.704	0.749	0.767	0.796	0.756	0.784	0.829	0.496	0.26
m4_daily/D/short	2.392	2.557	3.211	3.181	2.577	2.697	2.551	1.685	-0.02
m4_hourly/H/short	0.573	0.624	0.628	0.658	0.613	0.597	0.633	0.435	0.15
saugeen/D/short	2.553	2.978	2.907	3.107	2.852	2.975	3.038	1.888	0.17
solar/H/short	0.549	0.618	0.715	0.782	0.582	0.598	0.576	0.365	0.00
us_births/D/short	0.320	0.331	0.365	0.361	0.384	0.401	0.363	0.269	0.22
geo-mean ratio (GIFT)	1.000	1.074	1.165	1.193	1.091	1.116	1.126	0.740	0.13
synth/ar_seasonal	0.517	0.534	0.539	0.549	0.531	0.525	0.544	0.449	0.17
synth/garch	2.912	3.145	3.871	3.840	2.998	3.033	2.893	1.905	0.01
synth/gp_mix	1.444	1.477	1.968	1.841	1.474	1.449	1.486	0.905	-0.00
synth/level_shift	0.773	1.015	1.053	1.183	0.853	0.855	0.817	0.536	0.02
synth/regime_ar	1.134	1.212	1.304	1.342	1.140	1.135	1.194	0.890	0.04
geo-mean ratio (synthetic)	1.000	1.099	1.242	1.264	1.037	1.033	1.037	0.720	0.05

Scaling the backtest itself. A natural objection is that the verifier is weak only because a single k-step backtest is a noisy estimate. We re-run the study scoring every strategy by its *average* backtest reward over $m \in \{1, 2, 4\}$ disjoint k-step tails (`scripts/eval_ttc_m.py`; windows are restricted to series with $4k$ of history, which keeps 14 of the 15 configurations). If single-window noise were binding, ρ should grow like $\sqrt{m}$. It does not. The median fidelity across configurations is 0.116, 0.091 and 0.118 at m=$1, 2, 4$, and the mean moves only from 0.135 to 0.172, because a few strongly seasonal configurations improve (m4_hourly $0.16 \to 0.31$, hierarchical_sales $0.37 \to 0.55$) while others decline (covid_deaths $0.34 \to 0.23$, electricity daily $0.12 \to 0.07$). Every intervention stays net-harmful at m=4: the geometric-mean fair-CRPS ratio is 1.051 for selection, 1.044 for conformal scaling and 1.069 for bias correction, against an oracle headroom of 0.727. The synthetic families repeat the pattern at lower levels (median ρ rises only from 0.02 to 0.04; *ar_seasonal* is the one family with real growth, $0.17 \to 0.28$). Quadrupling the backtest data therefore does not buy a usable verifier: the fidelity ceiling reflects distribution shift between the recent past and the future, not estimation noise.

Learning the verifier. The remaining objection is that the *mean* is the wrong way to use the backtest. We therefore fit a learned verifier: a ridge regression predicting each strategy's future reward from all four backtest-tail rewards (z-scored within window), their mean, dispersion and rank, the strategy's context length, temperature and checkpoint identity, and interactions (`scripts/ttc_learned_fit.py`; the regularisation weight is chosen on a validation fold). We score it within configurations (5-fold cross-validation over windows) and across them (leave-one-configuration-out, the cleaner number, since a configuration's rolling windows overlap and make within-configuration folds optimistic). The learned verifier beats the mean-backtest verifier on 13 of 14 configurations (geometric-mean selection ratio $1.065 \to 1.027$ cross-validated, 1.033 transferred; $p \approx 0.003$), yet remains significantly worse than the zero-shot default ($p \leq 0.04$), and its median fidelity moves only from 0.11 to 0.13–0.14 ($p = 0.06$). Every conclusion is unchanged across the regularisation grid. Two controls locate what it learns. A prior-only verifier that sees the strategy attributes but *no backtest at all* attains the highest fidelity of the study (0.15–0.16), and a backtest-only verifier is indistinguishable from the mean ($\rho = 0.10$, selection 1.06). The learnable signal is thus mostly the static ordering of strategies, whose best single choice is the zero-shot default itself, while the window-level signal the backtest is supposed to supply behaves as noise. The synthetic families repeat the pattern (ρ $0.04 \to 0.12$, selection $1.04 \to 1.01$–1.02). Learning the verifier therefore does not rescue backtest-driven test-time compute either.

D.7 STRATIFIED ANALYSIS: TABLES

Tables 13 and 14 give the per-stratum CRPS changes and the within-configuration rank correlations discussed in Section 5.6. The window-level bootstrap CIs in Table 13 ignore clustering by configuration. Re-computing them with a configuration-level cluster bootstrap (resampling configurations with replacement, 4000 draws; `scripts/cluster_bootstrap.py`) widens them by roughly a factor of three and removes the significance of *every* stratum-level effect for all three models: the corrected run's short-term change of -2.4% has window CI $(-3.0, -1.7)$ but cluster CI $(-4.9, +0.4)$, the SFT control's -3.5% has cluster CI $(-7.5, +0.3)$, and the uncorrected run's long-term $+3.0\%$ has cluster CI $(-2.2, +9.7)$. The stratified percentages should therefore be read as descriptive, and the configuration-level Wilcoxon tests of the main tables as the operative inference.

Table 13: Where does post-training change the forecast? Relative change of per-window fair CRPS (normalised by the seasonal-naive error, winsorised at 50) versus the zero-shot model on the 27853 windows of 93 GIFT-Eval configurations evaluated for all listed models, stratified by term, domain and series features (tertiles over the evaluated windows; a tertile edge that coincides with the minimum is merged); 95% bootstrap CIs over windows; win = fraction of windows where the post-trained model is better. All models are Chronos-T5-small trained on GiftEval-Pretrain, $H = 64$, 2000 steps.

stratum	n	GRPO proper (std, $c = 1$)		GRPO proper (std, $c = 7/6$)		SFT control	
		Δ (95% CI)	win	Δ (95% CI)	win	Δ (95% CI)	win
all windows	27853	+0.9% (+0.3, +1.5)	–	-1.4% (-2.0, -0.9)	–	-2.3% (-3.0, -1.6)	–
short	17678	-0.1% (-1.0, +0.7)	0.50	-2.4% (-3.0, -1.7)	0.52	-3.5% (-4.4, -2.6)	0.49
medium	5198	+2.3% (+0.7, +3.7)	0.37	-0.8% (-2.4, +0.6)	0.52	-0.9% (-2.5, +0.5)	0.55
long	4977	+3.0% (+2.1, +3.8)	0.39	+1.3% (+0.3, +2.2)	0.50	+0.6% (-0.6, +1.7)	0.55
Econ/Fin	2745	+3.1% (+1.8, +4.5)	0.48	-0.6% (-1.8, +0.7)	0.49	-0.2% (-1.4, +1.3)	0.48
Energy	8180	+5.9% (+5.2, +6.6)	0.42	+4.6% (+3.9, +5.3)	0.45	+2.7% (+2.0, +3.4)	0.48
Healthcare	778	-6.7% (-9.9, -4.0)	0.56	-9.1% (-12.0, -6.6)	0.54	-13.8% (-17.8, -10.4)	0.54
Nature	3862	+3.1% (+2.3, +3.9)	0.43	-0.1% (-1.0, +0.6)	0.51	-1.7% (-2.6, -0.9)	0.53
Sales	1976	-1.2% (-1.9, -0.4)	0.56	-2.0% (-2.6, -1.3)	0.57	-1.0% (-1.7, -0.3)	0.54
Transport	5532	-2.5% (-3.4, -1.7)	0.49	-6.4% (-7.2, -5.5)	0.59	-2.4% (-3.4, -1.5)	0.58
Web/CloudOps	4780	-0.6% (-2.9, +1.4)	0.45	-1.7% (-3.3, -0.1)	0.51	-3.2% (-5.1, -1.5)	0.47
stl_seasonal < 0.504	18568	-0.9% (-1.7, -0.1)	0.48	-3.4% (-4.1, -2.7)	0.53	-3.9% (-4.8, -3.0)	0.52
stl_seasonal ≥ 0.504	9285	+5.7% (+5.1, +6.3)	0.42	+3.8% (+3.2, +4.3)	0.48	+2.1% (+1.5, +2.6)	0.50
stl_trend < 0.142	9284	+4.4% (+3.2, +5.3)	0.43	+2.6% (+1.5, +3.4)	0.50	+1.5% (+0.4, +2.3)	0.50
stl_trend 0.142-0.527	9284	+2.4% (+1.8, +3.0)	0.46	+0.2% (-0.4, +0.8)	0.52	-0.1% (-0.8, +0.7)	0.53
stl_trend ≥ 0.527	9285	-1.6% (-2.7, -0.5)	0.49	-4.2% (-5.1, -3.3)	0.52	-5.2% (-6.4, -4.1)	0.51
cp=0	15918	+1.3% (+0.6, +1.9)	0.45	-1.0% (-1.6, -0.4)	0.52	-1.0% (-1.7, -0.4)	0.52
cp=1	3838	-3.7% (-5.9, -1.9)	0.50	-4.7% (-6.1, -3.2)	0.53	-5.7% (-7.7, -3.7)	0.51
cp$\geq$2	8097	+2.3% (+1.2, +3.4)	0.45	-0.6% (-1.6, +0.4)	0.49	-2.3% (-3.6, -1.0)	0.48
spectral_entropy < 0.465	9283	+1.2% (+0.2, +2.3)	0.45	-1.1% (-2.1, -0.2)	0.48	-3.8% (-5.0, -2.5)	0.49
spectral_entropy 0.465-0.676	9285	+0.5% (-0.6, +1.3)	0.46	-1.7% (-2.4, -1.1)	0.53	-1.1% (-1.8, -0.5)	0.53
spectral_entropy ≥ 0.676	9285	+0.8% (-0.3, +1.7)	0.46	-1.5% (-2.7, -0.6)	0.54	-1.0% (-2.2, -0.0)	0.50

Table 14: Rank correlation between series features and the per-window log CRPS ratio (post-trained / zero-shot), both rank-transformed within each configuration so that only within-configuration variation contributes; same windows as Table 13. Negative = post-training helps more when the feature is high.

Model	spectral_entropy	hurst	stl_seasonal	stl_trend	n_changepoints	cv	lag1_acf
GRPO proper (std, $c = 1$)	-0.002	-0.020	+0.005	-0.029	+0.008	-0.020	-0.016
GRPO proper (std, $c = 7/6$)	+0.010	-0.003	-0.003	-0.027	+0.013	-0.005	-0.014
SFT control	+0.021	+0.002	-0.004	-0.008	+0.001	+0.000	-0.017