BOUNDARYWARP: REVEALING AND EXPLOITING PATCH-END VULNERABILITIES IN TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

State-of-the-art time-series foundation models divide input sequences into fixed, non-overlapping patches before forecasting, yet the robustness consequences of this design are not well understood. We identify a recurring tokenizer-relative patch-end vulnerability in which model sensitivity concentrates at the final position of a patch. To distinguish this pattern from general recency effects, we shift tokenizer alignment and patch length and find that the sensitivity maximum moves with the active patch grid rather than remaining at a fixed temporal position. Internal-boundary, unpatched, recency-controlled, and content-matched analyses further support a tokenizer-relative explanation. We then introduce BoundaryWarp, an untargeted temporal deformation attack that exploits this structural sensitivity through bounded resampling while preserving temporal order, endpoints, and context length. Across Chronos-Bolt, Moirai, and TimesFM on four forecasting benchmarks, BoundaryWarp consistently degrades forecasts under sub-sample temporal constraints. With a maximum displacement of half a sampling interval, it increases median mean squared forecasting error by up to 12.76% relative to clean predictions. Our results show that fixed patch geometry is not only a computational design choice but also a robustness-relevant architectural property that can create a systematic adversarial attack surface in time-series foundation models.

1 INTRODUCTION

Time-series foundation models increasingly rely on patch tokenization to process long temporal contexts efficiently (Das et al., 2024; Woo et al., 2024). Building on the success of patch-based representations in time-series forecasting (Nie et al., 2023), models such as TimesFM, Moirai, and Chronos-Bolt group consecutive observations into fixed, non-overlapping patches before applying their forecasting backbone (Das et al., 2024; Woo et al., 2024; Ansari et al., 2024b). This substantially reduces sequence length and enables scalable modeling of long histories. At the same time, patching imposes a repeated segmentation grid on the input: observations that are adjacent in time may be encoded within the same token or separated across two tokens solely because of their position relative to a patch boundary. Fixed patching is therefore computationally useful, but it need not be representationally neutral.

This observation raises a largely unexplored robustness question. If the representation of a local temporal pattern depends on its alignment with the patch grid, model sensitivity may also vary systematically with that alignment. In particular, a recurring sensitivity at the same relative position within different patches would indicate a property of the tokenizer geometry rather than sensitivity to a particular absolute timestamp or proximity to the forecast origin. Existing work has studied patch-based forecasters under non-adversarial input irregularities (Qiu et al., 2026; Yu et al., 2026), developed adversarial attacks for time-series forecasting (Liu et al., 2023; Su et al., 2026), and evaluated attacks against forecasting systems that include patch-based representations (Feng et al., 2025; Liu et al., 2025; Zhang et al., 2025). However, in these studies, patching remains a property of the victim architecture rather than the hypothesized source of a structured vulnerability. It therefore remains unknown whether fixed patch tokenization itself organizes a recurring sensitivity pattern, whether such a pattern follows the active tokenizer grid, and whether it can be deliberately exploited.

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

We show that it does. Across pretrained patch-based time-series foundation models, sensitivity on clean, undeformed inputs repeatedly concentrates at the terminal position of the patch. Crucially, this pattern follows controlled changes to the tokenizer geometry. When we shift the patch boundaries over the same input sequence, the sensitivity peak shifts with them and remains located at the end of a patch. Likewise, changing the patch length relocates the peak to the terminal position of the new patch size. The same behavior recurs under the native tokenizer configurations of independently pretrained models. Unpatched controls, recency-adjusted analyses, internal-boundary comparisons, and content-matched alignment experiments further distinguish this effect from artifacts of phase aggregation, absolute input position, increasing proximity to the forecast origin, and local signal content. Thus, we identify a *tokenizer-relative patch-end sensitivity* organized by fixed temporal segmentation and rule out alternative explanations.

We evaluate the exploitability of this structural sensitivity with BOUNDARYWARP, an untargeted temporal deformation attack under an offline robustness protocol. Here, offline means that candidate deformations are evaluated retrospectively against the realized future continuation. BoundaryWarp shifts the temporal coordinates of sampling points depending on their relative position within their patch while preserving temporal order, the covered input interval, context length, and a strict displacement budget (Fig. 1). This changes how local signal patterns align with the fixed patch grid without directly perturbing the observed values. A smooth positional prior derived from the victim model's native patch transitions biases the optimization toward regions associated with the identified sensitivity. To isolate the contribution of this patch-aware prior, we compare BoundaryWarp with a budget-matched random deformation and an otherwise identical boundary-unaware optimizer. These matched controls distinguish the effects of temporal deformation and adversarial optimization and test whether explicit knowledge of the patch grid provides additional leverage.

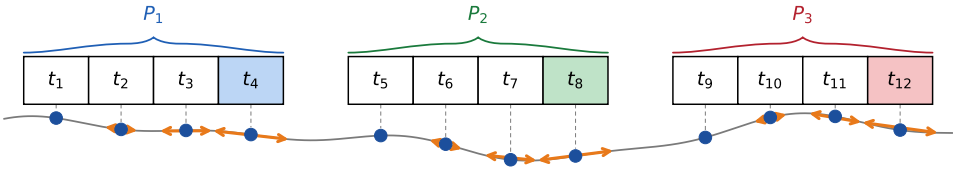

Figure 1: BoundaryWarp distorts the signal by shifting sampling points in time, with larger shifts near patch ends while preserving temporal order.

We evaluate BoundaryWarp on Chronos-Bolt, Moirai, and TimesFM (Ansari et al., 2024b; Woo et al., 2024; Das et al., 2024) across four forecasting benchmarks (Zhou et al., 2021; Trindade, 2015; Wu et al., 2021). At the primary sub-sample budget of $\epsilon = 0.5$, BoundaryWarp increases median squared forecasting error (MSE) by 4.71% to 12.76% and yields the strongest aggregate degradation for every evaluated victim model. The attack remains effective across temporal budgets and under restricted output-only access. These results identify fixed patch geometry as a robustness-relevant architectural factor: tokenization choices shape where model sensitivity concentrates and should therefore be evaluated not only for computational efficiency and forecasting accuracy but also for the attack surfaces they create.

Contributions. Our study has three contributions:

- **Discovery.** We identify a recurring patch-end sensitivity in time-series foundation models with fixed, non-overlapping tokenization and show that it emerges independently of the proposed attack across multiple pretrained models and forecasting datasets.
- **Mechanistic evidence.** We show that the sensitivity follows the active tokenizer geometry rather than absolute input position, proximity to the forecast origin, or local signal content through controlled tokenizer-alignment and patch-length interventions, together with unpatched controls, recency adjustment, internal-boundary comparisons, and content-matched analyses.
- **Exploitation.** We introduce BoundaryWarp, a patch-aware temporal-deformation adversarial attack, and demonstrate that small, order-preserving temporal deformations can exploit this structured sensitivity across three pretrained time-series foundation models and four forecasting benchmarks, including under restricted output-only access.

2 Background and Related Work

Temporal patching in time-series foundation models. Fixed patching has become a common design choice in modern time-series forecasting. PatchTST established patching as a way to reduce sequence length while preserving useful local temporal structure (Nie et al., 2023). This principle is now used by time-series foundation models such as TimesFM, Moirai, and Chronos-Bolt, which construct tokens from fixed groups of consecutive observations before applying their backbones (Das et al., 2024; Woo et al., 2024; Ansari et al., 2024b). Other models use alternative representations, including scalar tokenization in Chronos (Ansari et al., 2024a) and wavelet-based tokenization in WaveToken (Masserano et al., 2025). DeformableTST further showed that patch partitions can shape effective receptive fields and that Transformer forecasters may become over-reliant on patching (Luo & Wang, 2024), while controlled analyses of time-series foundation models demonstrate systematic temporal and geometric biases from design choices including patch size (Yu et al., 2026).

Recent work has consequently explored alternatives to fixed segmentation. Approaches include entropy-guided boundary placement in EntroPE, adaptive temporal granularity in TimeMosaic, and elastic time-frequency tokenization in PATK (Abeywickrama et al., 2025; Ding et al., 2026; Liao et al., 2026). Controlled comparisons indicate that adaptive segmentation does not uniformly outperform carefully configured fixed patches (Zucchi et al., 2026). These studies establish that patch size, boundary placement, and segmentation strategy can affect representation, forecasting behavior, and model inductive biases. However, the robustness consequences of fixed patch geometry remain much less understood, including how patch boundaries shape input sensitivity, whether such effects recur across tokenizer-relative positions, and whether they create exploitable weaknesses.

Robustness and adversarial attacks in forecasting. Robustness and adversarial characteristics of time-series forecasting models have been studied from complementary perspectives. First, robustness-oriented work on patch-based forecasting has considered degraded inputs, i.e. missing values, anomalies, distribution shifts, and noise (Qiu et al., 2026). These studies characterize how patch-based representations behave under corrupted or irregular observations. Second, adversarial attacks have been developed for time-series forecasting. Sparse indirect attacks manipulate selected input series to alter forecasts (Liu et al., 2023), while temporally unified perturbations enforce consistent modifications across overlapping forecasting windows (Su et al., 2026). These studies establish that forecasting models can exhibit systematic adversarial vulnerabilities under targeted input manipulation. Third, recent work has extended robustness analyses to modern forecasting systems. Frequency-aware attacks have been evaluated on models including PatchTST (Feng et al., 2025), gradient-free and black-box attacks have been studied for LLM-based forecasting systems (Liu et al., 2025), and systematic white- and black-box evaluations of time-series foundation models have identified position-dependent and architecture-dependent vulnerabilities (Zhang et al., 2025).

Prior work shows that forecasting models can be vulnerable to both corrupted and adversarial inputs, including in patch-based and foundation-model settings. However, the role of fixed patch geometry in structuring these vulnerabilities remains unexplored. In particular, it is unclear whether sensitivity is systematically organized by patch-relative position, boundary placement, tokenizer alignment, or patch length. This gap is highly relevant, as fixed patching is a common architectural component of time-series foundation models and may introduce structured vulnerabilities that are not captured by robustness analyses defined only over absolute input positions or perturbation magnitude.

3 Method

First, we characterize patch-relative sensitivity and test whether its location follows tokenizer geometry through controlled alignment and patch-length interventions. Second, we evaluate alternative explanations using controls for phase aggregation, recency, and local signal content. Third, we assess exploitability using BoundaryWarp.

3.1 Patch-Relative Sensitivity and Tokenizer Geometry

Let $\mathbf{x} = (x_0, \ldots, x_{C-1}) \in \mathbb{R}^C$ denote a forecasting context, $\mathbf{y} \in \mathbb{R}^H$ its observed continuation, and f_m a frozen forecasting model. For a model with non-overlapping patch length P_m and tokenizer

alignment o_m, we define the *patch phase* of input position t as

$$\phi_m(t) = (t + o_m) \bmod P_m, \tag{1}$$

where $\phi_m(t) = P_m - 1$ is the terminal position of a patch. Patch phase is distinct from absolute input position and distance to the forecast origin, which allows sensitivity to be analyzed relative to the repeated tokenizer grid.

To assess whether the repeated patch grid is reflected in the sensitivity of the unmodified forecasting model independently of any attack, we use gradient-based input sensitivity (Simonyan et al., 2014; Ismail et al., 2020) and measure, for each clean input position, how strongly the forecasting loss responds to a local change in that observation:

$$s_t = \left| \frac{\partial L_{\mathrm{rel}}\left(f_m(\mathbf{x}), \mathbf{y}\right)}{\partial x_t} \right|, \tag{2}$$

where L_{rel} is a window-normalized forecasting loss that balances differences in scale and clean forecasting error. No temporal deformation, patch prior, or attack optimization is used in this step.

To test whether sensitivity recurs at a particular position within different patches, we group observations by patch phase. For $\mathcal{I}_\phi = \{t : \phi_m(t) = \phi\}$,

$$S_\phi = \frac{1}{|\mathcal{I}_\phi|} \sum_{t \in \mathcal{I}_\phi} s_t. \tag{3}$$

A repeated maximum at $\phi = P_m - 1$ therefore identifies a recurring patch-end concentration on clean inputs. Whether this localization is tied to tokenizer geometry rather than fixed source positions is tested through controlled tokenizer interventions.

Tokenizer interventions. The alignment intervention tests whether the location of the sensitivity pattern is determined by the tokenizer grid itself. We shift the tokenizer relative to the same source sequence while holding the observed values, their order, effective context length, normalization basis, input-envelope length, and number of tokens fixed. For an alignment shift o, the source phase that coincides with the terminal tokenizer phase is

$$\phi^*(o) = (P - 1 - o) \bmod P. \tag{4}$$

This yields a direct prediction: if tokenizer geometry determines the sensitivity pattern, its maximum should move with the grid according to $\phi^*(o)$. If sensitivity is instead tied to fixed source positions, the maximum should remain at the same location.

We independently vary the active patch length while keeping pretrained model parameters, context length, and prediction horizon fixed. Under the same hypothesis, the maximum should occur at the corresponding terminal phase $P-1$ for every supported patch length rather than at one fixed absolute position. As changing P can also alter token count and local receptive structure, clean forecasting performance is evaluated for each configuration. We additionally replicate phase localization under the native tokenizer configurations of all evaluated patch-based foundation models.

3.2 Controls for Alternative Explanations

To assess whether the tokenizer-relative pattern can be explained by potential confounds, we test phase aggregation, recency, and local signal content using three complementary controls.

First, we test whether phase aggregation itself can produce a terminal maximum. For unpatched models, input positions are grouped *post hoc* into pseudo-patches of length P without affecting the model or its forecast. A systematic maximum at $P - 1$ would therefore indicate an aggregation artifact rather than an effect associated with tokenizer geometry.

Second, we test whether the effect is driven by recency. We fit a monotonically non-decreasing isotonic regression (Barlow & Brunk, 1972) to log-sensitivity as a function of absolute input position and construct phase profiles from the residuals. We separately compare adjacent positions on opposite sides of internal patch boundaries. The latter is a conservative test because the post-boundary observation is one step more recent; higher sensitivity at the preceding terminal position therefore cannot be explained by a simple monotonic recency effect.

Third, the content-matched crossover isolates local signal content from tokenizer placement. Unlike the alignment intervention, which tests whether the *location of the overall sensitivity maximum* moves with the grid, the crossover compares the *same source positions* across different alignments. Their observed value, local neighborhood, and forecasting target remain unchanged; only their phase relative to the active tokenizer changes. If identical source positions become more sensitive when placed at the terminal patch phase, local signal content alone cannot explain the effect.

3.3 BOUNDARYWARP

To assess the exploitability of the tokenizer-relative sensitivity, we apply BoundaryWarp under the oracle-based offline robustness protocol. BoundaryWarp keeps the patch partition fixed in sequence-index space and deforms the temporal sampling coordinates of the original signal within a displacement budget. This changes which parts of the underlying signal occupy the fixed patch-relative input positions while preserving temporal order, context length, and the covered input interval.

Bounded temporal deformation. Let $\mathbf{t} = (0, \ldots, C-1)$ denote the original, regularly spaced sampling coordinates and let $\boldsymbol{\tau} = (\tau_0, \ldots, \tau_{C-1})$ denote their warped counterparts. BoundaryWarp restricts the deformation to

$$\mathcal{D}_\epsilon = \left\{ \boldsymbol{\tau} \;\middle|\; \begin{array}{l} \tau_0 = 0, \quad \tau_{C-1} = C-1, \\ 0 \le \tau_t \le C-1, \quad \tau_{t+1} > \tau_t, \\ \|\boldsymbol{\tau} - \mathbf{t}\|_\infty \le \epsilon \end{array} \right\}. \tag{5}$$

These constraints define the admissible temporal deformations. The first condition preserves the observed time interval, the second preserves temporal order and prevents extrapolation, and the final condition limits every sampling coordinate to a displacement of at most ϵ. Thus, $\epsilon < 1$ corresponds to a sub-sample temporal deformation rather than a shift by a complete observation interval.

The attacked context $\widetilde{\mathbf{x}}$ is obtained by sampling the original signal at the warped coordinates $\boldsymbol{\tau}$ using linear interpolation. Instead of optimizing each warped coordinate independently, we optimize the positive gaps between neighboring coordinates. This guarantees that their order cannot be reversed. We then normalize the resulting coordinates so that the first and last positions remain fixed and scale the displacement if necessary to satisfy the L_∞ budget.

Patch-aware deformation. The feasible deformation above does not yet use information about the victim model's patch structure. BoundaryWarp introduces this information through a smooth positional prior. Let $\mathcal{B}_m = \{t : \phi_m(t) = 0\}$ denote the positions at which new patches begin. We assign each input position the weight

$$w_t = \exp\left(-\frac{d(t, \mathcal{B}_m)^2}{2(\rho P_m)^2} \right), \qquad \rho = 0.125, \tag{6}$$

where $d(t, \mathcal{B}_m)$ is the distance to the nearest patch transition. Weights are largest near patch boundaries and decrease with distance, making nearby temporal intervals more responsive to optimization. Because BoundaryWarp optimizes gaps between neighboring coordinates, we average the weights of the two adjacent positions for each interval. Since the optimized gaps span adjacent coordinates, weighting patch starts also emphasizes intervals connecting terminal phase $P_m - 1$ to phase 0.

Attack objective. Using the realized continuation $\mathbf{y}$ retrospectively, BoundaryWarp searches within the admissible deformation set for a smooth warp that maximizes forecasting error:

$$\max_{\boldsymbol{\tau} \in \mathcal{D}_\epsilon} \; L_{\mathrm{rel}}\left(f_m(\widetilde{\mathbf{x}}), \mathbf{y}\right) - \lambda R(\boldsymbol{\tau} - \mathbf{t}), \tag{7}$$

where L_{rel} is the normalized forecasting loss introduced above and R penalizes abrupt changes between neighboring displacements. The first term therefore encourages deformations that increase forecasting error, while the second favors smooth temporal warps. We use $\lambda = 0.1$ and optimize the deformation for ten Adam updates (Kingma & Ba, 2015).

Matched controls. To separate the contributions of temporal deformation, adversarial optimization, and explicit patch information, we compare three variants. Random Warp uses the same feasible deformation and displacement budget but does not optimize forecasting error. It therefore measures the effect of temporal resampling alone. Optimized Warp uses the same objective, parameterization, initialization, optimizer, update count, and candidate selection as BoundaryWarp, but removes the patch prior by assigning uniform interval weights. Comparing Optimized Warp with Random Warp isolates the effect of adversarial optimization, whereas comparing BoundaryWarp with Optimized Warp isolates the additional contribution of native patch geometry.

Restricted output-only access. Finally, we test whether exploiting the identified structure requires exact gradients. We replace gradient-based optimization with two-sided simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992), which estimates search directions from model outputs. Boundary-weighted SPSA uses the same boundary-localized Gaussian prior family as BoundaryWarp, while Uniform SPSA uses uniform search directions under identical query budgets and update procedures. This comparison tests whether patch information remains useful when model access is restricted to forecast outputs. The retrospective objective and deformation constraints remain unchanged, while access to model internals is removed.

3.4 Experimental Protocol

We conduct a comprehensive evaluation across representative time-series foundation models with native non-overlapping patch tokenization: Chronos-Bolt Tiny (Ansari et al., 2024b), Moirai 1.1-R-Small (Woo et al., 2024), and TimesFM 2.5-200M (Das et al., 2024). Evaluation uses ETTh1 and ETTm1 (Zhou et al., 2021), Electricity (Trindade, 2015), and Weather (Wu et al., 2021), with a common context length $C = 512$ and forecasting horizon $H = 96$. The main attack study uses $\epsilon \in \{0.25, 0.5, 1.0\}$, with $\epsilon = 0.5$ as the primary sub-sample budget. Three seeds and 25 sampled test windows per model–dataset–seed combination yield 8,100 attacked instances across the three attack variants and budgets. Attack strength is measured as the median per-window relative MSE degradation $D = 100(\mathrm{MSE}_{\mathrm{attack}}/\mathrm{MSE}_{\mathrm{clean}} - 1)$. Direct attack comparisons use paired differences in D, while attack success denotes the fraction of windows with $D > 0$.

We perform the alignment intervention with Chronos-Bolt because its mask-aware interface is the only one among the evaluated models that permits controlled shifts of the tokenizer alignment while preserving source composition and token count. We perform the patch-length intervention with Moirai because its pretrained checkpoint is the only one among the evaluated models that natively supports multiple patch lengths while keeping all learned parameters fixed, with $P \in \{8, 16, 32, 64\}$. Native-grid analyses are performed for all three foundation models across all four datasets. Unpatched negative controls comprise a learned ridge forecaster and a Seasonal Naive forecaster, to which artificial periods are assigned only after prediction. Restricted-access experiments are conducted for all three foundation models with query budgets of 8, 16, and 32.

Forecasting windows originating from the same time series are not treated as independent observations. All primary paired comparisons therefore use a series-cluster moving-block bootstrap with 10,000 resamples (Künsch, 1989), and multiplicity is controlled with Holm correction within predefined families of related claims (Holm, 1979). We additionally audit every generated deformation for the L_∞ budget, fixed endpoints, valid coordinate range, strict monotonicity, coordinate uniqueness, and exact reconstruction of the context used for inference. Additional implementation details, statistical procedures, model-specific configurations, preprocessing steps, and hyperparameters are provided in the appendix.

4 Results

4.1 Patch-Relative Sensitivity Concentrates at Patch Ends and Tracks Tokenizer Geometry

Across native and controlled tokenizer configurations, sensitivity consistently tracked the terminal patch position: it was the native-grid maximum in 11 of 12 model–dataset settings and followed every tested alignment shift and patch-length change. The median cluster-bootstrap argmax rate for $P - 1$ was 0.9965 (range 0.6978–1.0000).

Under the alignment intervention, the sensitivity maximum moved to the phase predicted by $\phi^*(o) = (P-1-o) \bmod P$ in all 28 dataset–offset settings (Fig. 2A). The maximum therefore followed the shifted tokenizer grid rather than remaining at a fixed source position.

The patch-length intervention showed the same tokenizer-relative pattern. The equal-cluster sensitivity maximum occurred at $P-1$ in all 16 dataset–patch-length settings (Fig. 2B). Clean forecasting performance varied: relative to $P = 16$, the median paired MSE changed by $+4.0\%$ for $P = 8$, -37.5% for $P = 32$, and -48.7% for $P = 64$. The terminal-phase maximum therefore persisted despite substantial variation in clean forecasting performance.

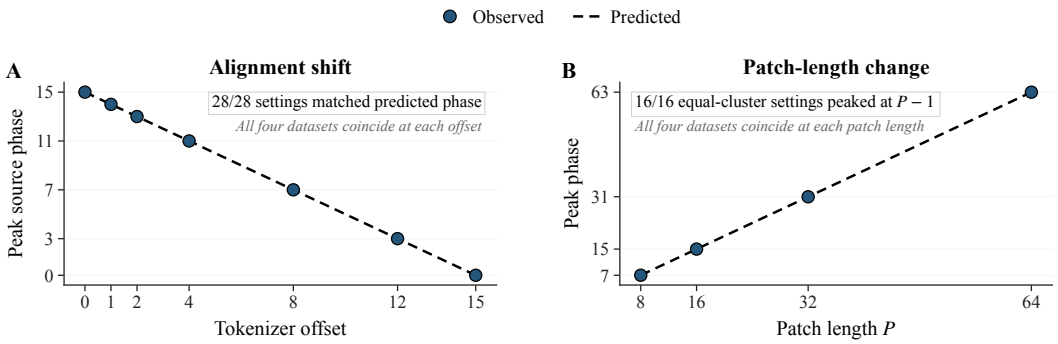

Figure 2: Sensitivity tracks the active tokenizer geometry under controlled interventions. (A) The alignment intervention shifts the observed sensitivity maximum with the predicted phase $\phi^*(o) = (P-1-o) \bmod P$ for Chronos-Bolt. (B) The patch-length intervention shifts the sensitivity maximum to the corresponding terminal phase $P-1$ for Moirai. Observed peak locations coincide across all four datasets in both interventions.

Taken together, across the native configurations and both tokenizer interventions, the descriptive sensitivity maximum aligned with the terminal tokenizer phase $P-1$ in every evaluated configuration. After equal weighting of the originating time-series clusters, $P-1$ remained the maximum in 89.6% of configurations. The median bootstrap argmax rates were 0.989 for Chronos-Bolt, 1.000 for Moirai, and 0.898 for TimesFM. These results show that elevated sensitivity is systematically organized by the active tokenizer geometry.

4.2 Tokenizer-Relative Sensitivity Persists Under Controls for Alternative Explanations

The tokenizer-relative sensitivity pattern persisted after controlling for the main potential confounds considered here. Unpatched controls rarely showed a terminal maximum, while recency-adjusted, internal-boundary, and content-matched analyses consistently favored the terminal patch position.

Phase aggregation. The unpatched controls did not reproduce the terminal-phase concentration. Only one of 20 control settings supported a maximum at the predefined terminal phase. The median cluster-bootstrap argmax rates were 0.004 for the learned unpatched forecaster and 0.000 for Seasonal Naive, compared with 0.898–1.000 for the patch-based victim models. Phase aggregation alone therefore does not reproduce the observed pattern.

Recency. After recency detrending, the terminal phase remained more sensitive in all 12 model–dataset settings. The median terminal–versus–other–phase contrast was 0.565 log units (range 0.231–1.366), corresponding to a $1.76\times$ multiplicative sensitivity difference at the median setting. All confidence intervals excluded zero and all tests remained significant after Holm correction, with adjusted $p \leq 0.0136$. The complementary internal-boundary comparison yielded the same conclusion locally. All 12 model–dataset contrasts were positive, with a median difference of 0.647 log units (range 0.422–1.048), corresponding to approximately $1.91\times$ greater sensitivity immediately before than immediately after an internal patch boundary. All confidence intervals excluded zero and all tests remained significant after Holm correction ($p_{\mathrm{adj}} \leq 0.0272$), despite the post-boundary observation being closer to the forecast origin.

Local signal content. The content-matched Chronos-Bolt crossover isolated tokenizer placement from local signal content. Across all four datasets, the same source positions were more sensitive when aligned with the terminal patch phase. The median contrast was 0.468 log units (range 0.247–1.076), corresponding to a $1.60\times$ multiplicative increase in sensitivity. All four confidence intervals excluded zero and all comparisons remained significant after Holm correction ($p_{\mathrm{adj}} = 0.0012$).

4.3 BOUNDARYWARP EXPLOITS TOKENIZER-RELATIVE SENSITIVITY

At the prespecified primary budget $\epsilon = 0.5$, BoundaryWarp increased median relative MSE by 4.71% to 12.76% across all three victim models and produced the largest aggregate degradation for every model (Table 1). This demonstrates that the identified tokenizer-relative sensitivity can be exploited through bounded temporal deformation.

Table 1: Attack effectiveness at the primary temporal budget $\epsilon = 0.5$, aggregated across datasets. Attack columns report median relative MSE degradation in percent with 95% series-cluster moving-block bootstrap confidence intervals. Δ BW–Opt. reports the mean paired BoundaryWarp-minus-Optimized-Warp difference in percentage points with its 95% confidence interval.

Model	Random	Optimized	BoundaryWarp	Δ BW–Opt.
Chronos-Bolt	0.42	6.18	**7.70**	2.31
	$[-0.10, 1.07]$	$[4.60, 8.48]$	$[5.08, 10.15]$	$[1.71, 3.12]$
Moirai	0.15	3.30	**4.71**	1.25
	$[-0.27, 0.96]$	$[2.66, 4.67]$	$[3.34, 6.60]$	$[0.70, 1.85]$
TimesFM	1.78	10.77	**12.76**	1.65
	$[0.30, 4.28]$	$[6.97, 14.50]$	$[9.56, 17.75]$	$[-1.30, 4.13]$

The matched controls clarify where this effect originates. Both BoundaryWarp and the boundary-unaware Optimized Warp exceeded Random Warp in all 36 model–dataset–budget settings after Holm correction, showing that the effect cannot be attributed to temporal resampling alone. Explicit patch information provided a further, but more heterogeneous, benefit: Under the paired, series-balanced comparison, BoundaryWarp exceeded Optimized Warp in 34 of 36 settings, with 20 remaining significant after Holm correction. Thus, adversarial temporal optimization drives the main degradation, while knowledge of the tokenizer geometry provides recurring additional leverage rather than a universal advantage.

The attack surface persisted beyond the primary white-box setting. Across the full budget sweep, BoundaryWarp achieved the highest aggregate median degradation in all nine model–budget combinations, with degradation increasing monotonically as the temporal budget increased. Under restricted output-only access, boundary-weighted SPSA remained effective across all three victim models. At 32 queries, median relative MSE degradation reached 7.20% for Chronos-Bolt, 4.86% for Moirai, and 11.96% for TimesFM. Aggregated across all three models, attack success reached 90.0%, and all 12 model–dataset effectiveness tests remained significant after Holm correction ($p_{\mathrm{adj}} \leq 0.0112$). Boundary-weighted SPSA did not significantly outperform Uniform SPSA in any of the 36 model–dataset–budget comparisons after multiplicity correction, indicating that the vulnerability remains accessible without gradients while the patch prior provides no statistically reliable advantage under restricted query access.

Finally, the observed degradation was not specific to the relative MSE metric and did not rely on invalid deformations. At $\epsilon = 0.5$, BoundaryWarp increased normalized root mean squared error (NRMSE) in all 12 model–dataset settings, and all 8,100 generated attacks satisfied the specified displacement, endpoint, range, uniqueness, and monotonicity constraints.

Complete per-setting results, confidence intervals, statistical tests, and additional validity analyses are reported in the appendix.

5 DISCUSSION AND LIMITATIONS

We show that fixed temporal patching creates a coordinate system in which model sensitivity is systematically structured relative to the patch grid. Observations that are close in absolute time

can occupy different representational roles solely because of their position within this grid. The tokenizer interventions and complementary controls show that the sensitivity pattern follows the active tokenizer geometry rather than fixed source coordinates, monotonic recency, or local signal content. Fixed patching should therefore be viewed not only as an efficiency and representation choice, but also as an architectural factor that shapes where sensitivity concentrates.

This distinction matters because clean forecasting performance and robustness geometry need not vary together. In the Moirai patch-length intervention, larger patch sizes substantially reduced clean forecasting error, yet the sensitivity maximum continued to track the terminal patch phase. Selecting a tokenizer configuration on the basis of predictive accuracy alone can therefore leave structured sensitivity entirely unobserved. More broadly, standard robustness analyses based on perturbation magnitude, absolute input position, or distance to the forecast origin may miss vulnerabilities that are defined relative to the representation itself. For patch-based forecasters, patch phase, alignment, boundary placement, and patch length provide additional axes along which robustness should be assessed.

BoundaryWarp demonstrates that this structure can be exploited, while the results also separate the existence of the vulnerability from the efficiency of a particular exploitation strategy. In the gradient-based setting, explicit patch information improved over the otherwise matched temporal optimizer in most evaluated settings, although the magnitude and statistical strength of this gain varied across victim models. Under restricted output-only access, however, boundary-weighted SPSA did not significantly outperform Uniform SPSA after multiplicity correction. This separation suggests that tokenizer-relative sensitivity is a broader architectural phenomenon than the advantage of any one boundary-aware search procedure. It also motivates a different class of robustness interventions: rather than only constraining input perturbations, future defenses may alter the segmentation mechanism itself. Overlapping patches, randomized alignments, ensembles over multiple alignments, or adaptive segmentation are natural candidates. Recent work has already shown that temporal segmentation can be made input-dependent (Abeywickrama et al., 2025; Ding et al., 2026; Ankireddy et al., 2025), although whether such designs reduce patch-relative sensitivity remains an open empirical question.

The results should be interpreted in light of three limitations. First, BoundaryWarp is evaluated under an oracle-based offline protocol. This design isolates whether the identified tokenizer structure is exploitable at all, independently of the quality of a particular online surrogate objective. The reported effects should therefore be interpreted as a measure of exploitable model sensitivity rather than the success of a deployment-ready attacker. Second, the study focuses on pretrained models with fixed, non-overlapping temporal patches because this architecture provides the controlled setting needed to test the patch-relative hypothesis directly. Whether analogous sensitivity structures arise under overlapping, adaptive, scalar, or frequency-domain tokenization is a separate empirical question. Third, the interventions show that the effect is tied to the active tokenizer geometry, but they do not reveal which internal model computations produce it. Identifying whether patch embeddings, positional information, attention, or later representations drive the effect requires more detailed internal analysis.

6 Conclusion

This study shows that fixed temporal patching does more than compress forecasting inputs. It imposes a repeated representational geometry that organizes model sensitivity and creates a structured attack surface. Across three pretrained time-series foundation models and four datasets, sensitivity tracked controlled changes to tokenizer alignment and patch length and remained concentrated near terminal patch positions under complementary controls. BoundaryWarp further showed that small, order-preserving temporal deformations can exploit this structure, increasing median relative MSE by 4.71%–12.76% at the primary sub-sample budget. Temporal tokenization should therefore be treated as a robustness-relevant architectural component rather than as a neutral preprocessing choice.

AI USE STATEMENT

Generative AI tools were used to assist with software implementation, including the drafting, editing, and debugging of portions of the experimental Python code. AI-assisted implementations were reviewed and tested by the authors against the specified methodology and experimental protocol. The authors take responsibility for the final implementation and all reported results.

ETHICS STATEMENT

This work studies adversarial robustness in time-series foundation models and therefore has potential dual-use implications. BoundaryWarp is evaluated under an oracle-based offline protocol intended to characterize model sensitivity rather than to demonstrate a deployment-ready attack. The study aims to identify robustness limitations of fixed patch tokenization and to inform the development and evaluation of more robust forecasting systems. No human subjects or personally identifiable data are involved in the experiments.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce the reported experiments throughout the main paper and appendix. Section 3.4 specifies the evaluated models and datasets, forecasting setup, attack budgets, sampling protocol, and statistical procedures. The Appendix provides detailed definitions of the forecasting objective and temporal-warp parameterization, model- and dataset-specific configurations, optimization and SPSA hyperparameters, sensitivity analyses, statistical inference procedures, and complete experimental results. We additionally report validity checks for the generated temporal deformations.

REFERENCES

Sachith Abeywickrama, Emadeldeen Eldele, Min Wu, Xiaoli Li, and Chau Yuen. Entropy Guided Dynamic Patch Segmentation for Time Series Transformers. arXiv preprint, 2025. Art. no. arXiv:2509.26157.

Sravan Kumar Ankireddy, Nikita Seleznev, Nam H. Nguyen, Yulun Wu, Senthil Kumar, Furong Huang, and C. Bayan Bruss. TimeSqueeze: Dynamic Patching for Efficient Time Series Forecasting. In *NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT2S)*, 2025.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*, 2024a.

Abdul Fatir Ansari, Caner Türkmen, Oleksandr Shchur, and Lorenzo Stella. Fast and Accurate Zero-Shot Forecasting with Chronos-Bolt and AutoGluon. AWS Machine Learning Blog, 2024b. Accessed: Jul. 21, 2026.

Richard E. Barlow and H. D. Brunk. The Isotonic Regression Problem and Its Dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972. doi: 10.1080/01621459.1972.10481216.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A Decoder-Only Foundation Model for Time-Series Forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. TimeMosaic: Temporal Heterogeneity Guided Time Series Forecasting via Adaptive Granularity Patch and Segment-Wise Decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 20790–20798, 2026.

Naifu Feng, Lixing Chen, Junhua Tang, Hua Ding, Jianhua Li, and Yang Bai. Fre-CW: Targeted Attack on Time Series Forecasting Using Frequency Domain Loss. In *Proceedings of the 21st International Conference on Intelligent Computing*, pp. 915–928, 2025. doi: 10.65286/icic.v21i1.32921.

Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Aya Abdelsalam Ismail, Mohamed Gunady, Héctor Corrada Bravo, and Soheil Feizi. Benchmarking Deep Learning Interpretability in Time Series Predictions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6441–6452. Curran Associates, Inc., 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

Hans R. Künsch. The Jackknife and the Bootstrap for General Stationary Observations. *The Annals of Statistics*, 17(3):1217–1241, 1989.

Huaizhang Liao, Zhixiong Yang, Jingyuan Xia, Yuheng Sun, Yue Zhang, Shengxi Li, and Yongxiang Liu. Dynamic Semantic Tokenization for Time Series via Elastic Sampling on Physics-aware Perception. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(28):23460–23468, 2026.

Fuqiang Liu, Sicong Jiang, Luis Miranda-Moreno, Seongjin Choi, and Lijun Sun. Adversarial Vulnerabilities in Large Language Models for Time Series Forecasting. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4672–4680, 2025.

Linbo Liu, Youngsuk Park, Trong Nghia Hoang, Hilaf Hasson, and Jun Huan. Robust Multivariate Time-Series Forecasting: Adversarial Attacks and Defense Mechanisms. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Donghao Luo and Xue Wang. DeformableTST: Transformer for Time Series Forecasting without Over-reliance on Patching. In *Advances in Neural Information Processing Systems*, volume 37, pp. 88003–88044. Curran Associates, Inc., 2024.

Luca Masserano, Abdul Fatir Ansari, Boran Han, Xiyuan Zhang, Christos Faloutsos, Michael W. Mahoney, Andrew Gordon Wilson, Youngsuk Park, Syama Sundar Rangapuram, Danielle C. Maddix, and Bernie Wang. Enhancing Foundation Models for Time Series Forecasting via Wavelet-Based Tokenization. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43248–43275, 2025.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Xiangfei Qiu, Xvyuan Liu, Tianen Shen, Xingjian Wu, Hanyin Cheng, Bin Yang, and Jilin Hu. SEER: Transformer-Based Robust Time Series Forecasting via Automated Patch Enhancement and Replacement. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations (ICLR 2014), Workshop Track Proceedings*, Banff, AB, Canada, 2014. International Conference on Learning Representations.

James C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. doi: 10.1109/9.119632.

Ruixian Su, Yukun Bao, and Xinze Zhang. Temporally Unified Adversarial Perturbations for Time Series Forecasting. arXiv preprint, 2026. Art. no. arXiv:2602.11940.

Artur Trindade. ElectricityLoadDiagrams20112014, 2015. URL https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014. Dataset; accessed: Aug. 6, 2026.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified Training of Universal Time Series Forecasting Transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430. Curran Associates, Inc., 2021.

Annan Yu, Danielle C. Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W. Mahoney, and Yuyang Wang. Understanding the Implicit Biases of Design Choices for Time Series Foundation Models. In *Proceedings of the 14th International Conference on Learning Representations (ICLR)*, 2026.

Jiawen Zhang, Zhenwei Zhang, Shun Zheng, Xumeng Wen, Jia Li, and Jiang Bian. Are Time-Series Foundation Models Deployment-Ready? A Systematic Study of Adversarial Robustness Across Domains. arXiv preprint, 2025. Art. no. arXiv:2505.19397.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 11106–11115, 2021.

Federico Zucchi, Yi Xie, Chao Zhang, Keyuan Luo, Thomas Lampert, and Ziyue Li. Adaptive Patching Is Harder Than It Looks For Time-Series Forecasting. arXiv preprint, 2026. Art. no. arXiv:2606.04074.

A Appendix

This appendix provides additional implementation details, statistical procedures, model and data configurations, and complete experimental results supporting the analyses in the main paper.

A.1 Implementation and Objective Details

Forecast loss. For forecasting window i, let y_i denote the observed continuation, $\hat{y}_i$ the corresponding forecast, H_i the prediction horizon, and m_{ih} a valid-target mask. We define the masked forecasting error as

$$E_i(\hat{y}_i, y_i) = \frac{\sum_{h=1}^{H_i} m_{ih}(\hat{y}_{ih} - y_{ih})^2}{\max(1, \sum_h m_{ih})}. \tag{8}$$

The normalized loss used for sensitivity computation and attack optimization is

$$d_i = \max\left[\text{stopgrad}\left\{E_i(\hat{y}_i^{\text{clean}}, y_i)\right\}, \max\left\{10^{-4}\,\text{Var}_{\text{pop}}(y_i), 10^{-8}\right\}\right], \tag{9}$$

$$\mathcal{L}_{\text{rel}} = \frac{1}{N}\sum_{i=1}^{N}\frac{E_i}{d_i}. \tag{10}$$

Population variance is used in the stabilization term, and the clean forecasting error in the denominator is detached from gradient computation. Missing target values are masked. This normalized objective is distinct from the relative MSE degradation reported as the main attack metric,

$$D_i = 100\left(\frac{E_i^{\text{attack}}}{\max(E_i^{\text{clean}}, 10^{-12})} - 1\right). \tag{11}$$

Temporal-warp parameterization. BoundaryWarp parameterizes a valid deformation through positive increments between adjacent temporal coordinates. Let $\theta \in \mathbb{R}^{C-1}$ denote the optimization variables and $w \in \mathbb{R}^C$ an optional positional prior. Location eights are converted to interval weights by

$$\bar{w}_i = \frac{w_i + w_{i+1}}{2}, \qquad i = 0, \ldots, C-2, \tag{12}$$

with $\bar{w}_i = 1$ when no prior is used. For $\eta = 10^{-4}$ and $b = \log(\exp(1-\eta) - 1)$, the positive increments are

$$q_i = \mathrm{softplus}\,(b + \theta_i \bar{w}_i) + \eta. \tag{13}$$

Defining

$$u_0 = 0, \qquad u_t = \sum_{j<t} q_j, \tag{14}$$

gives endpoint-normalized coordinates

$$\tilde{c}_t = (C-1)\frac{u_t}{u_{C-1}}, \qquad \tilde{\delta}_t = \tilde{c}_t - t. \tag{15}$$

The displacement budget is imposed with the global scaling factor

$$\alpha = \min\left(1, \frac{\epsilon}{\max_t |\tilde{\delta}_t|}\right), \tag{16}$$

where $\alpha = 1$ if the raw displacement is zero. The final warped coordinates are

$$\tau_t = t + \alpha\tilde{\delta}_t. \tag{17}$$

This construction fixes the first and last coordinates, preserves strict temporal order, and guarantees $\|\tau - t\|_\infty \le \epsilon$ by construction.

The attacked context is obtained by linearly sampling the original signal at the warped coordinates. For $\ell_t = \lfloor \tau_t \rfloor$, $r_t = \tau_t - \ell_t$, and $r'_t = \min(\ell_t + 1, C-1)$,

$$\tilde{x}_t = (1 - r_t)x_{\ell_t} + r_t x_{r'_t}. \tag{18}$$

Interpolation is performed in single precision. Coordinates are defensively restricted to the valid interval $[0, C-1]$ before sampling.

Patch-aware prior and regularization. Let

$$\mathcal{B} = \{t : \phi_m(t) = 0\} \tag{19}$$

denote the patch-start positions of the victim model. BoundaryWarp assigns

$$w_t = \exp\left[-\frac{d(t,\mathcal{B})^2}{2(\rho P_m)^2}\right], \qquad \rho = 0.125, \tag{20}$$

where $d(t, \mathcal{B})$ is the ordinary linear distance to the nearest patch transition. The main white-box experiments use the native patch length of each model and do not normalize these weights by their root mean square.

Writing $\delta_t = \tau_t - t$, temporal smoothness is encouraged through

$$R(\delta) = \frac{1}{C-1}\sum_{t=0}^{C-2}\left(\frac{\delta_{t+1} - \delta_t}{\epsilon}\right)^2. \tag{21}$$

BoundaryWarp therefore solves

$$\max_\theta \; \mathcal{L}_{\mathrm{rel}}\left(f_m(W_\theta(x)), y\right) - \lambda R(\delta), \qquad \lambda = 0.1. \tag{22}$$

Optimization and matched controls. The main white-box attacks use Adam with learning rate 0.05 for ten updates. Each run uses a single initialization $\theta \sim \mathcal{N}(0, 0.01^2)$ and global gradient clipping at norm 10. No early stopping is applied except when non-finite values are encountered. Candidate deformations are evaluated before each parameter update, and the clean input or evaluated candidate with the largest unnormalized target MSE is retained for each window.

Optimized Warp uses the same temporal parameterization, hard constraints, forecasting objective, regularizer, initialization, optimizer, update count, and candidate-selection procedure as BoundaryWarp, but assigns uniform interval weights. It therefore removes explicit information about the patch grid.

Random Warp draws independent $\theta_i \sim \mathcal{N}(0, 1)$, smooths the resulting vector by one-dimensional average pooling with kernel size 17, and passes it through the same monotone warp operator and displacement constraint. It performs no forecasting-error optimization. Random Warp is matched to the allowed displacement budget ϵ, rather than to the realized displacement of an optimized warp.

Restricted output-only optimization. For the restricted-access experiments, gradients of the forecasting model are replaced by two-sided simultaneous perturbation stochastic approximation (SPSA). Starting from $\theta = 0$, each iteration samples a Rademacher direction Δ and evaluates

$$J(\theta + c\Delta) \qquad \text{and} \qquad J(\theta - c\Delta), \tag{23}$$

with $c = 0.05$. The search direction is estimated as

$$\hat{g} = \frac{J(\theta + c\Delta) - J(\theta - c\Delta)}{2c}\Delta. \tag{24}$$

The estimate is RMS-normalized independently for each window before the update

$$\theta \leftarrow \theta + 0.08\hat{g}. \tag{25}$$

Query budgets $q \in \{8, 16, 32\}$ correspond to 4, 8, and 16 two-sided SPSA iterations, respectively. The clean forecast is not counted against the attack query budget, yielding 9, 17, and 33 total forward evaluations including the clean prediction.

Boundary-weighted SPSA uses the model-native patch length and the same Gaussian prior family and relative bandwidth $\rho = 0.125$ as BoundaryWarp. In the restricted-access implementation, the prior is centered with offset -1 and normalized to unit RMS before use. Uniform SPSA instead assigns equal weight to all directions. All other search settings and query budgets are identical.

Model access in this experiment is restricted to forecast outputs. The search does not access model parameters, activations, internal tokenizer states, or model gradients. The realized future continuation remains available for retrospective candidate scoring, so this experiment represents an oracle-target output-only robustness assessment rather than a label-free online attack.

A.2 Sensitivity Analyses and Statistical Inference

Patch-phase sensitivity. For each clean input, sensitivity is computed in a single backward pass as

$$s_t = \left| \frac{\partial \mathcal{L}_{\text{rel}}(f_m(x), y)}{\partial x_t} \right|. \tag{26}$$

For tokenizer offset o, the active patch phase is

$$\phi(t) = (t + o) \bmod P. \tag{27}$$

The corresponding phase profile is

$$S_\phi = \frac{1}{|\{t : (t + o) \bmod P = \phi\}|} \sum_{t:(t+o) \bmod P = \phi} s_t. \tag{28}$$

For the native alignment $o = 0$, this reduces to grouping positions by $t \bmod P$. Phase-profile normalization by each window's own phase mean is used only for visualization. Inferential analyses operate on unnormalized sensitivities or on explicitly defined log-sensitivity residuals.

Under an alignment shift o, the source residue expected to coincide with the terminal active tokenizer phase is

$$\phi^*(o) = (P - 1 - o) \bmod P. \tag{29}$$

Thus the alignment experiment can be represented either in source coordinates, where the predicted peak moves according to $\phi^*(o)$, or in active tokenizer coordinates, where the predicted maximum remains at phase $P - 1$.

Tokenizer-alignment intervention. The controlled alignment experiment uses Chronos-Bolt. A fixed sequence of 496 observed values is placed inside a 512-position model input envelope. For $o \in \{0, 1, 2, 4, 8, 12, 15\}$, the first o positions and the corresponding $16 - o$ suffix positions are masked. Consequently, the observed source values, their temporal order, effective observed length, input-envelope length, and number of patch tokens remain fixed while their alignment with the tokenizer changes. The experiment uses 100 matched windows for each offset and each of the four datasets.

Patch-length intervention. The patch-length experiment uses the same pretrained Moirai checkpoint while varying

$$P \in \{8, 16, 32, 64\}. \tag{30}$$

Context length remains $C = 512$ and forecasting horizon remains $H = 96$. All learned checkpoint parameters are held fixed, while the number of tokens is allowed to change with P. Each dataset–patch-length configuration uses 100 matched windows.

Controls for potential confounds. Three complementary analyses test whether the observed patch-relative pattern can instead be explained by phase aggregation, recency, or local signal content.

For the **phase-aggregation control**, artificial patch periods are assigned only after prediction to forecasters without patch tokenization. The learned control is a normalized direct ridge map from the context to the forecast horizon, trained with ridge coefficient 1.0 on $2{,}048$ windows and evaluated on 100 windows per dataset. Artificial periods $P \in \{8, 16, 32, 64\}$ are applied only during the subsequent sensitivity aggregation. A Seasonal Naive forecaster provides an additional unpatched control on ETTh1 for the same four artificial periods.

For the **recency control**, per-position sensitivities are transformed as

$$z_t = \log(s_t + \eta_i), \tag{31}$$

where

$$\eta_i = \max\left(10^{-6}\,\mathrm{median}\{s_t : s_t > 0\}, 10^{-30}\right). \tag{32}$$

A monotonically non-decreasing isotonic regression is fitted to z_t as a function of absolute input position, and phase profiles are constructed from the resulting residuals. The primary analysis excludes two patches at each edge of the context.

The complementary **internal-boundary comparison** evaluates adjacent positions around each retained internal boundary b. The primary contrast is the residual log-sensitivity immediately before the boundary, $b - 1$, minus that immediately after the boundary, b. Because b is one step closer to the forecast origin than $b - 1$, a positive contrast cannot be explained by a simple monotonically increasing recency effect.

For the **content-matched crossover**, the same Chronos-Bolt source positions are compared under different tokenizer alignments. Each source position therefore retains its observed value, local neighborhood, and forecasting target, while its tokenizer-relative phase changes. Terminal phase $P - 1$ is compared with phase 0, the central phase, and the mean of the remaining phases. The first and last two source patches are excluded.

The primary recency and internal-boundary analyses use one native patch configuration per victim model: $P = 16$ for Chronos-Bolt and Moirai and $P = 32$ for TimesFM. Each model is evaluated on all four datasets using 100 windows per model–dataset cell. The TimesFM controls are computed from previously stored per-position sensitivities and therefore require no additional model inference.

Series dependence and bootstrap inference. Forecasting windows originating from the same source series are not treated as independent. Paired analyses join observations by model, dataset, seed, window identifier, originating series, and context start. For each bootstrap replicate, originating series clusters are sampled with replacement. Within each selected cluster, observations are ordered by context start and circular moving blocks are sampled until the original cluster size is recovered. The primary analysis uses 10,000 resamples, block length $B = 5$, and percentile 95% confidence intervals. Block lengths $B = 1$ and $B = 10$ are retained as sensitivity analyses.

For the direct peak analysis, sensitivities are first reduced to one phase profile per originating series by taking the within-series median for each phase. Each originating series then contributes equally. The tested contrast is

$$T = S_{\text{target}} - \max_{\phi \neq \text{target}} S_\phi. \tag{33}$$

The one-sided bootstrap p-value is

$$p = \frac{1 + \#\{T_r^* \leq 0\}}{10{,}001}. \tag{34}$$

A direct peak is considered supported when the observed argmax equals the prespecified target, the lower bound of the 95% confidence interval for T is positive, and the Holm-adjusted p-value is below 0.05.

This series-balanced procedure should be distinguished from descriptive pooled phase profiles, which average sampled windows and therefore need not weight each originating series equally. The pooled profiles are used to visualize and describe geometry tracking, while the series-balanced analysis provides the direct inferential check reported in the main paper.

Multiplicity is controlled with Holm correction within predefined families of related tests. The direct peak family contains 48 tokenizer configurations. The expanded recency and internal-boundary families contain 136 and 272 tests, respectively, after inclusion of TimesFM. The content-matched crossover family contains 12 tests. Confidence intervals themselves are pointwise and are not multiplicity-adjusted.

A.3 Models, Data, and Experimental Configuration

Foundation models. Table 2 summarizes the exact model assets and native tokenization used in the experiments. All pretrained model parameters are frozen and all models are evaluated in inference mode.

Table 2: Foundation models and native configurations used in the experiments.

Model	Checkpoint	Native patch	Point forecast
Chronos-Bolt Tiny	`amazon/chronos-bolt-tiny`	$P = 16$	median
Moirai 1.1-R-Small	`Salesforce/moirai-1.1-R-small`	$P = 16$	distribution mean
TimesFM 2.5-200M	`google/timesfm-2.5-200m-transformers`	$P = 32$	mean prediction

Chronos-Bolt uses checkpoint revision `a0e552de83495b5c28c14c71c374f3e33280b340`, Moirai uses `0c24ab99db2c1a70ea2a0fc03bf113329772ac64`, and TimesFM uses `5a9806b9b291fad9233b5249d88263f1846304d3`. Chronos-Bolt uses masked instance normalization and a median point forecast. Moirai supports checkpoint-native patch sizes $P \in \{8, 16, 32, 64, 128\}$ and uses the distribution mean as its point forecast. TimesFM uses native patch length 32 and its checkpoint-native mean prediction.

Datasets and preprocessing. All four datasets are loaded from the same Time-Series-Library snapshot. Every numeric column is treated as an independent univariate series, and no dataset-level pre-standardization is applied. The primary experiments use context length $C = 512$ and horizon $H = 96$. Dataset sizes and split indices are reported in Table 3.

Partial missing values are masked, whereas an entirely missing context or target is rejected. One duplicate row in Weather is removed, leaving one 100-minute gap, which is not interpolated.

Table 3: Dataset configuration. The validation end is also the start of the held-out test region.

Dataset	Rows	Series	Train end	Test start	Cadence
ETTh1	17,420	7	8,640	11,520	1 h
ETTm1	69,680	7	34,560	46,080	15 min
Electricity	26,304	321	18,412	21,043	1 h
Weather	52,696	21	36,886	42,156	10 min

Candidate forecast starts range from $\max(\text{test start}, C)$ through test end $- H$. A seeded NumPy generator shuffles source series, assigns them in round-robin fashion, samples target starts independently and uniformly with replacement, and finally shuffles the resulting records. Consequently, overlapping windows and repeated start times are permitted. Across the primary attack seeds, the sampled windows cover 7, 7, 71, and 21 distinct source series for ETTh1, ETTm1, Electricity, and Weather, respectively.

Run sizes and random seeds. The main robustness study evaluates $\epsilon \in \{0.25, 0.5, 1.0\}$, with $\epsilon = 0.5$ designated as the primary budget. For each model and dataset, 25 test windows are evaluated under each of three seeds 0, 1, and 2. Across three victim models, four datasets, three attacks, and three budgets, this yields $8{,}100$ attacked instances.

The tokenizer-alignment and patch-length experiments use 100 matched windows per configuration with seed 73. The primary recency and internal-boundary controls use 100 windows per model–dataset cell. The learned unpatched model uses $2{,}048$ training windows and 100 evaluation windows with seeds 401 and 402. Restricted-access SPSA uses 50 windows per model–dataset–condition–query configuration with seed 211.

Software environment. The main environment records Python 3.11.15, PyTorch 2.13.0, NumPy 2.4.6, pandas 3.0.3, SciPy 1.17.1, Transformers 5.13.0, Chronos 2.3.1, and BoundaryWarp 0.1.0. Moirai is executed in a separately pinned environment with PyTorch 2.4.1, NumPy 1.26.4, pandas 2.1.4, SciPy 1.11.4, GluonTS 0.14.4, and uni2ts 2.0.0. Experiments use Apple arm64 macOS and MPS where supported. Random seeds and stored window manifests make the data selection reproducible, although bitwise determinism across library and MPS versions is not assumed.

A.4 Complete Results and Validity Audits

Main temporal-deformation results. Table 4 reports median relative MSE degradation for all model–dataset–budget combinations. Each cell lists Random Warp / Optimized Warp / BoundaryWarp.

Table 4: Median relative MSE degradation (%) for all main attack configurations. Each entry reports Random Warp / Optimized Warp / BoundaryWarp.

Model	Dataset	$\epsilon = .25$	$\epsilon = .50$	$\epsilon = 1.00$
Chronos-Bolt	Electricity	1.005 / 6.713 / 8.334	3.603 / 21.947 / 27.272	12.270 / 86.323 / 94.319
	ETTh1	−0.506 / 1.228 / 2.156	−0.641 / 4.072 / 5.364	1.454 / 12.026 / 12.601
	ETTm1	0.151 / 2.590 / 2.544	0.439 / 4.603 / 5.171	1.175 / 6.999 / 7.777
	Weather	−0.138 / 0.810 / 0.424	−0.223 / 1.144 / 1.233	−0.371 / 2.173 / 2.856
Moirai	Electricity	0.113 / 1.231 / 1.831	−0.205 / 2.739 / 3.776	−0.319 / 3.705 / 6.689
	ETTh1	−0.022 / 1.694 / 1.819	−0.111 / 3.339 / 4.286	0.011 / 6.361 / 6.615
	ETTm1	0.458 / 3.058 / 3.502	1.084 / 5.512 / 8.366	0.113 / 7.941 / 13.364
	Weather	0.301 / 1.161 / 1.965	0.788 / 2.319 / 3.651	1.235 / 6.907 / 7.917
TimesFM	Electricity	2.521 / 9.589 / 10.213	10.694 / 31.975 / 34.185	28.496 / 109.378 / 98.382
	ETTh1	0.012 / 3.856 / 3.666	−0.329 / 6.877 / 7.676	2.006 / 15.470 / 18.681
	ETTm1	1.341 / 3.954 / 4.687	1.755 / 6.089 / 9.898	3.333 / 11.328 / 15.314
	Weather	1.095 / 1.500 / 1.688	1.719 / 5.910 / 5.863	1.755 / 9.143 / 8.607

Aggregating across datasets, the Random / Optimized / BoundaryWarp medians at $\epsilon = 0.5$ are $0.42/6.18/7.70$ for Chronos-Bolt, $0.15/3.30/4.71$ for Moirai, and $1.78/10.77/12.76$ for TimesFM. Across all budgets, BoundaryWarp produces larger raw degradation than Optimized Warp in 34 of 36 model–dataset–budget configurations, with 20 paired comparisons remaining significant after Holm correction. At the primary budget, the mean paired BoundaryWarp-minus-Optimized-Warp effects are 2.31 percentage points $[1.71, 3.12]$ for Chronos-Bolt, 1.25 $[0.70, 1.85]$ for Moirai, and 1.65 $[-1.30, 4.13]$ for TimesFM.

Table 5 reports primary-budget estimates separately for each model and dataset using the series-cluster bootstrap estimand.

Table 5: Primary-budget ($\epsilon = 0.5$) median relative MSE degradation with 95% series-cluster bootstrap confidence intervals.

Model	Dataset	Random Warp	Optimized Warp	BoundaryWarp
Chronos-Bolt	Electricity	4.07 [1.62, 6.36]	21.42 [17.14, 34.23]	27.27 [21.84, 39.63]
	ETTh1	−0.87 [−1.46, 0.58]	3.83 [0.40, 6.07]	5.21 [4.50, 8.19]
	ETTm1	0.98 [−0.19, 1.80]	4.60 [2.37, 6.83]	7.16 [3.01, 7.55]
	Weather	−0.52 [−2.41, 0.46]	1.14 [0.33, 2.01]	1.92 [0.32, 2.41]
Moirai	Electricity	−0.11 [−0.87, 0.88]	2.67 [1.65, 4.21]	3.59 [2.14, 5.73]
	ETTh1	0.12 [−0.66, 1.69]	3.77 [1.97, 6.25]	4.29 [2.26, 8.12]
	ETTm1	1.54 [−1.41, 3.42]	5.17 [4.80, 9.18]	8.37 [4.91, 8.67]
	Weather	1.15 [−1.29, 2.17]	3.02 [1.28, 4.67]	5.39 [2.43, 7.40]
TimesFM	Electricity	10.69 [4.88, 16.73]	33.00 [25.92, 48.27]	35.52 [24.79, 43.41]
	ETTh1	−0.44 [−1.33, 4.00]	6.88 [4.08, 11.08]	7.62 [3.63, 12.79]
	ETTm1	1.28 [0.07, 3.01]	8.02 [4.88, 11.32]	9.90 [7.74, 15.56]
	Weather	−1.09 [−4.41, 4.90]	8.07 [0.11, 11.17]	10.14 [0.35, 18.37]

Tokenizer-geometry analyses. The descriptive pooled alignment profiles follow the predicted source residue $(P - 1 - o) \bmod P$ in all 28 Chronos-Bolt dataset–offset settings. Expressed in active tokenizer coordinates, all 28 pooled maxima therefore occur at terminal phase 15. For Moirai, the pooled sensitivity maximum occurs at $P - 1$ in all 16 dataset–patch-length settings.

The stricter series-balanced direct analysis evaluates 48 configurations, comprising the 28 Chronos-Bolt alignment settings, 16 Moirai patch-length settings, and four native TimesFM settings. It identifies the terminal phase as the exact maximum in $43/48$ configurations, or 89.6%. By model, the exact counts are $24/28$ for Chronos-Bolt, $16/16$ for Moirai, and $3/4$ for TimesFM. The corresponding native model–dataset subset contains $11/12$ exact terminal-phase maxima.

A direct peak is statistically supported under the joint argmax, confidence interval, and Holm criteria in $12/28$ Chronos-Bolt alignment settings, $13/16$ Moirai patch-length settings, and $1/4$ TimesFM native settings. These tests provide a stricter inferential characterization than the descriptive pooled profiles and explain why the pooled and series-balanced counts differ.

Clean performance across patch lengths. Changing the Moirai patch length alters clean forecasting performance in addition to changing tokenizer geometry. Table 6 therefore reports clean MSE for every patch-length configuration.

Table 6: Clean Moirai MSE across patch lengths. Entries report mean / median MSE over the matched clean windows.

Dataset	$P = 8$	$P = 16$	$P = 32$	$P = 64$
Electricity	1,624,172.68 / 36,526.57	1,466,531.94 / 35,368.96	273,020.03 / 6,080.84	246,315.75 / 5,858.38
ETTh1	21.530 / 1.878	20.121 / 1.888	11.434 / 1.207	10.597 / 1.139
ETTm1	21.690 / 1.705	24.450 / 1.807	17.867 / 1.864	15.963 / 1.155
Weather	5,264.63 / 8.505	20,214.28 / 10.378	7,353.11 / 5.393	8,954.21 / 5.438

Thus the patch-length intervention establishes the persistence of the tokenizer-relative peak within each configured model, rather than invariance of clean forecasting performance across patch lengths.

Controls for potential confounds. The unpatched Seasonal Naive control shows no terminal maximum in any of its four artificial-period settings. The learned ridge control shows one exact and statistically supported terminal maximum among 16 dataset–period settings. The combined series-balanced result is therefore $1/20$.

Table 7 reports the primary recency-detrended and internal-boundary effects for all 12 model–dataset settings. Positive values favor the terminal or immediately pre-boundary position.

Table 7: Primary controls for recency. Effects are differences in residual log-sensitivity with 95% confidence intervals.

Model	Dataset	Recency-detrended	Internal boundary
Chronos-Bolt	Electricity	0.396 [0.346, 0.445]	0.494 [0.423, 0.565]
	ETTh1	0.380 [0.244, 0.521]	0.487 [0.320, 0.661]
	ETTm1	0.624 [0.459, 0.800]	0.678 [0.483, 0.875]
	Weather	1.082 [0.895, 1.251]	1.048 [0.880, 1.204]
Moirai	Electricity	1.020 [0.963, 1.080]	0.653 [0.573, 0.733]
	ETTh1	0.963 [0.911, 1.027]	0.748 [0.613, 0.882]
	ETTm1	0.992 [0.902, 1.084]	0.641 [0.562, 0.721]
	Weather	1.366 [1.149, 1.610]	1.038 [0.790, 1.321]
TimesFM	Electricity	0.256 [0.185, 0.326]	0.469 [0.371, 0.567]
	ETTh1	0.231 [0.161, 0.301]	0.422 [0.337, 0.516]
	ETTm1	0.358 [0.262, 0.462]	0.596 [0.474, 0.709]
	Weather	0.506 [0.326, 0.705]	0.895 [0.712, 1.093]

All 12 primary recency-detrended contrasts remain significant after Holm correction in the expanded 136-test family ($p_{\mathrm{adj}} = 0.0136$ for these primary rows). All 12 primary internal-boundary contrasts likewise remain significant in the expanded 272-test family ($p_{\mathrm{adj}} = 0.0272$). The medians of the 12 cell-level mean contrasts are 0.565 and 0.647 log units, respectively. These summary values are descriptive medians over the cell-level effects rather than separately pooled bootstrap estimates. For the content-matched Chronos-Bolt crossover, the terminal-minus-mean-other effects are 0.247 $[0.217, 0.277]$ for Electricity, 0.346 $[0.239, 0.463]$ for ETTh1, 0.589 $[0.441, 0.734]$ for ETTm1, and 1.076 $[0.907, 1.221]$ for Weather. All four primary contrasts remain significant after Holm correction ($p_{\mathrm{adj}} = 0.0012$). Their median is 0.468 log units.

Restricted output-only results. Table 8 reports model-level restricted-access results. Increasing the query budget increases median degradation for all three victim models. Boundary weighting does not provide a statistically reliable advantage over Uniform SPSA after multiplicity correction.

Table 8: Restricted output-only results aggregated across datasets. RED is median relative MSE degradation with 95% confidence interval. Success is the fraction of windows with positive degradation.

Model	q	Boundary RED	Success	Uniform RED	Success
Chronos-Bolt	8	3.98 [3.12, 5.50]	.835	3.72 [2.60, 4.94]	.820
	16	5.49 [4.15, 7.64]	.880	4.83 [3.69, 6.59]	.885
	32	7.20 [5.40, 9.54]	.910	6.28 [5.26, 9.00]	.915
Moirai	8	2.75 [1.83, 3.63]	.800	2.76 [2.04, 3.61]	.775
	16	3.95 [3.10, 4.91]	.865	3.76 [3.04, 4.91]	.820
	32	4.86 [4.13, 6.02]	.905	5.06 [3.95, 6.41]	.900
TimesFM	8	6.04 [4.37, 8.27]	.810	6.10 [4.60, 8.50]	.805
	16	8.35 [6.39, 11.53]	.870	8.54 [6.87, 11.41]	.860
	32	11.96 [9.58, 15.17]	.885	11.74 [9.73, 16.24]	.895

For completeness, Table 9 reports all individual model–dataset–query configurations.

Table 9: Complete restricted-output results. Each RED entry reports median relative MSE degradation with 95% confidence interval.

Model	Dataset	q	Boundary RED	B succ.	Uniform RED	U succ.
Chronos-Bolt	Electricity	8	12.97 [8.85,15.84]	.98	11.67 [7.39,14.65]	.90
		16	16.54 [11.72,19.66]	.98	15.39 [10.12,20.32]	.98
		32	19.73 [15.55,23.21]	1.00	20.15 [14.06,23.00]	.98
	ETTh1	8	3.76 [1.47,4.86]	.82	3.65 [2.20,4.89]	.84
		16	5.45 [3.32,7.80]	.86	4.68 [3.52,6.48]	.90
		32	7.57 [4.39,9.49]	.90	6.05 [4.28,7.64]	.90
	ETTm1	8	3.31 [0.87,6.33]	.76	1.89 [1.05,5.31]	.80
		16	4.48 [1.64,8.20]	.84	3.63 [2.23,5.71]	.88
		32	5.70 [2.65,9.32]	.86	5.56 [2.63,9.99]	.94
	Weather	8	2.06 [1.19,3.64]	.78	1.66 [0.90,3.61]	.74
		16	3.28 [2.01,4.83]	.84	2.77 [1.32,4.17]	.78
		32	4.25 [2.43,6.64]	.88	3.99 [2.37,5.82]	.84
Moirai	Electricity	8	1.76 [1.00,2.65]	.76	1.88 [1.06,2.59]	.76
		16	2.36 [1.44,4.13]	.86	2.61 [1.62,4.47]	.80
		32	3.82 [2.46,4.84]	.92	4.20 [2.64,5.24]	.90
	ETTh1	8	2.60 [1.10,3.98]	.82	2.48 [1.04,3.63]	.76
		16	3.59 [1.93,5.01]	.90	3.16 [1.44,4.12]	.82
		32	4.50 [2.84,5.61]	.94	3.67 [2.39,6.15]	.90
	ETTm1	8	7.76 [3.63,11.51]	.86	5.97 [4.12,11.46]	.84
		16	9.46 [5.71,13.89]	.86	8.86 [5.48,14.30]	.86
		32	10.34 [6.53,16.20]	.90	10.69 [6.86,17.18]	.92
	Weather	8	2.05 [1.47,5.05]	.76	2.52 [1.82,5.77]	.74
		16	3.83 [1.93,6.64]	.84	3.30 [2.80,8.69]	.80
		32	4.61 [3.04,7.57]	.86	4.90 [3.50,11.65]	.88
TimesFM	Electricity	8	14.92 [10.14,23.46]	.94	16.00 [9.84,25.21]	.94
		16	19.08 [14.94,33.24]	.98	24.58 [16.03,30.22]	.98
		32	27.44 [22.05,39.70]	.98	28.38 [22.84,36.65]	.98
	ETTh1	8	6.03 [3.63,8.33]	.80	4.88 [2.75,9.29]	.82
		16	7.28 [3.65,10.50]	.88	6.99 [3.18,11.54]	.84
		32	10.21 [4.73,12.32]	.92	10.31 [6.09,13.59]	.90
	ETTm1	8	4.12 [0.99,6.54]	.76	3.93 [0.96,6.17]	.74
		16	5.66 [1.62,8.33]	.82	5.53 [1.37,8.57]	.82
		32	8.09 [4.14,11.81]	.84	8.08 [3.08,12.47]	.84
	Weather	8	4.29 [1.08,7.90]	.74	5.69 [2.92,8.38]	.72
		16	6.96 [3.09,11.80]	.80	7.90 [4.96,10.33]	.80
		32	10.94 [5.19,18.62]	.80	10.11 [6.84,19.40]	.86

At $q = 32$, effectiveness relative to zero degradation remains significant after Holm correction in all 12 model–dataset configurations, with largest adjusted $p = 0.0112$. Boundary-weighted SPSA does not significantly outperform Uniform SPSA in any of the 36 model–dataset–query comparisons after Holm correction. The pooled Boundary-minus-Uniform median differences are 0.31, 0.25, and 0.34 percentage points at $q = 8$, 16, and 32, respectively.

NRMSE robustness check. Table 10 reports an absolute-error robustness check at the primary budget. BoundaryWarp increases NRMSE in every model–dataset configuration.

The equal three-decimal clean and attacked values for TimesFM on ETTh1 result from rounding and do not imply a zero paired change.

Deformation-validity audit. All $8{,}100$ main-study deformations were independently re-evaluated against the hard validity conditions. Every deformation satisfied the requested L_∞ displacement budget, fixed endpoints, valid coordinate range, strict monotonicity, coordinate uniqueness, and reconstruction consistency. The audit uses tolerances of 10^{-5} for the displacement budget, 10^{-4}

Table 10: Series-cluster median clean and attacked NRMSE at $\epsilon = 0.5$. Parentheses show the paired change.

Model	Electricity	ETTh1	ETTm1	Weather
Chronos-Bolt	.332→.369 (+.042)	.800→.816 (+.024)	.746→.788 (+.023)	.707→.719 (+.003)
Moirai	1.013→1.034 (+.019)	1.071→1.103 (+.032)	1.062→1.094 (+.047)	1.108→1.119 (+.023)
TimesFM	.321→.365 (+.047)	.770→.770 (+.028)	.694→.836 (+.032)	.761→.766 (+.018)

for coordinates and endpoints, and 2×10^{-5} for reconstruction. Coordinate uniqueness additionally requires a minimum increment greater than 10^{-4}.

The maximum displacement reaches the requested budget in every model–attack–budget group, while the maximum recorded reconstruction error is zero. Across all audited conditions, the smallest observed coordinate increment ranges from 0.835785 for Moirai BoundaryWarp at $\epsilon = 1$ to 0.984131 for TimesFM Random Warp at $\epsilon = 0.25$. These checks confirm that the reported degradation is obtained from valid order-preserving temporal deformations rather than from endpoint violations, extrapolation, duplicated coordinates, or interpolation errors.

Computational cost. Random Warp requires two forward-equivalent evaluations per window, while the white-box optimized attacks require 12. Restricted SPSA requires 9, 17, or 33 forward evaluations including the clean forecast at query budgets 8, 16, and 32, respectively. Runtime depends strongly on the model and execution environment. The restricted TimesFM experiments require approximately 2.49, 4.70, and 9.20 seconds per window at the three query budgets, while the corresponding Chronos-Bolt and Moirai executions are substantially faster. These timings are reported only as implementation-level cost indicators and should not be interpreted as hardware-independent benchmarks.