Beyond Isolation: Neighbor-Consistent Data Pruning for Multivariate Time Series Forecasting

Zi Yang[1] **Jona Otholt**[2] **Weixing Wang**[2] **Haojin Yang**[3,4] **Guillermo Gallego**[1,5]
[1]Technische Universität Berlin, Einstein Center Digital Future, and Robotics Institute Germany
[2]Hasso Plattner Institute, University of Potsdam [3]GreenBit.AI
[4]German University of Digital Science [5]Science of Intelligence Excellence Cluster
zi.yang@gmx.de, jona.otholt@hpi.de

Abstract

Data pruning for multivariate time-series forecasting (MTSF) is an emerging topic that poses unique challenges due to the intricate temporal dependencies within and between training samples. The dominant data pruning scores have been designed for i.i.d. image classification tasks and rate each data sample using its own training trajectory *in isolation*. However, this misses a structural property of MTSF training: samples (obtained by windowing time series) can easily overlap, especially if strides down to 1 time step are used, thus breaking the standard training i.i.d. assumption. Indeed, our key observation is that this overlap drives neighbors' loss trajectories into lockstep across training. The few samples whose trajectories deviate from this lockstep are inconsistent with their overlapping neighbors, typically because they contain contaminated points, and are the ones worth pruning away. Hence, we propose **TWIN** (*Trajectory-Wise Inconsistency with Neighbors*), a multi-scale Wasserstein-2 score and pruning algorithm that is, to our knowledge, the first to compare each sample's sorted training-loss trajectory against those of its sliding-window neighbors. Specifically, our design is motivated by two complementary bounds: the pairwise discrepancy between neighboring loss trajectories is at most $\mathcal{O}(\delta)$ when both samples are clean, where δ bounds the one-step change of the clean series (Theorem 1), but at least $\Omega(\Delta^2)$ once one carries a contamination spike of magnitude Δ (Theorem 2). Whenever $\Delta \gg \delta$, these scaling orders separate, exposing an asymptotic scaling-order gap that TWIN's rank-based rule reads. A synthetic audit of $16{,}000$ trained models empirically confirms this gap, with a median $44.23\times$ separation between clean and contaminated populations. We further evaluate TWIN's effectiveness on eight MTSF benchmarks, four horizons, and five seeds against nine baseline methods, showing that it shortens the average retraining time to about 58% of the unpruned baseline and reduces the average Relative MSE by up to 1.46% under IQR self-calibration, the best results among all evaluated methods. Moreover, the score transfers across backbones: computed from a short run of a lightweight linear model, it also applies to heavier backbones.

1 Introduction

By training a model on a small, effective subset of the original dataset, data pruning has become an established route to faster training and better generalization in image classification [12, 20, 22]. The dominant sample-level scoring functions, EL2N [12], forgetting events [15], Avg Loss [20], and TDDS [22], were developed under the i.i.d. image-classification assumption: each derives a scalar from a single sample's training trajectory *in isolation* from the rest of the dataset. When this family of scores is transplanted to multivariate time-series forecasting (MTSF) without modification, the per-sample-independent design ignores a structural signal that the sampling protocol produces.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

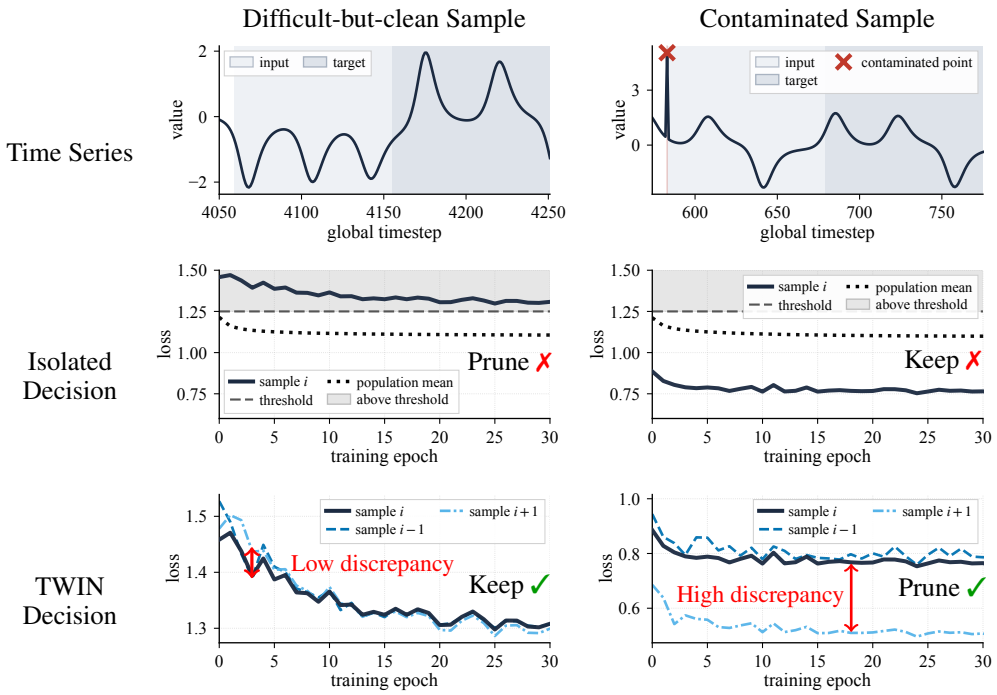


Figure 1: **An example showcasing the limitations of pruning decisions based only on per-sample loss trajectories.** The first row shows the time-series samples themselves: one is clean but follows a complex pattern, whereas the other follows a simple pattern but contains a contaminated point. Judging the two loss curves in isolation against a loss threshold results in wrongfully pruning the clean sample and keeping the contaminated one (middle row). However, by taking into account the loss trajectories of nearby samples (bottom row), we see that the second sample stands out from its neighbors, and the first one does not, allowing TWIN to make correct pruning decisions.

That signal is *sliding-window neighbor consistency*. Under the stride-1 sliding-window protocol used universally in MTSF training [9, 19], two consecutive input–target pairs $(\tilde{x}_i, \tilde{y}_i)$ and $(\tilde{x}_{i+1}, \tilde{y}_{i+1})$, of size $W+H$, share $W+H-1$ of their time steps (e.g., 191 out of 192 at the standard choice of $W=H=96$ time steps). With overlaps this large, the per-epoch losses of neighboring samples should move in lockstep across training, and decouple only when a sample is inconsistent with its neighbors, typically because it contains contaminated points. We therefore prune these inconsistent samples rather than redundant or merely difficult ones. Keeping such a sample in the training set hurts the forecaster, while removing it costs little, because most of its data still appears in the retained neighbors. Crucially, this signal is *relational*: it lives in the gap between adjacent loss trajectories, not in either trajectory alone. Any score that examines a single trajectory in isolation is therefore blind to it (Figure 1).

We turn this principle into **TWIN** (*Trajectory-Wise Inconsistency with Neighbors*). TWIN measures the Wasserstein-2 discrepancy between a sample's loss trajectory and those of its sliding-window neighbors at several offsets d, called dilations, and prunes the highest-scoring tail. Because a single contaminated time step lies in the windows of $W+H$ consecutive samples, it inflates their losses into a *contamination plateau*. Dilation $d=1$ resolves the plateau boundaries, while $d=W+H$ pairs every interior sample with a clean external neighbor. Together, they cover the edges and the interior of the plateau.

Two bounds motivate this design. Neighbor-consistent pairs admit an $\mathcal{O}(\delta)$ upper bound on the $\mathcal{W}_2$ distance between adjacent loss trajectories (δ: one-step temporal smoothness of the clean series; Theorem 1); contaminated–reference pairs admit an $\Omega(\Delta^2)$ lower bound (Δ: contamination magnitude; Theorem 2). When $\Delta \gg \delta$ the two populations separate by scaling order, which motivates what to compute (a discrepancy between neighbors' loss trajectories) and why a rank-based rule suffices (its ranking is invariant to every unknown constant). The bridge from the bounds to realized

population separation is established empirically: a synthetic audit of $16{,}000$ trained models plus 200 clean-only models (Section 6.2) verifies the upper bound pointwise on $961{,}600$ clean pairs, confirms the predicted $\Omega(\Delta^2)$ scaling, and records a median $44.23\times$ separation between the two populations. On eight MTSF benchmarks (Section 6.3), this separation translates into pruning gains.

In summary, our contributions are as follows:

- **A new pruning principle: scoring relative to neighbors.** To our knowledge, TWIN is the *first* sample-level pruner to exploit the overlap of the stride-1 sliding-window protocol. By scoring each MTSF sample against its neighbors rather than in isolation, TWIN captures the relational signal between adjacent loss trajectories that per-sample-independent scores cannot. We further improve performance by incorporating the Wasserstein distance and a multi-scale dilation set.
- **A design motivated by theoretical insights and validated empirically.** Theorems 1 and 2 show that the discrepancy between neighbors' loss trajectories is $\mathcal{O}(\delta)$ for clean pairs but $\Omega(\Delta^2)$ for contaminated ones. A synthetic audit verifies the upper bound, recovers the Δ^2 order of the lower one, and records a median $44.23\times$ empirical separation between the two populations—the gap that TWIN's rank rule exploits.
- **State-of-the-art accuracy and efficiency.** On eight MTSF benchmarks, four horizons, and five seeds, TWIN attains the best average Relative MSE and the shortest retraining time at every pruning rate, against nine baseline methods. Under interquartile-range (IQR) self-calibration, it reduces Relative MSE by 1.46% and shortens retraining time to about 58% of the unpruned baseline. It also outperforms strong baseline methods on PatchTST, DLinear, and MambaTS, and on univariate series. Its score transfers across backbones: computed with DLinear trained for only 20% of its epochs, it also applies to iTransformer and PatchTST.

2 Related Work

MTSF backbones and the case for sample-level pruning. MTSF has progressed through Transformer variants [9, 11, 19, 24], linear models [21], and selective state-space models [1] built on Mamba [5]. These forecasters are all trained on stride-1 sliding-window samples under a squared-error objective. Because the sample construction is identical across these backbones, TWIN can in principle attach to any forecaster exposing per-sample loss trajectories, with backbone generalization verified on iTransformer, PatchTST, DLinear, and MambaTS (Section 6.5).

Sample-level data pruning. Sample-level pruning [3, 12] aims to retain a compact, high-utility training subset. The dominant family, developed for image classification, derives a scalar from each sample's own training dynamics: EL2N from the norm of the per-sample error [12], forgetting score from how often a learned sample is later misclassified [15], TDDS from the temporal variance of per-epoch loss changes [22], and Avg Loss from the per-sample loss averaged over epochs, adapting the epoch-averaged logits of Yang et al. [20] to regression. Sliding-window adjacency lies outside their scope by construction: a per-sample-independent design treats samples i and $i+d$ as exchangeable and ignores the overlap between them, which is the important signal the MTSF protocol provides. TWIN utilizes this protocol to compare each sample with its neighboring samples and derive a score.

Time-series data selection and valuation. Most existing time-series methods judge data either from the raw series alone or against a reference: reference data, a validation set, or a reference model. Amplitude filters such as the median-absolute-deviation (MAD) filter and RobustTSF [2] flag deviations from a median or trend. Data valuation quantifies how much training data contribute to a learning task, at different granularities: Wang et al. [18] and TimeInf [23] use influence functions to value channels and individual time points, respectively, and TimeLAVA [8] values temporal segments by a wavelet-based Wasserstein discrepancy to reference data. FAF [14], which builds on RHO-Loss [10], selects samples online: in each mini-batch, it trains only on the samples whose loss most exceeds that of a reference model. We compare against its static pruning variant, FAF-S (Appendix E.5). None of these methods, to our knowledge, uses sliding-window neighbor consistency, and TWIN needs none of these references: each sample's reference is its own neighbors.

3 Problem Setup

Definition 1 (Forecasting sample pairs)**.** *Let $\{\tilde{z}_t\}_{t=1}^{T}$ be the observed D-channel time series of length T (the data we train on), with each $\tilde{z}_t \in \mathbb{R}^{1\times D}$. The look-back window and prediction horizon are W and H, respectively. Under the stride-1 sliding-window protocol, the i-th observed input–target pair is*

$$\tilde{x}_i := [\tilde{z}_i; \ldots; \tilde{z}_{i+W-1}] \in \mathbb{R}^{W\times D}, \qquad \tilde{y}_i := [\tilde{z}_{i+W}; \ldots; \tilde{z}_{i+W+H-1}] \in \mathbb{R}^{H\times D},$$

for $i \in \{1, \ldots, S\}$ with $S := T - W - H + 1$.

The tilde notation $\tilde{z}_t, \tilde{x}_i, \tilde{y}_i$ is reserved throughout the paper for observed quantities. Section 4 introduces a hypothesized clean (counterfactual) series $\{z_t\}$ (without tilde) and corresponding clean samples (x_i, y_i) defined by the same windowing protocol.

Definition 2 (Forecaster and per-sample loss)**.** *Let $f_\theta : \mathbb{R}^{W\times D} \to \mathbb{R}^{H\times D}$ be the forecaster with parameters θ. In Section 4 we use the mean squared error (MSE) loss, written without its constant factor $1/(HD)$: $\mathcal{L}(\theta, x, y) = \|f_\theta(x) - y\|_F^2$.*

Definition 3 (Training-dynamics matrix and empirical loss measure)**.** *Training proceeds for E epochs, yielding checkpoints $\theta_1, \ldots, \theta_E$. The training-dynamics matrix $\mathbf{M} \in \mathbb{R}^{S\times E}$ records the loss of every sample at every checkpoint, evaluated on the observed pairs:*

$$\mathbf{M}_{i,e} := \mathcal{L}(\theta_e, \tilde{x}_i, \tilde{y}_i), \qquad i \in \{1, \ldots, S\},\ e \in \{1, \ldots, E\}. \tag{1}$$

Sample i's loss trajectory is the vector $\mathbf{m}_i := (\mathbf{M}_{i,1}, \ldots, \mathbf{M}_{i,E})^\top \in \mathbb{R}^E$, and its empirical loss measure is the uniform discrete probability distribution on its E epoch losses, $\hat{\mu}_i := \frac{1}{E}\sum_{e=1}^{E} \boldsymbol{\delta}_{\mathbf{M}_{i,e}}$, where $\boldsymbol{\delta}_a$ denotes the Dirac mass at $a \in \mathbb{R}$. For any epoch-indexed quantity a_e, e.g., $a_e = g(\mathbf{M}_{i,e})$ for a function $g : \mathbb{R} \to \mathbb{R}$, we write $\overline{a_e} := \frac{1}{E}\sum_{e=1}^{E} a_e$ for the arithmetic mean over epochs.

Goal. Assign each sample a pruning score $\mathrm{Score}(i)$, remove the top-scored samples (those most inconsistent with their neighbors), and retrain the forecaster on the retained subset.

4 Theoretical Foundations

We introduce two bounds on adjacent loss trajectories, parameterized by the local temporal smoothness δ of the clean series and the magnitude Δ of any single-step contamination: an $\mathcal{O}(\delta)$ upper bound on neighbor-consistent pairs (Theorem 1) and an $\Omega(\Delta^2)$ lower bound on contaminated–reference pairs (Theorem 2). Whenever $\Delta \gg \delta$, these bounds establish an asymptotic scaling-order gap between the two populations. This gap motivates TWIN's rank-based pruning rule, and Section 6.2 measures it empirically.

4.1 Assumptions and the Δ-Spike Model

For every $d \geq 1$ and pair (i, j) with $|i - j| = d$, the *per-step Wasserstein discrepancy* is $\tau_d(i, j) := \mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_j)/d$, with $\hat{\mu}_i$ as in Definition 3. We make two regularity assumptions on the data and the trained model (full statements in Appendix A):

- **(A1) Local temporal smoothness.** There exists a constant $\delta > 0$ (the *one-step smoothness bound* of the clean series) such that $\|z_t - z_{t+1}\|_F \leq \delta$ for every $t \in \{1, \ldots, T-1\}$.
- **(A2)** C^1 **loss and Lipschitz prediction map along consecutive clean samples.** For each $i \in \{1, \ldots, S-1\}$, let Γ_i be the line segment joining the clean samples (x_i, y_i) and (x_{i+1}, y_{i+1}), and $\pi_x(\Gamma_i)$ its x-projection. For every epoch e, the loss $\mathcal{L}(\theta_e, \cdot, \cdot)$ is C^1 on a neighborhood of $\bigcup_i \Gamma_i$ with bounded loss–input and loss–target gradient suprema $K_x, K_y \in (0, \infty)$, and the prediction map f_{θ_e} is Lipschitz on each segment $\pi_x(\Gamma_i)$ with supremum $L_x \in (0, \infty)$.

Δ-spike model. The observed series $\{\tilde{z}_t\}$ differs from the clean (counterfactual) series $\{z_t\}$ only at a single contamination index $t^\star$: $\tilde{z}_{t^\star} = z_{t^\star} + \Delta\, u$ with $\|u\|_F = 1$ and $\Delta > 0$, and $\tilde{z}_t = z_t$ otherwise. Observed pairs $(\tilde{x}_i, \tilde{y}_i)$ are obtained by windowing $\{\tilde{z}_t\}$ via Definition 1.

Definition 4 (Contamination plateau)**.** *The contamination plateau $P := \{m \in \{1, \ldots, S\} : t^\star \in [m, m + W + H - 1]\}$ is a contiguous integer interval of cardinality at most $W+H$. A sample / pair / chain is plateau-external if all its members lie outside P, in which case observed and clean windows coincide for every member m ($\tilde{x}_m = x_m$ and $\tilde{y}_m = y_m$).*

4.2 The Two Bounds

Theorem 1 (Upper Bound)**.** *Under (A1)–(A2), for every plateau-external pair $(i, i+1)$ in the sense of Definition 4 (i.e., both samples i and $i+1$ are plateau-external),*

$$\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+1}) \;\le\; C_{\mathrm{adj}} \;:=\; (K_x\sqrt{W} + K_y\sqrt{H})\,\delta. \tag{2}$$

Proof sketch. A Mean Value Theorem on Γ_i under (A2) gives the per-epoch bound $|\mathbf{M}_{i,e} - \mathbf{M}_{i+1,e}| \le C_{\mathrm{adj}}$, which the 1-D optimal-transport closed form lifts to $\mathcal{W}_2$. Full proof in Appendix B.1 and chain extension to general d in Proposition 5. □

Theorem 2 (Lower Bound)**.** *Adopt the MSE loss and assume the spike falls in a target row of $\tilde{y}_{i^\star}$ for some contaminated sample $i^\star \in \{1, \dots, S\}$. Let $j \in \{1, \dots, S\}$ be a plateau-external clean reference at offset $d := |i^\star - j| \ge 1$ whose epoch-mean loss is bounded by some $\eta \ge 0$, i.e., $\overline{\mathbf{M}_{j,e}} \le \eta$, and define the clean-side fluctuation bound*

$$C_{\mathrm{clean}}(d) \;:=\; (L_x\sqrt{W} + \sqrt{H})\,d\,\delta. \tag{3}$$

Under (A1)–(A2),

$$\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \;\ge\; \big(\Delta - C_{\mathrm{clean}}(d) - \sqrt{\eta}\big)_+^2 \;-\; \eta. \tag{4}$$

The bound is non-trivial precisely when $\Delta > C_{\mathrm{clean}}(d) + 2\sqrt{\eta}$, in which case $\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) = \Omega(\Delta^2)$ as $\Delta \to \infty$ with d, δ, L_x, η held fixed.[1]

Proof sketch. For the prediction $\hat{y}_{i^\star}^{(e)} := f_{\theta_e}(x_{i^\star})$ at epoch e, a reverse triangle inequality gives $\|\hat{y}_{i^\star}^{(e)} - \tilde{y}_{i^\star}\|_F \ge \Delta - \|\hat{y}_{i^\star}^{(e)} - y_{i^\star}\|_F$, and two Jensen lifts (concave $\sqrt{\cdot}$, convex t^2), the hypothesis $\overline{\mathbf{M}_{j,e}} \le \eta$, and the 1-D quantile representation of $\mathcal{W}_2$ raise it to the d-independent form $\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \ge (\Delta - \sqrt{\eta_{\mathrm{c}}})_+^2 - \eta$, where η_{c} is the epoch-mean loss of $i^\star$ against its clean target. When the model does not fit the spike ($\eta_{\mathrm{c}} \ll \Delta^2$), the epoch-mean loss of $i^\star$ is therefore at least of order Δ^2. (A1)–(A2) chain telescoping through the consecutive clean inputs $x_{i^\star} \to \cdots \to x_j$ then bounds $\sqrt{\eta_{\mathrm{c}}} \le C_{\mathrm{clean}}(d) + \sqrt{\eta}$, which gives (4). Full proof in Appendix B.2. □

4.3 Implications for the Rank-Based Pruning Rule

Let τ_d^{cons} and τ_d^{div} denote τ_d realized on neighbor-consistent pairs (plateau-external chains) and neighbor-divergent pairs (contaminated–reference pairs at offset d), respectively. Theorems 1–2 separate the two populations: $\tau_d^{\mathrm{cons}} \le C_{\mathrm{adj}} = \mathcal{O}(\delta)$ (extended to general d via Proposition 5) and $\tau_d^{\mathrm{div}} \ge \Omega(\Delta^2/d)$. When $\Delta \gg \delta$ the two scaling orders are asymptotically separated. TWIN's rank-based pruning rule exploits this gap by aggregating τ_d over a fixed multi-scale dilation set $\{d_k\}_{k=1}^K$ and removing the top quantile of $\sum_k \tau_{d_k}$. None of the bound constants $\{K_x, K_y, L_x, \delta, \eta, \Delta\}$ enter the rule: they appear only in the analytical bounds, through C_{adj} and $C_{\mathrm{clean}}(d)$, never in the empirical τ_d computed from $\mathbf{M}$ (Corollary 7). We measure this gap by the *empirical separation ratio* $\hat{r}(d) := \mathrm{med}(\tau_d^{\mathrm{div}})/\mathrm{med}(\tau_d^{\mathrm{cons}})$, which contains none of the bound constants and reaches a median of $44.23\times$ across all 16,000 trained models of the synthetic audit (Section 6.2).

5 Pruning Method: TWIN

Section 4 establishes an asymptotic scaling-order gap between neighbor-consistent and contaminated populations in τ_d. We turn this gap into a concrete pruning algorithm (Figure 2) with two components: a multi-scale $\mathcal{W}_2$ score of how far each sample's loss trajectory lies from its neighbors' (Section 5.1), and a rank-based pruning rule that removes the highest scores (Section 5.2).

5.1 Multi-Scale Wasserstein Discrepancy Score

Per-dilation discrepancy. To quantify the discrepancy between a sample's training-loss trajectory and those of its sliding-window neighbors, for each training sample i and a set of dilations (neighbor

[1] The bound depends on the contamination only through $\|\tilde{y}_{i^\star} - y_{i^\star}\|_F$, so it extends, with Δ replaced by this norm, to multi-step patterns such as block drops, amplitude bursts, and noise blocks that leave the input window of $i^\star$ clean (Appendix C).

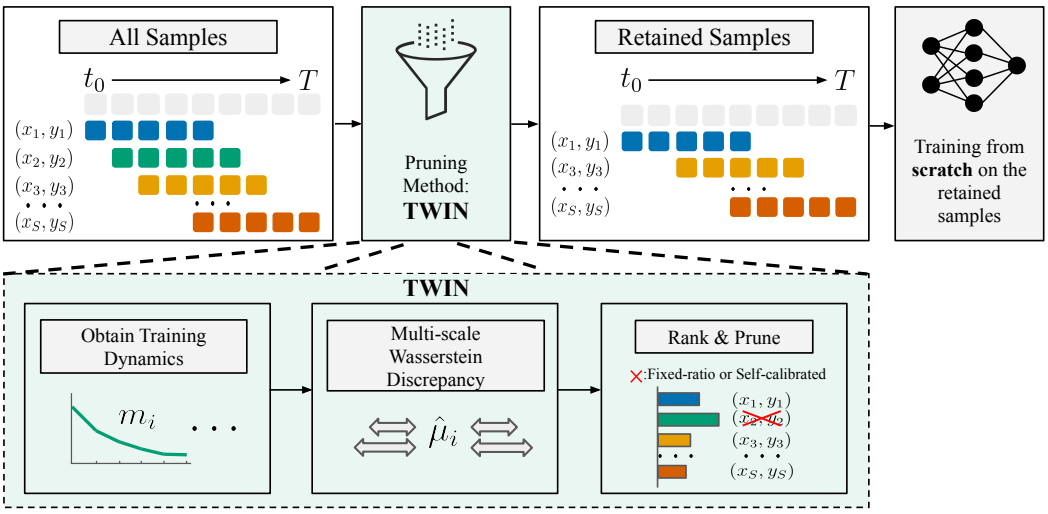


Figure 2: **Overview of the TWIN pipeline.** Starting from the full training set, we train a forecaster and record per-sample loss trajectories across epochs. For each sample, TWIN computes a multi-scale Wasserstein discrepancy between its loss trajectory and those of its sliding-window neighbors, capturing inconsistency in training dynamics. Samples are then ranked by their discrepancy score and pruned using a fixed ratio or self-calibrated threshold. The model is retrained from scratch on the retained subset.

offsets) $\mathcal{D} = \{d_k\}_{k=1}^{K}$, we use the per-step Wasserstein discrepancy (Section 4.1) at each dilation d_k:

$$\tau_k(i) \;:=\; \frac{1}{d_k}\,\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+d_k}) \;=\; \frac{1}{d_k\sqrt{E}}\,\|\mathrm{sort}(\mathbf{m}_i) - \mathrm{sort}(\mathbf{m}_{i+d_k})\|_2\,, \tag{5}$$

where the second equality is the closed-form 1-D optimal transport between empirical distributions with E equally-weighted atoms. In the notation of Section 4.1, $\tau_k(i) = \tau_{d_k}(i, i{+}d_k)$, and "multi-scale" refers to using several dilations. Sorting absorbs benign epoch-wise phase drift (mini-batch order, warm-up, initialization) and isolates the structural discrepancy that Theorems 1–2 bound.

Why multi-scale: the contamination plateau (of $W{+}H$ size). A single contaminated time step affects every sample whose input or target window covers it, producing a plateau of $W{+}H$ inflated rows in $\mathbf{M}$. Within this plateau, adjacent samples cover the same contaminated step, so their loss trajectories are inflated together and could remain mutually consistent. Thus $\tau_1(i)$ stays small, and $d{=}1$ alone can miss the plateau interior. Two dilations are therefore needed: $d{=}1$ detects the plateau *edges*, and $d{=}W{+}H$ pairs every sample in the plateau *interior* with a clean neighbor outside it.

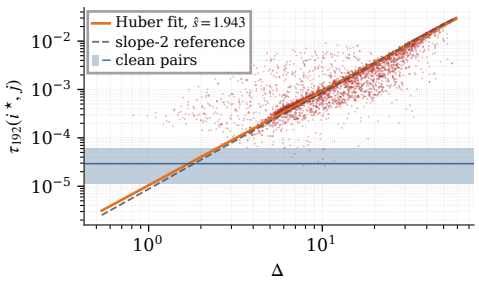
Aggregation. We sum the per-dilation discrepancies with uniform weights:

$$\mathrm{Score}(i) \;:=\; \sum_{k=1}^{K} \tau_k(i). \tag{6}$$

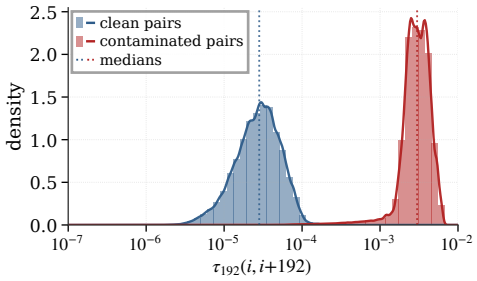
Uniform weights are the simplest choice that keeps the gap: all terms are non-negative, so one term from a contaminated–reference pair lifts $\mathrm{Score}(i)$ by its full value, while on clean chains the $1/d_k$ factor keeps every term at most C_{adj}.

5.2 Rank-Based Pruning Rule

Because none of the bound constants enter $\mathrm{Score}(i)$ (Section 4.3), a rule that prunes by rank needs no estimate of them. We expose two such variants. **Fixed-ratio:** given a target pruning rate $\mathrm{PR} \in (0, 1)$, $\mathcal{I}_{\mathrm{prune}}$ contains the top PR samples ranked by TWIN score, for budget-controlled comparison with baselines. **IQR (Tukey's fences [16]):** $\mathcal{I}_{\mathrm{prune}} = \{i : \mathrm{Score}(i) > Q_3 + 1.5\,\mathrm{IQR}\}$, removing only the upper tail with a data-chosen pruning rate. The retained set is $\mathcal{I}_{\mathrm{keep}} = \{1, \ldots, S\} \setminus \mathcal{I}_{\mathrm{prune}}$ (full pseudocode in Algorithm 1).



(a) **Illustration of Theorem 2**, $\Omega(\Delta^2)$: Log–log scatter plot of $\tau_{W+H}(i^\star, j) = \mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j)/(W{+}H)$ vs. $\Delta := \|\tilde{y}_{i^\star} - y_{i^\star}\|_F$ for contaminated samples $i^\star$ and $j = i^\star{+}W{+}H$ ($W{+}H{=}192$) on group DLinear/G. The Huber-regressed solid line (slope $\hat{s} = 1.943$) agrees well with the dashed slope-2 line of Theorem 2; the blue band spans clean pairs (10th–90th pct.).



(b) **Empirical separation of τ_{W+H} populations.** Histograms and kernel density estimates of τ_{W+H} ($d{=}W{+}H{=}192$, log x-axis) for clean pairs (blue) and contaminated–reference pairs (red) at perturbation level 10 of 20, pooled over 100 seeds. Dotted lines mark the medians, whose ratio is $\sim$106$\times$.

Figure 3: **Synthetic audit on the DLinear backbone with amplitude-burst perturbations (G).** Results for all eight (backbone, perturbation) groups (Huber slopes in $[1.94,\ 2.01]$, per-group median $\hat{r}(d) > 25$) are in Appendices D.3 and D.4.

6 Experiments

6.1 Experimental Setup

Setup. We evaluate on eight standard MTSF benchmarks at four horizons $H \in \{96, 192, 336, 720\}$: **Weather** [19] with 21 indicators, **Electricity** [19] with 321 clients, the four **ETT** datasets [24] ETTh1/2 and ETTm1/2, **Solar** [7] with 137 plants, and **Exchange** [7] with 8 countries. Table 1 covers all eight. Unless stated otherwise, the remaining experiments use the six benchmarks other than Solar and Exchange. The primary backbone is **iTransformer** [9] with look-back $W = 96$, implemented on top of Time-Series-Library.[2] The evaluation is extended to PatchTST, DLinear, and MambaTS (Section 6.5) and to univariate series (Appendix H.1). Training-pipeline, hardware, and baseline-score-formula details are in Appendix E.1.

Nine baselines. Four are data-side and do not consult the trained model: Random, matched-size stride thinning (Stride), a MAD amplitude filter, and RobustTSF [2]. Five are model-based. TimeInf [23] is a time-series data-valuation method, and FAF-S is a static pruning version of FAF [14] (Appendix E.5). EL2N [12], TDDS [22], and Avg Loss [20] originate in i.i.d. classification. All are evaluated under fixed-ratio (Pruning Rate, PR $\in \{10\%, 20\%, 30\%\}$, identical retained-set size) and IQR (Tukey's fences applied to each method's own score distribution) protocols, averaged over five seeds. IQR is omitted for Random and Stride. Appendix E.4 adds TimeLAVA.

6.2 Empirical Verification of the Two Bounds

We verify the two scaling-order predictions of Theorems 1–2 on a synthetic grid with known ground-truth contamination indices. This grid is based on a 7-channel sinusoidal series, yielding 16,000 trained models across two backbones (MLP, DLinear [21]), 100 seeds, and four perturbation families (20 levels each): (B) Bernoulli spikes, (D) block drops, (G) length-20 amplitude bursts, and (N) Gaussian noise blocks. See Appendix D for details.

Headline audit results. Three findings verify the upper bound pointwise, confirm the lower bound's predicted scaling, and quantify the empirical separation that the rank rule exploits. Figure 3 illustrates the latter two on a representative group.

- **(i) Upper bound holds:** across 961,600 neighbor-consistent pairs, $\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+1}) < C_{\text{adj}}$ on every pair (violation rate 0% in Appendix D.2).

[2] https://github.com/thuml/Time-Series-Library

- **(ii)** $\Omega(\Delta^2)$ **order confirmed:** per-group Huber slopes of $\log \tau_{W+H}$ on $\log \Delta$ average $\mathbf{1.99}$ (median 2.00), all in $[1.94, 2.01]$, matching the order 2 of Theorem 2 (Appendix D.3).
- **(iii) Empirical population separation:** the empirical separation ratio $\hat{r}(d) := \mathrm{med}(\tau_d^{\mathrm{div}})/\mathrm{med}(\tau_d^{\mathrm{cons}})$, which contains *none* of the bound constants, has median $\mathbf{44.23}$ across all $16{,}000$ trained models (Appendix D.4).

6.3 Main Pruning Results

Protocols and metrics. We report *Relative MSE* (percentage change vs. unpruned baseline, so negative values mean improvement) and *Retraining Time* (the time to retrain the forecaster on the pruned subset until it reaches the best unpruned test loss, as a fraction of the unpruned-baseline training time, so values below 100% mean retraining acceleration). Retraining Time excludes the one-off cost of computing the pruning score, which TWIN shares with every loss-dynamics-based baseline (EL2N, TDDS, Avg Loss). We adopt this decomposed cost-accounting convention [20], in contrast to the *end-to-end* convention of Selection via Proxy [3] (rationale in Appendix E.2, end-to-end cost with scoring included in Appendix H.5). Implementation details are described in Appendix E.1. All metrics are averaged over eight datasets, four horizons, and five seeds.

Table 1: Comparison across eight MTSF benchmarks, four horizons, and five seeds. **Left**: Avg Relative MSE (%) vs. unpruned baseline (negative = better than unpruned). **Right**: Retraining Time (% of unpruned-baseline training time, lower = faster). Columns: target pruning rate PR $= \{10\%, 20\%, 30\%\}$ and the self-calibrated pruning rate IQR. Statistical significance of the MSE differences is reported in Appendix F.

Method	**Avg Relative MSE (%) at Pruning Rate**				**Retraining Time (%) at Pruning Rate**			
	PR=10%	PR=20%	PR=30%	IQR	PR=10%	PR=20%	PR=30%	IQR
Random	+0.16	+0.32	+0.68	—	90.28	92.69	96.17	—
Stride	+0.15	+0.34	+0.61	—	91.02	95.15	95.64	—
RobustTSF [2]	+0.05	+0.40	+1.11	−0.05	91.02	89.74	93.98	90.96
MAD	+0.17	+0.47	+1.80	−0.17	90.90	86.66	84.89	85.81
EL2N [12]	−0.84	−0.65	−0.25	−0.83	70.19	69.17	72.98	70.84
Avg Loss [20]	−1.13	−1.06	−0.45	−1.10	64.43	62.33	66.02	64.97
TDDS [22]	−1.14	−0.52	+0.25	−0.96	61.77	70.50	80.24	64.79
TimeInf [23]	−0.32	−0.09	+0.48	−0.24	77.96	78.97	82.38	81.22
FAF-S [14]	−0.74	+0.47	+1.95	−0.62	66.78	76.35	87.50	69.34
TWIN (Ours)	**−1.41**	**−1.20**	**−0.69**	**−1.46**	**60.53**	**60.38**	**64.24**	**58.41**

Table 1 shows TWIN attaining the largest average MSE reduction and the shortest retraining time in every column. Appendix G shows that the gains concentrate on rare test samples and that the removed samples are enriched in contamination. Holm-corrected Wilcoxon tests on Relative MSE confirm that the gap is statistically significant against all nine baselines ($p \leq 2.0 \times 10^{-6}$, Appendix F).

6.4 Ablation Studies

6.4.1 Distance Function

Table 2 replaces $\mathcal{W}_2$ in Eq. (5) with L_1, L_2, and Dynamic Time Warping (DTW) [13] on the unsorted trajectories. $\mathcal{W}_2$ attains the largest average MSE reduction in every column, and the difference is statistically significant against all three alternatives. The L_2-vs-$\mathcal{W}_2$ gap (the two differ only in whether sorting is applied) isolates the contribution of sorting.

6.4.2 Multi-Scale Dilation Set

Table 3 compares four nested dilation subsets, with $\mathcal{D}_3$ as TWIN's default. The ranking $\mathcal{D}_3 \succ \mathcal{D}_4 \succ \mathcal{D}_2 \succ \mathcal{D}_1$ holds on average across the four pruning protocols, and every multi-scale variant beats single-scale $\mathcal{D}_1$ on average. The finer $\mathcal{D}_4$ brings no further gain (Holm $p = 0.45$ vs. $\mathcal{D}_3$).

6.5 Backbone Independence and Score Transfer

Because the score is computed from one forecaster's loss trajectories, we test in two ways whether the selection depends on that forecaster. First, we recompute $\mathbf{M}$ and the score on three further backbones:

Table 2: Distance-function ablation: Avg Relative MSE (%), means over six benchmarks, four horizons, and five seeds. Rightmost column: Holm–Bonferroni-corrected two-sided Wilcoxon signed-rank p vs. Wasserstein (TWIN).

Method	**Avg Relative MSE (%) at Pruning Rate**				**Holm p vs TWIN**
	PR=10%	PR=20%	PR=30%	IQR	
L_1	−1.25	−0.97	−0.74	−1.24	3.9×10^{-5}
L_2	−1.29	−1.06	−0.66	−1.31	2.1×10^{-5}
DTW [13]	−1.26	−1.08	−0.60	−1.33	3.9×10^{-5}
Wasserstein (TWIN)	**−1.41**	**−1.14**	**−0.78**	**−1.56**	— (ref)

Table 3: Dilation-set ablation: Avg Relative MSE (%), means over six benchmarks, four horizons, and five seeds. Rightmost column: Holm–Bonferroni-corrected two-sided Wilcoxon signed-rank p vs. $\mathcal{D}_3$ (TWIN's default).

Method	**Avg Relative MSE (%) at Pruning Rate**				**Holm p vs $\mathcal{D}_3$**
	PR=10%	PR=20%	PR=30%	IQR	
$\mathcal{D}_1 = \{1\}$ (single-step)	−1.22	−1.06	−0.66	−1.32	6.1×10^{-6}
$\mathcal{D}_2 = \{1, W{+}H\}$ (two extremes)	−1.36	−1.12	−0.74	−1.32	7.2×10^{-3}
$\mathcal{D}_3 = \{1, (W{+}H)/4, (W{+}H)/2, W{+}H\}$ (**TWIN**)	**−1.41**	**−1.14**	−0.78	**−1.56**	— (ref)
$\mathcal{D}_4 = \{1, (W{+}H)/8, (W{+}H)/4, (W{+}H)/2, W{+}H\}$	−1.31	−1.12	**−0.81**	−1.46	0.45

PatchTST, DLinear, and MambaTS (original protocol, $W{=}720$). TWIN attains the best average Relative MSE in every column on each of them, against Random and the two baselines that perform best on average in Table 1, Avg Loss and EL2N (Appendix H.2). Second, we score with a different backbone from the one we train: the TWIN score is computed with the cheapest backbone (DLinear), and iTransformer and PatchTST are then trained on the resulting retained set (Table 4). The scoring run can also be cut short: on the native backbone, scores computed from the first 20%, 40%, 60%, or 100% of the epochs give almost the same MSE (Appendix H.3). Both savings together, DLinear with 20% of its epochs, keep the Relative MSE within 0.13 points of the native results. The selection therefore depends little on the scoring backbone or the length of the scoring run, so the cost of scoring need not grow with the size of the target model (Appendix H.4).

Table 4: Cross-backbone score transfer: Avg Relative MSE (%), means over six benchmarks, four horizons, and five seeds. The TWIN score is computed with the target backbone itself (Native) and with DLinear trained for 100% and 20% of its epochs. The target backbone is then trained on each retained set.

Scoring source	**Target Backbone: iTransformer**				**Target Backbone: PatchTST**			
	PR=10%	PR=20%	PR=30%	IQR	PR=10%	PR=20%	PR=30%	IQR
Native	−1.41	−1.14	**−0.78**	**−1.56**	**−1.29**	−1.33	−1.31	**−1.16**
DLinear, 100% epochs	−1.42	−1.15	−0.76	−1.45	−1.23	−1.29	**−1.36**	−1.12
DLinear, 20% epochs	**−1.44**	**−1.18**	−0.75	−1.44	−1.16	**−1.34**	−1.20	−1.13

7 Conclusion

We addressed sample-level data pruning for MTSF by identifying the relational signal that pruning methods designed for image classification miss: under stride-1 sliding windows, adjacent loss trajecto ries tend to move in lockstep across training and decouple only when a sample is inconsistent with its neighbors, typically because of contamination. We formalized this principle with two bounds (Theorems 1–2): the discrepancy is $\mathcal{O}(\delta)$ between clean neighbors and $\Omega(\Delta^2)$ between a contaminated sample and a clean reference, so for $\Delta \gg \delta$ the lower bound far exceeds the upper bound. A synthetic audit verifies the upper bound pointwise, confirms the predicted $\Omega(\Delta^2)$ scaling, and records a median $44.23\times$ separation between the two populations. To our knowledge, TWIN is the *first* realization of this principle: it uses a multi-scale $\mathcal{W}_2$ score on sorted neighbor trajectories and a rank-based pruning rule that needs none of the unknown bound constants. Across eight MTSF benchmarks, four horizons, and five seeds, TWIN attains the best average Relative MSE in every

evaluated pruning protocol (a reduction of up to 1.46% relative to the unpruned baseline, achieved under IQR self-calibration), while shortening the average retraining time under IQR to about 58% of the unpruned baseline.

Limitations and future work. TWIN has two limitations. **(i) Pre-memorization regime:** the lower bound (Theorem 2) requires that the model has not fully fit the contaminated targets (the pre- or partial-memorization regime of Remark 2). Standard MTSF training practices such as dropout and early stopping work against full memorization, and in the scoring-window sweep, scores computed from the first 20% of training already give pruning results comparable to full-run scores. **(ii) Offline scope:** TWIN is a one-shot offline pruning method that consumes the full training-dynamics matrix $\mathbf{M}$ after a single training pass. Adapting it to streaming or continual settings, where samples accumulate over time and the multi-scale Wasserstein score must be updated incrementally, is a natural future direction. An extended discussion of these limitations is in Appendix I.

Acknowledgments and Disclosure of Funding

The project on which this report is based was funded by the Federal Ministry of Research, Technology and Space under the funding code "KI-Servicezentrum Berlin-Brandenburg" 16IS22092. Responsibility for the content of this publication remains with the authors.

References

[1] Xiuding Cai, Xueyao Wang, Yaoyao Zhu, and Yu Yao. MambaTS: Improved selective state space models for long-term time series forecasting. *Pattern Recognition*, 180:114536, 2026.

[2] Hao Cheng, Qingsong Wen, Yang Liu, and Liang Sun. RobustTSF: Towards theory and design of robust time series forecasting with anomalies. In *International Conference on Learning Representations (ICLR)*, 2024.

[3] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[4] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[5] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

[6] Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. *Inequalities*. Cambridge University Press, Cambridge, 2nd edition, 1952.

[7] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 95–104, 2018.

[8] Wenqin Liu, Weizhi Quan, Aoqi Zuo, Erdun Gao, Vu Nguyen, Dino Sejdinovic, Howard Bondell, and Mingming Gong. TimeLAVA: Learning-agnostic valuation for time series data. In *International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, pages 78066–78099, 2026.

[9] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024.

[10] Sören Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649, 2022.

[11] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

[12] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 20596–20607, 2021.

[13] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[14] Ege Onur Taga, Halil Alperen Gozeten, Kutay Tire, Rahul Dalvi, Reinhard Heckel, and Samet Oymak. Filter, augment, forecast: Online data selection for robust time series forecasting. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pages 1081–1089, 2026.

[15] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[16] John Wilder Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.

[17] Cédric Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2003.

[18] Muyao Wang, Zeke Xie, Bo Chen, Hongwei Liu, and James Kwok. Channel matters: Estimating channel influence for multivariate time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 12750–12775, 2025.

[19] 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 (NeurIPS)*, volume 34, pages 22419–22430, 2021.

[20] Zi Yang, Haojin Yang, Soumajit Majumder, Jorge Cardoso, and Guillermo Gallego. Data pruning can do more: A comprehensive data pruning approach for object re-identification. *Transactions on Machine Learning Research*, 2024.

[21] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pages 11121–11128, 2023.

[22] Xin Zhang, Jiawei Du, Yunsong Li, Weiying Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (TDDS) for enhanced dataset pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26223–26232, 2024.

[23] Yizi Zhang, Jingyan Shen, Xiaoxue Xiong, and Yongchan Kwon. TimeInf: Time series data contribution via influence functions. In *International Conference on Learning Representations (ICLR)*, 2025.

[24] 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 *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 11106–11115, 2021.

Appendix Contents

A Detailed Statements for Section 4: Notation, Assumptions, and Spike Model **13**

A.1 Notation Conventions 13

A.2 Assumption (A1): Local Temporal Smoothness 14

A.3 Assumption (A2): C^1 Loss and Lipschitz Prediction Map along Consecutive Clean Samples 14

A.4 The Δ-Spike Contamination Model 15

A.5 Contamination Plateau and Plateau-External Samples / Pairs / Chains 15

B Full Proofs for Section 4 **16**

B.1 Proof of Theorem 1 (Upper Bound) 16

B.2 Proof of Theorem 2 (Lower Bound) 16

B.3 Chain Extension and Strong-Separation Sufficient Condition 18

B.4 Rank Cuts and Support Separation 19

C Pattern-Agnosticism of Theorem 2: Step-by-Step Verification **20**

C.1 Generalized Contamination Data and Plateau 21

C.2 Pattern-Agnostic Form of Theorem 2 21

C.3 Plateau Structure for General $\mathcal{T}$ and the Chain Extension of Proposition 5 21

C.4 Caveats 21

D Synthetic Audit Details **22**

D.1 Protocol 22

D.2 Q1: Clean-Side Upper Bound 23

D.3 Q2: $\Omega(\Delta^2)$ Slope Verification 23

D.4 Q3: Empirical Separation of τ_d Populations 25

E Experimental Setup and Baselines **26**

E.1 Training and Scoring Pipeline 26

E.2 Reporting Convention 27

E.3 RobustTSF Baseline 27

E.4 Learning-Agnostic and Influence-Based Valuation: TimeLAVA and TimeInf 28

E.5 FAF-S: A Static Pruning Version of FAF 29

F Statistical Significance Analysis **30**

F.1 Test Protocol 30

F.2 Relative MSE Results on iTransformer (Corresponding to Table 1) 31

G Analysis: Where the Gains Come From **31**

G.1 Rarity-Stratified Evaluation 31

G.2 What TWIN Removes and What Remains 32

G.3 Clustered Contamination Stress Test 34

H Extended Evaluation: The Univariate Setting, Backbones, Score Transfer, and Cost **34**
H.1 Univariate Forecasting 34
H.2 Additional Backbones: PatchTST, DLinear, and MambaTS 35
H.3 Scoring-Window Robustness 35
H.4 Cross-Backbone Score Transfer 36
H.5 End-to-End Cost Accounting 36

I Limitations and Future Work (Extended) **37**
I.1 Limitations 37
I.2 Future Work 37

NeurIPS Paper Checklist **38**

A Detailed Statements for Section 4: Notation, Assumptions, and Spike Model

This appendix fixes the notation (Appendix A.1) and provides the precise definitions of assumptions **(A1)** and **(A2)**, the line segments $\Gamma_i, \pi_x(\Gamma_i)$ on which (A2) operates, the three Lipschitz/gradient constants K_x, K_y, L_x, and the Δ-spike contamination model and its plateau, all of which are stated in compact form in Section 4 and referenced throughout the proofs (Appendix B). Appendix A.3 also splits (A2) into the parts (A2) (i) and (A2) (ii) used by the proofs of Theorems 1 and 2 and relates K_x to L_x (Remark 1).

A.1 Notation Conventions

Notation. $\mathcal{L}$ denotes the per-sample loss, which for MSE is $\mathcal{L}(\theta, x, y) = \|f_\theta(x) - y\|_F^2$. The Frobenius inner product on matrix spaces is $\langle A, B\rangle_F := \mathrm{tr}(A^\top B)$, used to identify Fréchet derivatives of scalar-valued functions on matrix spaces.

Frobenius norm. On a single row, $\mathbb{R}^{1\times D}$, the Frobenius norm is the Euclidean norm, and for any matrix $\|A\|_F^2 = \sum_k \|A_{k,:}\|_F^2$. Both proofs use this row-wise decomposition.

The Wasserstein-2 distance. For any two probability measures μ, ν on $\mathbb{R}$ with finite second moments, the *Wasserstein-2 distance* is

$$\mathcal{W}_2(\mu,\nu) \;:=\; \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{R}\times\mathbb{R}} (x-y)^2 \, d\pi(x,y)\right)^{1/2}, \tag{7}$$

where $\Pi(\mu,\nu)$ is the set of couplings of μ and ν. On $\mathbb{R}$ this admits the equivalent *1-D quantile representation* [17]

$$\mathcal{W}_2^2(\mu,\nu) \;=\; \int_0^1 \left(F_\mu^{-1}(q) - F_\nu^{-1}(q)\right)^2 dq, \tag{8}$$

where F_μ^{-1}, F_ν^{-1} are the (left-continuous) quantile functions. For empirical measures $\hat\mu_i = \frac{1}{E}\sum_{e=1}^E \boldsymbol{\delta}_{\mathbf{M}_{i,e}}$ and $\hat\mu_j = \frac{1}{E}\sum_{e=1}^E \boldsymbol{\delta}_{\mathbf{M}_{j,e}}$, Eq. (8) reduces to the closed form

$$\mathcal{W}_2^2(\hat\mu_i, \hat\mu_j) \;=\; \frac{1}{E}\sum_{e=1}^{E} (r_{i,e} - r_{j,e})^2, \tag{9}$$

where $\mathbf{r}_i := \mathrm{sort}(\mathbf{m}_i) \in \mathbb{R}^E$ is the ascending sort of the loss trajectory of sample i, with components $r_{i,1} \le \cdots \le r_{i,E}$. We refer to (9) as the *sorted-vector* form.

A.2 Assumption (A1): Local Temporal Smoothness

(A1) Local temporal smoothness. There exists a constant $\delta > 0$, called the *one-step smoothness bound* of the clean series, such that

$$\|z_t - z_{t+1}\|_F \;\le\; \delta \qquad \text{for every } t \in \{1, \dots, T-1\}, \tag{10}$$

where $\{z_t\}_{t=1}^T$ is the hypothesized clean (counterfactual) series introduced in Section 4 (see Definition 1) and $\|\cdot\|_F$ is the Frobenius norm on $\mathbb{R}^{1\times D}$.

Scope. The constant δ is a fixed but unknown property of the data. (A1) concerns the clean series only: on the observed series it can fail at the contamination index $t^\star$ (Appendix A.4), where $\|\tilde{z}_{t^\star} - \tilde{z}_{t^\star\pm1}\|_F$ may exceed δ by an arbitrary amount.

A.3 Assumption (A2): C^1 Loss and Lipschitz Prediction Map along Consecutive Clean Samples

(A2) is a regularity assumption on the trained-model family $\{\mathcal{L}(\theta_e, \cdot, \cdot), f_{\theta_e}\}_{e=1}^E$ across all E epoch checkpoints. We state it in three blocks: (a) the clean-pair line segments Γ_i and their input-projections $\pi_x(\Gamma_i)$, on which (A2) operates; (b) the C^1 regularity and the loss-gradient constants K_x, K_y; (c) the prediction-map Lipschitz constant L_x. The proofs of Theorems 1 and 2 use blocks (b) and (c), respectively. We refer to them as **(A2) (i)** and **(A2) (ii)**.

(a) Clean-pair line segments and their input projections. For every $i \in \{1, \dots, S-1\}$, the *clean-pair line segment* in input–target space is the convex combination of the two clean samples (x_i, y_i) and (x_{i+1}, y_{i+1}):

$$\Gamma_i \;:=\; \big\{(\lambda x_i + (1-\lambda)x_{i+1},\; \lambda y_i + (1-\lambda)y_{i+1}) : \lambda \in [0,1]\big\} \;\subset\; \mathbb{R}^{W\times D} \times \mathbb{R}^{H\times D}. \tag{11}$$

The clean samples (x_i, y_i) are obtained from the clean series $\{z_t\}$ by the windowing protocol of Definition 1. In particular, $x_i = [z_i; \dots; z_{i+W-1}] \in \mathbb{R}^{W\times D}$ and $y_i = [z_{i+W}; \dots; z_{i+W+H-1}] \in \mathbb{R}^{H\times D}$. The *x-projection* of Γ_i keeps only the input component:

$$\pi_x(\Gamma_i) \;:=\; \big\{\lambda x_i + (1-\lambda)x_{i+1} : \lambda \in [0,1]\big\} \;\subset\; \mathbb{R}^{W\times D}. \tag{12}$$

The union $\bigcup_{i=1}^{S-1} \Gamma_i \subset \mathbb{R}^{W\times D} \times \mathbb{R}^{H\times D}$ is a polygonal path through the S clean samples, and $\bigcup_{i=1}^{S-1} \pi_x(\Gamma_i) \subset \mathbb{R}^{W\times D}$ is its input-side image. (A2) constrains $\mathcal{L}$ and f_{θ_e} only near these paths: the C^1 condition on an open neighborhood of them, and the Lipschitz condition on the input segments themselves.

(b) C^1 regularity and loss-gradient bounds K_x, K_y. For every epoch $e \in \{1, \dots, E\}$, the loss $\mathcal{L}(\theta_e, \cdot, \cdot) : \mathbb{R}^{W\times D} \times \mathbb{R}^{H\times D} \to \mathbb{R}_{\ge 0}$ is C^1 on an open neighborhood of $\bigcup_{i=1}^{S-1} \Gamma_i$ (where C^1 means continuously differentiable with continuous Fréchet derivatives in both arguments). The two constants

$$K_x \;:=\; \sup_{e\in\{1,\dots,E\},\ (x,y)\in\bigcup_i \Gamma_i} \|\nabla_x \mathcal{L}(\theta_e, x, y)\|_F \qquad (\textit{loss–input gradient bound}), \tag{13}$$

$$K_y \;:=\; \sup_{e\in\{1,\dots,E\},\ (x,y)\in\bigcup_i \Gamma_i} \|\nabla_y \mathcal{L}(\theta_e, x, y)\|_F \qquad (\textit{loss–target gradient bound}), \tag{14}$$

are assumed finite and strictly positive, $K_x, K_y \in (0, \infty)$. The Fréchet derivatives $\nabla_x \mathcal{L}$, $\nabla_y \mathcal{L}$ are identified with elements of $\mathbb{R}^{W\times D}, \mathbb{R}^{H\times D}$ respectively via the Frobenius inner product $\langle \cdot, \cdot \rangle_F$ (Appendix A.1).

(c) Prediction-map Lipschitz constant L_x. For every $e \in \{1, \dots, E\}$ and every $i \in \{1, \dots, S-1\}$, the restriction of the prediction map $f_{\theta_e} : \mathbb{R}^{W\times D} \to \mathbb{R}^{H\times D}$ to the input segment $\pi_x(\Gamma_i)$ is Lipschitz in the Frobenius norm:

$$\mathrm{Lip}\big(f_{\theta_e}\big|_{\pi_x(\Gamma_i)}\big) \;:=\; \inf\big\{L \ge 0 : \|f_{\theta_e}(x) - f_{\theta_e}(x')\|_F \le L\, \|x - x'\|_F \text{ for all } x, x' \in \pi_x(\Gamma_i)\big\}. \tag{15}$$

The supremum

$$L_x \;:=\; \sup_{e\in\{1,\dots,E\},\ i\in\{1,\dots,S-1\}} \mathrm{Lip}\big(f_{\theta_e}\big|_{\pi_x(\Gamma_i)}\big) \tag{16}$$

is assumed finite and strictly positive, $L_x \in (0, \infty)$.

When (A2) holds. (A2) is a local condition: it concerns only the finitely many segments Γ_i and $\pi_x(\Gamma_i)$, which form a bounded closed set, and finitely many epochs. The derivatives of a C^1 map are continuous and therefore bounded on such a set, so K_x, K_y, and L_x are finite for every forecaster that is a C^1 map of its input with finite weights, e.g., a linear model such as DLinear or a network with smooth activations such as GELU. The synthetic audit estimates K_x and K_y for its two backbones with the Lipschitz probes of Appendix D.1.

Remark 1 (Two distinct constants)**.** *For the MSE loss, $\nabla_x \mathcal{L} = 2\,[Df_{\theta_e}(x)]^*(f_{\theta_e}(x) - y)$ and $\nabla_y \mathcal{L} = -2\,(f_{\theta_e}(x) - y)$, where $Df_{\theta_e}(x)$ is the Fréchet derivative of f_{θ_e} at x and $[\,\cdot\,]^*$ its adjoint under the Frobenius inner product. Hence $K_y = 2\sup \|f_{\theta_e}(x) - y\|_F$ and $K_x \le 2\sup \|Df_{\theta_e}(x)\|_{\mathrm{op}} \cdot \sup \|f_{\theta_e}(x) - y\|_F$, the suprema taken over the epochs and the segments Γ_i: both loss-gradient constants shrink with the training residual, whereas L_x does not. (A2) (ii) controls only the derivative of f_{θ_e} along each segment $\pi_x(\Gamma_i)$, not $\|Df_{\theta_e}(x)\|_{\mathrm{op}}$, so this relation does not bound K_x by L_x, and the proofs of Theorems 1–2 use K_x and L_x as independent constants.*

A.4 The Δ-Spike Contamination Model

Definition 5 (Δ-Spike Contamination Model)**.** *The observed series $\{\tilde{z}_t\}_{t=1}^T$ agrees with the clean series $\{z_t\}_{t=1}^T$ everywhere except at a single contamination index $t^\star \in \{1, \dots, T\}$, where*

$$\tilde{z}_{t^\star} \;=\; z_{t^\star} + \Delta\, u, \qquad \tilde{z}_t = z_t \text{ for every } t \neq t^\star, \tag{17}$$

with $u \in \mathbb{R}^{1\times D}$, $\|u\|_F = 1$ (a unit-Frobenius row encoding the spike direction) and $\Delta > 0$ (the contamination magnitude). This is exactly the spike model of Section 4.1. The observed pairs $(\tilde{x}_i, \tilde{y}_i)$ are obtained by applying the windowing protocol of Definition 1 to $\{\tilde{z}_t\}$. The contamination plateau P (Definition 4, stated in full as Eq. (18) *in Appendix A.5) collects the sample indices whose window covers $t^\star$. For plateau-external i, $(\tilde{x}_i, \tilde{y}_i) = (x_i, y_i)$. On the plateau, exactly one row of $\tilde{y}_i - y_i$ (target-spike sub-model: $t^\star \in [i+W, i+W+H-1]$) or of $\tilde{x}_i - x_i$ (input-window sub-model: $t^\star \in [i, i+W-1]$) equals Δu, contributing Δ in Frobenius norm. The two sub-models are disjoint because the input window $[i, i+W-1]$ and the target window $[i+W, i+W+H-1]$ of sample i are disjoint.*

A.5 Contamination Plateau and Plateau-External Samples / Pairs / Chains

Definition 6 (Contamination plateau, full statement)**.** *The contamination plateau associated with the contamination index $t^\star$ is the set of sample indices whose input or target window covers $t^\star$:*

$$\begin{aligned} P \;&:=\; \big\{m \in \{1,\dots,S\} : t^\star \in [m,\, m+W+H-1]\big\} \\ &=\; \big\{m \in \{1,\dots,S\} : t^\star - W - H + 1 \le m \le t^\star\big\}, \end{aligned} \tag{18}$$

and after boundary truncation (since $m \in \{1, \dots, S\}$),

$$P \;=\; [\max(1, t^\star - W - H + 1),\; \min(S, t^\star)] \cap \mathbb{Z}. \tag{19}$$

Cardinality and contiguity. P is a contiguous integer interval of cardinality at most $W+H$, with equality precisely when $W+H \le t^\star \le S$. For boundary contamination indices $t^\star \le W + H - 1$ or $t^\star \ge S + 1$, $|P|$ is strictly smaller than $W+H$ due to the truncation in (19).

Plateau-external samples, pairs, and chains. A sample $m \in \{1, \dots, S\}$ is *plateau-external* if $m \notin P$, i.e., neither its input window $[m, m+W-1]$ nor its target window $[m+W, m+W+H-1]$ contains $t^\star$, in which case

$$\tilde{x}_m = x_m, \qquad \tilde{y}_m = y_m, \tag{20}$$

so $\mathbf{M}_{m,e} = \mathcal{L}(\theta_e, x_m, y_m)$ for every epoch e. A pair or chain is plateau-external if all of its members are.

Generalization to non-singleton contamination sets. The above is the single-spike plateau. Appendix C.1 treats the generalized plateau for any contamination set $\mathcal{T} \subseteq \{1, \dots, T\}$, which may have multiple connected components.

B Full Proofs for Section 4

This appendix gives the full proof of Theorem 1, the proof of Theorem 2 (restated as Lemma 3) and its d-independent residual form (Lemma 4), the chain extension and the strong-separation sufficient condition (Proposition 5), and the rank-cut results (Proposition 6 and Corollary 7), which show that the rank rule's guarantee needs no estimate of the bound constants.

B.1 Proof of Theorem 1 (Upper Bound)

Proof. We prove $\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+1}) \le C_{\mathrm{adj}}$ in three steps. Since both samples are plateau-external, $\mathbf{M}_{i,e} = \mathcal{L}(\theta_e, x_i, y_i)$ and $\mathbf{M}_{i+1,e} = \mathcal{L}(\theta_e, x_{i+1}, y_{i+1})$ for every e by (20), so the proof works on the clean segment Γ_i throughout.

Step 1 (Joint input–target distances). By Definition 1 (applied to the clean series), the k-th row of $x_i - x_{i+1}$ equals $z_{i+k-1} - z_{i+k}$ for $k \in \{1, \dots, W\}$. Therefore

$$\|x_i - x_{i+1}\|_F^2 \;=\; \sum_{k=1}^{W} \|z_{i+k-1} - z_{i+k}\|_F^2 \;\overset{\text{(A1)}}{\le}\; W\delta^2. \tag{21}$$

Taking square roots, $\|x_i - x_{i+1}\|_F \le \sqrt{W}\,\delta$. Analogously $\|y_i - y_{i+1}\|_F \le \sqrt{H}\,\delta$.

Step 2 (Per-epoch loss difference via the joint Mean Value Theorem). Fix $e \in \{1, \dots, E\}$ and define

$$g(\lambda) \;:=\; \mathcal{L}\big(\theta_e,\; \lambda x_i + (1-\lambda)x_{i+1},\; \lambda y_i + (1-\lambda)y_{i+1}\big), \qquad \lambda \in [0,1]. \tag{22}$$

By **(A2) (i)**, $\mathcal{L}(\theta_e, \cdot, \cdot)$ is C^1 on a neighborhood of Γ_i, so $g \in C^1([0,1])$. By the Mean Value Theorem there exists $\lambda^* \in (0,1)$ with $\mathbf{M}_{i,e} - \mathbf{M}_{i+1,e} = g(1) - g(0) = g'(\lambda^*)$. Set $\xi^* := (\lambda^* x_i + (1-\lambda^*)x_{i+1},\; \lambda^* y_i + (1-\lambda^*)y_{i+1}) \in \Gamma_i$. The chain rule, identifying Fréchet derivatives via the Frobenius inner product (Appendix A.1), gives

$$g'(\lambda^*) \;=\; \langle \nabla_x \mathcal{L}(\theta_e, \xi^*), x_i - x_{i+1} \rangle_F + \langle \nabla_y \mathcal{L}(\theta_e, \xi^*), y_i - y_{i+1} \rangle_F . \tag{23}$$

The triangle inequality, Cauchy–Schwarz term-by-term, **(A2) (i)**, and Step 1 yield

$$\begin{aligned} |\mathbf{M}_{i,e} - \mathbf{M}_{i+1,e}| \;&\le\; \|\nabla_x \mathcal{L}\|_F \cdot \|x_i - x_{i+1}\|_F + \|\nabla_y \mathcal{L}\|_F \cdot \|y_i - y_{i+1}\|_F \\ &\le\; K_x\sqrt{W}\,\delta + K_y\sqrt{H}\,\delta \;=\; C_{\mathrm{adj}}. \end{aligned}$$

This holds for every e, so

$$(\mathbf{M}_{i,e} - \mathbf{M}_{i+1,e})^2 \;\le\; C_{\mathrm{adj}}^2, \qquad \forall\, e \in \{1, \dots, E\}. \tag{24}$$

Step 3 (Lift to $\mathcal{W}_2$ via the 1-D closed form and rearrangement). By the sorted-vector form (9), $\mathcal{W}_2^2(\hat{\mu}_i, \hat{\mu}_{i+1}) = \frac{1}{E}\sum_{e=1}^{E}(r_{i,e} - r_{i+1,e})^2$. By the Hardy–Littlewood–Pólya rearrangement inequality [6], pairing two sequences in the same sorted order minimizes $\sum_e (a_e - b_{\pi(e)})^2$ over all permutations π, so

$$\sum_{e=1}^{E}(r_{i,e} - r_{i+1,e})^2 \;\le\; \sum_{e=1}^{E}(\mathbf{M}_{i,e} - \mathbf{M}_{i+1,e})^2 \;\overset{(24)}{\le}\; E \cdot C_{\mathrm{adj}}^2, \tag{25}$$

and taking square roots gives $\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+1}) \le C_{\mathrm{adj}}$, which is (2). □

B.2 Proof of Theorem 2 (Lower Bound)

We restate Theorem 2 as a self-contained lemma for clarity, then prove it. The equivalent form $c_0(d)\Delta^2 - \eta$ with $c_0(d) := (1 - (C_{\mathrm{clean}}(d) + \sqrt{\eta})/\Delta)_+^2$ is used in Proposition 5, which introduces the per-step lower-bound value $L(d) := (c_0(d)\Delta^2 - \eta)/d$.

Lemma 3 (Lower Bound, restatement of Theorem 2)**.** *Under assumptions (A1)–(A2), the MSE loss, and the target-spike sub-model of Definition 5 (so the spike falls in a target row of $\tilde{y}_{i^\star}$, equivalently $t^\star \in [i^\star + W,\; i^\star + W + H - 1]$), let j be a plateau-external clean reference at offset $d := |i^\star - j| \ge 1$ with $\overline{\mathbf{M}_{j,e}} \le \eta$, and define*

$$C_{\mathrm{clean}}(d) \;:=\; (L_x\sqrt{W} + \sqrt{H})\, d\, \delta, \qquad c_0(d) \;:=\; \Big(1 - \tfrac{C_{\mathrm{clean}}(d) + \sqrt{\eta}}{\Delta}\Big)_+^2. \tag{26}$$

Then

$$\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \;\geq\; \big(\Delta - C_{\text{clean}}(d) - \sqrt{\eta}\big)_+^2 - \eta \;=\; c_0(d)\,\Delta^2 - \eta, \tag{27}$$

where the two right-hand sides are equal by direct expansion. Here $c_0(d) > 0 \Leftrightarrow \Delta > C_{\text{clean}}(d) + \sqrt{\eta}$*, and the bound is non-trivial (its right-hand side is positive) precisely when* $\Delta > C_{\text{clean}}(d) + 2\sqrt{\eta}$*.*

Proof. For a fixed epoch e, write $\hat{y}_i := f_{\theta_e}(x_i)$ for the prediction at the *clean* input x_i. Step 5 averages over epochs. Since the spike sits in the target window of $i^\star$, $\tilde{x}_{i^\star} = x_{i^\star}$, and plateau-externality of j gives $\tilde{x}_j = x_j$ and $\tilde{y}_j = y_j$, so

$$\mathbf{M}_{i^\star,e} \;=\; \|\hat{y}_{i^\star} - \tilde{y}_{i^\star}\|_F^2\,, \qquad \mathbf{M}_{j,e} \;=\; \|\hat{y}_j - y_j\|_F^2\,. \tag{28}$$

We write the proof for $i^\star < j = i^\star + d$. For $j < i^\star$, Steps 1–2 run along the chain from j to $i^\star$ instead, and nothing else changes.

Step 1 (Window-pair distances at offset d, on the clean series). The four windows $x_{i^\star}, y_{i^\star}, x_j, y_j$ are all built from the clean series $\{z_t\}$: $x_{i^\star}$ because the spike lies in the target window of $i^\star$, x_j and y_j by plateau-externality of j, and $y_{i^\star}$ as the clean target window (we explicitly distinguish $y_{i^\star}$ from $\tilde{y}_{i^\star}$). Iterating **(A1)** on the clean series and applying the triangle inequality,

$$\|z_t - z_{t+d}\|_F \;\leq\; \sum_{k=0}^{d-1} \|z_{t+k} - z_{t+k+1}\|_F \;\leq\; d\delta \tag{29}$$

for every $t \in \{1, \ldots, T-d\}$. Applying the row-wise decomposition of Appendix A.1 to $A = x_{i^\star} - x_j$ (whose k-th row is $z_{i^\star+k-1} - z_{j+k-1}$, a d-step temporal gap on the clean series, hence $\leq d\delta$ in Frobenius norm) and $A = y_{i^\star} - y_j$ yields

$$\|x_{i^\star} - x_j\|_F \;\leq\; \sqrt{W} \cdot d\delta, \qquad \|y_{i^\star} - y_j\|_F \;\leq\; \sqrt{H} \cdot d\delta. \tag{30}$$

Step 2 (Lipschitz transfer along the segment chain). For each $k \in \{i^\star, i^\star + 1, \ldots, j-1\}$, both x_k and x_{k+1} are endpoints of $\pi_x(\Gamma_k)$, on which f_{θ_e} is L_x-Lipschitz by **(A2) (ii)**, and Step 1 (with $d = 1$ on this single link) gives $\|x_k - x_{k+1}\|_F \leq \sqrt{W}\delta$. Telescoping the prediction differences along the chain and applying the triangle inequality,

$$\|\hat{y}_{i^\star} - \hat{y}_j\|_F \;\leq\; \sum_{k=i^\star}^{j-1} \|f_{\theta_e}(x_k) - f_{\theta_e}(x_{k+1})\|_F \;\leq\; \sum_{k=i^\star}^{j-1} L_x\, \|x_k - x_{k+1}\|_F \;\leq\; L_x\sqrt{W} \cdot d\delta. \tag{31}$$

The telescoping uses Lipschitzness only on the segments themselves, which is all that **(A2) (ii)** provides.

Step 3 (Reverse triangle on the contaminated target). Since $\tilde{y}_{i^\star} - y_j = (\tilde{y}_{i^\star} - y_{i^\star}) + (y_{i^\star} - y_j)$ with $\|\tilde{y}_{i^\star} - y_{i^\star}\|_F = \Delta$ (Definition 5), the reverse triangle inequality and (30) give

$$\|\tilde{y}_{i^\star} - y_j\|_F \;\geq\; \Delta - \|y_{i^\star} - y_j\|_F \;\geq\; \Delta - \sqrt{H} \cdot d\delta. \tag{32}$$

Step 4 (Triangle chain on the prediction error of $i^\star$). The algebraic identity $y_j - \tilde{y}_{i^\star} = (y_j - \hat{y}_j) + (\hat{y}_j - \hat{y}_{i^\star}) + (\hat{y}_{i^\star} - \tilde{y}_{i^\star})$, combined with the triangle inequality and rearrangement, gives

$$\|\hat{y}_{i^\star} - \tilde{y}_{i^\star}\|_F \;\geq\; \|\tilde{y}_{i^\star} - y_j\|_F - \|\hat{y}_j - \hat{y}_{i^\star}\|_F - \|y_j - \hat{y}_j\|_F\,. \tag{33}$$

Substituting (31) (upper bound on the second term) and (32) (lower bound on the first term),

$$\|\hat{y}_{i^\star} - \tilde{y}_{i^\star}\|_F \;\geq\; \Delta - C_{\text{clean}}(d) - \|y_j - \hat{y}_j\|_F\,, \tag{34}$$

with $C_{\text{clean}}(d) = (L_x\sqrt{W} + \sqrt{H})\, d\delta$ as in (26).

Step 5 (Epoch averaging via two Jensen applications). Take the epoch average $\overline{\cdot}$ of (34). Since $\sqrt{\cdot}$ is concave on $\mathbb{R}_{\geq 0}$, Jensen's inequality and the hypothesis $\overline{\mathbf{M}_{j,e}} \leq \eta$ give

$$\overline{\|y_j - \hat{y}_j\|_F} \;\leq\; \sqrt{\overline{\|y_j - \hat{y}_j\|_F^2}} \;=\; \sqrt{\overline{\mathbf{M}_{j,e}}} \;\leq\; \sqrt{\eta}, \tag{35}$$

hence

$$\overline{\|\hat{y}_{i^\star} - \tilde{y}_{i^\star}\|_F} \;\geq\; \Delta - C_{\text{clean}}(d) - \sqrt{\eta}. \tag{36}$$

If $\Delta > C_{\text{clean}}(d) + \sqrt{\eta}$, i.e., $c_0(d) > 0$, both sides of (36) are positive, and squaring and Jensen's inequality for the convex map $t \mapsto t^2$ give

$$\overline{\mathbf{M}_{i^\star,e}} \;=\; \overline{\|\hat{y}_{i^\star} - \tilde{y}_{i^\star}\|_F^2} \;\geq\; \left(\overline{\|\hat{y}_{i^\star} - \tilde{y}_{i^\star}\|_F}\right)^2 \;\geq\; \left(\Delta - C_{\text{clean}}(d) - \sqrt{\eta}\right)^2 \;=\; c_0(d)\Delta^2, \quad (37)$$

where the last equality is (26). Otherwise, $c_0(d) = 0$ and $\overline{\mathbf{M}_{i^\star,e}} \geq c_0(d)\Delta^2$ holds trivially. In both cases,

$$\overline{\mathbf{M}_{i^\star,e}} - \overline{\mathbf{M}_{j,e}} \;\geq\; c_0(d)\Delta^2 - \eta. \quad (38)$$

Step 6 (From mean shift to $\mathcal{W}_2$). By the quantile representation (8), Jensen's inequality for $t \mapsto t^2$ on the probability space $([0,1], dq)$, and $\int_0^1 F_a^{-1}(q)\, dq = \overline{\mathbf{M}_{a,e}}$ for the quantile function F_a^{-1} of $\hat{\mu}_a$,

$$\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \;\geq\; \left|\overline{\mathbf{M}_{i^\star,e}} - \overline{\mathbf{M}_{j,e}}\right| \;\geq\; \overline{\mathbf{M}_{i^\star,e}} - \overline{\mathbf{M}_{j,e}}. \quad (39)$$

Combining (39) with (38) gives (27). The asymptotic $\mathcal{W}_2 = \Omega(\Delta^2)$ as $\Delta \to \infty$ (with d, δ, L_x, η held fixed) follows because $C_{\text{clean}}(d) + \sqrt{\eta}$ is then fixed. □

Lemma 4 (Residual form of the lower bound)**.** *Under the MSE loss and the target-spike sub-model of Definition 5, let j be any sample with $\overline{\mathbf{M}_{j,e}} \leq \eta$, and let $\eta_{\text{c}} := \overline{\|f_{\theta_e}(x_{i^\star}) - y_{i^\star}\|_F^2}$ be the epoch-mean loss of $i^\star$ against its clean target. Then, without (A1)–(A2),*

$$\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \;\geq\; \left(\Delta - \sqrt{\eta_{\text{c}}}\right)_+^2 - \eta. \quad (40)$$

If, in addition, j is a plateau-external reference at offset $d := |i^\star - j|$ and (A1)–(A2) hold, then $\sqrt{\eta_{\text{c}}} \leq C_{\text{clean}}(d) + \sqrt{\eta}$, so (40) *implies* (27)*.*

Proof. Fix an epoch e and write $\hat{y}_{i^\star}^{(e)} := f_{\theta_e}(x_{i^\star})$, so that $\mathbf{M}_{i^\star,e} = \left\|\hat{y}_{i^\star}^{(e)} - \tilde{y}_{i^\star}\right\|_F^2$ as in the proof of Lemma 3. Since $\hat{y}_{i^\star}^{(e)} - \tilde{y}_{i^\star} = (\hat{y}_{i^\star}^{(e)} - y_{i^\star}) - (\tilde{y}_{i^\star} - y_{i^\star})$ and $\|\tilde{y}_{i^\star} - y_{i^\star}\|_F = \Delta$, the reverse triangle inequality gives $\left\|\hat{y}_{i^\star}^{(e)} - \tilde{y}_{i^\star}\right\|_F \geq \Delta - \left\|\hat{y}_{i^\star}^{(e)} - y_{i^\star}\right\|_F$. Averaging over epochs and applying Jensen's inequality to the concave $\sqrt{\cdot}$, $\overline{\left\|\hat{y}_{i^\star}^{(e)} - y_{i^\star}\right\|_F} \leq \sqrt{\eta_{\text{c}}}$, hence $\overline{\left\|\hat{y}_{i^\star}^{(e)} - \tilde{y}_{i^\star}\right\|_F} \geq \Delta - \sqrt{\eta_{\text{c}}}$. If $\Delta > \sqrt{\eta_{\text{c}}}$, squaring both positive sides and applying Jensen's inequality to the convex $t \mapsto t^2$ give $\overline{\mathbf{M}_{i^\star,e}} \geq (\Delta - \sqrt{\eta_{\text{c}}})^2$. Otherwise, $(\Delta - \sqrt{\eta_{\text{c}}})_+^2 = 0 \leq \overline{\mathbf{M}_{i^\star,e}}$. In both cases $\overline{\mathbf{M}_{i^\star,e}} - \overline{\mathbf{M}_{j,e}} \geq (\Delta - \sqrt{\eta_{\text{c}}})_+^2 - \eta$. As in the proof of Lemma 3, the quantile representation of $\mathcal{W}_2$, which involves only the two empirical measures, gives $\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \geq \overline{\mathbf{M}_{i^\star,e}} - \overline{\mathbf{M}_{j,e}}$, which is (40).

For the second claim, plateau-externality of j gives $\mathbf{M}_{j,e} = \left\|\hat{y}_j^{(e)} - y_j\right\|_F^2$ with $\hat{y}_j^{(e)} := f_{\theta_e}(x_j)$. As in the proof of Lemma 3, (A1)–(A2) give $\|y_{i^\star} - y_j\|_F \leq \sqrt{H}\, d\delta$ and $\left\|\hat{y}_{i^\star}^{(e)} - \hat{y}_j^{(e)}\right\|_F \leq L_x\sqrt{W}\, d\delta$. The triangle inequality on $\hat{y}_{i^\star}^{(e)} - y_{i^\star} = (\hat{y}_{i^\star}^{(e)} - \hat{y}_j^{(e)}) + (\hat{y}_j^{(e)} - y_j) + (y_j - y_{i^\star})$ then gives $\left\|\hat{y}_{i^\star}^{(e)} - y_{i^\star}\right\|_F \leq C_{\text{clean}}(d) + \left\|\hat{y}_j^{(e)} - y_j\right\|_F$. Squaring, averaging over epochs, and using $\overline{\left\|\hat{y}_j^{(e)} - y_j\right\|_F} \leq \sqrt{\overline{\mathbf{M}_{j,e}}} \leq \sqrt{\eta}$ and $\overline{\left\|\hat{y}_j^{(e)} - y_j\right\|_F^2} = \overline{\mathbf{M}_{j,e}} \leq \eta$ give $\eta_{\text{c}} \leq (C_{\text{clean}}(d) + \sqrt{\eta})^2$. Since $t \mapsto (\Delta - t)_+^2$ is non-increasing on $[0, \infty)$, (40) implies (27). □

Remark 2 (Scope of Theorem 2)**.** *The bound applies in two training regimes relevant to TWIN: pre-memorization (clean samples have converged to a small loss floor but contaminated samples have not) and partial memorization (epoch-mean loss of $i^\star$ retains an $\Omega(\Delta^2)$ residual). The residual form* (40) *makes the regime explicit: full memorization is the limit in which the model fits the contaminated target, so $\eta_{\text{c}} \to \Delta^2$ and the bound becomes vacuous.*

B.3 Chain Extension and Strong-Separation Sufficient Condition

We now prove (i) the chain extension $\tau_d(i, i+d) \leq C_{\text{adj}}$ on every plateau-external chain (referenced after Theorem 1 in the main text), and (ii) the sufficient condition for pointwise support separation used in Corollary 7.

Proposition 5 (Chain extension and strong separation)**.** *Under (A1)–(A2) and the Δ-spike model: (a) For every $i \in \{1, \dots, S-d\}$ such that the chain $\{i, i+1, \dots, i+d\}$ lies entirely outside the plateau, $\tau_d(i, i+d) \le C_{\mathrm{adj}}$. (b) For every contaminated $i^\star \in P$ such that $t^\star \in [i^\star + W,\ i^\star + W + H - 1]$ (i.e., the spike falls in the target window of $i^\star$, the target-spike sub-model of Definition 5), and every plateau-external clean reference j at offset $d := |i^\star - j|$ with $\overline{\mathbf{M}_{j,e}} \le \eta$, $\tau_d(i^\star, j) \ge L(d) := (c_0(d)\Delta^2 - \eta)/d$ where $c_0(d)$ is from* (26)*.*

Together with $\tau_d \ge 0$, the contaminated population lies in $[\max(0, L(d)), \infty)$, while the clean chain population lies in $[0, C_{\mathrm{adj}}]$. Since $C_{\mathrm{adj}} > 0$ (as $K_x, K_y, \delta > 0$), the (strict) strong-separation condition $L(d) > C_{\mathrm{adj}}$ is equivalent to

$$\Delta \ > \ C_{\mathrm{clean}}(d) + \sqrt{\eta} + \sqrt{dC_{\mathrm{adj}} + \eta}, \tag{41}$$

and under (41)*, any threshold $\gamma \in (C_{\mathrm{adj}}, L(d))$ classifies the two populations exactly: every realized $\tau_d^{\mathrm{cons}} \le C_{\mathrm{adj}} < \gamma < L(d) \le \tau_d^{\mathrm{div}}$.*

Proof. Part (a): Clean-pair upper bound. Every member of the chain is plateau-external, so each adjacent pair $(i+k, i+k+1)$, $k \in \{0, \dots, d-1\}$, is plateau-external and Theorem 1 gives $\mathcal{W}_2(\hat{\mu}_{i+k}, \hat{\mu}_{i+k+1}) \le C_{\mathrm{adj}}$. By the triangle inequality for $\mathcal{W}_2$, $\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+d}) \le \sum_{k=0}^{d-1} \mathcal{W}_2(\hat{\mu}_{i+k}, \hat{\mu}_{i+k+1}) \le d\, C_{\mathrm{adj}}$, and dividing by d gives $\tau_d(i, i+d) \le C_{\mathrm{adj}}$.

Part (b): Contaminated-pair lower bound. Direct from Lemma 3: $\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \ge c_0(d)\Delta^2 - \eta$, then divide by d to get $\tau_d(i^\star, j) \ge L(d)$.

Strict strong-separation equivalence. By Parts (a) and (b) and $\tau_d \ge 0$, the two populations lie in $[0, C_{\mathrm{adj}}]$ and $[\max(0, L(d)), \infty)$. Since $d \ge 1$, $L(d) > C_{\mathrm{adj}}$ is equivalent to $c_0(d)\Delta^2 > dC_{\mathrm{adj}} + \eta$. The right-hand side is positive ($C_{\mathrm{adj}} > 0$), so this forces $c_0(d) > 0$, i.e., $\Delta - C_{\mathrm{clean}}(d) - \sqrt{\eta} > 0$ and $c_0(d)\Delta^2 = (\Delta - C_{\mathrm{clean}}(d) - \sqrt{\eta})^2$. Taking square roots of the two positive sides gives (41), and squaring gives the converse. □

B.4 Rank Cuts and Support Separation

A *labeled finite set of scores* is a finite index set $I = I_X \sqcup I_Y$ with $|I_X| = n_c \ge 1$ and $|I_Y| = n_n \ge 1$, together with a value map $h : I \to \mathbb{R}$. In Corollary 7, I_X and I_Y index the divergent and the consistent pairs, and h is the realized τ_d. A *rank cut of size k*, $k \in \{0, \dots, n_c + n_n\}$, is a subset $C \subseteq I$ with $|C| = k$ such that $h(\omega) \ge h(\omega')$ for every $\omega \in C$ and every $\omega' \in I \setminus C$, i.e., a set of k largest values (unique up to ties at the boundary). A fixed-ratio rule removes a rank cut of prescribed size.

Proposition 6 (Rank cuts and support separation)**.** *Let (I_X, I_Y, h) be a labeled finite set of scores. Then: (i) (Thresholds are rank cuts.) For every $\gamma \in \mathbb{R}$, the sets $\{\omega \in I : h(\omega) > \gamma\}$ and $\{\omega \in I : h(\omega) \ge \gamma\}$ are rank cuts. (ii) (Support-separation characterization.) The following are equivalent: (a) $\min_{a \in I_X} h(a) > \max_{b \in I_Y} h(b)$; (b) every rank cut C of size $k \le n_c$ satisfies $C \subseteq I_X$, and every rank cut C of size $k \ge n_c$ satisfies $C \supseteq I_X$; (c) I_X is the unique rank cut of size n_c.*

Proof. **(i) Thresholds are rank cuts.** Let $C := \{\omega \in I : h(\omega) > \gamma\}$. For every $\omega \in C$ and $\omega' \in I \setminus C$, $h(\omega) > \gamma \ge h(\omega')$, so C is a rank cut of size $|C|$. For $C := \{\omega \in I : h(\omega) \ge \gamma\}$ the same argument gives $h(\omega) \ge \gamma > h(\omega')$.

(ii) Support-separation characterization.

(a)⇒(b). Let C be a rank cut of size $k \le n_c$ and suppose C contains some $b \in I_Y$. Then $|C \cap I_X| \le k - 1 < n_c$, so some $a \in I_X$ lies outside C, and the rank-cut property gives $h(b) \ge h(a)$, contradicting (a). Hence $C \subseteq I_X$. Now let C be a rank cut of size $k \ge n_c$ and suppose some $a \in I_X$ lies outside C. Then $|C \cap I_X| \le n_c - 1 < k$, so C contains some $b \in I_Y$, and the rank-cut property gives $h(b) \ge h(a)$, again contradicting (a). Hence $C \supseteq I_X$.

(b)⇒(c). A rank cut of size n_c exists: order I by non-increasing h, breaking ties arbitrarily, and take the first n_c elements. By (b) with $k = n_c$, every rank cut of size n_c is both contained in and contains I_X, hence equals I_X.

(c)⇒(a). Suppose (a) fails, and pick $a^\star \in \arg\min_{a \in I_X} h(a)$ and $b^\star \in \arg\max_{b \in I_Y} h(b)$, so $h(a^\star) \le h(b^\star)$. Since I_X is a rank cut and $b^\star \notin I_X$, also $h(a^\star) \ge h(b^\star)$, hence $h(a^\star) = h(b^\star)$.

Then $C' := (I_X \setminus \{a^\star\}) \cup \{b^\star\}$ is a rank cut of size n_c: every $a \in I_X \setminus \{a^\star\}$ satisfies $h(a) \geq h(a^\star) = h(b^\star) \geq h(b)$ for all $b \in I_Y$, and $b^\star$ satisfies $h(b^\star) = h(a^\star)$ and $h(b^\star) \geq h(b)$ for all $b \in I_Y$. Since $C' \neq I_X$, this contradicts the uniqueness in (c). □

Proposition 6 is why the rank rule's guarantee needs no estimate of the bound constants: both pruning variants of Section 5.2 remove rank cuts, the fixed-ratio rule one of prescribed size and the IQR rule $\{i : \mathrm{Score}(i) > Q_3 + 1.5\,\mathrm{IQR}\}$, and whenever the two populations are support-separated, every rank cut either consists only of elements of I_X or contains all of I_X, whatever its size and wherever the gap lies. Corollary 7 instantiates the proposition with the two bounds.

Corollary 7 (Rank-based pruning under the two bounds)**.** *Fix a realization of the training trajectory $\{\theta_e\}_{e=1}^{E}$ and an offset $d \geq 1$. Let*

$$T_d^{\mathrm{div}} := \{\tau_d^{\mathrm{div},a}\}_{a=1}^{n_c}, \qquad T_d^{\mathrm{cons}} := \{\tau_d^{\mathrm{cons},b}\}_{b=1}^{n_n} \tag{42}$$

be the realized values of τ_d on contaminated–reference pairs (Lemma 3) and on plateau-external chains (Proposition 5), respectively, and consider the rank cuts of their union in the sense of Proposition 6, with the divergent pairs as I_X and the consistent pairs as I_Y. Then: (i) (Partial separation.) The union admits a rank cut that contains every divergent pair and exactly $N_{\mathrm{FP}} := \left|\{b : \tau_d^{\mathrm{cons},b} \geq L(d)\}\right|$ consistent pairs, and a rank cut that contains no consistent pair and exactly $N_{\mathrm{TP}} := \left|\{a : \tau_d^{\mathrm{div},a} > C_{\mathrm{adj}}\}\right|$ divergent pairs. (ii) (Strong separation ⇒ every cut is pure or complete.) If $L(d) > C_{\mathrm{adj}}$ (equivalently (41)*), every rank cut of size $k \leq n_c$ contains only divergent pairs, every rank cut of size $k \geq n_c$ contains all divergent pairs, and the divergent pairs form the unique rank cut of size n_c. (iii) (No bound constants needed.) The rank cuts are determined by the realized τ_d, computed from $\mathbf{M}$ alone. The constants $\{K_x, K_y, L_x, \delta, \eta, \Delta\}$ enter only the two bounds C_{adj} and $L(d)$, which locate the gap $(C_{\mathrm{adj}}, L(d))$. The guarantee of (ii) holds for every cut size k, so a rank-based rule enjoys it without estimating any of these constants, whereas a bound-based threshold $\gamma \in (C_{\mathrm{adj}}, L(d))$ would require them.*

Proof. By Proposition 5, every $\tau_d^{\mathrm{cons},b} \leq C_{\mathrm{adj}}$ and every $\tau_d^{\mathrm{div},a} \geq L(d)$.

Part (i): Partial separation. By Proposition 6, $C_1 := \{\text{pairs with } \tau_d \geq L(d)\}$ is a rank cut. It contains every divergent pair, since $\tau_d^{\mathrm{div},a} \geq L(d)$, and exactly the N_{FP} consistent pairs with $\tau_d^{\mathrm{cons},b} \geq L(d)$. Likewise $C_2 := \{\text{pairs with } \tau_d > C_{\mathrm{adj}}\}$ is a rank cut. It contains no consistent pair, since $\tau_d^{\mathrm{cons},b} \leq C_{\mathrm{adj}}$, and exactly the N_{TP} divergent pairs with $\tau_d^{\mathrm{div},a} > C_{\mathrm{adj}}$.

Part (ii): Strong separation. If $L(d) > C_{\mathrm{adj}}$, then $\tau_d^{\mathrm{div},a} \geq L(d) > C_{\mathrm{adj}} \geq \tau_d^{\mathrm{cons},b}$ for every (a, b), so $\min_a \tau_d^{\mathrm{div},a} > \max_b \tau_d^{\mathrm{cons},b}$. Proposition 6 then gives all three claims. Consistently with Part (i), $N_{\mathrm{FP}} = 0$ and $N_{\mathrm{TP}} = n_c$ in this case, so both cuts C_1 and C_2 coincide with the set of divergent pairs.

Part (iii): No bound constants needed. The realized τ_d is computed from $\mathbf{M}$ by Eq. (5), in which none of the constants appears, and the conclusion of Part (ii) holds for every cut size k. □

Scope. Corollary 7 is stated for a single offset d and for pairs (i, j), the level at which the two bounds hold. TWIN's Score aggregates τ_{d_k} over several offsets and assigns each value to a sample. The corollary is the pair-level principle behind this design (Section 4.3), not a sample-level guarantee for the aggregated score.

C Pattern-Agnosticism of Theorem 2: Step-by-Step Verification

This appendix substantiates Theorem 2 by verifying that the proof of Theorem 2 (Appendix B.2) does not anywhere rely on the single-row support of the spike pattern. We work throughout with a generalized contamination model: the observed series $\{\tilde{z}_t\}_{t=1}^{T}$ differs from the clean series $\{z_t\}_{t=1}^{T}$ (with T the time-series length of Definition 1) exactly on a *contamination index set* $\mathcal{T} := \{t \in \{1, \ldots, T\} : \tilde{z}_t \neq z_t\} \subseteq \{1, \ldots, T\}$, with no constraint on $|\mathcal{T}|$ or on the structure of the (necessarily non-zero) perturbation entries $\{\tilde{z}_t - z_t : t \in \mathcal{T}\}$. The single-spike instance ($|\mathcal{T}| = 1$ with $\tilde{z}_{t^\star} - z_{t^\star} = \Delta\, u$ for a unit row u) is the special case of Definition 5. Throughout this appendix, we relax both restrictions.

C.1 Generalized Contamination Data and Plateau

For sample $i \in \{1, \dots, S\}$, define the input-window perturbation $U_i := \tilde{x}_i - x_i \in \mathbb{R}^{W \times D}$ and the target-window perturbation $V_i := \tilde{y}_i - y_i \in \mathbb{R}^{H \times D}$. Their supports satisfy

$$U_i \neq 0 \iff [i,\, i+W-1] \cap \mathcal{T} \neq \emptyset, \qquad V_i \neq 0 \iff [i+W,\, i+W+H-1] \cap \mathcal{T} \neq \emptyset.$$

Sample $i^\star$ is *target-only contaminated* if $U_{i^\star} = 0$ and $V_{i^\star} \neq 0$, with magnitude $\Delta_{i^\star} := \|V_{i^\star}\|_F > 0$. A reference sample j is *plateau-external* if $U_j = 0$ *and* $V_j = 0$, equivalently $[j, j+W+H-1] \cap \mathcal{T} = \emptyset$.

Generalized plateau. The plateau is

$$P \;:=\; \{m \in \{1, \dots, S\} : [m,\, m+W+H-1] \cap \mathcal{T} \neq \emptyset\}. \tag{43}$$

For general $\mathcal{T}$, P is a finite union of contiguous integer intervals. *Specifically*, P is the union, over the connected components $[a, b] \subseteq \mathcal{T}$, of the intervals

$$[\max(1,\, a-W-H+1),\; \min(S,\, b)], \tag{44}$$

i.e., each component $[a, b]$ is extended leftward by $W + H - 1$ (its left endpoint moves from a to $a - W - H + 1$) while its right endpoint b is unchanged, and the result is clipped to $\{1, \dots, S\}$. The intervals of components separated by fewer than $W+H$ clean steps merge into one component of P. The widening is therefore *asymmetric* (one-sided, leftward only): a single time index $t^\star \in \mathcal{T}$ contaminates the $W+H$ samples whose windows reach $t^\star$, namely $\{t^\star - W - H + 1, \dots, t^\star\}$, with no rightward extension. The single-spike case $|\mathcal{T}| = 1$ produces a single contiguous plateau of length at most $W+H$, recovering the plateau of Definition 6.

C.2 Pattern-Agnostic Form of Theorem 2

Let $i^\star$ be target-only contaminated ($U_{i^\star} = 0$, $V_{i^\star} \neq 0$) with magnitude $\Delta_{i^\star} := \|V_{i^\star}\|_F$, and let j be plateau-external at offset $d := |i^\star - j| \geq 1$ with $\overline{\mathbf{M}_{j,e}} \leq \eta$. Inspection of the proof of Lemma 3 (Appendix B.2) shows that $V_{i^\star}$ enters only through its Frobenius magnitude: the window distances on the clean series, the Lipschitz transfer along the clean-input chain, the triangle decomposition, the two applications of Jensen's inequality, and the lift to $\mathcal{W}_2$ do not involve $V_{i^\star}$, and the reverse triangle inequality on $\tilde{y}_{i^\star} - y_j = V_{i^\star} + (y_{i^\star} - y_j)$ depends on $V_{i^\star}$ only via $\|V_{i^\star}\|_F = \Delta_{i^\star}$. The substitution $\Delta \mapsto \Delta_{i^\star}$ therefore propagates verbatim through every step of the proof, yielding the *pattern-agnostic form*

$$\mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j) \;\geq\; c_0(d, \Delta_{i^\star})\, \Delta_{i^\star}^2 - \eta, \qquad c_0(d, \Delta) := \Big(1 - \tfrac{C_{\mathrm{clean}}(d) + \sqrt{\eta}}{\Delta}\Big)_+^2, \tag{45}$$

for any $V_{i^\star}$ supported on multiple rows (multi-step contamination), partial rows (per-coordinate contamination), or any modulated / sparse pattern with $\|V_{i^\star}\|_F = \Delta_{i^\star}$. The single-spike case of Theorem 2 is the special instance $V_{i^\star} = \Delta\, e_r\, u$ with $\|V_{i^\star}\|_F = \Delta$, where $e_r \in \mathbb{R}^{H \times 1}$ is the r-th standard basis vector and $r := t^\star - i^\star - W + 1$ is the target row of the spike.

C.3 Plateau Structure for General $\mathcal{T}$ and the Chain Extension of Proposition 5

For a general contamination set $\mathcal{T}$, a sample m is plateau-external when $[m, m+W+H-1] \cap \mathcal{T} = \emptyset$ (Appendix C.1). Every adjacent pair of a chain of such samples is plateau-external, so the proof of Theorem 1 and the chain extension of Proposition 5 apply verbatim and $\tau_d(i, i+d) \leq C_{\mathrm{adj}}$ on every such chain.

C.4 Caveats

- *Per-sample $\Delta_{i^\star}$ varies for block contamination.* For a contiguous block $\mathcal{T} = [a, b]$, the target-only contaminated samples are $i^\star \in [\max(1,\, a-W-H+1),\; \min(S,\, a-W)]$, and $\Delta_{i^\star}^2 = \|V_{i^\star}\|_F^2$ sums only over the rows of $V_{i^\star}$ whose time steps fall in $[a, b]$. Samples at the edge of the block therefore carry a smaller $\Delta_{i^\star}$ and may fall into the collapsed regime $c_0(d, \Delta_{i^\star})\, \Delta_{i^\star}^2 \leq \eta$. The bound is most informative when the block lies substantially inside the target window.

D Synthetic Audit Details

This appendix expands Section 6.2 with the full audit protocol, the per-group numerical tables, and the supporting analysis for the three audit questions Q1–Q3.

Purpose of the synthetic audit. The synthetic audit verifies the structural predictions of Theorems 1–2. All four perturbation families inject anomalies into a smooth sinusoidal series, so the underlying δ is small by construction and the perturbation locations are either single-step or short-block. The audit factors into three complementary checks:

- **Q1 (upper-bound inequality, Theorem 1).** On a dedicated 200-trial *clean-only* subset, the bound $\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+1}) \leq C_{\mathrm{adj}}$ is verified *pointwise* on every adjacent pair. Perturbation families do not enter because the bound concerns clean pairs alone.
- **Q2 (lower-bound scaling order, Theorem 2).** On each of the eight (backbone, perturbation) groups, the log–log Huber slope of τ_{W+H} on Δ is regressed and compared to the predicted exponent 2. A slope check is the natural test here: Δ is the audit's deliberately swept axis, and the exponent is invariant to the unknown constants L_x, η in the lower-bound expression.
- **Q3 (realized population separation).** On the same eight groups, the empirical ratio $\hat{r}(d) := \mathrm{med}(\tau_d^{\mathrm{div}}) \,/\, \mathrm{med}(\tau_d^{\mathrm{cons}})$ quantifies the realized gap between the two τ_d populations (Section 4.3).

All three checks pass uniformly across the four perturbation families (B: Bernoulli spikes, D: block drops, G: amplitude bursts, N: Gaussian noise blocks): zero violations of $\mathcal{W}_2 \leq C_{\mathrm{adj}}$ on 961,600 clean pairs (Q1), Huber slopes uniformly in $[1.943, 2.009]$ (Q2), and per-group median $\hat{r}(d)$ exceeding 25 with pooled median $\mathbf{44.23}\times$ (Q3).

D.1 Protocol

Series. The clean series is a 7-channel sinusoidal signal $z_t^{(c)} = \sum_{k=1}^{10} A_{k,c} \sin(2\pi(k/T_p)t + \phi_{k,c}) + \varepsilon_t^{(c)}$ with base period $T_p = 100$, amplitudes $A_{k,c} \sim \mathcal{U}(0.5, 1.0)$, phases $\phi_{k,c} \sim \mathcal{U}(0, 2\pi)$, and observation noise $\varepsilon_t^{(c)} \sim \mathcal{N}(0, 0.01^2)$. The total length is $T_{\mathrm{total}} = 8000$ steps, partitioned into a 500-step warm-up, 5000 training steps, a 500-step gap, and 2000 test steps. Each channel is z-scored independently. With $W = H = 96$ and stride 1, the sliding-window construction yields $S_{\mathrm{train}} = 4{,}809$ training samples per seed.

Backbones. We use two backbones. The first is a 3-layer MLP with hidden width 512, GELU activations, dropout 0.1, and 0.95M parameters. The second is a DLinear model with a kernel-25 trend–seasonality decomposition. Both are trained with AdamW (β_1=0.9, β_2=0.999, learning rate 10^{-3}, weight decay 10^{-4}), batch size 256, MSE loss, $E = 50$ epochs, a cosine-annealing schedule, and patience-5 early stopping. The dynamics matrix $\mathbf{M} \in \mathbb{R}^{S \times E}$ records the per-epoch loss of every training sample.

Perturbation families. We inject four families of perturbations into the z-scored training series, each controlled by a single magnitude parameter swept over 20 log-spaced levels: (B) Bernoulli spikes at probability $p = 0.003$ with per-spike magnitude $k \in [2, 8]$, (D) block drops of length $L \in [5, 50]$ (block content set to zero), (G) length-20 amplitude bursts with multiplicative scale in $[1.5, 5]$, and (N) additive isotropic Gaussian noise with standard deviation $\sigma \in [0.2, 5]$ on a length-20 block.

Lipschitz probes. At every epoch we estimate K_x and K_y from per-sample loss-gradient norms over 128 random adjacent pairs and 8 interpolation points along Γ_i. The clean-pair upper bound is $C_{\mathrm{adj}} = (K_x\sqrt{W} + K_y\sqrt{H})\,\delta_{\max}$, where $\delta_{\max} = \max_t \|z_{t+1} - z_t\|_F$ is the empirical worst-case one-step jump on the training segment.

Scale. The main grid comprises 2 backbones $\times$ 4 perturbation families $\times$ 20 levels $\times$ 100 seeds $=$ 16,000 trained cells. Beyond the main grid, we run 200 clean-only trials for Theorem 1 verification, which yield 961,600 adjacent pairs in total, and the contamination-ratio grid below for the bound of Theorem 2.

Contamination-ratio grid. A second grid is organized by the contamination ratio $\Delta/\delta_{\max}$ and uses the same generator, backbones, and hyperparameters as the main grid. It covers 5 families: a single target spike plus B/D/G/N. The single spike, B, G, and N each take all 8 target ratios $\{2, 5, 10, 20, 50, 100, 200, 500\}$. Family D takes only 4 levels within its attainable range: zeroing a block removes at most the signal itself, so its ratio cannot exceed $\Delta/\delta_{\max} \approx 12.6$. This gives $4 \times 8 + 4 = 36$ (family, level) combinations, each trained on 2 backbones with 100 seeds, for 7,200 trained cells in total. Each run also records the clean loss matrix, from which η_c is computed. Contaminated–reference pairs are drawn at up to 64 target-only contaminated samples per (run, offset), each paired with every plateau-external reference at that offset: 4,628,798 pairs at eleven offsets from $d = 1$ to 300, including TWIN's dilations 1, 48, 96, and 192.

D.2 Q1: Clean-Side Upper Bound

The upper-bound inequality $\mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_{i+1}) < C_{\text{adj}}$ holds on all 961,600 adjacent clean pairs, across both backbones and every one of the 100 seeds (Table 5).

Table 5: Theorem 1 clean-side verification. Empirical $\mathcal{W}_2/C_{\text{adj}}$ ratio per backbone. The bound holds on every adjacent pair.

Backbone	$\mathcal{W}_2/C_{\text{adj}}$ median	p95	#pairs	#violations
DLinear	0.0014	0.0038	480,800	0
MLP	0.0016	0.0044	480,800	0
Combined	**0.0015**	**0.0041**	**961,600**	**0**

Rank-based interpretation. The rank rule requires only the validity of the upper bound on every clean pair (Corollary 7), not its numerical tightness: by the same corollary, C_{adj} never enters the empirical computation of τ_d, so the rank ordering produced by TWIN is independent of the magnitude of C_{adj}. Q1 reports exactly what that validity hinges on: the violation rate, which is 0 across all 961,600 adjacent clean pairs.

The observed median ratio $\mathcal{W}_2/C_{\text{adj}} \approx 1.5\times10^{-3}$ reflects two properties of the worst-case bound, not of the data. *(i) Uniform-over-epochs requirement.* The proof of Theorem 1 bounds $|\mathbf{M}_{i,e} - \mathbf{M}_{i+1,e}| \leq C_{\text{adj}}$ at *every* epoch, so the gradient suprema (K_x, K_y) in C_{adj} are taken over all epochs, including the early ones in which the loss and its gradients are largest, whereas $\mathcal{W}_2$ averages the differences between sorted losses over all E epochs, most of which come late in training, where these differences are much smaller. *(ii) Worst-case alignment in the proof.* The Cauchy–Schwarz step assumes adversarial alignment between loss gradient and temporal displacement, and the per-step substitution $\|z_t - z_{t+1}\|_F \leq \delta_{\max}$ uses the global maximum increment. Both bounds are tight only in the worst case. The two factors compound to the two-to-three orders of magnitude observed in Table 5. This gap is a property of the bound's worst-case construction, not of the data, and it does not affect the rank ordering of τ_d.

D.3 Q2: $\Omega(\Delta^2)$ Slope Verification

For each (backbone, perturbation) group we collect every $(i^\star, j)$ pair in which $i^\star$ is target-only contaminated (its target window contains contaminated time steps and its input window is clean), and regress $\log \tau_{W+H}(i^\star, j)$ on $\log \Delta$, with $\Delta := \|\tilde{y}_{i^\star} - y_{i^\star}\|_F$ as in Figure 3. We use a Huber estimator with threshold 1.35. Figure 4 shows the fits, and Table 6 lists the slopes.

Where the bound is active. The slope checks the order of Theorem 2, while the bound itself is checked in its residual form (40), which implies (27) and does not depend on d (Lemma 4). On the contamination-ratio grid of Appendix D.1, the bound is never violated on the 4,628,798 contaminated–reference pairs. At $d{=}W{+}H$, the bound is non-trivial (its right-hand side is positive) on 82.9% of the 712,017 pairs, and on these pairs the realized $\mathcal{W}_2$ exceeds the bound by a median factor of 1.19.

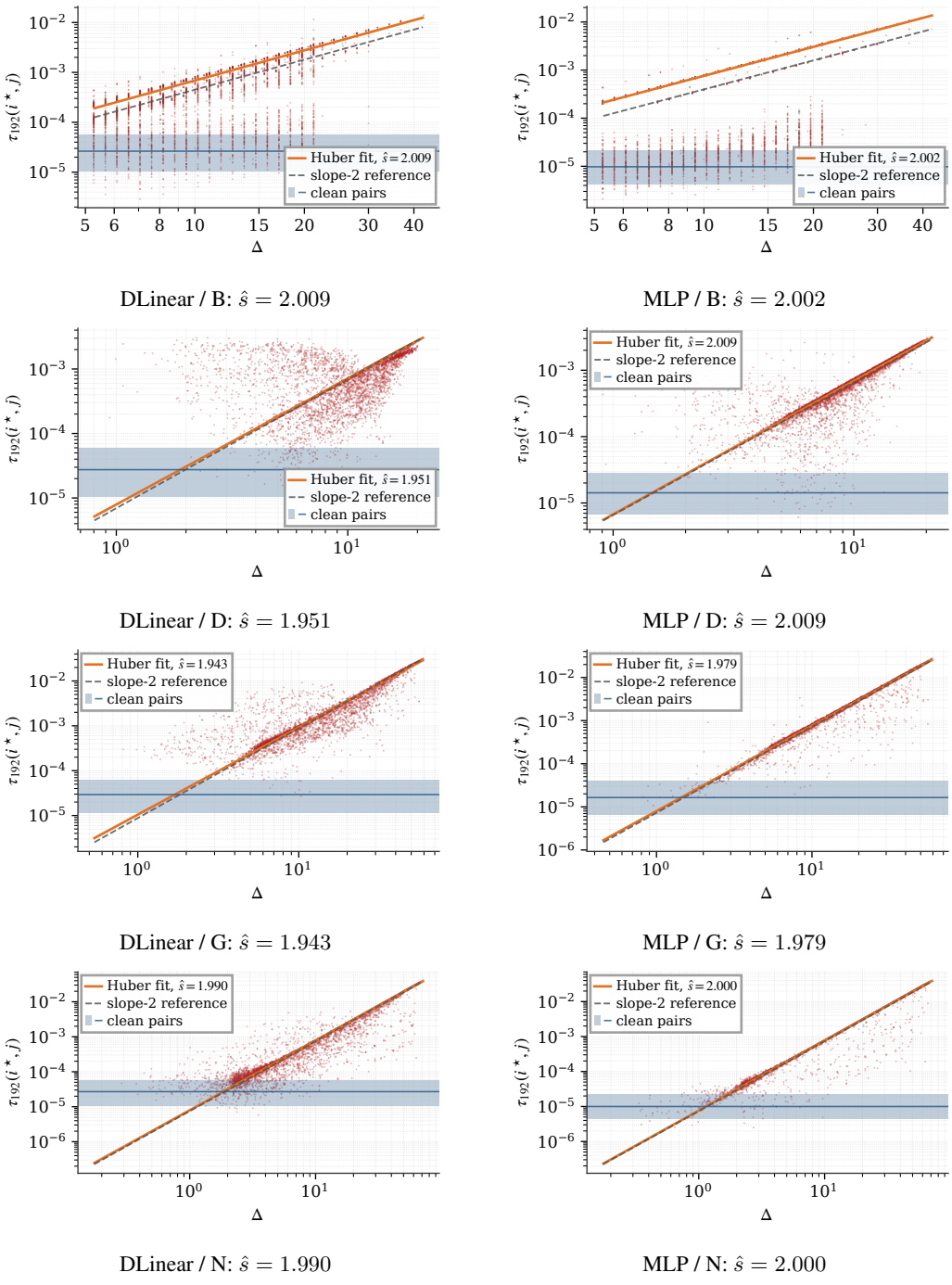


Figure 4: **Per-group verification of the $\Omega(\Delta^2)$ scaling law on every (backbone, perturbation) group of the synthetic audit.** Rows: perturbation family (from top: B: Bernoulli spike, D: block drop, G: amplitude burst, N: Gaussian noise block). Columns: backbone (left: DLinear, right: MLP). Each point is one pair $(i^\star, j)$ plotted as $(\Delta, \tau_{W+H}(i^\star, j))$ on log–log axes, where $i^\star$ is a target-only contaminated sample (clean input window, contaminated target window), $j = i^\star + W + H$, $\Delta := \|\tilde{y}_{i^\star} - y_{i^\star}\|_F$, and $\tau_{W+H}(i^\star, j) := \mathcal{W}_2(\hat{\mu}_{i^\star}, \hat{\mu}_j)/(W+H)$ with $W+H = 192$. For legibility each panel shows a random 2.5% of the n pairs of Table 6; the fit uses all of them. The orange line is the Huber regression with reported slope $\hat{s}$, the dashed line is a slope-2 reference (arbitrary intercept), and the blue band and line give the 10th–90th percentiles and median of τ_{W+H} on clean pairs. The eight estimated slopes lie in $[1.943,\ 2.009]$ (mean 1.985, median 1.995), inside the ± 0.1 band around the theoretical value 2.

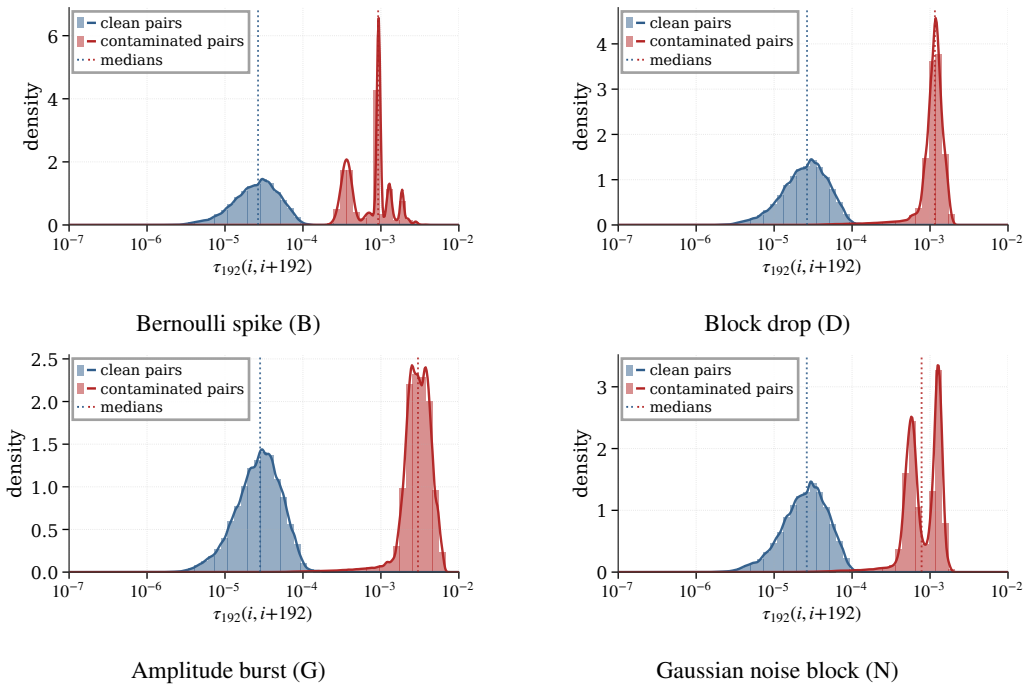


Figure 5: **Empirical separation of clean and contaminated τ_{W+H} distributions, one panel per perturbation family.** Each panel shows the histogram and kernel density estimate of $\tau_{W+H}(i,j) = \mathcal{W}_2(\hat{\mu}_i, \hat{\mu}_j)/(W{+}H)$ at the canonical cross-plateau dilation $d = W{+}H = 192$, on DLinear at perturbation level 10 (of 20; a representative mid-range magnitude). Each estimate pools data from 100 seeds. Within every panel, the blue density is for clean pairs (τ_d^{cons} in the notation of Section 4.3) and the red density is for contaminated pairs (τ_d^{div}). The dotted vertical lines mark the empirical medians of the two populations. Across all four perturbation families the contaminated density sits at τ_{W+H} values 1–2 orders of magnitude above the clean density. Aggregated separation ratios for all eight (backbone, perturbation) groups are reported in Table 7.

D.4 Q3: Empirical Separation of τ_d Populations

We measure the separation between the two τ_d populations using the *empirical separation ratio*

$$\hat{r}(d) \;:=\; \frac{\operatorname{med}\big(\tau_d^{\text{div}}\big)}{\operatorname{med}\big(\tau_d^{\text{cons}}\big)}, \tag{46}$$

which depends on neither the bound constants nor the multiplicative scale of $\mathcal{W}_2$. Computing $\hat{r}(d)$ requires knowing which pairs are divergent and which are consistent, i.e., the ground-truth contamination indices. $\hat{r}(d)$ is the typical-case counterpart of the ratio $L(d)/C_{\text{adj}}$ of the lower to the upper bound, which exceeds 1 exactly under the strong-separation condition of Proposition 5: for the populations of Corollary 7, $\operatorname{med}(\tau_d^{\text{div}}) \geq \min_a \tau_d^{\text{div},a} \geq L(d)$ and $\operatorname{med}(\tau_d^{\text{cons}}) \leq \max_b \tau_d^{\text{cons},b} \leq C_{\text{adj}}$, so $\hat{r}(d) \geq L(d)/C_{\text{adj}}$ whenever $L(d) > 0$. Table 7 reports $\hat{r}(d)$, and Figure 5 shows the two τ_{W+H} populations for each family on DLinear.

Table 6: Per-group Huber slope of $\log \tau_{W+H}$ on $\log \Delta$ at the canonical cross-plateau dilation $d = W+H = 192$. The theoretical prediction is 2. All eight groups lie inside the strict band $[1.9, 2.1]$ around the prediction. n is the number of $(i^\star, j)$ pairs in each regression. We report the Huber slope as the headline.

Backbone	**Pert.**	**Huber slope**	n
DLinear	B	**2.009**	400,000
DLinear	D	**1.951**	395,200
DLinear	G	**1.943**	400,000
DLinear	N	**1.990**	400,000
MLP	B	**2.002**	400,000
MLP	D	**2.009**	395,200
MLP	G	**1.979**	400,000
MLP	N	**2.000**	400,000
Mean	—	**1.985**	—
Median	—	**1.995**	—

Table 7: Empirical separation ratio $\hat{r}(d)$ aggregated within each (backbone, perturbation) group over 20 perturbation levels and 100 seeds. The summary row pools individual $\hat{r}(d)$ values across the full 16,000-cell grid. Every per-row median exceeds 25.

Backbone	**Pert.**	$\hat{r}$ **med.**	$\hat{r}$ **p95**	$\hat{r}$ **max**
DLinear	B	32.50	122.0	194.2
DLinear	D	39.15	88.1	140.2
DLinear	G	**94.93**	299.3	516.3
DLinear	N	25.36	342.0	642.7
MLP	B	89.03	289.2	576.3
MLP	D	29.06	69.2	150.5
MLP	G	53.41	141.9	294.3
MLP	N	39.68	561.2	1178.1
Pooled (16,000 cells)	—	**44.23**	—	1178.1

E Experimental Setup and Baselines

E.1 Training and Scoring Pipeline

Codebase and hardware. iTransformer, PatchTST, and DLinear are trained on top of the publicly available Time-Series-Library,[3] which supplies both the canonical model implementations and the standard MTSF data pipeline (stride-1 sliding windows, fixed train/val/test splits). MambaTS follows its original protocol with look-back $W{=}720$ under the same stride-1 sliding-window protocol. For every backbone and dataset we keep the respective default hyperparameters (learning rate, training epochs, optimizer, dropout, batch size, and so on). No per-method tuning is performed, so the performance gaps between pruners reflect the pruning signal itself and not backbone-specific tuning. All training, scoring, and retraining runs are performed on NVIDIA L40S GPUs and RTX 3090 GPUs.

Baseline score formulas. Every loss-dynamics-based baseline is computed on the per-sample dynamics matrix $\mathbf{M} \in \mathbb{R}^{S \times E}$ of Definition 3. EL2N [12] is the Frobenius norm of the per-sample residual at a fixed scoring epoch. TDDS [22] scores each sample by the temporal variance of its loss differences between consecutive epochs. Avg Loss is the per-sample mean loss over epochs, our regression adaptation of Yang et al. [20]. We tune nothing beyond the scoring-epoch and scoring-prefix conventions specified in the original papers. Two data-side baselines of Table 1 are zero-cost controls. Matched-size stride thinning raises the stride of the sliding-window protocol until the retained count matches the target size. The MAD filter flags amplitude spikes. In each channel of

[3] https://github.com/thuml/Time-Series-Library

Algorithm 1 TWIN: Trajectory-Wise Inconsistency with Neighbors

Require: Training dynamics $\mathbf{M} \in \mathbb{R}^{S \times E}$ (raw losses); dilation set $\mathcal{D} = \{d_k\}_{k=1}^{K}$; mode $\in \{\text{FIXED-RATIO}(\text{PR}),\ \text{IQR}\}$ // PR used only in Fixed-ratio

$\mathbf{r}_i \leftarrow \text{sort}(\mathbf{m}_i)$ for all i // absorbs benign phase drift

for $k = 1$ to K **do**

  for $i = 1$ to $S - d_k$ **do**

    $\tau_k(i) \leftarrow \dfrac{1}{d_k\sqrt{E}} \left\|\mathbf{r}_i - \mathbf{r}_{i+d_k}\right\|_2$ // Eq. (5)

  end for

end for

$\text{Score}(i) \leftarrow \sum_{k=1}^{K} \tau_k(i)$ for all i // multi-scale aggregation

if mode = FIXED-RATIO **then**

  $\mathcal{I}_{\text{prune}} \leftarrow$ top PR samples by Score

else

  Compute Q_1, Q_3, IQR on $\{\text{Score}(i)\}$; $\mathcal{I}_{\text{prune}} \leftarrow \{i : \text{Score}(i) > Q_3 + 1.5\,\text{IQR}\}$

end if

return $\mathcal{I}_{\text{keep}} = \{1, \ldots, S\} \setminus \mathcal{I}_{\text{prune}}$

the training split, it subtracts a centered rolling median of width 25 and converts the residual v_t into the robust z-score $0.6745\ |v_t - \text{med}(v)|\ /\text{MAD}(v)$, where the median and the MAD are taken over the whole split. A sample's score is the largest such z-score over all channels and the $W+H$ time steps it spans, so a single spike sets the score. The highest-scoring samples are removed. TimeInf and FAF-S are specified in Appendices E.4 and E.5.

Lightweight in-loop logging. EL2N, TDDS, Avg Loss, and TWIN all consume the same dynamics matrix $\mathbf{M}$, so we share its construction across the four scores. A naive implementation would persist every training checkpoint $\{\theta_e\}_{e=1}^{E}$ to disk and, after training finishes, run E separate forward passes over the full training set to populate $\mathbf{M}_{i,e} = \mathcal{L}(\theta_e, \tilde{x}_i, \tilde{y}_i)$. We instead adopt the in-loop protocol of Avg Loss: the training loop itself emits each sample's per-batch loss at the end of every epoch, and the values stream directly into $\mathbf{M}$ during training. The two implementations produce similar matrices, but the in-loop variant removes both the E checkpoint snapshots on disk and the post-hoc forward-pass stage. The only marginal cost over a standard training run is a single $\mathbb{R}^{S \times E}$ log buffer in memory.

Pseudocode. Algorithm 1 gives TWIN's full pseudocode.

E.2 Reporting Convention

Retraining time excludes scoring cost. The retraining-time accounting in Section 6.3 measures the time required to retrain the forecaster on the pruned subset and excludes the one-off scoring cost. A run that never reaches the best unpruned test loss is recorded at 100%, the unpruned-baseline training time. Within the data-pruning literature, timing conventions vary: Avg Loss [20] reports retrain-only after separating scoring from retraining, while Selection via Proxy [3] reports end-to-end including proxy training. We adopt the decomposed convention because the four loss-dynamics-based pruners we compare (EL2N, TDDS, Avg Loss, and TWIN) share the same per-sample training-dynamics matrix $\mathbf{M}$ constructed during a single training pass, enabling fair timing comparison among them. In single-deployment workflows, the scoring cost is paid once. In periodic retraining, hyperparameter search, and neural architecture search, where the same $\mathbf{M}$ is reused across many candidate retrainings, the cost amortizes. Appendix H.5 complements this convention with a full end-to-end accounting that charges the scoring run to a single training run.

E.3 RobustTSF Baseline

RobustTSF [2] scores each training sample $\tilde{x}_i$ by the residual of an L1 trend filter at the last point of its input window. The method was originally derived for univariate forecasting at a single target step. To run it on our multivariate long-horizon benchmark ($D \in \{7, 8, 21, 137, 321\}$, $H \in \{96, 192, 336, 720\}$), we make the smallest changes that still recover the original score when $D{=}1$

and $W{=}16$: (i) L1 trend filtering is run per channel via ADMM, then reduced across channels with $\mathrm{agg} \in \{\max, \mathrm{mean}\}$; (ii) the Dirac-window width is $w_x = \mathrm{round}(W/16) = 6$ for our $W{=}96$; (iii) the time spacing is unit, which preserves the paper's interpretation of λ; (iv) the hard threshold is dropped in favor of raw per-sample scores. The resulting score is

$$A(\tilde{x}_i) \;=\; \tfrac{1}{w_x} \sum_{t=W-w_x}^{W-1} \mathrm{agg}_c \big| \tilde{z}_{i+t}^{(c)} - s_{i+t}^{(c)} \big|, \tag{47}$$

where $s_t^{(c)}$ is the trend that L1 trend filtering estimates on channel c. We sweep $\lambda \in \{0.1, 0.3, 1.0, 1.5, 2.0\}$ and both aggregations on ETTh2, ETTm2, and Weather across the four horizons (Table 8). The best single configuration is $\max$ with $\lambda{=}1.5$, which gives -0.17% Relative MSE at PR$=10\%$ and -0.18% under IQR on this three-dataset subset.

The limited improvement at higher pruning rates is consistent with the method's original design scope. The sample-robustness argument concerns anomalies in the input window and says nothing about the long target windows used here ($H \leq 720$). The L1-fidelity score is also purely data-side: it sees neither the optimizer nor the trained model state, so it cannot diagnose the loss-dynamics signals that other baselines and TWIN rely on.

Table 8: RobustTSF hyperparameter sweep on three datasets (ETTh2, ETTm2, Weather) and four horizons. Avg Relative MSE (%), lower is better. The best single configuration on this subset is $\max$ with $\lambda{=}1.5$.

Method	**Avg Relative MSE (%) at Pruning Rate**			
	PR=10%	PR=20%	PR=30%	IQR
Random	+0.14	+0.25	+0.40	—
RobustTSF $\max$, $\lambda{=}0.1$	+0.33	+0.40	+0.80	+0.20
RobustTSF $\max$, $\lambda{=}0.3$	+0.01	+0.25	+0.90	+0.17
RobustTSF $\max$, $\lambda{=}1.0$	−0.11	**−0.03**	+0.57	−0.15
RobustTSF $\max$, $\lambda{=}1.5$	**−0.17**	+0.08	+0.53	**−0.18**
RobustTSF $\max$, $\lambda{=}2.0$	0.00	+0.21	+0.52	−0.14
RobustTSF mean, $\lambda{=}0.1$	+0.39	+0.52	+0.90	+0.22
RobustTSF mean, $\lambda{=}0.3$	+0.19	+0.48	+0.69	+0.29
RobustTSF mean, $\lambda{=}1.0$	+0.19	+0.29	+0.55	+0.10
RobustTSF mean, $\lambda{=}1.5$	+0.08	+0.25	**+0.38**	+0.19
RobustTSF mean, $\lambda{=}2.0$	+0.07	+0.35	+0.70	−0.08

E.4 Learning-Agnostic and Influence-Based Valuation: TimeLAVA and TimeInf

TimeLAVA [8] and TimeInf [23] are data-valuation methods for time series: each assigns a value to parts of the training series, and pruning removes the parts valued least. TimeLAVA trains no model: it values temporal segments by their marginal contribution to a wavelet-based Wasserstein discrepancy, computed with optimal transport, between the evaluated series and reference data, and aggregates the segment values into pointwise scores. TimeInf values individual time points: it averages influence functions over all time blocks that contain a point and, for forecasting, measures the influence on a held-out loss, so the lowest-valued points are those that worsen it most. Both methods thus value time points or segments rather than sliding-window samples. We run both as sample-level pruning methods in our protocol, with the validation set as TimeLAVA's reference data: iTransformer, six benchmarks $\times$ four horizons $\times$ five seeds (Table 9). TimeInf is additionally evaluated on the full eight-benchmark grid in Table 1.

Common protocol. Both methods are run once per dataset and horizon on the standardized training split. Each method assigns a value to every time point of the split, with lower values marking less useful points, and a sample's value is the mean value over the $W{+}H$ time points it spans. The samples valued least are removed: under the fixed-ratio protocol, the lowest-valued PR of the samples, and under IQR, those more than $1.5\,\mathrm{IQR}$ below the first quartile Q_1 of the sample values (Tukey's lower fence). The forecaster is then retrained from scratch on the retained samples.

TimeLAVA implementation. We use the official implementation[4] with the setting of its data-pruning experiments: non-overlapping segments of length 96, db4 wavelet features at decomposition level 2, and the remaining hyperparameters as set there. All other options are left at their package defaults. Valuation is run on each channel separately, the values are averaged across channels, and each time point takes the value of the segment that contains it.

TimeInf implementation. We use the official implementation[5] with its linear autoregressive model and a block length of 100, as in its data-pruning experiments. In each channel, a training block is valued by its influence on the validation loss, averaged over all validation blocks. The block values are averaged across channels, and each time point takes the mean value of the blocks that contain it.

Table 9: Comparison with time-series data-valuation methods: Avg Relative MSE (%), iTransformer, six benchmarks, four horizons, five seeds. Holm p: Holm-corrected two-sided Wilcoxon signed-rank p-value against TWIN on the 96 paired (dataset, horizon, pruning-rate) cells, corrected over the two comparisons.

Method	PR=10%	PR=20%	PR=30%	IQR	**Holm p vs TWIN**
TimeInf	-0.54	-0.05	$+0.78$	-0.51	5.8×10^{-15}
TimeLAVA	$+0.64$	$+1.48$	$+2.23$	0.00	4.3×10^{-17}
TWIN (Ours)	$\mathbf{-1.41}$	$\mathbf{-1.14}$	$\mathbf{-0.78}$	$\mathbf{-1.56}$	— (ref)

TimeLAVA. The method assumes a clean reference set. As in its forecasting experiments, we use the validation set as the reference. In those experiments, the validation set is clean because only training segments are contaminated, whereas on our real benchmarks it is not guaranteed to be clean. TWIN needs no external reference, since it compares each sample against its own sliding-window neighbors.

TimeInf. Its criterion is close to what our protocol measures, and it does improve on the unpruned baseline at the lowest pruning rate (-0.54 at PR$=10\%$). The two behave differently as the pruning rate rises: between PR$=10\%$ and PR$=30\%$ TimeInf's Relative MSE rises by 1.32 points while TWIN's rises by 0.63, and at PR$=30\%$ TimeInf no longer improves on the unpruned baseline.

E.5 FAF-S: A Static Pruning Version of FAF

FAF [14] extends RHO-Loss [10] to time-series forecasting and selects samples online: inside every mini-batch it ranks samples by the reducible loss, i.e., the loss of the model being trained (the target model) minus the loss of a reference model on the same sample. It updates the target model only on the top-ranked ones. Where RHO-Loss keeps the reference model fixed, FAF's selection component (AdaRho) adapts it during training. A large target loss says the sample is not yet learned, and a large reference loss says it is hard to learn at all (noise, anomalies). Their difference discounts the second from the first. An online rule of this kind cannot be placed in Table 1, where every method must fix one retained set before retraining and is evaluated at identical retained-set sizes. We therefore derive **FAF-S** (S for static): the same reducible loss, computed once per sample and then used exactly like every other pruning score (fixed-ratio and IQR protocols).

Three stages. Each stage is run once per (dataset, horizon) pair. *(T) Target model.* The unpruned iTransformer is trained with the default hyperparameters, and its weights are saved at the end of every epoch, giving checkpoints $\theta_1, \ldots, \theta_E$. *(R) Reference model.* We draw 25% of the samples uniformly without replacement, train the same backbone with the same hyperparameters on this subset only, and keep the early-stopping checkpoint with the lowest validation loss, θ^{ref}, as in FAF's reference-model protocol (FAF's default multivariate reference is a pretrained TimePFN, but FAF reports similar results with an iTransformer reference). *(I) Inference.* With the unpruned training set in its original order, in evaluation mode and without gradients, we compute the per-sample loss of Definition 2 (the

[4] https://github.com/lww28/TimeLAVA
[5] https://github.com/yzhang511/TimeInf

training criterion) for every checkpoint of the target model and for the reference model,

$$\ell_e^{\mathrm{tgt}}(i) := \mathcal{L}(\theta_e, \tilde{x}_i, \tilde{y}_i), \qquad \ell^{\mathrm{ref}}(i) := \mathcal{L}(\theta^{\mathrm{ref}}, \tilde{x}_i, \tilde{y}_i), \qquad i \in \{1, \dots, S\},\ e \in \{1, \dots, E\}.$$

The reference loss is evaluated on all S samples, including the 25% the reference model was trained on.

Score. For every checkpoint e, the reducible loss ρ_e is standardized robustly across samples, and the standardized values are averaged along the training trajectory:

$$\rho_e(i) := \ell_e^{\mathrm{tgt}}(i) - \ell^{\mathrm{ref}}(i), \qquad \zeta_e(i) := \frac{\rho_e(i) - \mathrm{med}_e}{\mathrm{IQR}_e}, \qquad \text{FAF-S}(i) := \frac{1}{E}\sum_{e=1}^{E} \zeta_e(i),$$

where med_e and IQR_e are the median and the interquartile range (third minus first quartile) of $\{\rho_e(i)\}_{i=1}^{S}$. The standardization is per checkpoint because the overall loss level is large early in training and small late, so an unstandardized average would be dominated by the early checkpoints. It uses the median and the interquartile range rather than the mean and the standard deviation so that the extreme samples do not set the scale. After standardization ζ_e is dimensionless, which makes checkpoints, horizons, and datasets comparable.

What is pruned. As with TWIN, the highest-scoring samples are removed: those on which the target model's loss stays high along the whole trajectory relative to the reference model's loss. The online rule of RHO-Loss and FAF prioritizes such samples for parameter updates rather than discarding them, so the FAF-S results in Table 1 measure pruning by the static FAF-S score, not the effect of FAF's online selection. In this role FAF-S improves on the unpruned baseline at a low pruning rate (-0.74 at PR$=10\%$ and -0.62 under IQR) and degrades as the rate grows ($+0.47$ at PR$=20\%$, $+1.95$ at PR$=30\%$), where TWIN remains at -1.20 and -0.69.

F Statistical Significance Analysis

This appendix details the statistical-significance test used in Table 1 of the main paper. It also lists the p-values that this table defers to the appendix.

F.1 Test Protocol

Statistic. For each baseline we run a two-sided Wilcoxon signed-rank test with the null

$$H_0\colon\ \mathrm{med}\big(\text{Relative MSE}_{\mathrm{TWIN}} - \text{Relative MSE}_{\mathrm{baseline}}\big)\ =\ 0$$

against the alternative H_1: the median is non-zero. The paired observations are the per-cell mean Relative MSE values, each averaged over the cell's five seeds.

Pairing unit. A *cell* here is one (dataset, horizon, pruning-rate) combination (distinct from the (backbone, perturbation, level, seed) cells of the synthetic audit in Appendix D). The iTransformer grid in Table 1 contains $8 \times 4 \times 4 = 128$ such cells (eight datasets, four horizons, and four pruning protocols PR $\in \{10\%, 20\%, 30\%, \text{IQR}\}$), and $8 \times 4 \times 3 = 96$ for Random and Stride, which have no IQR column. The ablation grids use the six benchmarks other than Solar and Exchange and contain $6 \times 4 \times 4 = 96$ cells. Each cell contributes a single paired observation $(\text{Relative MSE}_{\mathrm{TWIN}},\ \text{Relative MSE}_{\mathrm{baseline}})$.

Why cell-level rather than seed-level. A naive within-cell test on the five seeds is structurally underpowered: at $n = 5$, the two-sided Wilcoxon signed-rank distribution has minimum p-value $1/16 \approx 0.0625$, so rejection at the 0.05 level is mathematically unreachable regardless of effect size. Working at the cell level instead lifts the paired-sample size to $n \in \{96, 128\}$ and follows Demšar [4] in applying the Wilcoxon signed-rank test to paired per-problem scores, with (dataset, horizon, pruning-rate) cells in place of datasets.

Multiple-comparison correction. For Table 1, the raw two-sided p-values across the family of nine baseline-vs-TWIN comparisons are corrected by the step-down Holm–Bonferroni procedure, which controls the family-wise Type-I error rate at the 0.05 level.

Sign convention. Both Relative MSE (in %) and the underlying MSE are oriented so that a *lower* value is better. Hence Relative MSE_{TWIN} − Relative $\text{MSE}_{\text{baseline}} < 0$ means TWIN attains a lower Relative MSE on that cell.

F.2 Relative MSE Results on iTransformer (Corresponding to Table 1)

Table 10 reports, for each baseline, the Holm-corrected p-value against TWIN on the grid of Table 1.

Table 10: Relative MSE significance on the iTransformer grid. The Relative MSE values are those of Table 1. The rightmost column reports the Holm-corrected two-sided Wilcoxon signed-rank p-value against TWIN (family of nine comparisons), computed on the 128 paired (dataset, horizon, pruning-rate) cell-mean observations of the grid (96 for Random and Stride, which have no IQR column).

Method	**Avg Relative MSE (%) at Pruning Rate**				**Holm p vs TWIN**
	PR=10%	PR=20%	PR=30%	IQR	
Random	+0.16	+0.32	+0.68	—	1.1×10^{-14}
Stride	+0.15	+0.34	+0.61	—	1.2×10^{-14}
RobustTSF	+0.05	+0.40	+1.11	−0.05	5.2×10^{-21}
MAD	+0.17	+0.47	+1.80	−0.17	1.0×10^{-14}
EL2N	−0.84	−0.65	−0.25	−0.83	2.0×10^{-6}
Avg Loss	−1.13	−1.06	−0.45	−1.10	2.0×10^{-6}
TDDS	−1.14	−0.52	+0.25	−0.96	2.0×10^{-6}
TimeInf	−0.32	−0.09	+0.48	−0.24	1.0×10^{-14}
FAF-S	−0.74	+0.47	+1.95	−0.62	4.1×10^{-11}
TWIN (Ours)	**−1.41**	**−1.20**	**−0.69**	**−1.46**	— (ref)

G Analysis: Where the Gains Come From

Table 1 compares the methods by Relative MSE averaged over all test samples. Such an average shows that TWIN outperforms the strongest baseline, but not where its margin comes from. This appendix locates the source: the margin concentrates on rare test samples (Appendix G.1), and the samples TWIN removes carry measurable contamination (Appendix G.2). Appendix G.3 stress-tests the score on clustered contamination.

G.1 Rarity-Stratified Evaluation

Avg Loss ranks a sample by the magnitude of its loss alone, and that magnitude does not explain why the loss is large: a contaminated sample and a difficult-but-clean sample look the same to it. TWIN measures something different: not how large a sample's loss is, but how far its loss trajectory sits from those of its neighbors. If one sample is difficult, its neighbors are difficult too, their losses are all large, the gaps between them are small, and TWIN keeps them. A contaminated sample is the opposite case: its loss is large where its clean neighbors' are not. The two scores therefore disagree mainly on difficult-but-clean samples: Avg Loss removes them, TWIN keeps them. These samples often carry the rarer patterns of the series, which yields a testable prediction: on ordinary test samples the two methods should be close, and on rare ones TWIN should pull ahead.

Setup. We train iTransformer models on ETTh2. We define the rarity of a test sample as its nearest-neighbor distance to the training set: each test sample is flattened into a vector in z-scored space, and we record its L_2 distance to the nearest neighbor among a stride-10 subsample of training samples. Test samples are then stratified into deciles of this rarity. We prune the training set once with Avg Loss and once with TWIN, retrain a model on each, and evaluate both on the same test samples. Table 11 reports the upper half of the deciles, d6 (mid-rare) to d10 (most rare). MSE is computed on the samples of each decile, Relative MSE (%) is relative to the unpruned baseline on those same samples (negative = better), and margin = Relative MSE of Avg Loss − Relative MSE of TWIN (positive means TWIN is better).

Table 11: Rarity-stratified test evaluation on ETTh2 (iTransformer): Relative MSE (%) per rarity decile.

Decile	Rarity range	TWIN	Avg Loss	Margin
d6	30.60–31.34	−2.94	−1.94	+1.00
d7	31.34–32.23	−2.17	−1.32	+0.85
d8	32.24–33.37	−6.25	−3.53	+2.72
d9	33.37–36.34	−10.50	−4.83	+5.67
d10 (most rare)	36.36–42.45	−6.36	−3.09	+3.27

The margin is positive in all five deciles, but it is far from uniform: on the three rarest it reaches $+2.72$, $+5.67$, and $+3.27$. On d9 the two methods are not close at all, with TWIN at -10.50% against -4.83% for Avg Loss. Rare test samples make up only a small share of the test set, so a gain concentrated on them is diluted when the error is averaged over all test samples. If TWIN were pruning difficult training samples, the rarest deciles would show its smallest gains. Instead, they show the largest gains in Table 11, consistent with TWIN removing contaminated rather than difficult samples.

G.2 What TWIN Removes and What Remains

We characterize the removed set on ETTh2 and ETTm2 at all four horizons. The removed set is what the TWIN score selects under the IQR protocol (which removes 9.2% to 10.8% of the training samples). Every comparison below is that removed set against the retained one.

The removed samples are enriched in contamination. For each of four real-valued indicators, Table 12 reports the mean and the median of both sets on the indicator's own scale, with their ratios. Six further indicators count the occurrence of specific events inside a sample. Table 13 reports occurrence rates, the share of samples containing at least one such event. Here $|z|$ denotes the absolute value of an entry of the training series after per-channel standardization with the mean and standard deviation of the training split.

In Table 13, the separation grows with the threshold: $2.96\times$ (median ratio) at $|z| \geq 3$, $17.5\times$ at $|z| \geq 4$, and $27.9\times$ at $|z| \geq 6$, so the IQR rule responds to how severe the contamination is, not only to whether it is present.

Table 12: Real-valued contamination indicators of removed vs. retained samples (ETTh2 and ETTm2, four horizons, IQR protocol). Mean and Median are the unweighted means, over the eight (dataset, horizon) pairs, of the per-pair mean and median of the indicator over the samples of each set; each ratio is computed from the unrounded removed and retained values.

Indicator	Mean			Median		
	Removed	Retained	Ratio	Removed	Retained	Ratio
largest $\|z\|$ in the sample	4.32	2.55	1.69×	3.73	2.33	1.60×
share of a sample's entries outside Tukey's fences on its values	0.225	0.0667	3.37×	0.192	0.00247	78.0×
within-sample dispersion	0.769	0.389	1.98×	0.583	0.350	1.67×
off-seasonal residual RMS	0.994	0.464	2.14×	0.810	0.387	2.09×

The diversity of what remains. Nine set-level indicators are measured on the full training set and on the set TWIN's IQR rule retains (Table 14), all embedded in the same 512-dimensional space by flattening each sample and applying a random (Johnson–Lindenstrauss-style) projection. Most values equal 1.000 on the full set by construction, so 1.000 means unchanged. The two exceptions are the coverage 95th percentile, in units of the full set's own median spacing, and the orphan share, in percent of the full set's samples.

How each metric is computed (an "independent" neighbor is one sharing no time step with the reference sample):

Table 13: Event occurrence rates in removed vs. retained samples (ETTh2 and ETTm2, four horizons, IQR protocol). For each (dataset, horizon) pair, a set's rate is the share of its samples that contain at least one such event; Mean and Median are the unweighted mean and the median of the eight per-pair rates, and each ratio is computed from the unrounded rates.

Event	**Mean**			**Median**		
	Removed	Retained	Ratio	Removed	Retained	Ratio
contains an $\lvert z\rvert \geq 3$ entry	77.8%	29.5%	2.64×	78.7%	26.6%	2.96×
contains an $\lvert z\rvert \geq 4$ entry	34.5%	3.2%	10.8×	32.6%	1.9%	17.5×
contains an $\lvert z\rvert \geq 6$ entry	27.0%	2.3%	11.7×	27.5%	1.0%	27.9×
contains a saturated (clipped) run	66.9%	24.2%	2.76×	68.6%	19.4%	3.54×
contains a $\geq 6\sigma$ level step	70.9%	41.7%	1.70×	74.2%	38.8%	1.91×
contains a reverting impulse	88.2%	69.1%	1.28×	89.9%	73.8%	1.22×

Table 14: Set-level diversity of the retained set relative to the full training set. Each entry is the unweighted mean over the eight (dataset, horizon) pairs (ETTh2 and ETTm2, four horizons, IQR protocol).

What is measured	**Full set**	**TWIN retains**
Is any region of the space of flattened samples now empty?		
coverage (median)	1.000	**1.000**
coverage (95th pct.)	2.626	**2.630**
orphan share (%)	0.000	**0.395**
survivor spacing	1.000	**1.000**
How much total spread does the set still have?		
effective directions	1.000	**0.920**
subspace alignment	1.000	**0.949**
bounding range	1.000	**0.854**
total variance	1.000	**0.838**
effective sample size	1.000	**0.891**

- **coverage (median)**: for each of the S full-set samples, the distance to its nearest independent survivor, reported as the median of those S distances and divided by the same median on the full set.
- **coverage (95th pct.)**: the same distances at the 95th percentile, in units of the full set's median spacing.
- **orphan share**: the percentage of full-set samples whose nearest survivor lies farther away than the full set's own 99th-percentile spacing.
- **survivor spacing**: the nearest-independent-neighbor distance measured from the survivors only, divided by what it was for those same samples in the full set (median).
- **effective directions**: the perplexity $\exp(-\sum_i p_i \log p_i)$ of the covariance's eigenvalue shares, divided by the full set's value.
- **subspace alignment**: the mean cosine of the principal angles between the retained and the full set's top-32 eigenvector subspaces.
- **bounding range**: the retained per-dimension range (max $-$ min) as a fraction of the full set's, averaged over the 512 embedding dimensions.
- **total variance**: the trace of the retained covariance divided by the trace of the full one.
- **effective sample size**: $(\operatorname{tr} A)^2 / \|A\|_F^2$ with $A = Z^\top Z$, where Z is the matrix of retained embeddings, divided by the full set's value.

Coverage is essentially unchanged: the median distance from a sample to its nearest surviving independent neighbor is what it was before, and so is the spacing between survivors. The 95th percentile shifts by less than 0.2%, and only 0.395% of samples (fewer than one in 250) no longer

have a survivor within the full set's 99th-percentile spacing. Dispersion drops a little, which is what removing extreme samples does: the bounding range and total variance are computed from the outermost points of the set, and those are exactly the points the IQR rule removes.

In short, the removed samples are enriched in contamination: spikes, saturated runs, and level steps appear in them 1.2 to 3.5 times as often as in the retained samples, and up to 27.9 times as often at the most extreme amplitudes. The retained set still covers almost everything the full set covered.

G.3 Clustered Contamination Stress Test

Theorem 2 is stated for a single contamination event, and Appendix C generalizes the bound to arbitrary contamination sets for samples whose input window is clean. Clustered contamination is nevertheless the right stress test for a neighbor-consistency score, because contaminated samples surrounded by contaminated neighbors might in principle remain mutually consistent.

Setup. We train iTransformer on ETTh2 with $W{=}H{=}96$ under five configurations that hold the total contaminated mass fixed at about $1{,}000$ time steps and redistribute it as $(m, L) \in \{(5, 200), (4, 250), (3, 333), (2, 500), (1, 1000)\}$, where m is the number of clusters and L their length. Each contaminated step receives independent Gaussian noise ($\sigma = 3$ in per-channel z-scored space) on all channels. A sample is labeled INTERIOR when it lies entirely inside one cluster (every point contaminated, the case of most concern), BOUNDARY when it contains both clean and contaminated points, and CLEAN otherwise. The CLEAN class excludes samples in which some channel is naturally constant over the whole 192-step span (stuck-sensor segments of ETTh2). Table 15 reports rate-insensitive separability as the area under the ROC curve (AUROC), with the contaminated class as positives and means over ten seeds.

Table 15: Clustered contamination: AUROC of the TWIN score for separating contaminated from clean samples (iTransformer, ETTh2, ten seeds).

Configuration ($m \times L$)	**INTERIOR vs. CLEAN**	**BOUNDARY vs. CLEAN**	**all-contam. vs. CLEAN**
5×200	0.805	0.729	0.731
4×250	0.901	0.789	0.804
3×333	0.941	0.873	0.891
2×500	0.835	0.714	0.768
1×1000	0.898	0.756	0.852
mean	0.876	0.772	0.809

INTERIOR samples become more separable under clustering, not less: AUROC(INTERIOR) exceeds AUROC(BOUNDARY) on every one of the five configurations, by $+0.068$ to $+0.142$ (mean $+0.104$). TWIN therefore detects contamination inside a cluster: across the five configurations it separates INTERIOR samples from CLEAN ones with a mean AUROC of 0.876 (0.809 for all contaminated samples).

H Extended Evaluation: The Univariate Setting, Backbones, Score Transfer, and Cost

This appendix extends the evaluation of Section 6 along three axes: the univariate setting (Appendix H.1), additional backbones (Appendix H.2), and the cost of the scoring run, which can be shortened (Appendix H.3) and moved to a cheaper backbone (Appendix H.4), with the full end-to-end accounting in Appendix H.5. In the univariate setting and on the additional backbones, TWIN is compared against Random and the baselines that perform best on average in Table 1: Avg Loss and EL2N.

H.1 Univariate Forecasting

TWIN is not restricted to multivariate forecasting: what the score exploits is the stride-1 sliding-window protocol, not the multivariate structure of the series, so the algorithm applies unchanged in the univariate setting. We verify this directly by training iTransformer under the standard univariate

Table 16: Univariate forecasting: Avg Relative MSE (%), iTransformer, four ETT datasets, four horizons, five seeds.

Method	PR=10%	PR=20%	PR=30%	IQR
Random	+0.19	+0.08	+0.22	—
EL2N	−0.85	−0.27	−0.20	−1.06
Avg Loss	−3.12	−3.38	−3.54	−3.21
TWIN (Ours)	**−4.24**	**−4.79**	**−4.52**	**−3.56**

Table 17: PatchTST backbone: Avg Relative MSE (%), six benchmarks, four horizons, five seeds.

Method	PR=10%	PR=20%	PR=30%	IQR
Random	+0.08	+0.17	+0.09	—
EL2N	−0.62	−0.75	−0.79	−0.60
Avg Loss	−0.90	−1.01	−1.11	−0.99
TWIN (Ours)	**−1.29**	**−1.33**	**−1.31**	**−1.16**

Table 18: DLinear backbone: Avg Relative MSE (%), six benchmarks, four horizons, five seeds.

Method	PR=10%	PR=20%	PR=30%	IQR
Random	+0.10	+0.38	+0.56	—
EL2N	−3.49	−3.57	−3.67	−3.45
Avg Loss	−3.46	−3.63	−3.64	−3.45
TWIN (Ours)	**−3.91**	**−4.12**	**−4.22**	**−3.90**

Table 19: MambaTS backbone: Avg Relative MSE (%), three datasets (Weather, ETTh2, ETTm2), four horizons, five seeds.

Method	PR=10%	PR=20%	PR=30%	IQR
Random	+0.02	−0.45	−0.52	—
EL2N	−1.54	−1.93	−1.27	−1.45
Avg Loss	−1.55	−1.78	−2.17	−1.38
TWIN (Ours)	**−2.05**	**−2.31**	**−2.55**	**−2.14**

protocol on the four ETT datasets $\times$ four horizons $\times$ five seeds (Table 16). TWIN attains the largest MSE reduction at every pruning rate in the univariate setting, consistent with the mechanism above.

H.2 Additional Backbones: PatchTST, DLinear, and MambaTS

Beyond iTransformer, we evaluate three further backbones, so that the four backbones span three design paradigms: the patch-based Transformer **PatchTST** [11], the linear model **DLinear** [21], and the selective state-space model **MambaTS** [1], whose linear-complexity state transitions differ substantially from iTransformer. TWIN attains the best average Relative MSE in every column on all three backbones (Tables 17–19).

H.3 Scoring-Window Robustness

TWIN's scores are computed on the training-dynamics matrix $\mathbf{M}$, whose construction requires E epochs of training. We sweep the scoring window $E^\star = \alpha E$ with $\alpha \in \{20\%, 40\%, 60\%, 100\%\}$ and recompute the multi-scale Wasserstein score on the truncated dynamics matrix $\mathbf{M}_{:,1:E^\star}$. Table 20 shows that the average Relative MSE changes little with α: in every column, each truncated window differs from $\alpha = 100\%$ by at most 0.20 percentage points, with no consistent direction, and Holm-corrected Wilcoxon tests detect no difference ($p \geq 0.92$). Scoring on the first 20% of the epochs thus gives pruning results comparable to scoring on the full run.

Table 20: Scoring-window robustness: TWIN under the $E^\star = \alpha E$ scoring window. Avg Relative MSE (%) across six benchmarks, four horizons, and five seeds. Bold marks the best α within each column. The penultimate column reports the Holm-corrected two-sided Wilcoxon signed-rank p-value against the default $\alpha = 100\%$, computed on the 96 paired (dataset, horizon, pruning-rate) cells.

α	Avg Relative MSE (%) at Pruning Rate				Holm p vs $\alpha = 100\%$	Scoring Cost
	PR=10%	PR=20%	PR=30%	IQR		
$\alpha = 20\%$	$-$**1.46**	$-$**1.26**	$-$**0.88**	-1.41	1.00	0.20×
$\alpha = 40\%$	-1.41	-1.16	-0.79	-1.36	0.92	0.40×
$\alpha = 60\%$	-1.39	-1.23	-0.78	-1.44	1.00	0.60×
$\alpha = 100\%$	-1.41	-1.14	-0.78	$-$**1.56**	— (ref)	1.00×

H.4 Cross-Backbone Score Transfer

If TWIN captured an idiosyncrasy of the scoring backbone, its selection would not transfer across backbones. We therefore compute the training-dynamics matrix and the TWIN score with the cheapest backbone available (DLinear), prune with it, and train the heavier models on that retained set. In the same spirit as the scoring-window sweep of Appendix H.3, we additionally repeat the transfer using only the first 20% of the DLinear epochs. Table 4 in Section 6.5 reports Relative MSE (%) against the unpruned baseline for both target models, averaged over six benchmarks $\times$ four horizons $\times$ five seeds.

Across the eight settings, the 20%-epoch DLinear score differs from native scoring by at most 0.13, with a mean absolute difference of 0.06. Two consequences follow. First, the selection depends little on the scoring backbone: the neighbor inconsistency that TWIN measures is largely reproduced by a much simpler forecaster, which supports the backbone-agnostic claim. Second, the method becomes substantially cheaper in practice: the scoring run can be carried out on a linear model and the resulting subset reused for an arbitrarily heavy target model, so the one-off cost of collecting the training-dynamics matrix does not scale with the size of the model one actually wants to train. The configuration we recommend in practice is the cheapest one: DLinear with 20% of the epochs (end-to-end cost accounting: Appendix H.5).

H.5 End-to-End Cost Accounting

The retraining-time convention of Section 6.3 excludes the one-off scoring run. This subsection charges it in full. Two independent findings reduce the scoring run to a small fraction of a single training run: the score does not depend on the backbone that produced it, and it does not need the full training run (both verified in Table 4), so the recommended configuration computes the training-dynamics matrix with DLinear for 20% of its epochs.

Scoring cost. All timings are measured on Electricity with an Intel Xeon Silver 4214R CPU and a single RTX 3090, and TWIN's scoring run uses the GPU. The 20%-epoch scoring time is estimated as 20% of the 100%-epoch time.

Table 21: One-off scoring cost on Electricity.

Method	Cost
TWIN, DLinear, 100% epochs	70.5 s
TWIN, DLinear, 20% epochs	**14.1** s

The recommended configuration takes 14.1 s (Table 21), which amounts to 3.1% of one iTransformer training run.

End-to-end pipeline. On iTransformer (Table 22), training falls from 457.2 s to 251.8 s, a reduction of 44.9%. Adding the 14.1 s scoring run, the whole pipeline costs 265.9 s against 457.2 s, a saving of 41.8%. The scoring run is charged in full to a single training run here, and the pipeline is still

Table 22: End-to-end cost (scoring + training) on Electricity.

Pipeline	Scoring	Training	Total
iTransformer, unpruned	—	457.2 s	457.2 s
iTransformer, TWIN	14.1 s	251.8 s	**265.9** s

cheaper than training once on the full dataset. This accounting is also the pessimistic one: scoring and pruning are performed once and are independent of what is trained afterward, so the same retained set is reused across hyperparameter search, multiple seeds, and multiple target backbones. The 14.1 s is paid once and amortized over all of them, while the per-run saving recurs.

I Limitations and Future Work (Extended)

I.1 Limitations

Pre-memorization regime. TWIN's lower bound (Theorem 2) requires the pre-/partial-memorization regime: once a model fully memorizes the training set, per-sample loss differences vanish at zero training loss, and every loss-dynamics-based pruner (EL2N, TDDS, Avg Loss, and TWIN alike) could lose ranking information. Standard MTSF training pipelines work against full memorization through (a) dropout, which makes it harder for the network to fit individual contaminated samples, and (b) early stopping on validation loss, the default protocol in Time-Series-Library used in all our experiments. In the scoring-window sweep (Appendix H.3, $\alpha \in \{20\%, 40\%, 60\%, 100\%\}$), scores computed from the first 20% of training give pruning results comparable to full-run scores (no difference detected, $p \geq 0.92$), consistent with the pruning signal forming early, before memorization could set in. Outside this regime (e.g., under prolonged training without early stopping), the score could become uninformative and TWIN should not be deployed.

Offline scope. TWIN is a one-shot offline pruning method: it consumes the full training-dynamics matrix $\mathbf{M} \in \mathbb{R}^{S \times E}$ after a single training pass and produces a single pruning decision over a fixed training set. Streaming and continual settings fall outside this scope. In such settings the training set accumulates over time, S grows unboundedly, and pruning decisions must be made before training completes. We discuss the corresponding extension in the Future Work subsection below.

I.2 Future Work

Online and streaming settings. Extending TWIN to the streaming/continual settings noted as a limitation above requires incremental updates to the multi-scale Wasserstein score: sorted-trajectory maintenance with quantile insertion, plus synchronization across the dilation set $\{d_k\}_{k=1}^{K}$ as new samples shift the relative neighbor offsets. We do not develop this online formulation here, and leave it as a natural extension.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a thorough limitation discussion in Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Section 4.1 and Appendix A state the assumptions, and Appendix B gives the complete proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all information needed to reproduce the main experimental results. Appendices D.1 and E.1 describe the experimental setup.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets and backbone codebases used are publicly available, and we will release our code for synthetic audit experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Appendix E.1 describes the training and test setup, and Appendix D.1 gives the full protocol of the synthetic audit.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report two-sided Wilcoxon signed-rank tests with Holm correction, computed on paired (dataset, horizon, pruning-rate) cell means over five seeds, in Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.1 lists the GPUs used.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We confirm the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper presents foundational efficiency research with no direct path to negative societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the creators of all datasets, codebases, and baselines and give URLs for the code we use (Section 6.1, Appendix E.1).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We used LLMs mainly to improve the writing, such as polishing the language (grammar and expression) and improving the logical flow between sentences.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.