TIME-SERIES FOUNDATION MODELS ANCHOR TO CORRUPTED CONTEXT: A DIAGNOSIS, A BLIND FIX, AND THE LIMITS OF FIXING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series foundation models forecast from a window of recent history with no training or data audit in between, transferring responsibility for data quality to the model. Operational series carry sentinel values, filled gaps and transient level artifacts, and a defect near the forecast origin forces an implicit decision about whether to trust them. We study what such defects do to zero-shot forecasts, how much of the damage a blind test-time repair removes, what produces the sensitivity, and where repair meets a limit no detector can cross. We audit ordinary defects across foundation models from several organizations and architectures, reporting aggregates on normalized errors with equal dataset weights rather than raw pooling, which one dataset dominates. We then propose a blind, training-free sanitizer pairing robust clipping with a gated level correction whose onset is estimated rather than assumed, and localize the mechanism by intervening on where the level is perturbed. Because a transient artifact and a just-started regime change are observationally identical at the origin, no context-only policy is accurate in both worlds, and an equal-weight mixture of the two readings is minimax-optimal. Fragility concentrates in level structure and grows with model scale, the sensitivity is extrapolation of the recent level rather than normalization, and the mixture alone keeps its stated uncertainty in either world. Code is available at `https://anonymous.4open.science/r/tsfm-iclr27-AEB1/`.

1 INTRODUCTION

A long-standing goal of time-series research is a forecaster that works on any series without task-specific training. Time-series foundation models have brought this within reach: pretrained on large corpora, they span discretized-token models (Ansari et al., 2024), patched decoders (Das et al., 2024), any-variate masked encoders (Woo et al., 2024), lag-feature decoders (Rasul et al., 2023), continuous in-context regressors (Le Naour et al., 2026), and mixture-of-experts variants (Liu et al., 2025; Shi et al., 2025), and they are increasingly deployed as drop-in forecasters behind dashboards, capacity planners and anomaly monitors. Benchmark evaluations of these models report accuracy on curated series under one implicit assumption: that the context window handed to the model is trustworthy.

In deployment that assumption fails quietly. Operational series carry sentinel fill values written by acquisition software (weather.csv, a standard benchmark, ships with -9999 entries in its target channel), gaps patched by forward-fill, and transient level artifacts from sensor recalibration, unit changes or logging errors that are corrected upstream later. When such a defect falls inside the context window the model must implicitly decide what to believe: extrapolate the observed recent level, or discount it. A human analyst faced with a suspicious jump in the last few hours of history would ask whether it is real before trusting it; a zero-shot forecaster answers that question implicitly, on every window, with no opportunity for review.

The decision is not merely unsupervised, it is underdetermined. A level shift confined to the recent context is consistent with two futures that the context cannot separate: a transient artifact that will

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.

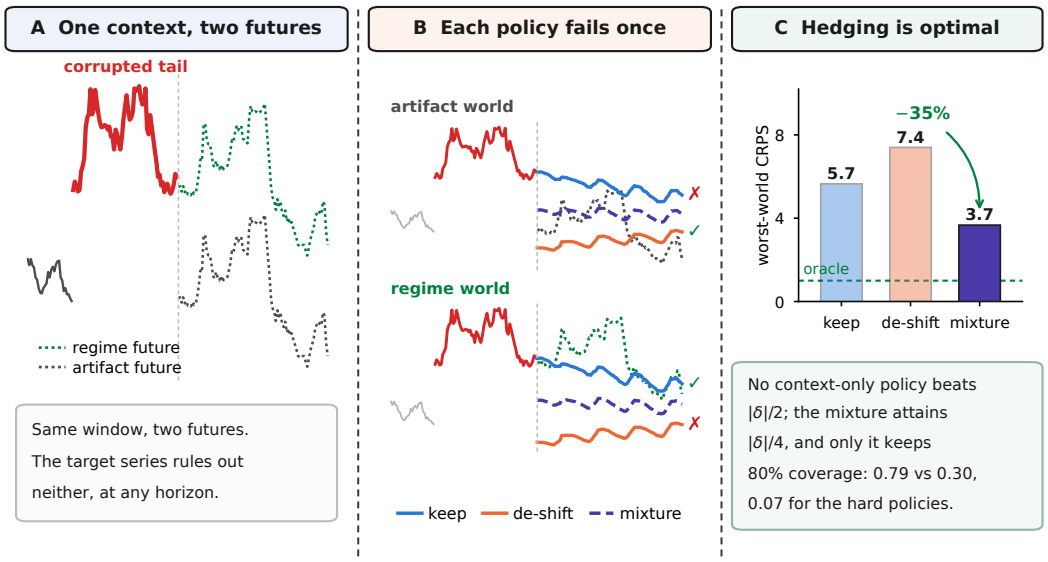

Figure 1: (a) a real ETTh1 window whose last 20% carries a $+3\sigma$ level shift, with the two futures it admits; (b) Chronos-2 forecasts in each world, where keeping the context fails under an artifact and de-shifting fails under a regime change; (c) worst-world CRPS, normalized by the clean-context oracle and averaged over the seven probabilistic models.

be corrected, in which case the forecast should revert to the body level, and a regime change that has just begun, in which case it should follow the new level. We refer to this as the *artifact–regime dilemma*, and it is the reason a corrupted context cannot be treated purely as a detection problem. Figure 1(a) shows it on a real window: the same context precedes two futures, and nothing in the target series prefers either.

Three lines of work bear on corrupted contexts, and none of them settles the question. The first is the classical treatment of outliers and structural change in time series. Fox (1972) separated additive from innovational outliers, Tsay (1988) extended the taxonomy to level shifts and variance changes, and Chen & Liu (1993) showed that forecasts are acutely sensitive to outliers near the forecast origin and that a temporary change and a level shift there are both easy and costly to confuse. That literature already contains the conceptual core of the dilemma above. What it does not contain is the measurement for pretrained zero-shot networks, where there is no fitting stage, no diagnostic step and no analyst reading residuals, and where the decision is taken silently on every window.

The second line treats level as a nuisance to be removed. Reversible instance normalization (Kim et al., 2022) and its descendants normalize the context and restore its statistics after decoding, and test-time adaptation (Wang et al., 2021) updates parameters against distribution shift. Our mechanism experiment bears directly on this reading: the normalization statistics are a real causal channel, but they reallocate level sensitivity rather than create it, so adding normalization neither causes nor cures the fragility we report (Sec. 4.3). Test-time adaptation addresses the persistent world and has no mechanism for the transient one, which makes it complementary to a context sanitizer rather than an alternative to one.

The third line supplies the machinery for acting once the ambiguity is admitted. Changepoint detection (Page, 1954; Aminikhanghahi & Cook, 2017) locates structural moves, and we use exactly such a statistic to estimate where a defect began; but detection is not a policy, since knowing that the level moved does not say whether to follow it. Forecast combination (Bates & Granger, 1969), the linear pooling of predictive distributions (Gneiting & Ranjan, 2013) and adaptive conformal recalibration under shift (Gibbs & Candès, 2021) give the form of an answer, and the hedge we arrive at is a linear pool in the sense of Gneiting & Ranjan (2013), taken over *readings of the data* under a single model rather than over models, which is why it helps when only one forecaster is available. We ask a different question. Specifically, for pretrained zero-shot forecasters used with no diagnostic step and no analyst in the loop, how large is the damage from ordinary context defects, how much of it

can be removed without labels or retraining, what produces it, and what should a deployment output where no amount of detection can decide?

Motivated by these questions, we build a corruption audit with explicit scale-free aggregation, a blind test-time sanitizer that needs no corruption labels and no model access beyond forecasting, an intervention that localizes the mechanism, and a decision-theoretic account of the limit. We validate all four across eight TSFMs spanning five organizations, two release generations and six architecture styles, and nine standard datasets at 100 forecast origins each, with every threshold swept on a global grid rather than tuned per dataset.

Key contributions.

- We formulate context corruption for zero-shot forecasting as a measurable audit with explicitly stated aggregation weights, and show that fragility concentrates in level structure, degrading MAE 4.6–$5.9\times$ on every model while spikes and filled gaps cost at most 6%, and growing with model scale within a family.
- We propose a blind, training-free sanitizer, robust clipping plus a gated de-shift whose correction offset is near zero on clean data, and show it recovers 70–92% of the damage but only where the level is stable, and that it requires a changepoint-estimated onset rather than an assumed one.
- We identify the mechanism by intervening on where the level is perturbed, finding extrapolation of the recent level rather than normalization by window statistics, against classical controls that contain no normalization at all.
- We prove that context-only policies face a $|\delta|/2$ worst-case floor that distributional hedging halves at the minimax optimum $|\delta|/4$, and validate the complete routing policy end to end, including where it does not help.

2 Problem formulation

We first define what is measured. Let $x \in \mathbb{R}^L$ be a context window and $y \in \mathbb{R}^H$ the future to be forecast, with $L = 336$ and $H = 96$ throughout. A forecaster f maps a context to a predictive distribution over y, and a *policy* π is a rule that may transform the context before calling f. The level-shift corruption adds $m\sigma$ to the last fraction p of the context,

$$\tilde{x}_i \;=\; x_i + m\sigma \cdot \mathbf{1}\{i > (1-p)L\}, \qquad \sigma = \mathrm{sd}(x), \tag{1}$$

with defaults $m = 3$ and $p = 0.2$. The taxonomy also includes $\pm 8\sigma$ spikes on 5% of points, a 5% contiguous sentinel block at -100σ, and a 10% gap repaired by forward-fill, which mimics naive practitioner handling.

Writing MAE_o for the error of f on the clean context, MAE_c for its error on $\tilde{x}$, and MAE_s for a sanitized context, the three quantities we report are degradation, recovery and clean cost,

$$\text{degradation} = \frac{\mathrm{MAE}_c}{\mathrm{MAE}_o}, \quad \text{recovery} = \frac{\mathrm{MAE}_c - \mathrm{MAE}_s}{\mathrm{MAE}_c - \mathrm{MAE}_o}, \quad \text{cost} = \frac{\mathrm{MAE}_s^{\mathrm{cl}} - \mathrm{MAE}_o}{\mathrm{MAE}_o}. \tag{2}$$

Aggregation. Errors on the nine targets carry different units and differ by orders of magnitude, so an average of raw MAE across them summarizes whichever dataset has the largest numbers rather than the method: under unit-weighted pooling, electricity alone contributes 92–94% of the total injected damage. Every aggregate here therefore divides each window's errors by the mean clean-context error of its own dataset and weights datasets equally, so a degradation of $5\times$ means five times the accuracy the model would have had on an uncorrupted context, averaged over datasets on equal footing. Because the weights are a modeling choice, we report per-dataset values alongside every aggregate, give the range over datasets in the main tables, and label any unit-weighted number as such (App. D).

Two worlds. The same corrupted context $\tilde{x}$ is paired with two futures: the clean future y, meaning the shift was a transient artifact, and the shifted future $y + m\sigma\mathbf{1}$, meaning a regime change that persists. Because the two worlds share the context by construction, this isolates exactly the information a context-only policy has.

Probabilistic scoring. Families expose their predictive distribution differently: Chronos and the Moirai family return 20 Monte-Carlo sample paths, while Chronos-2, both TimesFM versions and TS-ICL return nine quantiles at levels $0.1, \ldots, 0.9$. The two are not interchangeable: taking the empirical 0.1 and 0.9 quantiles of nine quantile atoms returns roughly the 0.18 and 0.82 quantiles of the underlying distribution, so a nominal 80% interval is evaluated as a 64% one. We therefore reconstruct each family's quantile function natively and score with a censored CRPS, $\mathrm{CRPS}_{[0.1,0.9]} = 2\int_{0.1}^{0.9} \mathrm{QL}_\tau(F^{-1}(\tau), y)\, d\tau$, which is proper on the range every family reports natively, together with coverage of the native interval $[F^{-1}(0.1), F^{-1}(0.9)]$. TimeMoE is point-only and is excluded from distributional metrics.

3 BLIND SANITIZATION AND HEDGING

The method has three parts: a point-defect filter, a level gate with a correction, and a policy for what to do when the gate fires but the context cannot say why. None of them uses corruption labels, training, or model access beyond forecasting.

3.1 ROBUST CLIPPING AND THE LEVEL GATE

Points beyond 8 robust standard deviations (median absolute deviation, MAD) from the context median are removed and linearly interpolated. This stage repairs sentinels and spikes and never fires on a 3σ level shift, whose points are individually unremarkable. Given a boundary b splitting the context into a body $x_{1:b}$ and a tail $x_{b+1:L}$, the model forecasts the tail from the body and the gate statistic is the deviation of the observed tail level from that backcast, in robust units of the body,

$$z \;=\; \frac{\left|\bar{x}_{\mathrm{tail}} - \bar{\hat{x}}_{\mathrm{tail}}\right|}{s_{\mathrm{body}}}, \qquad s_{\mathrm{body}} = 1.4826 \cdot \mathrm{MAD}(x_{\mathrm{body}}), \tag{3}$$

and the tail is flagged when $z > \tau$ (default $\tau = 1$). When the gate fires, the estimated offset $\hat{\delta} = \bar{x}_{\mathrm{tail}} - \bar{\hat{x}}_{\mathrm{tail}}$ is subtracted from the tail before forecasting. The correction is self-calibrating: on clean data $\hat{\delta} \approx 0$, so a false fire is cheap. Every input to the gate, the body it backcasts from included, is taken from the observed and possibly corrupted context.

3.2 WHERE THE BOUNDARY COMES FROM

A deployment does not know where a defect began, so we evaluate three ways of setting b: a constant $\hat{p} = 0.2$ fixed in advance, a boundary estimated from the observed context as the argmax of a CUSUM-style two-sample level statistic over splits in the last half of the window, and an oracle boundary equal to the true onset. Only the first two are deployable, and Section 4.2 shows the choice is not a detail.

3.3 POLICIES UNDER AMBIGUITY

With covariates available, the de-shift fires only if the target's tail move is not corroborated across channels: the statistic is the median over covariate channels of the robust z of each channel's tail against its own body, and firing requires $z > \tau$ and corroboration below a threshold c. The rationale is physical, since a broad regime change moves many channels while a single-sensor artifact does not. When the gate fires and corroboration is unavailable or inconclusive, we forecast the equal-weight *linear pool* $\frac{1}{2}F_{\mathrm{keep}} + \frac{1}{2}F_{\text{de-shift}}$ of the two predictive distributions (Gneiting & Ranjan, 2013), formed by averaging the two CDFs on a value grid and inverting, not by pooling quantile atoms. The complete policy is then: robust-clip; run the gate; if it does not fire, keep; if it fires and corroboration is decisive, route to keep (corroboration $> c_{\mathrm{hi}}$) or de-shift (corroboration $< c_{\mathrm{lo}}$); and otherwise, in the inconclusive band $c_{\mathrm{lo}} \le \mathrm{corr} \le c_{\mathrm{hi}}$ or when no covariates exist, forecast the mixture. We use $c_{\mathrm{lo}} = 1.0$ and $c_{\mathrm{hi}} = 2.0$.

3.4 WHAT NO POLICY CAN DO

Hedging is not a heuristic here; it is what the structure of the problem forces.

Proposition 1 (context-only identifiability). *Fix a corrupted context $\tilde{x}$ and let the transient world have future y and the persistent world future $y + \delta\mathbf{1}$, both arising with the same $\tilde{x}$. Let π be any policy that is a function of the context alone, emitting a predictive distribution $F = \pi(\tilde{x})$ scored by CRPS, of which mean absolute error is the special case F degenerate. Then, per horizon step, $\max\{\mathrm{CRPS}(F, y), \mathrm{CRPS}(F, y+\delta)\} \geq |\delta|/4$, with equality attained by the equal-weight mixture of point masses at y and $y + \delta$; and if F is restricted to be a point forecast, the bound is $|\delta|/2$, attained at the midpoint.*

The proof is in App. A. Two consequences matter. The bound is a statement about the data, not the model: no architecture, however it estimates level, can make two identical observations reveal which future will occur. And the gap between $|\delta|/2$ and $|\delta|/4$ is the entire case for hedging, since moving from the best possible point forecast to the mixture halves the worst-case score and does so optimally. What our measurements add is how far real models sit from these idealized bounds.

4 EXPERIMENTS

Zero-shot forecasters are deployed on data nobody audits, so the practical question is not whether a curated benchmark is predicted well but what happens when the context is not clean, and what a deployment can do about it. We organize these studies around the following four Research Questions (RQs):

(RQ1) Which context defects damage zero-shot forecasts, and by how much?

(RQ2) How much of the damage can a blind repair remove, and what does it cost on clean data?

(RQ3) What produces the sensitivity to a corrupted level?

(RQ4) What should a deployment output where no detector can decide?

We use nine standard benchmarks: ETTh1/2 and ETTm1/2 (Zhou et al., 2021), weather (Wu et al., 2021), and electricity, traffic, exchange-rate and solar (Lai et al., 2018), chosen to span regimes that stress different parts of the method: strongly periodic bounded series (traffic, electricity), drifting 15-minute series with natural level moves (ETTm1/2), a near-random-walk (exchange-rate) and a zero-inflated intermittent series (solar). The target is the OT channel where present and the last column otherwise; the remaining channels serve as covariates. We evaluate eight zero-shot TSFMs: Chronos-t5-base (Ansari et al., 2024) and Chronos-2 (Ansari et al., 2025) (Amazon), TimesFM-1.0-200m and TimesFM-2.0-500m (Das et al., 2024) (Google), TS-ICL (Le Naour et al., 2026), Moirai-1.1-R-base (Woo et al., 2024) and Moirai-MoE (Liu et al., 2025) (Salesforce), and TimeMoE-200M (Shi et al., 2025), plus Chronos-t5 tiny/small/large and Chronos-Bolt for the scale study. Unless otherwise stated we forecast $H = 96$ from $L = 336$ at 100 evenly spaced origins per dataset (40 for the scale and mechanism studies, 25 for the onset study), treat origins as the unit of replication, and run every model exactly as released with no fine-tuning; all experiments are inference-only. Weather's native -9999 entries are removed for injected-corruption experiments so that "clean" means clean, and retained for the in-the-wild study. Appendix B gives the full protocol and the scope conditions behind the corruption design. In the tables that follow, ✓ marks a row that recovers at least 65% of the injected damage and ✗ one that does not, a clean cost is green at or below 5% and red above it, and in Table 4 green marks an improvement over the better hard policy.

4.1 RQ1: WHICH DEFECTS DAMAGE ZERO-SHOT FORECASTS

We first answer RQ1 by running the corruption taxonomy against every model and reading the result per dataset rather than in aggregate. Fragility concentrates entirely in level structure: spikes cost at most 5% and forward-filled gaps at most 6% (Table 6), while the 3σ tail shift degrades MAE 4.6–$5.9\times$ on every one of the eight models (Table 1). Sentinel blocks are model-dependent and diagnostic of the input interface: tokenizer-based Chronos clips them ($1.02\times$), while normalization-based TS-ICL degrades $2.1\times$ because one extreme block corrupts its instance statistics.

The per-dataset structure is the informative part. Bounded, strongly periodic series are hit hardest: on traffic, where a 3σ shift is enormous relative to the 0–1 occupancy scale, degradation reaches $19.9\times$ on average and $23.1\times$ at worst. Smooth low-noise series (ETTm1/2) degrade 3.2–$3.6\times$, and zero-inflated solar least ($2.2\times$), because its scale statistics are dominated by the day–night cycle. That

Table 1: Level-shift fragility and blind-sanitizer recovery, by model and by dataset.

	Damage	Repair		Cost	
Model / dataset	Degrad.	Recov. (%)	Range	Clean (%)	Worst / fire
By model, equal weight over the nine datasets; last column is the worst single-dataset clean cost					
Chronos-base	5.80×	74	9–94	+7.8	+16.0
TimesFM	5.75×	92	59–141	+8.1	+25.4
TS-ICL	5.10×	70	−5–98	+7.0	+20.5
Moirai	5.73×	77	18–99	**+4.8**	+17.5
Chronos-2	5.94×	77	14–95	+5.2	+15.5
TimesFM-2.0	5.94×	91	39–127	+6.7	+17.0
Moirai-MoE	4.64×	83	34–98	+5.1	+14.2
TimeMoE	4.62×	**92**	57–98	+8.2	+24.9
By dataset, averaged over the eight models; last column is the clean fire rate					
ETTh1	3.05×	78	67–87	+4.5	21%
ETTh2	3.39×	81	64–90	+6.0	14%
ETTm1	3.23×	54	33–72	+14.9	26%
ETTm2	3.56×	41	−5–79	+10.9	25%
electricity	6.29×	95	91–97	+0.5	1%
weather	3.73×	62	22–89	+3.1	28%
traffic	19.86×	**97**	92–100	**+0.0**	0%
exchange	3.65×	39	9–72	+17.2	27%
solar	2.19×	88	41–141	+2.6	60%

spread is what makes the aggregation rule consequential: unit-weighted pooling assigns electricity 92–94% of the total damage and reports its 6.3× as the nine-dataset result.

Neither a newer release generation nor more capacity helps. The 2025–26 models fail like their predecessors, and within the Chronos family, compared on identical data, degradation grows monotonically with scale from 3.6× (tiny) through 4.1× (small) and 4.3× (base) to 4.4× (large), with the differently decoded Chronos-Bolt at 4.1× (Table 9). Larger models are better forecasters and, for that reason, more faithful extrapolators of a corrupted level. The ordering across *families* is not stable under the aggregation rule, so we draw this conclusion only from the within-family ladder, which uses identical data throughout. Nor is the fragility confined to injected corruptions: forecasting weather.csv windows whose lookback contains the dataset's native -9999 sentinels degrades MAE 19–44×, and a one-line forward-fill recovers essentially all of it (Table 7).

These results answer RQ1: the damage is concentrated in level structure, universal across architectures, and larger for larger models.

Takeaway 1

Robustness to context defects is not implied by benchmark accuracy. The corruptions classical cleaning pipelines target are the ones these models already tolerate, the one that is not is structural rather than point-wise, and it grows with model scale, so an audit of the context belongs alongside accuracy in TSFM evaluation.

4.2 RQ2: HOW MUCH A BLIND REPAIR REMOVES, AND AT WHAT COST

We next answer RQ2 by applying the sanitizer at a single global operating point and reading both what it recovers and what it costs on uncorrupted data. At $\tau = 1$ it recovers 70–92% of the injected damage at a normalized clean cost of 5–8% (Table 1). Both halves of that sentence are weaker than unit-weighted pooling suggests (87–96% at under 3%), and the gap is not noise: the datasets the correction handles best are also the ones with the largest raw errors.

The per-dataset breakdown explains the method and its failure mode at once. On electricity and traffic, whose level is stable, recovery is 95–97% and the gate essentially never fires on clean data, so the clean cost is 0.0–0.5%. On ETTm1/ETTm2 and exchange-rate it fires on 25–27% of *uncorrupted*

Table 2: Recovery when the onset of the shift is unknown.

Model	Assumed boundary	10%	20%	30%	40%	Clean cost (%)	τ
Chronos-base	fixed $\hat{p} = 0.2$	63	76	9	6	+7.9	0.42
Chronos-base	estimated (CUSUM)	85	84	81	78	+7.6	2.13
Chronos-base	oracle $\hat{p} = p$	69	76	73	71	+7.9	0.43
TS-ICL	fixed $\hat{p} = 0.2$	53	68	6	0	+7.8	1.04
TS-ICL	estimated (CUSUM)	91	93	92	90	+7.9	1.66
TS-ICL	oracle $\hat{p} = p$	61	69	67	67	+7.6	1.06
TimesFM-2.0	fixed $\hat{p} = 0.2$	62	92	32	6	+8.0	0.57
TimesFM-2.0	estimated (CUSUM)	89	95	94	94	+6.6	1.82
TimesFM-2.0	oracle $\hat{p} = p$	89	92	92	91	+7.8	0.73

contexts, which contain genuine level moves, recovery falls to 39–54% and the correction costs 11–17%; four of nine datasets recover less than 65%. This is not a tuning failure but the artifact–regime dilemma appearing inside the audit.

The boundary must be estimated, not assumed. Holding the assumed boundary at a constant $\hat{p} = 0.2$ while the true onset varies separates performance from a correctly specified boundary (Table 2). At matched clean cost, recovery is 68–92% when the shift begins at the assumed boundary and 53–63% when it begins later, but falls to 6–32% when it begins at 30% of the context and 0–6% at 40%. The method does not degrade, it switches off: once the shift starts before the assumed boundary the assumed body contains part of it, the backcast of the tail is computed from an already elevated body and so predicts an elevated tail, the gate statistic shrinks below τ, and the correction never fires. Estimating the boundary instead, as the argmax of a CUSUM changepoint statistic on the observed context, removes the problem: it recovers the onset to within one index of the truth and restores 78–95% recovery at every onset for the same clean cost, matching or exceeding what knowing the true onset would have bought.

Simple detectors suffice. At a matched normalized clean-cost budget of 5%, swept over a fine threshold grid, the best model-free detector matches or exceeds the model-in-the-loop backcast gate on seven of eight models and trails it by 2 points on the eighth (Table 10). The practical recipe therefore does not depend on privileged model access, and we present the diagnosis and the recipe, not detector novelty, as the contribution. At a stricter 2% budget the picture is mixed rather than systematic, the backcast gate leading on four models by 5–21 points and trailing on four by 5–20.

These results answer RQ2: a blind repair removes most of the damage where level moves are artifacts and little of it where they are real, and it needs an estimated onset.

Takeaway 2

A blind repair is close to free and close to complete only where the level is known to be stable; where level moves are real it fires on genuine ones and costs more than it saves. It must also locate the defect rather than assume it, since a boundary fixed in advance switches the correction off entirely once the shift begins earlier than assumed.

4.3 RQ3: WHAT PRODUCES THE SENSITIVITY

We answer RQ3 by intervening on *where* the level is perturbed, which separates two accounts that a single anchoring regression cannot. Regressing the change in mean forecast level on the injected offset gives coefficients near one, but that is not evidence for normalization: a persistence forecaster, which contains none, has an anchoring coefficient of exactly one. Shifting the whole window is a calibration check any level-equivariant forecaster must pass with $\alpha = 1$, and every model returns $\alpha = 0.99$–1.00. The discriminating comparison is between a shift on the last 20% of the context and one on the first 80%: a forecaster that normalizes by window statistics must respond about four times as strongly to the body shift, which moves the window mean by 0.8δ against the tail shift's 0.2δ. Every model does the opposite ($\alpha_{\mathrm{tail}} = 0.63$–$0.81$ against $\alpha_{\mathrm{body}} = 0.06$–$0.32$, a ratio of 2.2

Table 3: Anchoring coefficient by perturbation location, against classical controls.

Forecaster	Tail	Body	Whole	Norm. fixed	R^2 (tail)
Chronos-base	0.81	0.06	0.99	0.46	0.97
TimesFM	0.77	0.12	1.00	0.40	0.95
TS-ICL	0.70	0.32	1.00	0.35	0.96
TimesFM-2.0	0.73	0.28	1.00	0.39	0.97
TimeMoE	0.63	0.31	1.00	0.30	0.95
TimeMoE (robust norm.)	0.44	0.57	1.00	0.18	0.88
Controls with no normalization anywhere, on the same windows					
Persistence (last value)	1.00	0.00	1.00	0.57	–
Seasonal naive	1.00	0.00	1.00	0.56	–
Exp. smoothing (α=0.3)	1.00	0.00	1.00	0.57	–
Linear trend extrapolation	0.82	0.18	1.00	0.42	–
Window mean	0.20	0.80	1.00	0.00	–

to 13.5 in favor of the tail), matching the persistence and exponential-smoothing controls (1 and 0) and reversing the window-mean control (0.20 and 0.80); see Table 3 and Fig. 11.

Normalization is not the dominant channel but it is not absent. Holding the window mean and standard deviation at their clean values while keeping the tail elevated reduces anchoring to 0.30–0.46, against 0.57 for the persistence control under the same transformation. A direct intervention is possible on TimeMoE, the one family that normalizes at the interface rather than inside the network: replacing its window mean and standard deviation with the median and MAD of the context *body*, which a tail defect cannot move, shifts the sensitivity from tail to body exactly as the account predicts. Here α_{tail} falls from 0.63 to 0.44 while α_{body} rises from 0.31 to 0.57, tail-shift degradation falls from $3.6\times$ to $3.0\times$ while body-shift degradation rises from $2.2\times$ to $3.4\times$, and the two coefficients still sum to one.

These results answer RQ3: the sensitivity is extrapolation of the recent level, and normalization reallocates it rather than causing it.

Takeaway 3

The fragility is not a normalization artifact that better normalization would remove. These models track the recent level, which is what makes zero-shot transfer work, and the normalization statistics reallocate that sensitivity rather than create it, so a defect in the recent level is transcribed into the forecast whatever sits in front of the decoder.

4.4 RQ4: WHAT TO OUTPUT WHERE NO DETECTOR CAN DECIDE

We answer RQ4 by measuring how far real models sit from the bounds of Proposition 1, and then by scoring the complete policy end to end. Figure 1(b) shows the shape of the problem on one window and Fig. 1(c) its aggregate cost. In units of each dataset's clean-context oracle, always-keep costs 4.6–$5.9\times$ oracle error in the transient world and always-de-shift 5.5–$7.8\times$ in the persistent world, each being near-oracle (1.1–$2.6\times$) in the world it was built for, while the point hedge splits the difference at 3.1–$4.9\times$ worst-world, as the proposition predicts a midpoint must (Table 14). The dilemma is not synthetic: the gate fires on 25–26% of *clean* ETTm1/ETTm2 contexts and 27% of clean exchange-rate contexts, which contain genuine persistent level moves.

Two natural single-series vetoes fail (App. N): vetoing when the covariate median moves reduces ETTm1's clean damage only from $+29\%$ to $+18\%$ while collapsing recovery, and vetoing unless the tail move is step-shaped fails because natural moves include true steps whose jump statistics overlap the injected ones. With covariates that do carry the information the corroboration gate reaches best-of-both-worlds selection, firing on 81–82% of artifacts and on 2 of 900 windows (0.2%) in the regime world. That result must be read with its construction in mind: the synthetic regime moves every covariate and the synthetic artifact moves none, so we supplied precisely the evidence the gate then discovers. Real regimes may move only the target and real acquisition faults may knock out

Table 4: Worst-world CRPS of the hypothesis mixture, normalized by the clean-context oracle.

	Worst-world CRPS			Natural gate-fired	
Model	Keep	De-shift	Mixture	Δ (%)	keep/de-shift/mix
Chronos-base	6.4	7.5	**3.3**	+49	1.07/1.38/1.10
TimesFM	6.0	7.8	**3.7**	+38	1.01/1.37/1.10
TS-ICL	5.2	7.0	**3.5**	+34	1.07/1.49/1.16
Moirai	5.8	6.1	**3.2**	+44	1.12/1.36/1.06
Chronos-2	5.4	7.9	**4.0**	+27	0.97/1.33/1.03
TimesFM-2.0	6.0	8.9	**4.5**	+25	1.04/1.35/1.13
Moirai-MoE	4.7	6.6	**3.5**	+26	1.06/1.24/1.04

several channels at once, in which case the statistic points the wrong way. The defensible claim is conditional, and absence of covariate movement does not certify an artifact.

Detect and hedge. The mixture is the only policy whose uncertainty statements survive both worlds (Table 4). In normalized CRPS, always-keep costs 4.7–$6.4\times$ oracle in its bad world and always-de-shift 6.2–$8.9\times$ in its bad world, while the mixture holds 3.2–$4.5\times$: an improvement of 25–49% over the better hard policy, on every model with probabilistic output. Coverage is sharper still. Keep's nominal 80% intervals cover 12–49% in the transient world and de-shift's 4–13% in the persistent world, while the mixture holds 70–86% and 67–88%, above the clean-context oracle's own 50–74% because pooling two hypotheses widens intervals these models make overconfident. On natural gate-fired windows the mixture costs -6 to $+8\%$ against always-keep where hard de-shift costs $+17$ to $+38\%$. Scored end to end, the full recipe reaches 1.5–$2.4\times$ oracle worst-world against 4.7–$6.4\times$ for the naive forecaster at a cost of 3–6% on natural contexts, routing 54–55% of windows to keep, 36% to de-shift and 9% to the mixture; the covariate-free fallback of gate-plus-mixture reaches 3.2–$4.5\times$ at -1 to $+2\%$ (App. M).

These results answer RQ4: where the context cannot decide, the right output is a distribution spanning both readings.

Takeaway 4

At the forecast origin the context cannot say whether a level move will persist, so no detector settles it and no point forecast beats $|\delta|/2$. Reporting a distribution over both readings is therefore not a hedge of convenience but the minimax-optimal answer, and the only one whose stated uncertainty survives either world.

5 Conclusion

Eight TSFMs across two release generations share a structural fragility to recent-context level corruption; a blind sanitizer removes most of the injected damage on series whose level is stable and misfires where it is not; the sensitivity is extrapolation of the recent level rather than the normalization machinery it is often attributed to; and the limit of any correction is the artifact–regime ambiguity, which standard benchmarks already contain and which hedging provably handles better than any point policy. Robustness to context corruption should therefore join accuracy as a standard axis of TSFM evaluation, and the property that makes zero-shot transfer work is the same one that transcribes a recent data defect into forecast error, which predicts that better forecasters will be more sensitive rather than less. Our evidence has stated limits: injected shifts are additive and extend to the context end, the corroboration result supplies its distinguishing evidence by construction, CRPS is censored to the quantile range every family reports natively, and the end-to-end evaluation uses synthetic worlds throughout. Validating the policy on independently identified natural incidents, where the ground truth about artifact versus regime comes from outside the series, remains future work.

Reproducibility Statement

Every experiment is inference-only and uses each model exactly as released. The corruption model is given by Eq. 1, the reported quantities by Eq. 2, the gate by Eq. 3, and the aggregation rule, scope conditions and exact package versions in App. B. App. C gives the audit loop, the four detectors and the end-to-end policy as executable listings, together with one window carried through all of them. Detector scores, gate statistics and the error of every policy are stored per window, so all threshold sweeps, aggregation rules and scoring rules are recomputed from those files without further inference. The code, the per-window CSV outputs and the scripts that regenerate every table and figure in this paper are available at `https://anonymous.4open.science/r/tsfm-iclr27-AEB1/`.

Ethics Statement

This work is an audit of publicly released forecasting models on standard public benchmark datasets. It involves no human subjects, no personally identifying information and no new data collection, and every experiment is inference-only: the models are used exactly as released, with no training and no new parameters, so the study cannot produce behaviour the released models do not already have. The corruptions we inject are ordinary data-quality defects, sentinel fill values, forward-filled gaps and level shifts of the kind acquisition pipelines produce, rather than adversarial perturbations, and injecting them into one's own evaluation is a robustness check rather than a capability for harm. The deployment risk we see runs the other way: a context sanitizer applied where level moves are genuine will suppress real regime changes and degrade forecasts, which matters in settings such as load or demand forecasting where a missed regime change is costly. We therefore report per-dataset behaviour alongside every aggregate, state the conditions under which hard correction is unsafe, and recommend hedging rather than correction as the default.

Use of Large Language Models

During the preparation of this paper, we used a large language model to draft and revise text and to improve its grammar, word choice and conciseness. We also used it to help identify relevant literature; the authors read each suggested work and verified every citation against its original source before inclusion. We used Claude Code to assist with implementing the analysis scripts, running the experiments and producing the tables and figures. The authors reviewed the code and verified all reported numerical results against the saved per-window outputs, which are released with the paper. All claims are the authors' own, and we take full responsibility for the content of this paper.

References

Samaneh Aminikhanghahi and Diane J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, 2017. doi: 10.1007/s10115-016-0987-z.

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`.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

J. M. Bates and C. W. J. Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969. doi: 10.1057/jors.1969.103.

Chung Chen and Lon-Mu Liu. Forecasting time series with outliers. *Journal of Forecasting*, 12(1): 13–35, 1993. doi: 10.1002/for.3980120103.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024.

A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(3):350–363, 1972. doi: 10.1111/j.2517-6161.1972.tb00912.x.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1660–1672, 2021.

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 and Roopesh Ranjan. Combining predictive distributions. *Electronic Journal of Statistics*, 7:1747–1782, 2013. doi: 10.1214/13-EJS823.

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*, 2022.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018. doi: 10.1145/3209978.3210006.

Etienne Le Naour, Tahar Nabil, and Adrien Petralia. TS-ICL: A flexible time-indexed foundation model for time series via in-context learning. *arXiv preprint arXiv:2606.05878*, 2026.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-MoE: Empowering time series foundation models with sparse mixture of experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38940–38962, 2025.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. doi: 10.1093/biomet/41.1-2.100.

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. *arXiv preprint arXiv:2310.08278*, 2023.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025.

Ruey S. Tsay. Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7 (1):1–20, 1988. doi: 10.1002/for.3980070102.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

APPENDIX OVERVIEW

The appendix reports the complete result sets and analyses behind the aggregates in the main text: reproducibility details (App. B), the protocol, the detectors and the complete policy written out as code, with one worked window (App. C), the effect of the aggregation rule on every headline number (App. D), the corruption taxonomy (App. E), per-dataset fragility and sanitizer behavior (App. F), detector operating characteristics (App. G), the magnitude and onset sweep with a correctly specified boundary (App. H), the unknown-onset study (App. I), the mechanism intervention (App. J), the corroboration threshold study (App. K), full probabilistic results under both the corrected and the superseded scoring (App. L), the complete policy scored end to end (App. M), and the veto-failure analysis (App. N). App. A proves Proposition 1.

A PROOF OF PROPOSITION 1

The policy sees identical inputs in the two worlds, so it emits the same F. For a point forecast a, $|a-y|+|a-(y+\delta)| \geq |\delta|$ by the triangle inequality, so the larger of the two is at least $|\delta|/2$, with equality at $a = y + \delta/2$. For a distribution, take the two worlds as equally likely; CRPS is a strictly proper scoring rule (Gneiting & Raftery, 2007), so the average of the two scores is minimized by the true predictive distribution P, the equal mixture of point masses at y and $y+\delta$. Its CDF equals $1/2$ on $[y, y+\delta)$ and is exact elsewhere, so its CRPS in either world is $\int_y^{y+\delta}(1/2)^2\,\mathrm{d}x = |\delta|/4$. Since the maximum of two numbers is at least their average, every F has worst-case score at least $|\delta|/4$. The point-forecast case is the restriction of the same argument to degenerate F, for which CRPS reduces to absolute error.

B REPRODUCIBILITY AND COMPUTE DETAILS

All experiments are inference-only. Each (model, experiment) pair ran on a single A100 GPU with per-dataset checkpointing; the complete campaign (eight models, nine datasets, seven experiment types, plus the scale study) totals roughly 1.4M forecast calls. Chronos models were used through the `chronos-forecasting` package, TimesFM through the official `timesfm` package (the 2.0 checkpoint with 50 layers, no positional embedding, context 2048), the Moirai family through `uni2ts` (Moirai-MoE with patch size 16, 20 samples), TS-ICL through its released package, and TimeMoE through its published checkpoint with mean–variance normalization at the interface. Detector scores, gate statistics and the error of every policy are stored per window, so all threshold sweeps, aggregation rules and scoring rules are recomputed post hoc from those CSVs without further inference; the detector comparison of Table 10 uses a 200-point grid of quantiles of each detector's own score distribution rather than a hand-picked threshold list. No per-dataset tuning is performed anywhere; all operating points reported are global across the nine datasets, because a deployed sanitizer cannot assume labeled corruptions for calibration. The injected shift is additive and extends to the context end because that is the observationally hardest case: a shift that reverts inside the context is visible as such, whereas one still active at the forecast origin is exactly what a deployed system sees mid-incident. We corrupt only the target channel because single-sensor faults are the common failure mode, and because that choice is what gives the corroboration gate its information, a limitation discussed in Sec. 4.4. Code, the per-window CSV outputs, and the scripts that regenerate every table and figure in this paper from those CSVs are available at `https://anonymous.4open.science/r/tsfm-iclr27-AEB1/`.

C PROTOCOL, SANITIZER, AND ONE WORKED WINDOW

This section states in code what the main text states in prose: the loop that produces every audited number (Fig. 2), the four gate statistics compared in Table 10 (Fig. 3), the complete policy of Sec. 3 from raw context to predictive distribution (Fig. 4), and one real window carried through all of it, with the numbers the results file records (Fig. 5). Every listing follows the released implementation (`https://anonymous.4open.science/r/tsfm-iclr27-AEB1/`); ASCII names replace the symbols of the main text (τ is `tau`, δ is `delta`, p is `p`).

Audit protocol: one (model, dataset) cell

```
INPUT   series y_1..y_N from one CSV (target channel OT, else the last column; entries <= -9000 dropped)
        lookback L = 336;  horizon H = 96;  windows per dataset W = 100;  assumed tail fraction p = 0.20
        shift magnitude m sigma (m = 3 in every headline table);  ONE global operating point, no
        per-dataset tuning anywhere, because a deployed sanitizer has no labelled corruptions to tune on

tl = int(L * p) = 67                    # length of the corrupted tail
b  = L - tl     = 269                   # index at which the corruption is assumed to start

for each origin o in linspace(L + 5, N - H - 1, W):          # W origins spaced evenly over the series
  lookback = y[o-L : o]                 # what the model is given
  truth    = y[o : o+H]                 # what actually happened; NEVER corrupted, and never shown

  ## 1. Corrupt the context.  Additive level shift on the last tl points only, so the sensor is
  ##    wrong and the world is not.  This is the corruption model of the main text.
  shifted = lookback;   shifted[b:] += m * std(lookback)

  ## 2. Clean both contexts the way a deployed pipeline would.  This is the BASELINE, not our
  ##    method: any point further than 8 * 1.4826 * MAD from the median is dropped and linearly
  ##    interpolated.  It removes sentinels and spikes.  A 3 sigma tail shift survives untouched.
  x_clean = robust_clip(lookback);   x_shift = robust_clip(shifted)

  ## 3. Forecast H steps from each context and from each candidate repair.
  keep_clean = MAE(truth, model(x_clean, H))     # oracle:  the model with an uncorrupted context
  keep_shift = MAE(truth, model(x_shift, H))     # naive:   the model with the corrupted context
  mds_shift  = MAE(truth, model(deshift(x_shift), H))       # backcast-de-shifted context
  sds_shift  = MAE(truth, model(detrend_deshift(x_shift), H))

OUTPUT  one row per window holding every detector score and every policy error, so thresholds,
        aggregation rules and scoring rules are swept post hoc with no further inference

AGGREGATION (every table in the paper)
  Divide each window's errors by the per-dataset mean of keep_clean.  Each dataset then has
  reference mean 1, so pooling weights the nine datasets equally instead of by their units.
  Write o, b, s for the equal-weight means of the clean, corrupted and sanitized errors, and c
  for the sanitized error on clean contexts:
        degradation = b / o        recovery = 100 * (b - s) / (b - o)        clean cost = 100 * (c - o) / o
  Recovery is 100% when the sanitizer restores the clean-context error and 0% when it changes
  nothing; it is negative when the repair costs more than the corruption.
```

Figure 2: The loop behind every audited number.

D AGGREGATION, AND HOW MUCH IT CHANGES THE HEADLINE

The nine target series carry different units and differ in scale by two orders of magnitude, so an average of raw MAE across them is a weighted average whose weights are an accident of measurement units. Table 5 places the two rules side by side. Under unit-weighted pooling of raw errors, electricity alone accounts for 91–95% of the total injected damage on every model, so the pooled degradation, recovery and clean cost are close to electricity's own values: 90–97% recovery at under 3% clean cost. Under equal dataset weights on normalized errors, the same experiments give 70–92% recovery at 4.8–8.2% clean cost. Neither rule is uniquely correct, but the difference is large enough that the conclusion depends on it, which is why we state the weights explicitly, report per-dataset values next to every aggregate, and give the range across datasets in the main tables. Two further consequences are worth flagging. First, the ranking of models is not stable across the two rules: TimesFM-2.0 is the most fragile model under unit-weighted pooling ($7.38\times$) and mid-pack under equal weights ($5.94\times$), which is why Sec. 4.1 draws its scaling conclusion only from the within-family Chronos ladder evaluated on identical data. Second, the direction of the correction

Gate statistics: the four detectors

```
rscale(v) = 1.4826 * median(|v - median(v)|) + 1e-8          # MAD-based robust sigma, used throughout
body = x[:b]        tail = x[b:]                             # split at the assumed or estimated onset

backcast     model-in-the-loop.  The model backcasts its own tail from the body, and the gate
             measures how far the observed tail sits from that backcast:
                 z = |mean(tail) - mean(model(body, tl))| / rscale(body)

robust-z     model-free.  Tail level against body level:
                 z = |mean(tail) - median(body)| / rscale(body)

detrended-z  model-free, trend-aware.  Fit a degree-1 polynomial on the body, extrapolate it over
             the tl tail positions, and score the tail residual against the body residual:
                 z = |mean(tail - trend_tail)| / rscale(body - trend_body)

CUSUM        model-free changepoint.  Maximize a robust two-sample level statistic over every split
             in the last 40% of the context, keeping at least 5 points on the right:
                 z = max over c of  |mean(x[c:]) - mean(x[:c])| / (rscale(x) * sqrt(1/c + 1/(n-c)))

A gate FIRES when z > tau.  The backcast gate runs at tau = 1.0 everywhere in the main text.  The
four statistics live on different scales, so we compare them not at a shared tau but at a
matched CLEAN fire rate, sweeping a 200-point grid of quantiles of each detector's own score
distribution.  The maximization makes CUSUM a poor detector and the best available locator: it
needs a high threshold to reach a given false-fire rate, but its argmax lands on the onset.
```

Figure 3: The four gate statistics, as implemented.

is the same for every model, so the qualitative findings (fragility is universal, level-structured, and only partly repairable) survive the change; it is the magnitudes that do not.

Table 5: Effect of the aggregation rule on every headline number.

Model	Unit-weighted pooling			Equal weight, normalised			Elec. share
	Degrad.	Recov.	Clean	Degrad.	Recov.	Clean	(%)
Chronos-base	6.03×	90.0	+3.08	5.80×	73.8	+7.80	94
TimesFM	6.41×	95.6	+0.83	5.75×	91.6	+8.07	95
TS-ICL	5.22×	87.3	+1.95	5.10×	70.4	+7.05	92
Moirai	6.31×	89.5	-0.88	5.73×	76.9	+4.81	94
Chronos-2	5.92×	93.2	-0.29	5.94×	77.4	+5.18	94
TimesFM-2.0	7.38×	97.2	+0.94	5.94×	91.3	+6.73	94
Moirai-MoE	4.18×	93.5	+1.10	4.64×	83.1	+5.13	91
TimeMoE	4.93×	94.2	+2.71	4.62×	91.7	+8.21	94

E CORRUPTION TAXONOMY

Table 6 reports the full taxonomy on the three base datasets. Two observations motivate the paper's focus. First, the corruptions that classical cleaning pipelines target are the ones TSFMs already survive: $\pm 8\sigma$ spikes on 5% of points cost at most 5% MAE and forward-filled 10% gaps at most 6%, on either audited model. Second, the two corruptions that do damage, sentinel blocks and level shifts, act through the same channel, the model's reading of the current level, but differ in observability: a -100σ sentinel is trivially separable from data and is fully repaired by the robust clip, while a 3σ tail shift is statistically unremarkable point by point and survives every point-wise filter. The taxonomy also explains the model dependence: Chronos tokenizes and effectively clips extreme values ($1.02\times$ under sentinels), while TS-ICL's instance normalization lets a single extreme block corrupt the statistics of the whole window ($2.14\times$).

Table 7 reports the one case in this paper where the corruption was not injected by us: weather.csv ships with -9999 sentinel entries in its target channel, and forecasting from a lookback that contains them degrades MAE 19–$44\times$. Because this is a single dataset, no cross-unit aggregation is involved and the raw MAEs are directly comparable. A one-line forward-fill removes essentially all of the damage, which is why we present the level-shift results, not the sentinel results, as the part of the audit that motivates a new method.

The complete policy, raw context to predictive distribution

```
INPUT   observed context x of length L, covariate channels C (L x J), horizon H, a frozen backbone
PARAM   gate threshold tau = 1.0;  corroboration band [c_lo, c_hi] = [1.0, 2.0];  mixture weight w = 0.5
        Nothing below is trained, and nothing below sees the future or a corruption label.

## 1. Clip.  Sentinels and spikes leave here; a level shift does not.
x = robust_clip(x, k = 8)

## 2. Locate the onset, rather than assume it.  CUSUM argmax over the last half of the context.
##    A FIXED boundary L - int(0.2 * L) appears only in the ablation that assumes the onset known.
b = argmax over c of  |mean(x[c:]) - mean(x[:c])| / (rscale(x) * sqrt(1/c + 1/(L-c)))

## 3. Gate.  The body is the part of the context the corruption is estimated not to reach, so the
##    model's own backcast of the tail is an estimate of what the tail should have been.
pred_tail = model(x[:b], L - b)
delta     = mean(x[b:]) - mean(pred_tail)                  # estimated level offset, signed
if |delta| / rscale(x[:b]) <= tau:  return model(x, H)     # KEEP: nothing looks wrong, do nothing

## 4. Certify against channels the corruption did not touch.  This is the only step that uses
##    information outside the target series, and the only reason the dilemma below is escapable.
corr = median over j of  |mean(C[b:, j]) - median(C[:b, j])| / rscale(C[:b, j])

## 5. Route on that evidence.
x_ds = x;   x_ds[b:] -= delta                              # the de-shifted context
if corr <  c_lo:  return model(x_ds, H)      # ISOLATED move: the target jumped alone -> repair it
if corr >  c_hi:  return model(x,    H)      # BROAD move:    every channel jumped     -> believe it
                                            # neither: the evidence does not decide -> HEDGE

## 6. Hedge.  Mix the two hypotheses as DISTRIBUTIONS, not as forecasts.
F1 = predictive CDF of model(x,    H)                      # "the level is real"
F2 = predictive CDF of model(x_ds, H)                      # "the level is an artifact"
return quantile function of  w * F1 + (1 - w) * F2         # linear pool (Gneiting & Ranjan, 2013)
       Averaging the two quantile FUNCTIONS instead would return a forecast at the midpoint of two
       hypotheses, which is the |delta|/2 point forecast the impossibility result rules out.  The
       pooled CDF is flat between the two modes and every value there is a valid quantile, so the
       inversion takes the MIDPOINT of that plateau, which is symmetric in the two hypotheses.

SCORING
  Sample families (Chronos, Chronos-Bolt, Moirai, Moirai-MoE) are read at the empirical quantiles of
  their sample paths.  Quantile-head families (Chronos-2, TimesFM 1.0 and 2.0, TS-ICL) report 9 native
  levels 0.1..0.9 and are interpolated IN TAU, never by taking quantiles of quantiles.  Scores are a
  censored CRPS, 2 * integral over tau in [0.1, 0.9] of the pinball loss, on a 0.05 grid: the range
  every family represents natively, so no family is scored where it must extrapolate.
```

Figure 4: The policy of Sec. 3, end to end.

Table 6: Mean MAE degradation for each corruption in the taxonomy.

Model	Spikes	Missing+ffill	Sentinel	Level shift
Chronos-base	1.01×	1.04×	1.02×	4.46×
TS-ICL	1.05×	1.06×	2.14×	3.95×

F PER-DATASET FRAGILITY AND SANITIZER BEHAVIOR

Figure 6 maps degradation across all 72 (model, dataset) cells, and Table 8 adds recovery, clean cost, and clean fire rate. Three regularities emerge. First, the dataset, not the model, is the dominant factor: the traffic column (15.7–23.1×) and electricity column (4.4–8.1×) dwarf between-model variation, because on bounded, strongly periodic series a 3σ shift is enormous relative to the residual scale the models work against. Second, the mild columns are informative: solar (1.5–2.8×) is protected by its zero-inflated day–night cycle, which dominates σ and makes the injected shift comparatively small, and exchange (2.1–4.1×) is protected by being a near-random-walk whose forecasts track the recent level anyway. Third, this spread is the reason the aggregation rule of App. D matters so much: the datasets with the largest degradation ratios are not the datasets with the largest raw errors, so the two rules weight different phenomena.

The clean-cost and fire-rate columns are the more consequential half of the table. The three datasets with genuine level moves (ETTm1, ETTm2, exchange) show 19–39% clean fire rates and +6 to +25% clean cost under hard correction, on every model, while the strongly periodic datasets show

One window: input, gate, routing, and the recorded result row

```
INPUT       dataset ETTh1 (target OT), model Chronos-2, origin o = 2914, one of the 100 audited windows
            context y[2578:2914]  (L = 336)          future y[2914:3010]  (H = 96, never corrupted)

CORRUPTION  m = 3  ->  offset = 3 * std(context) = 11.78, added to the last 67 points (onset b = 269)

CONTEXT     median(body)              16.95      rscale(body)               3.76
            mean(tail), clean         11.59      mean(tail), corrupted     23.37
            does the robust clip fire?  no       the shift is 3.1 robust sigma, well inside the k = 8 rule

GATE        z_backcast on the CLEAN context      0.31   -> below tau = 1.0, the gate does not fire
            z_backcast on the CORRUPTED context  2.82   -> FIRES;  estimated offset 2.82 * 3.76 = 10.60
                                                                    against a true offset of 11.78
            the other three, on the corrupted context:  robust-z 1.71   detrended-z 8.35   CUSUM 11.73

CERTIFY     6 covariate channels.  Median covariate z, and where it sends this window:
            0.27   only the target moved      -> below c_lo -> DE-SHIFT
            3.02   every channel moved        -> above c_hi -> KEEP
            A window landing between 1.0 and 2.0 would be hedged; 7 of this dataset's 100 windows do.

RESULT ROW  censored CRPS on [0.1, 0.9], raw units, exactly as the results file records it:
                                       transient future     persistent future      worst world
            keep (no sanitizer)                 5.43                  2.29               5.43
            always de-shift                     1.34                  7.77               7.77
            mixture (linear pool)               2.72                  4.38               4.38
            clean-context oracle                1.36                  1.36               1.36

            Each hard policy wins the world it assumes and loses the other by more than it won.  The
            mixture is beaten by de-shift where de-shift is right and by keep where keep is right, and
            still has the best worst case of the three.  That trade is the content of the impossibility
            result: no point forecast escapes half the offset, and hedging pays a quarter of it.
```

Figure 5: One real window, from raw context to recorded scores.

Table 7: Forecasts from weather.csv contexts containing native -9999 sentinels.

Model	Raw	Ffill	Robust clip	Full sanitizer
Chronos-base	239.6	8.93	9.57	11.18
TimesFM	380.3	9.30	10.43	10.53
TS-ICL	489.5	10.42	10.66	11.10
Moirai	278.5	13.07	14.12	14.45

0–2% fire at essentially zero cost. The gate is not misbehaving on the former: it is correctly detecting level moves whose future is simply unknowable from the target series. A reader who intends to deploy hard de-shift should read this table rather than the aggregate, and should expect the aggregate only on data whose level is known to be stable.

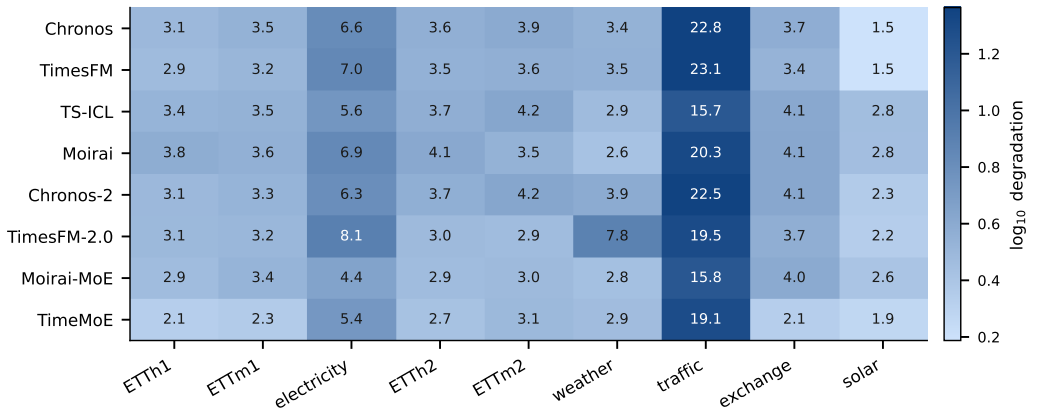

Figure 6: Level-shift degradation for every (model, dataset) pair.

Table 8: Per-dataset fragility, recovery, clean cost and clean fire rate, for every model.

Model	Dataset	Degr.	Rec. %	Cost %	Fire %	Model	Dataset	Degr.	Rec. %	Cost %	Fire %
Chronos-base	ETTh1	3.12×	67.4	+5.11	22	Chronos-2	ETTh1	3.10×	75.4	+2.32	19
Chronos-base	ETTh2	3.61×	68.8	+7.40	13	Chronos-2	ETTh2	3.65×	63.8	+4.82	15
Chronos-base	ETTm1	3.47×	45.9	+16.05	25	Chronos-2	ETTm1	3.34×	49.9	+8.97	20
Chronos-base	ETTm2	3.91×	26.3	+12.39	24	Chronos-2	ETTm2	4.25×	14.1	+7.51	19
Chronos-base	electricity	6.56×	92.8	+2.27	2	Chronos-2	electricity	6.33×	95.3	-1.11	1
Chronos-base	exchange	3.71×	8.5	+13.30	24	Chronos-2	exchange	4.14×	23.9	+15.54	23
Chronos-base	solar	1.54×	94.3	+8.23	96	Chronos-2	solar	2.27×	92.9	+1.16	21
Chronos-base	traffic	22.79×	92.2	+0.00	0	Chronos-2	traffic	22.49×	95.3	+0.00	0
Chronos-base	weather	3.45×	31.1	+5.48	34	Chronos-2	weather	3.90×	69.6	+7.40	22
TimesFM	ETTh1	2.94×	84.3	+6.85	25	TimesFM-2.0	ETTh1	3.09×	86.1	+7.60	22
TimesFM	ETTh2	3.49×	85.6	+5.62	12	TimesFM-2.0	ETTh2	3.00×	90.1	+5.14	14
TimesFM	ETTm1	3.21×	70.0	+15.74	23	TimesFM-2.0	ETTm1	3.17×	72.1	+16.97	28
TimesFM	ETTm2	3.60×	71.3	+13.39	22	TimesFM-2.0	ETTm2	2.92×	76.3	+10.07	23
TimesFM	electricity	7.04×	96.1	+0.00	0	TimesFM-2.0	electricity	8.09×	97.5	+0.00	0
TimesFM	exchange	3.37×	59.5	+25.43	30	TimesFM-2.0	exchange	3.70×	38.9	+13.31	25
TimesFM	solar	1.54×	141.0	+1.10	97	TimesFM-2.0	solar	2.17×	126.5	+2.09	97
TimesFM	traffic	23.08×	99.5	+0.00	0	TimesFM-2.0	traffic	19.54×	99.6	+0.00	0
TimesFM	weather	3.50×	83.5	+4.50	18	TimesFM-2.0	weather	7.78×	89.2	+5.35	8
TS-ICL	ETTh1	3.35×	71.0	+6.24	20	Moirai-MoE	ETTh1	2.91×	85.1	+1.68	21
TS-ICL	ETTh2	3.69×	82.5	+2.62	9	Moirai-MoE	ETTh2	2.93×	88.3	+8.32	18
TS-ICL	ETTm1	3.45×	40.9	+14.34	25	Moirai-MoE	ETTm1	3.36×	51.7	+13.09	25
TS-ICL	ETTm2	4.24×	-5.1	+14.40	26	Moirai-MoE	ETTm2	3.02×	33.9	+10.68	30
TS-ICL	electricity	5.59×	91.1	+1.53	2	Moirai-MoE	electricity	4.44×	96.6	+0.91	1
TS-ICL	exchange	4.06×	34.0	+20.54	25	Moirai-MoE	exchange	3.97×	71.8	+14.21	25
TS-ICL	solar	2.82×	47.3	+1.65	24	Moirai-MoE	solar	2.55×	41.0	-1.19	98
TS-ICL	traffic	15.74×	97.8	+0.00	0	Moirai-MoE	traffic	15.81×	98.5	+0.00	0
TS-ICL	weather	2.94×	39.4	+2.14	33	Moirai-MoE	weather	2.75×	74.0	-1.49	38
Moirai	ETTh1	3.77×	69.7	+2.51	19	TimeMoE	ETTh1	2.15×	87.1	+3.91	22
Moirai	ETTh2	4.06×	82.1	+9.83	18	TimeMoE	ETTh2	2.66×	87.8	+3.87	12
Moirai	ETTm1	3.57×	33.1	+17.54	29	TimeMoE	ETTm1	2.26×	65.5	+16.57	32
Moirai	ETTm2	3.46×	34.0	+6.19	32	TimeMoE	ETTm2	3.09×	78.8	+12.31	22
Moirai	electricity	6.89×	92.0	-1.36	2	TimeMoE	electricity	5.38×	94.8	+2.03	2
Moirai	exchange	4.13×	17.9	+10.01	22	TimeMoE	exchange	2.14×	56.8	+24.94	39
Moirai	solar	2.80×	67.5	+2.98	25	TimeMoE	solar	1.86×	89.7	+4.75	20
Moirai	traffic	20.28×	98.9	+0.00	0	TimeMoE	traffic	19.15×	98.1	+0.00	0
Moirai	weather	2.61×	21.9	-4.42	44	TimeMoE	weather	2.88×	83.7	+5.53	23

Table 9: Fragility across the Chronos scale ladder on identical data.

Model	Degrad.	Recov. (%)	Clean cost (%)
Chronos-tiny	3.55×	70	+9.2
Chronos-small	4.07×	71	+7.6
Chronos-base	4.34×	74	+7.6
Chronos-large	4.37×	77	+9.6
Chronos-Bolt	4.09×	75	+8.2

G DETECTOR OPERATING CHARACTERISTICS

Figure 8 plots, for each model, the operating frontier of the four detectors: normalized recovery against clean fire rate as the threshold sweeps. The main-text comparison at a matched clean-cost budget (Table 10) corresponds to a vertical slice of each panel, and the frontiers show the comparison is not an artifact of that slice. For most models the backcast, robust-z and detrended-z frontiers are nearly coincident above fire rates of about 10%, which is the honest core of the detector claim: at usable operating points the detectors are interchangeable and the recipe needs no privileged model access. The exceptions are systematic rather than random. The plain robust z collapses at low fire rates on Chronos-base and TimeMoE, the two models whose forecasts drift on trending windows, because a trending body shifts the body median away from the tail and inflates the statistic on clean data; the detrended variant repairs exactly this. CUSUM is dominated nearly everywhere as a *detector*, because its statistic is maximized over candidate changepoints and therefore needs a far higher threshold to reach the same false-fire rate; App. I shows that the same maximization makes it the best available *locator*, which is a different job.

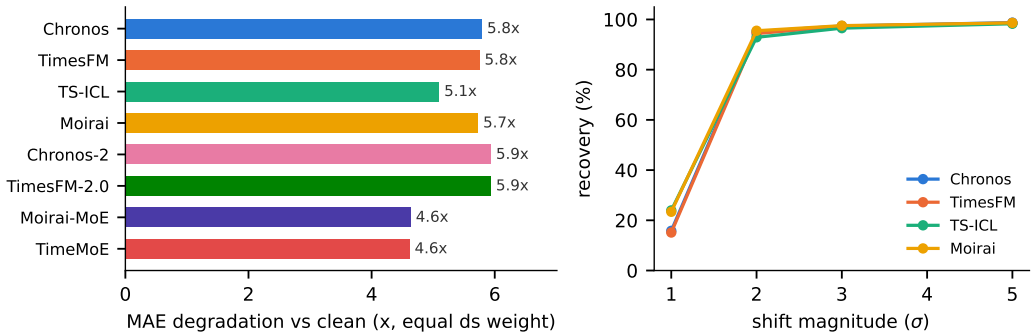

Figure 7: (a) level-shift degradation across the eight models; (b) recovery against shift magnitude with a correctly specified boundary.

We withdraw a claim made in an earlier version of this work, that the model-in-the-loop backcast gate's advantage grows with backbone quality. That reading rested on a single comparison (TimesFM-2.0, 97.0% versus 89.4%) computed under unit-weighted pooling and over a six-point threshold grid. Under equal dataset weights and a fine grid, TimesFM-2.0 shows 90% for the backcast gate against 92% for robust-z at a matched 5% clean-cost budget, and at a stricter 2% budget the backcast gate leads on four models and trails on four, with no relation to backbone quality that we can detect.

Table 10: Best recovery at a matched clean-cost budget for four level detectors.

Model	Backcast	Robust-z	Detrend-z	CUSUM	Δ at $\leq$2%
Chronos-base	70	74	73	60	+10
TimesFM	89	90	87	73	+8
TS-ICL	73	71	70	59	+21
Moirai	83	92	79	78	-20
Chronos-2	77	78	52	75	-5
TimesFM-2.0	90	92	87	85	+5
Moirai-MoE	85	85	84	83	-8
TimeMoE	90	89	90	73	-8

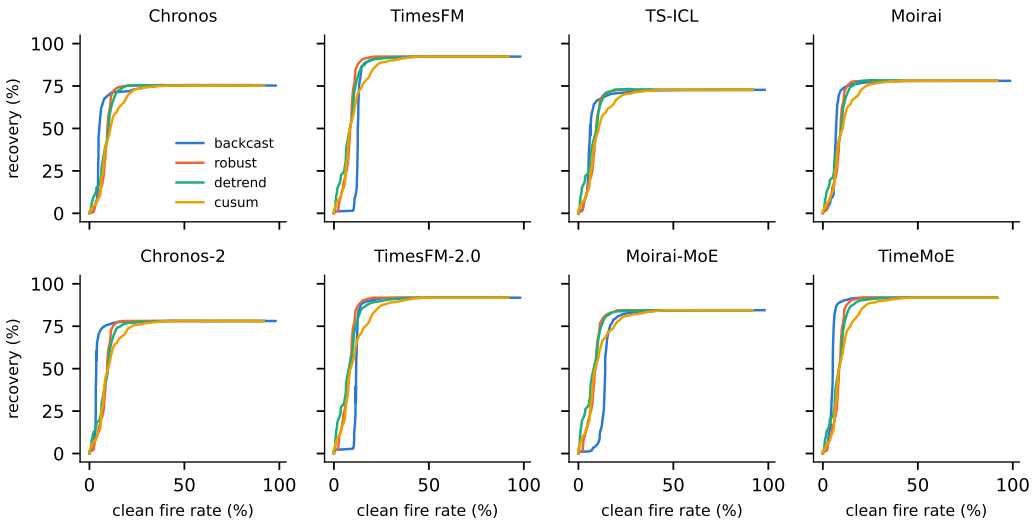

Figure 8: Detector operating frontiers per model: normalized recovery against clean fire rate.

H MAGNITUDE AND ONSET SWEEP WITH A CORRECTLY SPECIFIED BOUNDARY

Table 11 summarizes recovery by magnitude at the default onset, and Table 12 expands the full magnitude-by-onset grid with per-onset clean cost. These numbers are *privileged*: in this sweep the method's assumed boundary equals the injected onset, so they bound what a correctly specified boundary can achieve and should not be read as deployment performance. App. I reports the deployable setting. Within that caveat, the grid isolates magnitude as the interesting margin. Figure 9 shows that onset position is nearly irrelevant when the boundary is known, because the gate statistic compares levels rather than durations, and Figure 10 explains the magnitude dependence mechanically: the gate statistic z separates cleanly from its clean distribution at 2σ and above, while the 1σ distribution straddles the default threshold $\tau = 1$. This is a detection frontier rather than a cliff: at $\tau = 1$ the gate fires on only 47–52% of 1σ shifts and recovers 15–24% of their damage, while moving τ to 0.75 lifts that to 61–66% for one to two further points of clean damage. The practical reading is that the sanitizer removes most of the catastrophic damage, which comes from large shifts, and degrades gracefully on small ones, whose absolute harm is also the smallest.

Table 11: Recovery (%) by shift magnitude (onset = last 20%, τ=1).

Model	1σ	2σ	3σ	5σ
Chronos-base	29	98	99	99
TimesFM	26	97	98	99
TS-ICL	29	96	98	99
Moirai	49	99	99	100

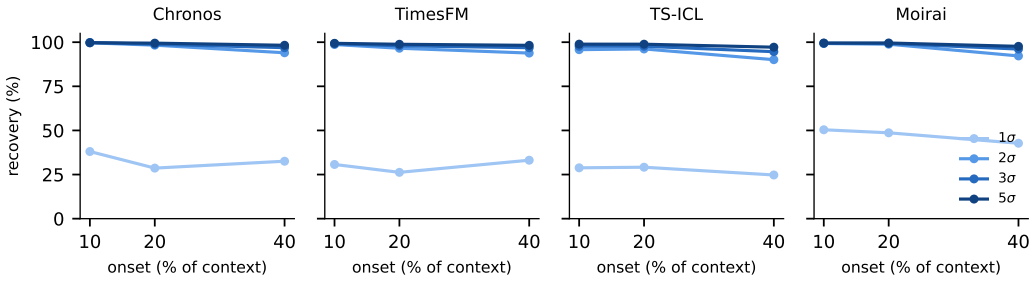

Figure 9: Recovery against onset fraction for each magnitude, with the boundary correctly specified.

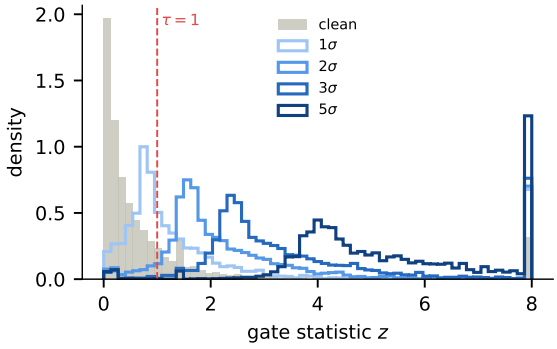

Figure 10: Distribution of the gate statistic z on clean and shifted contexts.

Table 12: Recovery by shift magnitude and onset fraction with a correctly specified boundary.

Model	Onset	1σ	2σ	3σ	5σ	Clean (%)
Chronos-base	10%	19	98	98	99	+2.81
Chronos-base	20%	16	95	97	99	+8.08
Chronos-base	40%	6	88	94	97	+19.98
TimesFM	10%	16	97	98	99	+3.49
TimesFM	20%	15	94	97	99	+8.07
TimesFM	40%	9	89	94	97	+18.20
TS-ICL	10%	28	95	98	99	+3.10
TS-ICL	20%	24	93	97	98	+7.05
TS-ICL	40%	12	87	93	96	+16.59
Moirai	10%	31	97	98	99	+3.43
Moirai	20%	23	96	98	99	+4.56
Moirai	40%	11	86	92	96	+17.85

I Unknown onset

Table 2 reports the deployable setting, in which the injected onset varies independently of the boundary the method assumes and every detector input, the backcast body included, is taken from the corrupted context. The three boundary rules are calibrated to a common clean-cost budget, because the CUSUM-located split maximizes the level jump by construction and would otherwise be compared at a looser operating point.

The failure of a fixed boundary is structural, and the fire rates show the mechanism directly. With $\hat{p} = 0.2$ held constant, the gate fires on 72–92% of corrupted windows when the true onset is at 10% or 20% of the context, but on only 26–42% when it is at 30% and 13–16% when it is at 40%; recovery falls correspondingly, from 53–92% to 6–32% and then to 0–6%. Once the shift begins before the assumed boundary, the assumed body contains part of the shift, the model's backcast of the tail is computed from an already elevated body and therefore predicts an elevated tail, the gate statistic shrinks below threshold, and the correction never fires. The method does not degrade; it switches off.

Estimating the boundary removes the problem entirely. A CUSUM changepoint search over the last half of the context recovers the onset almost exactly: on all three models the median estimated boundary is within one index of the truth at every onset (302 against a true 303, 268 against 269, 235 against 236, 201 against 202), and recovery is 78–95% at every onset at matched clean cost. The estimated boundary matches or outperforms the *oracle* boundary, strikingly so on TS-ICL (90–93% against 61–69%), because a split chosen to maximize the level jump produces a larger gate statistic than the nominal onset and so crosses the threshold more reliably. We therefore treat boundary estimation as part of the method rather than a detail, and the detector that is worst at deciding *whether* the level moved turns out to be the one that is best at deciding *where*.

J Locating the level sensitivity

Table 3 and Figure 11 report the intervention on the location of the perturbation, and Table 13 breaks the coefficients down by dataset. The logic of the design is that the two candidate mechanisms make opposite predictions about a body shift. A forecaster whose sensitivity runs through normalization by window statistics must respond in proportion to how much the perturbation moves those statistics, and a shift on the first 80% of the context moves the window mean by 0.8δ against a tail shift's 0.2δ; the window-mean control reproduces exactly this ratio ($\alpha = 0.20$ for a tail shift and 0.80 for a body shift). A forecaster that extrapolates the recent level responds to the tail and ignores the body, which is what the persistence, seasonal-naive and exponential-smoothing controls do ($\alpha = 1.00$ and 0.00). Every model tested falls on the second side.

Three details support reading the measurement rather than the story. First, the whole-window shift is a calibration check that any level-equivariant forecaster must pass with $\alpha = 1$, and the models return 0.99–1.00, so the regression is measuring what it claims to. Second, the tail-shift fits are

tight ($R^2 \geq 0.95$ on every released model), so the near-linear anchoring model is not an artifact of averaging. Third, the per-dataset coefficients are stable, so the conclusion is not carried by one series.

Normalization is not absent, only secondary. Restoring the window mean and standard deviation to their clean values while keeping the tail elevated reduces the models' anchoring, but it also mechanically compresses the elevation that the input presents, which is why the persistence controls are reduced by a similar factor under the same transformation; the fair comparison is the ratio of the model's response to the control's, which falls only modestly. The clean case is TimeMoE, the one family whose normalization we control directly at the interface rather than inside the network: replacing its window mean and standard deviation with the median and MAD of the context body, which a tail defect cannot move, is a direct intervention on the normalization statistics, and Table 3 reports what it does to both the anchoring coefficient and the degradation.

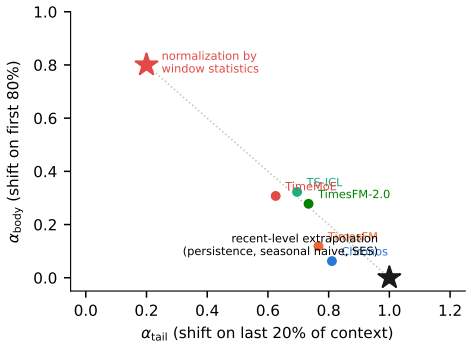

Figure 11: Each model placed by its response to a shift on the last 20% of the context and on the first 80%; normalization by window statistics predicts $(0.2, 0.8)$ and recent-level extrapolation predicts $(1, 0)$.

K Corroboration threshold study

Table 15 and Figure 12 sweep the corroboration threshold c. The two worlds respond asymmetrically, which is what makes the gate usable without tuning: in the artifact world (covariates clean) the fire rate rises steeply and saturates around 82% by $c = 1.5$, while in the regime world (covariates co-shifted) it is exactly 0% up to $c = 1$, 0.2% (2 windows in 900) at $c = 1.5$, and only then rises, to 3.8–3.9% at $c = 2$ and 29% at $c = 3$. Consequently the achieved error sits on the always-de-shift reference in the transient world and on the always-keep reference in the persistent world across the plateau $c \in [1, 1.5]$, on all eight models, and degrades gracefully rather than abruptly beyond it. The residual 18% of artifacts that never fire are windows whose covariates happen to drift naturally at the moment of corruption; they are the price of a near-zero-false-fire operating point, and the mixture policy covers exactly this remainder.

The separation is wide, but it is worth restating what produces it. In this experiment the synthetic regime change shifts every covariate channel and the synthetic artifact shifts none, so the sweep measures how cleanly a median-covariate statistic separates a co-movement signal that we injected. It establishes that the routing step works when regimes are covariate-visible and faults are single-channel; it does not establish how often that condition holds, and the condition is not checkable from the series itself. The inconclusive band used by the complete policy exists precisely because we do not want the routing step to commit when the covariate evidence is weak.

L Full probabilistic results, corrected and superseded

Table 16 reports normalized CRPS in both worlds together with coverage of the native central 80% interval, and, alongside it, the coverage an earlier version of this work reported. Figure 13 visualizes both.

Table 13: Anchoring coefficient per dataset and perturbation location.

Model	Dataset	Tail	Body	Whole	Tail/Body
Chronos-base	ETTh1	0.80	0.12	0.98	6.5
Chronos-base	ETTh2	0.79	0.19	1.02	4.2
Chronos-base	ETTm1	0.93	0.04	0.99	22.5
Chronos-base	ETTm2	0.95	0.14	1.02	6.8
Chronos-base	electricity	0.81	0.06	0.99	13.0
Chronos-base	exchange	1.01	0.03	1.00	39.1
Chronos-base	solar	0.28	0.02	0.99	16.8
Chronos-base	traffic	0.91	-0.01	1.00	913510.0
Chronos-base	weather	1.00	0.24	1.01	4.2
TimesFM	ETTh1	0.90	0.16	1.00	5.8
TimesFM	ETTh2	0.75	0.12	1.00	6.3
TimesFM	ETTm1	0.90	0.20	1.00	4.5
TimesFM	ETTm2	0.74	-0.03	1.00	739380.8
TimesFM	electricity	0.77	0.12	1.00	6.5
TimesFM	exchange	0.92	-0.04	1.00	924567.9
TimesFM	solar	0.37	0.39	1.00	1.0
TimesFM	traffic	0.88	0.16	1.00	5.6
TimesFM	weather	0.56	0.45	1.00	1.3
TS-ICL	ETTh1	0.79	0.25	1.00	3.2
TS-ICL	ETTh2	0.75	0.28	1.00	2.7
TS-ICL	ETTm1	0.84	0.25	1.00	3.3
TS-ICL	ETTm2	0.85	0.13	1.00	6.8
TS-ICL	electricity	0.70	0.32	1.00	2.2
TS-ICL	exchange	0.96	0.12	1.00	8.1
TS-ICL	solar	0.82	0.18	1.00	4.6
TS-ICL	traffic	0.72	0.24	1.00	3.1
TS-ICL	weather	0.86	0.22	1.00	3.9
TimesFM-2.0	ETTh1	0.75	0.15	1.00	5.0
TimesFM-2.0	ETTh2	0.61	0.38	1.00	1.6
TimesFM-2.0	ETTm1	0.86	0.14	1.00	6.0
TimesFM-2.0	ETTm2	0.68	0.14	1.00	5.0
TimesFM-2.0	electricity	0.73	0.28	1.00	2.6
TimesFM-2.0	exchange	0.99	0.01	1.00	171.6
TimesFM-2.0	solar	0.68	0.05	1.00	13.5
TimesFM-2.0	traffic	0.71	0.22	1.00	3.3
TimesFM-2.0	weather	0.69	0.26	1.00	2.6
TimeMoE	ETTh1	0.50	0.42	1.00	1.2
TimeMoE	ETTh2	0.54	0.29	1.00	1.8
TimeMoE	ETTm1	0.59	0.28	1.00	2.2
TimeMoE	ETTm2	0.66	0.33	1.00	2.0
TimeMoE	electricity	0.63	0.31	1.00	2.0
TimeMoE	exchange	0.71	0.26	1.00	2.8
TimeMoE	solar	0.36	0.68	1.00	0.5
TimeMoE	traffic	0.77	0.21	1.00	3.6
TimeMoE	weather	0.66	0.42	1.00	1.6
TimeMoE (robust norm.)	ETTh1	0.28	0.65	1.00	0.4
TimeMoE (robust norm.)	ETTh2	0.40	0.49	1.00	0.8
TimeMoE (robust norm.)	ETTm1	0.35	0.66	1.00	0.5
TimeMoE (robust norm.)	ETTm2	0.52	0.63	1.00	0.8
TimeMoE (robust norm.)	electricity	0.44	0.57	1.00	0.8
TimeMoE (robust norm.)	exchange	0.39	0.60	1.00	0.7
TimeMoE (robust norm.)	solar	0.10	0.98	1.00	0.1
TimeMoE (robust norm.)	traffic	0.69	0.39	1.00	1.8
TimeMoE (robust norm.)	weather	0.48	0.56	1.00	0.9

The correction matters because the two are not a constant apart. The superseded implementation treated every family's atoms as equally weighted samples and then took empirical quantiles of them. For the sample-based families (Chronos, Moirai, Moirai-MoE) that is the right estimator and nothing changes. For the quantile-head families (Chronos-2, both TimesFM versions, TS-ICL) the atoms *are*

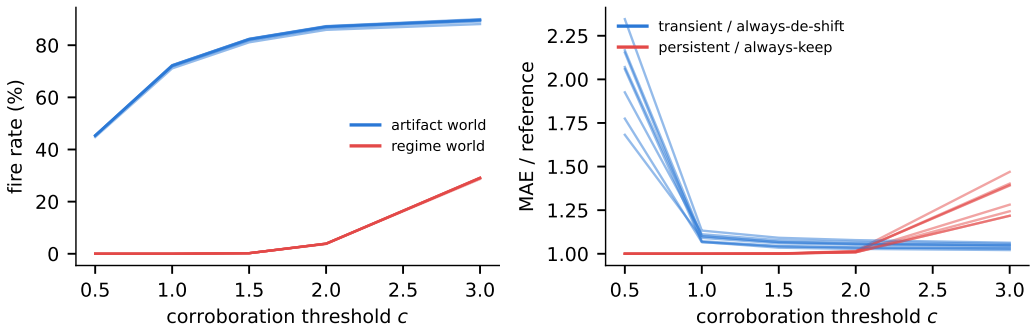

Figure 12: Corroboration threshold sweep, one line per model: (a) fire rates in the two worlds; (b) achieved error against the world-appropriate reference.

Table 14: Transient and persistent error per policy, normalized by the clean-context oracle.

	Hard policies		Hedged and gated		
Model	Keep	De-shift	Hedge	Corr-gate	Oracle
Chronos-base	5.7/1.7	1.1/6.7	3.0/3.9	1.4/1.7	1.0/1.0
TimesFM	5.7/2.0	1.1/6.9	3.0/4.2	1.4/2.0	1.0/1.0
TS-ICL	5.1/2.0	1.1/6.2	2.8/3.8	1.5/2.0	1.0/1.0
Moirai	5.8/2.3	1.1/5.5	3.1/2.9	1.5/2.4	1.0/1.0
Chronos-2	5.9/2.0	1.1/7.0	3.1/4.2	1.5/2.0	1.0/1.0
TimesFM-2.0	5.9/2.5	1.1/7.8	3.1/4.9	1.5/2.5	1.0/1.0
Moirai-MoE	4.6/2.1	1.1/5.9	2.5/3.8	1.4/2.1	1.0/1.0
TimeMoE	4.6/2.6	1.1/6.8	2.5/4.6	1.3/2.6	1.0/1.0

quantiles at levels $0.1, \ldots, 0.9$, and taking the empirical 0.1 and 0.9 quantiles of nine such atoms returns values near the 0.18 and 0.82 quantiles of the underlying distribution: a nominal 80% interval evaluated as a 64% one. The correction behaves exactly as that account predicts, which is the check that makes it credible: for Moirai, whose atoms are 20 sample paths, the native and superseded coverages agree to three decimals on every pure policy (0.290 vs 0.290 for keep, 0.633 vs 0.633 for de-shift, 0.683 vs 0.683 for the oracle), while for the three quantile-head families every entry moves by 13–20 points in the direction the geometry requires (TS-ICL's oracle, for instance, from 0.600 to 0.738). The mixture is also reconstructed differently: the superseded version pooled quantile atoms into one bag, whereas the linear pool averages the two CDFs and inverts, which is the mixture distribution the policy actually forecasts (Gneiting & Ranjan, 2013). That changes the mixture row even for the sample-based families, because the pooled CDF is flat between the two hypotheses and every value there is a valid quantile; we take the midpoint of that flat region rather than its left edge, since the left edge would silently bias the mixture toward whichever hypothesis predicts lower values.

The qualitative conclusion is unchanged and, if anything, strengthened: hard policies fail catastrophically at uncertainty quantification in the world they were not built for, keep's intervals collapsing in the transient world and de-shift's in the persistent world, while the mixture is the only policy whose stated uncertainty remains usable without knowing which world holds. What changes is that the corrected numbers are interpretable as calibration statements, which the superseded ones were not.

M THE COMPLETE POLICY, END TO END

Table 17 scores the full recipe of Sec. 4.4 against its components, in normalized CRPS, together with the fraction of windows routed to each branch. Reporting the recipe rather than its parts matters because the parts are evaluated on different subsets: the gate's clean cost is measured on uncorrupted contexts, the corroboration gate on the two synthetic worlds, and the mixture on gate-fired windows, and none of those numbers composes into a deployment guarantee on its own. The two rows to

Table 15: Corroboration threshold sweep in the artifact and regime worlds.

Model	cthr	Transient	Persistent	Fire art. (%)	Fire reg. (%)
Chronos-base	0.5	2.9	1.7	45	0
Chronos-base	1.0	1.6	1.7	72	0
Chronos-base	1.5	1.4	1.7	82	0
Chronos-base	2.0	1.2	1.8	87	4
Chronos-base	3.0	1.2	2.6	89	29
TimesFM	0.5	2.8	2.0	45	0
TimesFM	1.0	1.6	2.0	72	0
TimesFM	1.5	1.4	2.0	82	0
TimesFM	2.0	1.3	2.1	87	4
TimesFM	3.0	1.2	2.8	90	29
TS-ICL	0.5	2.9	2.0	45	0
TS-ICL	1.0	1.8	2.0	72	0
TS-ICL	1.5	1.5	2.0	82	0
TS-ICL	2.0	1.4	2.1	87	4
TS-ICL	3.0	1.3	2.8	90	29
Moirai	0.5	3.0	2.3	45	0
Moirai	1.0	1.8	2.3	71	0
Moirai	1.5	1.5	2.4	81	0
Moirai	2.0	1.4	2.5	86	4
Moirai	3.0	1.3	3.0	88	29
Chronos-2	0.5	3.0	2.0	45	0
Chronos-2	1.0	1.7	2.0	72	0
Chronos-2	1.5	1.5	2.0	82	0
Chronos-2	2.0	1.3	2.1	87	4
Chronos-2	3.0	1.3	2.9	90	29
TimesFM-2.0	0.5	3.1	2.5	45	0
TimesFM-2.0	1.0	1.8	2.5	72	0
TimesFM-2.0	1.5	1.5	2.5	82	0
TimesFM-2.0	2.0	1.3	2.7	87	4
TimesFM-2.0	3.0	1.3	3.6	90	29
Moirai-MoE	0.5	2.6	2.1	45	0
Moirai-MoE	1.0	1.7	2.1	72	0
Moirai-MoE	1.5	1.4	2.1	82	0
Moirai-MoE	2.0	1.3	2.2	87	4
Moirai-MoE	3.0	1.3	2.8	89	29
TimeMoE	0.5	2.4	2.6	45	0
TimeMoE	1.0	1.5	2.6	72	0
TimeMoE	1.5	1.3	2.6	82	0
TimeMoE	2.0	1.2	2.6	87	4
TimeMoE	3.0	1.2	3.2	90	29

compare are the full recipe, which uses covariate evidence, and gate-plus-mixture, which does not and is the fallback wherever covariates are uninformative or absent. The gap between them is the value of side information, and the gap between the full recipe and the clean-context oracle is what remains unresolved even with it.

N VETO-FAILURE ANALYSIS

Table 18 quantifies the two single-series vetoes discussed in the main text, evaluated on ETTm1, the dataset where natural persistent shifts are densest. The covariate-median veto at its tightest setting reduces the clean cost of hard correction only from $+29\%$ to $+18\%$ while removing most transient recovery, because ETT covariates drift continually and the median covariate statistic is high on ordinary windows regardless of whether the target's move is real. The step-shape veto (fire only when the tail move contains a sharp jump) fails differently: at thresholds tight enough to protect natural windows it also rejects genuine injected steps, because ETTm1's natural moves include true steps and the jump-statistic distributions overlap. Both failures are instances of Proposition 1: the distinguishing information is absent from the target series, so any single-series statistic trades

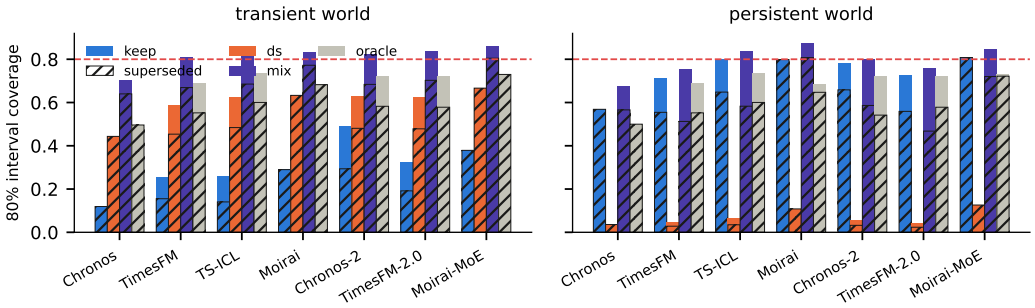

Figure 13: Coverage of nominal 80% intervals per policy and world, under native (solid) and superseded (hatched) scoring.

Table 16: Full probabilistic results under corrected and superseded scoring.

Model	Policy	CRPS (t)	CRPS (p)	Cov. (t)	Cov. (p)	Cov. (t) old	Cov. (p) old
Chronos-base	keep	6.42	1.47	0.12	0.57	0.12	0.57
Chronos-base	ds	1.13	7.48	0.44	0.04	0.44	0.04
Chronos-base	mix	2.55	3.26	0.70	0.67	0.64	0.57
Chronos-base	oracle	1.00	1.00	0.50	0.50	0.50	0.50
TimesFM	keep	6.03	1.81	0.25	0.71	0.16	0.55
TimesFM	ds	1.16	7.80	0.59	0.05	0.45	0.03
TimesFM	mix	2.51	3.72	0.81	0.75	0.67	0.51
TimesFM	oracle	1.00	1.00	0.69	0.69	0.55	0.55
TS-ICL	keep	5.24	1.76	0.26	0.80	0.14	0.65
TS-ICL	ds	1.14	6.98	0.62	0.06	0.48	0.03
TS-ICL	mix	2.28	3.47	0.83	0.84	0.69	0.58
TS-ICL	oracle	1.00	1.00	0.74	0.74	0.60	0.60
Moirai	keep	5.77	2.29	0.29	0.80	0.29	0.80
Moirai	ds	1.10	6.15	0.63	0.11	0.63	0.11
Moirai	mix	2.44	3.23	0.83	0.88	0.77	0.81
Moirai	oracle	1.00	1.00	0.68	0.68	0.68	0.65
Chronos-2	keep	5.44	1.92	0.49	0.78	0.29	0.66
Chronos-2	ds	1.14	7.93	0.63	0.05	0.48	0.03
Chronos-2	mix	2.36	3.99	0.82	0.80	0.68	0.59
Chronos-2	oracle	1.00	1.00	0.72	0.72	0.58	0.54
TimesFM-2.0	keep	6.01	2.33	0.32	0.73	0.19	0.56
TimesFM-2.0	ds	1.14	8.87	0.62	0.04	0.48	0.02
TimesFM-2.0	mix	2.51	4.53	0.84	0.76	0.70	0.47
TimesFM-2.0	oracle	1.00	1.00	0.72	0.72	0.58	0.58
Moirai-MoE	keep	4.68	1.95	0.38	0.81	0.38	0.81
Moirai-MoE	ds	1.10	6.60	0.67	0.13	0.67	0.13
Moirai-MoE	mix	2.09	3.48	0.86	0.85	0.80	0.72
Moirai-MoE	oracle	1.00	1.00	0.73	0.73	0.73	0.72

one error for the other. The corroboration gate escapes only because it imports information from channels the corruption did not touch, and only to the extent that those channels carry it.

Table 17: The complete policy scored end to end, in normalized CRPS.

Model	Policy	Transient	Persistent	Worst	Natural
Chronos-base	Naive (no sanitizer)	6.42	1.47	6.42	1.00
Chronos-base	Always de-shift	1.13	7.48	7.48	1.15
Chronos-base	Gate + mixture	2.56	3.26	3.26	1.01
Chronos-base	Full recipe	1.45	1.50	1.50	1.05
Chronos-base	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	55% / 36% / 9%				
TimesFM	Naive (no sanitizer)	6.03	1.81	6.03	1.00
TimesFM	Always de-shift	1.16	7.80	7.80	1.16
TimesFM	Gate + mixture	2.51	3.72	3.72	1.02
TimesFM	Full recipe	1.43	1.85	1.85	1.06
TimesFM	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	55% / 36% / 9%				
TS-ICL	Naive (no sanitizer)	5.24	1.76	5.24	1.00
TS-ICL	Always de-shift	1.14	6.98	6.98	1.14
TS-ICL	Gate + mixture	2.29	3.47	3.47	1.01
TS-ICL	Full recipe	1.53	1.79	1.79	1.05
TS-ICL	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	55% / 36% / 9%				
Moirai	Naive (no sanitizer)	5.77	2.29	5.77	1.00
Moirai	Always de-shift	1.10	6.15	6.15	1.10
Moirai	Gate + mixture	2.44	3.22	3.22	0.99
Moirai	Full recipe	1.48	2.32	2.32	1.03
Moirai	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	55% / 36% / 9%				
Chronos-2	Naive (no sanitizer)	5.44	1.92	5.44	1.00
Chronos-2	Always de-shift	1.14	7.93	7.93	1.14
Chronos-2	Gate + mixture	2.36	3.99	3.99	1.01
Chronos-2	Full recipe	1.45	1.97	1.97	1.04
Chronos-2	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	54% / 36% / 9%				
TimesFM-2.0	Naive (no sanitizer)	6.01	2.33	6.01	1.00
TimesFM-2.0	Always de-shift	1.14	8.87	8.87	1.14
TimesFM-2.0	Gate + mixture	2.51	4.53	4.53	1.02
TimesFM-2.0	Full recipe	1.49	2.41	2.41	1.05
TimesFM-2.0	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	55% / 36% / 9%				
Moirai-MoE	Naive (no sanitizer)	4.68	1.95	4.68	1.00
Moirai-MoE	Always de-shift	1.10	6.60	6.60	1.10
Moirai-MoE	Gate + mixture	2.09	3.47	3.47	1.00
Moirai-MoE	Full recipe	1.40	1.98	1.98	1.04
Moirai-MoE	Clean-context oracle	1.00	1.00	1.00	1.00
routing shares (keep/de-shift/mix)	55% / 36% / 9%				

Table 18: Two single-series vetoes on ETTm1's natural persistent shifts.

Model	Veto	Clean cost (%)	Fire (%)
TS-ICL	no veto	+28.5	38
TS-ICL	corroboration <1.0	+19.7	32
TS-ICL	step-shape >3.0	+7.8	8
TimesFM	no veto	+27.2	42
TimesFM	corroboration <1.0	+16.7	35
TimesFM	step-shape >3.0	+8.0	12