ARE NEURAL MODELS ALWAYS NECESSARY FOR IRREGULAR TIME SERIES FORECASTING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Irregular multivariate time-series (IMTS) forecasting is commonly assumed to require specialized neural architectures for modeling asynchronous observations. We audit this assumption by asking how much standard benchmark performance can already be explained by simple structure in the observed history. We introduce the ANCHOR family, a set of transparent, gradient-free diagnostic controls built from fixed history-derived forecasting rules. Its strongest adaptive variant, AUTOANCHOR, selects among these rules using only non-test data and structural diagnostics. Across global IMTS benchmarks and neural baselines under matched protocols, ANCHOR variants achieve the best standard-window error in every dataset–metric setting, while AUTOANCHOR remains competitive across lookback–horizon sweeps and paired entity-level tests. Perturbation, thinning, bootstrap, and ablation analyses further show that different benchmarks reward different forms of simple historical structure, while neural models regain advantages when history is insufficient or forecasting conditions become more demanding. These results suggest that current IMTS benchmarks can overstate the necessity of complex neural representations, motivating stronger non-neural diagnostic controls before attributing performance gains to architectural complexity. The source code is available at: https://anonymous.4open.science/r/Anchor-Family/.

1 INTRODUCTION

Irregular multivariate time series (IMTS) forecasting has motivated a broad class of specialized neural architectures. Unlike regularly sampled sequences, IMTS data contain asynchronous observations, variable-specific missingness, and uneven time gaps, making it difficult to directly apply standard sequence models. Existing approaches address these challenges through missingness-aware recurrence (Che et al., 2018), interpolation (Shukla & Marlin, 2019), continuous-time attention (Shukla & Marlin, 2021; Chowdhury et al., 2023), latent continuous dynamics (Chen et al., 2018; Rubanova et al., 2019; Kidger et al., 2020; Biloš et al., 2021; Schirmer et al., 2022), graph message passing (Zhang et al., 2022; Mercatali et al., 2024; Yalavarthi et al., 2024), and adaptive patching (Zhang et al., 2024; Liu et al., 2026). Improvements on standard benchmarks are consequently often interpreted as evidence that richer representations of irregular temporal dynamics lead to better forecasting. However, benchmark performance alone does not reveal *why* a model predicts well. The forecasting task is to predict future queried values from the available observed history, and that history can contain predictive structure of very different complexity. Success may require modeling asynchronous temporal interactions and cross-variable dependencies, but it may also arise because future values can already be estimated well from simple patterns in the observed history. These possibilities can lead to similar aggregate benchmark errors while implying very different requirements for the forecasting model. Consequently, a leaderboard improvement does not by itself establish that the benchmark requires the complex representation that a model is designed to learn.

This paper therefore asks a basic question: *how much of the performance on today's IMTS benchmarks actually requires complex learned representations?* We hypothesize that a substantial part of benchmark performance may be explained by history-derived structure: predictive information that can be extracted directly from the observed history using simple, transparent forecasting rules. If such structure already accounts for much of the benchmark score, architectural improvements should be interpreted only after controlling for it under matched benchmark evaluation protocols.

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

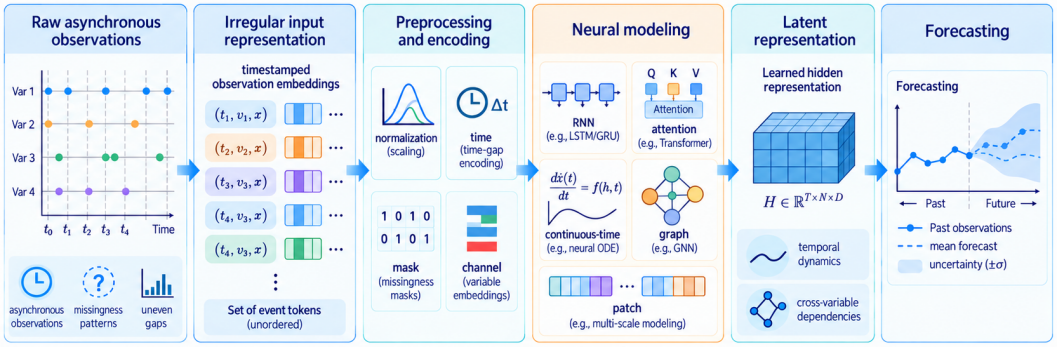


Figure 1: General modeling pipeline for irregular multivariate time series by neural models. Raw asynchronous observations are normalized and enriched with explicit timing and missingness information, then embedded into a learned latent representation that captures temporal dynamics and cross-variable dependencies. This pattern is finally decoded for downstream forecasting tasks.

To answer this question, we frame IMTS forecasting as a *benchmark audit* and introduce the ANCHOR family, a hierarchy of transparent diagnostic controls for history-derived structure. Rather than learning a latent representation, ANCHOR methods forecast using a fixed library of simple history-based rules. The family ranges from basic fixed controls to adaptive variants that select among these rules using only non-test data. Its strongest member, AUTOANCHOR, combines empirical rule selection with generic diagnostics of the observed history and freezes the resulting specification before test evaluation. AUTOANCHOR therefore serves as a strong non-neural control: if it can match a neural model under the same benchmark protocol, then the benchmark score alone provides limited evidence that the neural model's richer representation was necessary for forecasting. Figure 1 illustrates the representation-learning pipeline whose necessity our benchmark audit examines

Our experiments reveal that this control is surprisingly strong. Under matched benchmark protocols, ANCHOR variants match or outperform representative neural IMTS models in many standard settings, while AUTOANCHOR remains competitive across changes in lookback and forecasting horizon. More importantly, the results show that the explanation is not a single trivial heuristic. Different datasets favor different forms of history-derived structure, and paired tests, perturbations, history thinning, and ablations expose substantial differences in what information each benchmark actually requires. At the same time, the ANCHOR family has clear failure regimes: neural models regain advantages when available history becomes insufficient or when forecasting conditions require richer temporal information. These cases are important because they help distinguish benchmarks that can largely be explained by simple historical structure from those that genuinely reward additional modeling capacity under controlled, reproducible, and matched evaluation conditions.

This perspective complements recent findings in regularly sampled forecasting, where simple models have challenged the presumed necessity of increasingly complex architectures (Zeng et al., 2023; Qiu et al., 2024; Chen et al., 2025). Our goal is not to argue that neural models are generally unnecessary for IMTS forecasting. Rather, we argue that benchmark performance alone is insufficient evidence that complex irregular-time representations are required. Ultimately, a more informative evaluation asks what performance remains unexplained after strong, transparent controls for history-derived structure have been applied under the same benchmark protocol and splits.

Our contributions are:

- We formulate IMTS forecasting as a benchmark-audit problem and introduce the Anchor family, a hierarchy of transparent controls for measuring predictive signal available from history-derived structure without neural representation learning under matched evaluation.

- We conduct a matched evaluation against representative neural IMTS models under common splits, preprocessing, masks, scaling, and metrics, complemented by paired tests, lookback–horizon sweeps, perturbations, history thinning, and ablations for interpretation.

- We characterize both sides of the audit: standard regimes in which simple history-derived controls explain much of benchmark performance, and more demanding regimes in which neural models recover measurable advantages across distinctive benchmark conditions.

2 Related Work

Neural modeling for IMTS. IMTS forecasting has motivated a wide range of neural architectures designed to represent asynchronous observations, missingness, uneven temporal gaps, and dependencies across variables. Early approaches explicitly incorporate observation patterns into recurrent models: GRU-D (Che et al., 2018) decays inputs and hidden states according to elapsed time, while interpolation-prediction networks (Shukla & Marlin, 2019) construct continuous representations from sparse observations. Attention-based methods such as mTAN (Shukla & Marlin, 2021) operate over continuous-time embeddings, whereas SeFT (Horn et al., 2020) represents irregular observations as sets. Transformer-based models further capture temporal and cross-variable interactions, including CoFormer (Wei et al., 2023) and signature-based Rough Transformers (Moreno-Pino et al., 2024). Another line of work models latent trajectories in continuous time using Neural ODEs (Chen et al., 2018; Rubanova et al., 2019), neural controlled differential equations (Kidger et al., 2020), Neural Flows (Biloš et al., 2021), and continuous recurrent units (Schirmer et al., 2022). More recent methods incorporate graph structure and multi-scale temporal representations (Zhang et al., 2022; Mercatali et al., 2024; Yalavarthi et al., 2024; Zhang et al., 2024; Liu et al., 2026), as well as channel-harmony modeling (Liu et al., 2025), hierarchical patch graphs (Luo et al., 2025), hypergraph dependencies (Li et al., 2025), and canonical pre-alignment (Zhou et al., 2026). In general, these methods primarily seek increasingly effective representations of irregular temporal structure and cross-variable dependencies; in contrast, our work asks how much standard IMTS benchmark performance can be explained without learning such representations under matched protocols.

Simple models and forecasting benchmarks. A related line of research has questioned whether increasing architectural complexity is always necessary for strong forecasting performance. In regularly sampled time-series forecasting, Transformer-based models were developed to capture long-range temporal dependencies (Zhou et al., 2021), yet subsequent studies showed that substantially simpler models can remain competitive under comparable evaluation settings (Zeng et al., 2023; Chen et al., 2025). Large-scale benchmarking efforts further demonstrate that conclusions about forecasting performance can change when datasets, preprocessing, prediction horizons, hyperparameter selection, and evaluation protocols are standardized across methods (Qiu et al., 2024). These studies emphasize the importance of strong baselines and controlled evaluation when interpreting architectural improvements. Our work extends this perspective to irregular forecasting, but with a different objective: rather than proposing another lightweight predictor, we use simple history-derived rules as diagnostic controls to examine what current IMTS benchmarks actually require.

3 The Anchor Family

Task and notation. For each sample b, the observed history consists of irregularly timestamped measurements across variables. For variable v, we denote its observed history by

$$\mathcal{H}_{b,v} = \{(t_{b,v,j}, x_{b,v,j})\}_{j=1}^{n_{b,v}}, \qquad t_{b,v,1} < \cdots < t_{b,v,n_{b,v}}. \tag{1}$$

Given $\mathcal{H}_{b,v}$, the forecasting task is to predict future targets $y_{b,k,v}$ at query times $t^*_{b,k,v}$. Because future targets may be only partially observed, evaluation uses the benchmark-provided target mask $m_{b,k,v} \in \{0,1\}$. Our objective is not to learn another representation of the irregular history. Instead, we ask how much predictive information can be extracted directly from the observed history using transparent, non-neural forecasting rules. We therefore introduce the Anchor family, a hierarchy of diagnostic controls for history-derived structure.

Shared candidate class. All Anchor variants operate on the same fixed candidate library $\mathcal{A}$. A candidate $a \in \mathcal{A}$ combines a small number of primitive history-based rules,

$$f_a(\mathcal{H}, t^*) = \sum_{\ell=1}^{J_a} w_{a,\ell} g_{a,\ell}(\mathcal{H}, t^*), \qquad w_{a,\ell} \geq 0, \quad \sum_{\ell=1}^{J_a} w_{a,\ell} = 1. \tag{2}$$

Table 1: **Overview of the Anchor family.** All variants share the fixed candidate class $\mathcal{A}$ and differ only in how the per-variable specification $\theta_v = (a_v, \beta_v)$ is determined. Fixed variants isolate particular forms of history-derived structure, ERMANCHOR selects among the shared candidates using non-test empirical risk, and AUTOANCHOR additionally applies structural diagnostics. Once calibrated, the specification is frozen and applied through the common test-time forecasting operator.

Shared anchor class

$$\mathcal{A} = \left\{ a : f_a(\mathcal{H}_v, t^*) = \sum_{\ell=1}^{J_a} w_{a,\ell} g_{a,\ell}(\mathcal{H}_v, t^*), \quad w_{a,\ell} \geq 0, \ \sum_{\ell} w_{a,\ell} = 1 \right\}.$$

Variant	Calibration rule for variable v	Frozen specification
NaiveAnchor	Prescribe the persistence candidate.	$\theta_v = (\mathrm{Last}, 1)$
ExpoAnchor	Prescribe an EMA (Exponential Moving Average) candidate with horizon-aware smoothing, $\alpha = \mathrm{clip}\{\mathrm{round}(10\sqrt{K/L})/10, 0.1, 0.9\}$.	$\theta_v = (\mathrm{EMA}_\alpha, 1)$
SparseAnchor	Apply a sparse-history diagnostic $s_v = (p_v, u_v)$, where p_v is the dominant-value frequency and u_v is the rounded-value cardinality; choose a mode–phase candidate when the diagnostic activates.	$\theta_v = (a_v^{\mathrm{sp}}, 1)$
ERMAnchor	Select by non-test empirical risk: $(a_v, \beta_v) = \arg\min_{a \in \mathcal{A},\, \beta \in B_a} \widehat{R}_v(a, \beta)$.	$\theta_v = (a_v^{\mathrm{ERM}}, \beta_v)$
AutoAnchor	Use an ordered structural prior $\pi_v(\mathcal{H})$ when it fires; otherwise retain the ERM solution: $\theta_v = \begin{cases} (\pi_v(\mathcal{H}), 1), & \pi_v \neq \varnothing, \\ \theta_v^{\mathrm{ERM}}, & \text{otherwise.} \end{cases}$	$\theta_v = (a_v^{\mathrm{prior}}, 1)$ or θ_v^{ERM}

Shared test-time operator $\hat{y}_{b,k,v} = P_{b,v}\left(\beta_v f_{a_v}(\mathcal{H}_{b,v}, t^*_{b,k,v})\right)$, where $P_{b,v}$ denotes the Anchor output projection after inverse scaling.

The primitive rules capture forms of low-complexity predictability, including the most recent observation, recent-level summaries, trimmed means, mode, exponential smoothing, bounded local trend, and phase-nearest lookup. The candidate library is fixed before evaluation and shared across datasets and variables. Therefore, its purpose is not to construct an optimal hand-engineered forecaster, but to provide a common set of controls for predictive structure already available in the observed history.

Non-test calibration. Some ANCHOR variants use non-test data to select a candidate and, when applicable, a scalar coefficient β. Let $\mathcal{D}_{\mathrm{cal}}$ denote the calibration data drawn from the training and validation splits. For variable v, we define the masked empirical risk

$$\widehat{R}_v(a, \beta) = \frac{\sum_{b \in \mathcal{D}_{\mathrm{cal}}} \sum_k m_{b,k,v} \left(\widetilde{y}_{b,k,v} - \widetilde{f}_{a,\beta}(\mathcal{H}_{b,v}, t^*_{b,k,v}) \right)^2}{\sum_{b \in \mathcal{D}_{\mathrm{cal}}} \sum_k m_{b,k,v}}. \tag{3}$$

Accordingly, tildes denote quantities in the benchmark-scaled space, and $\widetilde{f}_{a,\beta}$ denotes the corresponding Anchor prediction after the same deterministic output processing used at evaluation. The selected per-variable specification $\theta_v = (a_v, \beta_v)$ is frozen before test evaluation; no candidate search or sample-wise routing is performed using test targets.

The Anchor ladder. The five variants differ only in how the frozen specification θ_v is chosen. NAIVEANCHOR fixes the last-value rule and therefore provides a persistence control. EXPOANCHOR fixes an exponential moving-average rule, with smoothing determined by the lookback–horizon configuration. SPARSEANCHOR uses a simple diagnostic of repeated values to choose

among last-value, mode, and mode–phase rules. These three variants isolate specific forms of history-derived structure without empirical search over the full candidate library. ERMANCHOR provides a stronger adaptive control by selecting the candidate and optional scale coefficient that minimize non-test empirical risk under the same frozen evaluation protocol:

$$(a_v^{\mathrm{ERM}}, \beta_v) = \arg \min_{a \in \mathcal{A},\ \beta \in \mathcal{B}_a} \widehat{R}_v(a, \beta), \qquad \theta_v^{\mathrm{ERM}} = (a_v^{\mathrm{ERM}}, \beta_v), \tag{4}$$

$\mathcal{B}_a$ denotes the allowed calibration coefficients for candidate a. ERMANCHOR asks how strong a control can be obtained by selecting among the fixed history-derived rules using non-test targets.

AutoAnchor. AutoAnchor is the strongest adaptive member of the family. It starts from the ERMAnchor solution and supplements it with an ordered structural diagnostic that detects pronounced repetition, long-window short-horizon smoothing, or phase-like recurrence in the calibration histories. Let a_v^{P} denote the candidate proposed when such a diagnostic activates. AutoAnchor uses

$$\theta_v^{\mathrm{auto}} = \begin{cases} (a_v^{\mathrm{P}}, 1), & \text{if a structural diagnostic activates,} \\ \theta_v^{\mathrm{ERM}}, & \text{otherwise.} \end{cases} \tag{5}$$

The diagnostics do not use dataset or variable identities. Their thresholds are fixed and disclosed in the implementation, but are treated as exploratory design choices rather than universal constants. We therefore interpret AUTOANCHOR together with the full Anchor ladder and corresponding ablations.

Frozen test-time forecasting. Once $\theta_v = (a_v, \beta_v)$ has been determined, all test examples use the same specification for variable v:

$$\hat{y}_{b,k,v} = P_{b,v}\big(\beta_v f_{a_v}(\mathcal{H}_{b,v}, t^*_{b,k,v})\big)\,, \tag{6}$$

Accordingly, $P_{b,v}$ denotes the deterministic ANCHOR output projection after inverse scaling. Thus, the ANCHOR family progressively strengthens the non-neural control while keeping the test-time forecasting procedure frozen. Table 1 summarizes ANCHOR hierarchy and its shared primitives.

3.1 BENCHMARK AUDIT PROTOCOL

Benchmark datasets. We evaluate on four widely used IMTS forecasting benchmarks: PhysioNet Challenge 2012 (P12) (Silva et al., 2012), MIMIC (Johnson et al., 2016), HumanActivity (Liu et al., 2026), and USHCN (Menne & Williams, 2012). They span clinical, activity, and climate domains and differ substantially in sampling density, target availability, temporal scale, and variable dynamics, providing diverse regimes for testing when history-derived controls are sufficient.

Neural baselines. We compare against representative IMTS models spanning recurrent, set-based, attention-based, continuous-time, graph-based, and patch-based modeling: PrimeNet (Chowdhury et al., 2023), NeuralFlows (Biloš et al., 2021), CRU (Schirmer et al., 2022), mTAN (Shukla & Marlin, 2021), SeFT (Horn et al., 2020), GNeuralFlow (Mercatali et al., 2024), GRU-D (Che et al., 2018), Raindrop (Zhang et al., 2022), Warpformer (Zhang et al., 2023), tPatchGNN (Zhang et al., 2024), GraFITi (Yalavarthi et al., 2024), and APN (Liu et al., 2026). We follow the common benchmark protocol of Liu et al. (2026), keeping data splits, preprocessing, sequence configurations, masks, scaling, prediction horizons, and evaluation procedures fixed across methods. Hence, neural baselines use their released configurations together with validation-based tuning, and results are averaged over seeds 2024–2026. No method uses test labels for tuning or sample-wise test routing.

Evaluation metrics. Let $s_v(\cdot)$ denote the scaling transform estimated from the training split for variable v. Predictions and targets are evaluated in scaled space,

$$\widetilde{y}_{b,k,v} = s_v(y_{b,k,v}), \qquad \widetilde{\hat{y}}_{b,k,v} = s_v(\hat{y}_{b,k,v}). \tag{7}$$

Following the benchmark convention, we report globally masked MAE and MSE:

$$\mathrm{MAE} = \frac{\sum_{b,k,v} m_{b,k,v} |\widetilde{\hat{y}}_{b,k,v} - \widetilde{y}_{b,k,v}|}{\sum_{b,k,v} m_{b,k,v}}, \ \mathrm{MSE} = \frac{\sum_{b,k,v} m_{b,k,v} (\widetilde{\hat{y}}_{b,k,v} - \widetilde{y}_{b,k,v})^2}{\sum_{b,k,v} m_{b,k,v}}. \tag{8}$$

Because aggregate metrics can obscure heterogeneity across entities, we complement them with paired entity-level tests and bootstrap confidence intervals.

Audit design. The benchmark audit addresses four questions. First, can transparent history-derived controls match neural IMTS models at standard benchmark settings? Second, does this conclusion persist across temporal configurations and individual entities? Third, which information in the observed history supports Anchor performance? Finally, under what conditions do neural models recover a measurable advantage? We address these questions through standard-window comparisons and paired tests, lookback–horizon sweeps, temporal perturbations and history thinning, and uncertainty and candidate-level analyses, respectively, under a unified evaluation framework.

4 BENCHMARK AUDIT RESULTS

4.1 CAN HISTORY-DERIVED CONTROLS MATCH NEURAL MODELS?

Standard benchmark settings. Table 2 compares the ANCHOR family with neural baselines at the standard window of each benchmark. An ANCHOR variant achieves the best or tied-best error among the evaluated methods in all eight dataset–metric settings, with AUTOANCHOR doing so in six. The magnitude of the differences, however, varies considerably. Relative to the strongest neural baseline, improvements are small on HumanActivity MSE and P12 MSE/MAE, but larger on USHCN MAE and MIMIC MSE/MAE. The main result is therefore not that ANCHOR uniformly dominates neural models, but that strong benchmark performance can often be matched by transparent controls that do not learn neural representations. The full ANCHOR ladder also shows that this result is not explained by a single heuristic. HumanActivity is largely captured by horizon-aware smoothing, P12 and MIMIC by empirical selection over the shared rule library, and USHCN by sparse and phase-sensitive structure. Thus, different benchmarks reward different forms of history-derived structure.

Paired entity-level comparison. Aggregate metrics can conceal heterogeneity across patients, stations, or activity windows. Table 3 compares AUTOANCHOR with the three representative neural baselines using matched entities. After Holm correction within each metric, AUTOANCHOR retains great MAE improvements in 10 of 12 comparisons. MSE is more mixed: three of 12 differences remain significant, with two favoring AUTOANCHOR and one favoring GraFITi on P12. These results reinforce the audit interpretation: several apparent leaderboard differences are not statistically distinguishable at the entity level, while a smaller set shows consistent advantages in either direction.

4.2 ROBUSTNESS ACROSS FORECASTING CONFIGURATIONS

Lookback and prediction horizon. Figure 2 tests whether the standard-window finding depends on a particularly favorable temporal configuration. Across the 19 evaluated lookback–horizon settings, AUTOANCHOR obtains the lowest MSE in 13 and the lowest MAE in 16. It remains strongest across all six MIMIC settings and in four of five P12 settings for both metrics. The deviations are equally informative. On USHCN, increasing the lookback from $(L, H) = (100, 3)$ to $(150, 3)$ reduces AUTOANCHOR MSE from 1.1591 to 0.1565, indicating that its phase-sensitive behavior depends strongly on sufficient historical context. At fixed $L = 150$, extending the horizon from $H = 3$ to $H = 7$ allows neural models to recover the MSE advantage, although AUTOANCHOR retains the MAE advantage. Small neural gains also appear in several HumanActivity configurations. Hence, the standard-window result is not confined to a single configuration, but neither is the ANCHOR advantage universal: shorter histories and more demanding horizons expose regimes in which learned models become more competitive under increasingly constrained temporal information settings.

4.3 WHAT INFORMATION SUPPORTS ANCHOR PERFORMANCE?

Temporal perturbations. Table 4 probes sensitivity to different components of the observed history. The reported deltas measure degradation under an intervention and should not be interpreted as a model ranking. On P12, MIMIC, and HumanActivity, timestamp shuffling changes AUTOANCHOR MSE by only approximately 0.001, 0.010, and 0.000, respectively, whereas several neural models degrade more substantially. This suggests that competitive ANCHOR performance on these benchmarks does not depend strongly on exact timestamp geometry. USHCN behaves differently. For AUTOANCHOR, timestamp shuffling increases MSE by 0.313 and swapping the two halves of the history increases it by 0.672. Its strong performance on USHCN therefore relies substantially on temporal arrangement, consistent with the phase-sensitive candidates selected for this dataset.

Table 2: **Standard-window benchmark comparison on four IMTS datasets.** Neural models report mean ± standard deviation over three runs (seeds 2024–2026), while ANCHOR methods are deterministic. **Bold** and underlining indicate the best and second-best results, respectively.

Method	HumanActivity		USHCN		P12		MIMIC	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PrimeNet	4.251±0.004	1.702±0.001	0.493±0.002	0.495±0.002	0.795±0.000	0.686±0.000	0.907±0.000	0.661±0.000
NeuralFlows	0.172±0.009	0.315±0.009	0.209±0.026	0.316±0.019	0.406±0.003	0.447±0.003	0.609±0.010	0.531±0.007
CRU	0.139±0.007	0.261±0.009	0.217±0.016	0.318±0.025	0.618±0.005	0.578±0.003	0.590±0.009	0.515±0.005
mTAN	0.099±0.003	0.222±0.005	0.556±0.202	0.502±0.097	0.381±0.004	0.429±0.004	0.941±0.113	0.676±0.046
SeFT	1.379±0.002	0.976±0.001	0.335±0.002	0.408±0.008	0.772±0.002	0.676±0.003	0.923±0.002	0.663±0.001
GNeuralFlow	0.394±0.159	0.454±0.084	0.221±0.042	0.329±0.041	0.821±0.031	0.676±0.010	0.896±0.021	0.645±0.007
GRU-D	0.189±0.063	0.325±0.049	0.210±0.049	0.305±0.031	0.342±0.003	0.399±0.001	0.476±0.010	0.453±0.006
Raindrop	0.092±0.007	0.211±0.007	0.204±0.034	0.303±0.026	0.348±0.002	0.404±0.002	0.675±0.183	0.544±0.087
Warpformer	0.045±0.001	0.123±0.002	0.189±0.060	0.294±0.059	0.306±0.001	0.366±0.002	0.430±0.004	0.403±0.001
tPatchGNN	0.044±0.001	0.125±0.003	0.189±0.040	0.308±0.048	0.313±0.005	0.370±0.005	0.443±0.012	0.408±0.009
GraFITi	0.044±0.001	0.122±0.002	0.169±0.009	0.278±0.025	0.308±0.002	0.364±0.004	0.436±0.046	0.414±0.030
APN	0.042±0.000	0.116±0.001	0.159±0.014	0.261±0.017	0.309±0.001	0.365±0.003	0.429±0.003	0.402±0.002
NAIVEANCHOR	0.061	0.123	0.364	0.329	0.399	0.395	0.456	0.371
EXPOANCHOR	**0.042**	0.115	0.592	0.512	0.329	0.375	0.643	0.497
SPARSEANCHOR	0.061	0.123	0.266	0.281	0.399	0.395	0.456	0.371
ERMANCHOR	0.043	**0.113**	0.166	0.263	**0.302**	**0.359**	**0.378**	0.366
AUTOANCHOR	**0.042**	0.115	**0.157**	**0.219**	**0.302**	**0.359**	0.379	**0.366**

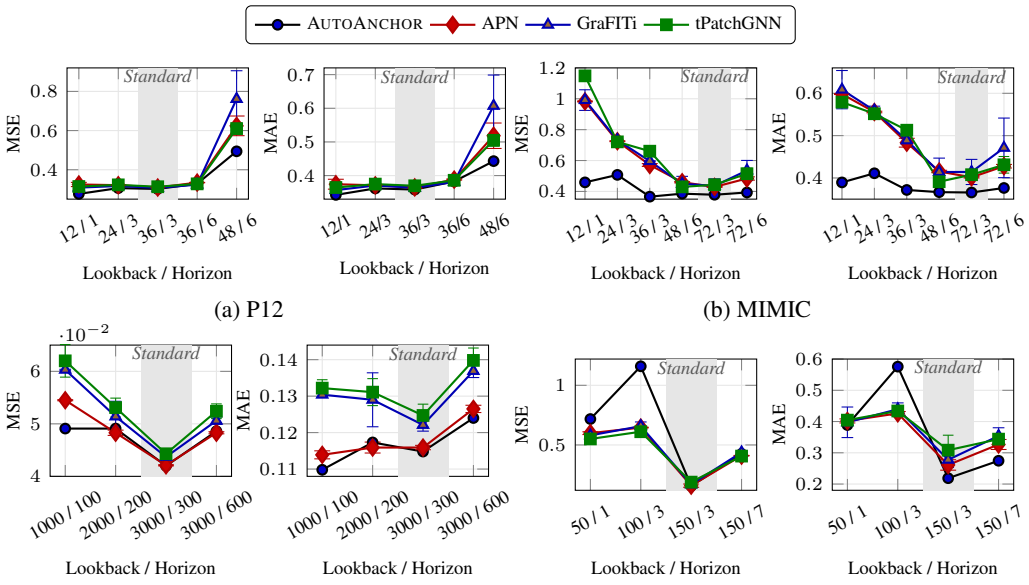
Table 3: **Paired entity-level comparison with neural baselines.** Results are computed at the standard benchmark windows using matched entities and the common valid target-coordinate set. Deltas are AUTOANCHOR minus neural error, so negative values favor AUTOANCHOR. Reported p-values are unadjusted paired entity-level sign-flip tests; Holm-corrected results are summarized in the text.

Dataset	Model	Auto MAE	Neural MAE	Δ MAE	p	Auto MSE	Neural MSE	Δ MSE	p
P12	APN	0.3585	0.3649	**-0.0063**	$< 10^{-4}$	0.3020	0.3032	**-0.0012**	0.4953
	GraFITi	0.3585	**0.3581**	+0.0004	0.6687	0.3020	**0.2953**	+0.0067	0.0002
	tPatchGNN	0.3585	0.3628	-0.0043	$< 10^{-4}$	0.3020	0.3006	+0.0014	0.9509
MIMIC	APN	0.3657	0.3737	-0.0080	$< 10^{-4}$	0.3785	**0.3931**	-0.0147	0.2684
	GraFITi	0.3657	0.4150	**-0.0493**	$< 10^{-4}$	0.3785	0.4365	**-0.0580**	$< 10^{-4}$
	tPatchGNN	0.3657	**0.3676**	-0.0019	0.0020	0.3785	0.3948	-0.0163	0.1895
USHCN	APN	0.2187	0.2844	**-0.0657**	$< 10^{-4}$	0.1565	0.1611	-0.0046	0.8642
	GraFITi	0.2187	0.2730	-0.0543	$< 10^{-4}$	0.1565	0.1669	**-0.0104**	0.6827
	tPatchGNN	0.2187	**0.2668**	-0.0481	0.0011	0.1565	**0.1596**	-0.0031	0.9600
HumanActivity	APN	0.1148	**0.1156**	-0.0009	0.1383	0.0420	**0.0421**	-0.0001	0.8420
	GraFITi	0.1148	0.1226	-0.0078	$< 10^{-4}$	0.0420	0.0439	-0.0019	0.0085
	tPatchGNN	0.1148	0.1323	**-0.0175**	$< 10^{-4}$	0.0420	0.0470	**-0.0051**	$< 10^{-4}$

Small negative perturbation deltas for some neural models should be interpreted as finite-sample or intervention effects rather than evidence that discarding temporal information improves forecasting.

History thinning. Figure 3 evaluates test-time robustness when only a fraction of the observed history is retained. Neural checkpoints and ANCHOR specifications are both obtained from the original, unthinned histories; the intervention is applied only at evaluation time. The results reveal markedly different regimes. On P12, AUTOANCHOR remains competitive even with 10% of the history, narrowly outperforming GraFITi. On MIMIC, its advantage is larger at the same retention level. In contrast, USHCN reverses under aggressive thinning, with neural models substantially outperforming AUTOANCHOR, while HumanActivity shows small APN advantages at several intermediate retention levels. These results should be interpreted as a test of robustness to missing history, rather than as evidence about performance after retraining under sparse observations. Nevertheless, the reversals identify settings in which neural models are more robust than frozen history-derived controls when historical information becomes severely limited.

Uncertainty and candidate attribution. Table 5 reports entity-bootstrap confidence intervals for the full ANCHOR family. The improvements over NAIVEANCHOR are substantial in several settings: the relative MSE reduction reaches 24.3% on P12, 57.0% on USHCN, 30.6% on HumanActivity, and about 17% on MIMIC. This confirms that the ANCHOR results cannot generally be reduced to last-value persistence. The candidate-removal analyses in Section A further identify the structures



(a) P12 (b) MIMIC

(c) HumanActivity (d) USHCN

Figure 2: Lookback–horizon sweep for AUTOANCHOR, APN, GraFITi, and tPatchGNN. The shaded region marks the standard benchmark window. Neural results include error bars showing one standard deviation over three runs, whereas AUTOANCHOR is deterministic.

Table 4: Cross-model temporal perturbations at the standard benchmark windows. Original MSE is the unperturbed error; each Δ is the mean MSE change under the perturbation. Positive values denote degradation and measure sensitivity, not ranking quality.

Dataset	Model	Orig. MSE	Shuffle Δ	Swap Δ	Last-only Δ	Gaps-only Δ
P12	AUTOANCHOR	**0.3020**	+0.0007	+0.2240	+0.0575	+0.4997
	APN	0.3093±0.0011	+0.1467	+0.2023	+0.0558	+0.4873
	GraFITi	0.3075±0.0015	+0.1471	+0.1900	+0.0464	+0.4906
	tPatchGNN	0.3133±0.0053	+0.4549	+0.1826	+0.0416	+0.4827
MIMIC	AUTOANCHOR	**0.3785**	+0.0102	+0.3756	+0.0076	+0.6575
	APN	0.4292±0.0027	+0.1843	+0.2344	+0.0454	+0.5408
	GraFITi	0.4359±0.0455	+0.1802	+0.2359	+0.0679	+0.5348
	tPatchGNN	0.4431±0.0115	+0.1809	+0.2372	+0.0444	+0.5170
USHCN	AUTOANCHOR	**0.1565**	+0.3125	+0.6718	+0.2075	+0.3362
	APN	0.1590±0.0137	+0.0023	+0.0045	+0.1401	+0.0804
	GraFITi	0.1691±0.0093	-0.0015	-0.0019	+0.0473	+0.1010
	tPatchGNN	0.1885±0.0403	-0.0195	+0.3605	+0.0595	+0.1267
HumanActivity	AUTOANCHOR	**0.0420**	+0.0000	+0.0611	+0.0185	+4.2071
	APN	0.0421±0.0001	+0.0349	+0.0581	+0.0166	+4.0887
	GraFITi	0.0437±0.0005	+0.0347	+0.0488	+0.1217	+3.5537
	tPatchGNN	0.0443±0.0009	+0.3731	+0.0554	+0.2989	+3.7591

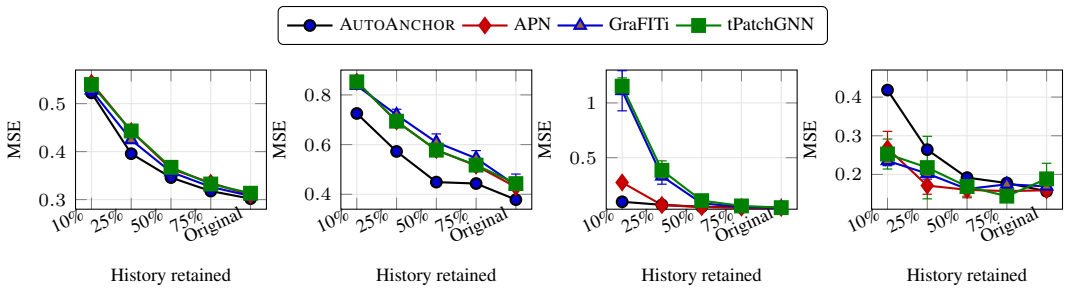
responsible for these gains. P12 and MIMIC are largely explained by empirical selection over the shared rule library; HumanActivity is largely captured by the horizon-aware EMA rule; and USHCN depends strongly on sparse and phase-sensitive candidates. The benchmark audit therefore reveals substantial heterogeneity: similar aggregate forecasting performance can arise from quite different forms of low-complexity historical structure across datasets with distinct temporal characteristics.

4.4 WHERE DO NEURAL MODELS REGAIN AN ADVANTAGE?

The reversals across the previous analyses define the boundary of the ANCHOR claim. Neural models regain advantages on USHCN when historical context is shortened, the prediction horizon is extended, or the available history is aggressively thinned. Smaller reversals also occur on HumanActivity across several lookback–horizon and thinning settings. These regimes are consistent with the limitations of the variable-wise ANCHOR family, which does not explicitly learn latent state,

Table 5: Entity-bootstrap 95% confidence intervals over patients, stations, or activity windows for all ANCHOR-family global metrics. MSE red. is the relative MSE reduction against NAIVEANCHOR within each dataset; negative values indicate higher MSE than NAIVEANCHOR.

Dataset	Method	MAE (95% CI)	MSE (95% CI)	MSE red. (%)
P12	NaiveAnchor	0.3951 [0.3834, 0.4076]	0.3992 [0.3741, 0.4273]	0.0%
	ExpoAnchor	0.3753 [0.3653, 0.3856]	0.3286 [0.3078, 0.3521]	17.7%
	SparseAnchor	0.3951 [0.3834, 0.4076]	0.3992 [0.3741, 0.4273]	0.0%
	ERMAnchor	0.3585 [0.3488, 0.3686]	0.3020 [0.2820, 0.3245]	24.3%
	AUTOANCHOR	0.3585 [0.3488, 0.3686]	0.3020 [0.2820, 0.3245]	24.3%
MIMIC	NaiveAnchor	0.3708 [0.3569, 0.3853]	0.4561 [0.4136, 0.5034]	0.0%
	ExpoAnchor	0.4974 [0.4809, 0.5139]	0.6431 [0.5850, 0.7080]	-41.0%
	SparseAnchor	0.3709 [0.3571, 0.3855]	0.4560 [0.4137, 0.5033]	0.0%
	ERMAnchor	0.3658 [0.3537, 0.3785]	0.3782 [0.3396, 0.4242]	17.1%
	AUTOANCHOR	0.3657 [0.3536, 0.3784]	0.3785 [0.3397, 0.4242]	17.0%
USHCN	NaiveAnchor	0.3291 [0.2787, 0.3837]	0.3640 [0.2640, 0.4791]	0.0%
	ExpoAnchor	0.5117 [0.4519, 0.5751]	0.5923 [0.4955, 0.6994]	-62.8%
	SparseAnchor	0.2812 [0.2382, 0.3266]	0.2661 [0.2029, 0.3367]	26.9%
	ERMAnchor	0.2634 [0.2315, 0.2963]	0.1662 [0.1289, 0.2074]	54.3%
	AUTOANCHOR	0.2187 [0.1868, 0.2528]	0.1565 [0.1209, 0.1958]	57.0%
HumanActivity	NaiveAnchor	0.1225 [0.1116, 0.1341]	0.0605 [0.0480, 0.0746]	0.0%
	ExpoAnchor	0.1148 [0.1054, 0.1248]	0.0420 [0.0342, 0.0504]	30.6%
	SparseAnchor	0.1225 [0.1116, 0.1341]	0.0605 [0.0480, 0.0746]	0.0%
	ERMAnchor	0.1131 [0.1039, 0.1230]	0.0428 [0.0346, 0.0519]	29.2%
	AUTOANCHOR	0.1148 [0.1054, 0.1248]	0.0420 [0.0342, 0.0504]	30.6%



(a) P12 (b) MIMIC (c) HumanActivity (d) USHCN

Figure 3: **Robustness to history thinning.** Global scaled MSE after retaining different fractions of the observed history at evaluation time. Curves show mean performance, with error bars denoting one standard deviation over three neural training runs; AUTOANCHOR is deterministic.

cross-variable dependencies, or anticipatory regime dynamics. The audit therefore does not imply that neural modeling is unnecessary for IMTS forecasting. Rather, it separates settings in which benchmark performance is largely explained by history-derived structure from settings in which the additional modeling capacity of neural approaches yields measurable gains.

5 CONCLUSION

This paper audits what current IMTS forecasting benchmarks actually require from a predictive model. By treating the ANCHOR family as a set of transparent diagnostic controls, we show that substantial benchmark performance can often be explained by simple history-derived structure under matched evaluation protocols. At the same time, the lookback–horizon, perturbation, and history-thinning analyses identify regimes in which these controls break down and neural models recover measurable advantages. These findings do not argue against neural modeling; rather, they clarify when its additional representational capacity is supported by benchmark evidence. More broadly, progress in IMTS forecasting should be demonstrated not only by lower aggregate error, but by improvements beyond strong historical controls in settings that genuinely demand irregular temporal and cross-variable reasoning across diverse datasets, horizons, and observation sparsity regimes.

AI USE STATEMENT

In this work, ChatGPT, a Generative AI application, was used only to improve the clarity, fluency, and grammatical quality of the manuscript, as the authors are non-native English speakers. The authors provided the original words, technical content, arguments, structure, and key contexts, while AI assistance was limited to linking sentences, rephrasing passages, and polishing language. It was not used to generate the research ideas, methodology, experiments, results, or scientific claims.

REFERENCES

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 21325–21337, 2021.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8:6085, 2018.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6572–6583, 2018.

Ranak Roy Chowdhury, Jiacheng Li, Xiyuan Zhang, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. Primenet: Pre-training for irregular multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 7184–7192, 2023.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4353–4363, 2020.

Alistair Johnson, Tom Pollard, and Roger Mark. MIMIC-III Clinical Database. *PhysioNet*, September 2016. doi: 10.13026/C2XW26. Version 1.4.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6696–6707, 2020.

Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. HyperIMTS: Hypergraph neural network for irregular multivariate time series forecasting. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 35502–35518, 2025.

Jiexi Liu, Meng Cao, and Songcan Chen. Timecheat: A channel harmony strategy for irregularly sampled multivariate time series analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 18861–18869, 2025.

Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 23873–23881, 2026.

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-patch: Hierarchical patch GNN for irregular multivariate time series. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 41494–41519, 2025.

Matthew J. Menne and Claude N. Jr. Williams. United States Historical Climatology Network (USHCN), Version 2.5.x, 2012.

Giangiacomo Mercatali, Andre Freitas, and Jie Chen. Graph neural flows for unveiling systemic interactions among irregularly sampled time series. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 57183–57206, 2024.

Fernando Moreno-Pino, Álvaro Arroyo, Harrison Waldon, Xiaowen Dong, and Álvaro Cartea. Rough transformers: Lightweight and continuous time series modelling through signature patching. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 106264–106294, 2024.

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

Yulia Rubanova, Ricky T. Q. Chen, and David K. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5321–5331, 2019.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 19388–19405, 2022.

Satya Narayan Shukla and Benjamin M. Marlin. Interpolation-prediction networks for irregularly sampled time series. *arXiv preprint arXiv:1909.07782*, 2019.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Ikaro Silva, George Moody, Daniel J. Scott, Leo A. Celi, and Roger G. Mark. Predicting in-hospital mortality of patients in icu: The physionet/computing in cardiology challenge 2012. In *Computing in Cardiology*, volume 39, pp. 245–248, 2012.

Yuxi Wei, Juntong Peng, Tong He, Chenxin Xu, Jian Zhang, Shirui Pan, and Siheng Chen. Compatible transformer for irregularly sampled multivariate time series. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 1409–1414, 2023.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. Grafiti: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 16255–16263, 2024.

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

Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 3273–3285, 2023.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 60179–60196, 2024.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

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

Ziyu Zhou, Yiming Huang, Yanyun Wang, Yuankai Wu, James Kwok, and Yuxuan Liang. Revitalizing canonical pre-alignment for irregular multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 29115–29123, 2026.

A ADDITIONAL ANCHOR-FAMILY ANALYSES

This technical appendix expands the benchmark audit along three axes. First, it decomposes which parts of the Anchor family are responsible for the standard-window results. Second, it reports operational properties, including deterministic CPU inference cost. Third, it studies whether the Anchor gains over pure persistence are concentrated in particular volatility or observation-density regimes. These analyses are not intended to introduce a second leaderboard; they are included to make the diagnostic interpretation of the main results auditable.

Candidate attribution. Tables 6 and 8 show that the standard-window Anchor results are not produced by one universal heuristic. For P12, ERM selection over the shared candidate library accounts for the improvement over NaiveAnchor: using rules only reverts to the NaiveAnchor error, while removing trend has essentially no effect. MIMIC follows a similar pattern, with ERMAnchor and AUTOANCHOR nearly identical and substantially below the rules-only condition in MSE. HumanActivity is different: the horizon-aware EMA rule already reaches the best MSE, and removing EMA increases MSE from 0.0420 to 0.0483. USHCN is the clearest case where the structural components matter: removing sparse and phase-sensitive candidates increases MSE by 0.0283 and 0.0616, respectively. Thus, all four standard settings are substantially matched by history-derived controls, but the responsible low-complexity structure differs by dataset.

Computational efficiency. Table 7 reports CPU inference measurements for the Anchor variants. The absolute runtime differs by dataset size and number of predicted coordinates, but all variants remain lightweight deterministic procedures. The adaptive variants add modest overhead relative to NaiveAnchor because they evaluate a richer rule set, yet AUTOANCHOR still produces hundreds to thousands of predictions per second on CPU. This matters for the audit framing: the controls are not only non-neural, but also computationally simple enough to serve as routine diagnostic baselines.

Temporal-information probes. Table 9 isolates the temporal information used by AUTOANCHOR itself. On P12, MIMIC, and HumanActivity, timestamp shuffling causes little degradation, whereas stronger interventions such as swapping history segments or retaining only time gaps can substantially increase error. This indicates that competitive Anchor performance on these datasets does not depend strongly on precise timestamp geometry alone. USHCN behaves differently: shuffling timestamps, swapping history halves, and retaining only the last observation all substantially increase error, supporting the main-text finding that AutoAnchor's USHCN performance depends much more strongly on temporal arrangement.

Improvement over persistence. Table 10 compares AUTOANCHOR directly with NaiveAnchor at the entity level. The gains over persistence are large and statistically clear for P12, USHCN, and HumanActivity, and MIMIC shows a strong MSE improvement even though the MAE interval includes zero. This paired comparison is important because it separates the claim "history is useful" from the stronger claim made by the Anchor family: selecting among several transparent history-derived structures gives a meaningfully stronger diagnostic control than last-value persistence alone.

Volatility and density strata. Tables 11 and 12 show where AUTOANCHOR improves over NaiveAnchor. Across all four datasets, the MSE reduction is positive in every volatility and density stratum. The largest relative reductions often occur in more difficult strata: high-volatility targets in P12, MIMIC, USHCN, and HumanActivity, and low- or high-density targets in USHCN. These stratified results suggest that the Anchor improvements are not confined to trivially stable or densely observed cases. Instead, transparent history-derived rules continue to explain a substantial fraction of error even when recent history is volatile or observation density changes.

Benchmark framework reproducibility. We provide the complete source code and reproducible evaluation pipeline at the anonymous repository https://anonymous.4open.science/r/Anchor-Family/. The released implementation includes the ANCHOR variants and the scripts required to reproduce the benchmark analyses on local machines.

In conclusion, the appendix reinforces the main diagnostic finding: strong performance across current IMTS benchmarks can often be explained by simple, transparent structures already present in

Table 6: Candidate-removal ablations. Entries are global scaled MSE/MAE; deltas are relative to full AUTOANCHOR on the same dataset.

Dataset	Condition	MSE	ΔMSE	MAE	ΔMAE
P12	full AUTOANCHOR	0.3020	+0.0000	0.3585	+0.0000
	ERM only	0.3020	+0.0000	0.3585	+0.0000
	rules only	0.3992	+0.0972	0.3951	+0.0366
	minus EMA	0.3157	+0.0137	0.3683	+0.0098
	minus trend	0.3019	-0.0001	0.3585	-0.0001
USHCN	full AUTOANCHOR	0.1565	+0.0000	0.2187	+0.0000
	ERM only	0.1662	+0.0097	0.2634	+0.0447
	rules only	0.1565	+0.0000	0.2187	+0.0000
	minus sparse	0.1848	+0.0283	0.2719	+0.0532
	minus phase	0.2181	+0.0616	0.2605	+0.0418
HumanActivity	full AUTOANCHOR	0.0420	+0.0000	0.1148	+0.0000
	ERM only	0.0428	+0.0009	0.1131	-0.0016
	rules only	0.0420	+0.0000	0.1148	+0.0000
	minus EMA	0.0483	+0.0064	0.1193	+0.0046
	minus sparse/trend/phase	0.0420	+0.0000	0.1148	+0.0000
MIMIC	full AUTOANCHOR	0.3785	+0.0000	0.3657	+0.0000
	ERM only	0.3782	-0.0002	0.3658	+0.0001
	rules only	0.4560	+0.0775	0.3709	+0.0053
	minus EMA	0.3878	+0.0094	0.3684	+0.0027
	minus sparse	0.3782	-0.0002	0.3658	+0.0001
	minus trend	0.3796	+0.0011	0.3673	+0.0016
	minus phase	0.3782	-0.0003	0.3660	+0.0003

Table 7: Anchor-family computational efficiency across datasets. **Bold** marks the best value for each dataset and metric, while underlining marks the second best. Lower is better for CPU Total and CPU ms/Pred.; higher is better for Pred./s.

Dataset	Metric	NaiveAnchor	ExpoAnchor	SparseAnchor	ERMAnchor	AUTOANCHOR
P12	CPU Total (s)	9.1592	9.3953	**9.1545**	9.4662	9.6304
	CPU ms/Pred.	0.4245	0.4354	**0.4242**	0.4387	0.4463
	Pred./s	2356.0	2296.8	**2357.2**	2279.6	2240.7
USHCN	CPU Total (s)	0.3055	**0.2843**	0.3069	0.3250	0.3414
	CPU ms/Pred.	0.8301	**0.7725**	0.8339	0.8832	0.9278
	Pred./s	1204.7	**1294.4**	1199.2	1132.2	1077.8
HumanActivity	CPU Total (s)	1.5235	1.6863	**1.4389**	1.6848	1.6722
	CPU ms/Pred.	0.2578	0.2853	**0.2435**	0.2851	0.2829
	Pred./s	3879.1	3504.7	**4107.2**	3507.8	3534.3
MIMIC	CPU Total (s)	**18.6622**	18.7083	19.1243	19.1358	19.4374
	CPU ms/Pred.	**1.4117**	1.4151	1.4466	1.4475	1.4703
	Pred./s	**708.4**	706.6	691.3	690.9	680.1

the observed history. The specific mechanism varies across datasets, ranging from persistence and smoothing to recurrence and adaptive rule selection, but these controls remain lightweight and consistently improve over pure persistence across diverse volatility and observation-density regimes. At the same time, the temporal probes reveal meaningful dataset differences, with some benchmarks requiring little precise timestamp structure and others depending strongly on temporal arrangement.

Table 8: Anchor-family ablation ladder. Lower is better for MSE and MAE. Bold marks the best Anchor-family value within each dataset; underline marks the second best. Positive deltas indicate improvement over NaiveAnchor.

Dataset	Method	MSE	Δ MSE vs Naive	MAE	Δ MAE vs Naive
P12	NaiveAnchor	0.3992	+0.0000	0.3951	+0.0000
	ExpoAnchor	0.3286	+0.0706	0.3753	+0.0198
	SparseAnchor	0.3992	+0.0000	0.3951	+0.0000
	ERMAnchor	**0.3020**	+0.0972	**0.3585**	+0.0366
	AUTOANCHOR	**0.3020**	+0.0972	**0.3585**	+0.0366
USHCN	NaiveAnchor	0.3640	+0.0000	0.3291	+0.0000
	ExpoAnchor	0.5923	-0.2284	0.5117	-0.1826
	SparseAnchor	0.2661	+0.0978	0.2812	+0.0479
	ERMAnchor	0.1662	+0.1978	0.2634	+0.0657
	AUTOANCHOR	**0.1565**	+0.2074	**0.2187**	+0.1104
HumanActivity	NaiveAnchor	0.0605	+0.0000	0.1225	+0.0000
	ExpoAnchor	**0.0420**	+0.0185	0.1148	+0.0078
	SparseAnchor	0.0605	+0.0000	0.1225	+0.0000
	ERMAnchor	0.0428	+0.0177	**0.1131**	+0.0094
	AUTOANCHOR	**0.0420**	+0.0185	0.1148	+0.0078
MIMIC	NaiveAnchor	0.4561	+0.0000	0.3708	+0.0000
	ExpoAnchor	0.6431	-0.1870	0.4974	-0.1266
	SparseAnchor	0.4560	+0.0002	0.3709	-0.0001
	ERMAnchor	**0.3782**	+0.0779	0.3658	+0.0050
	AUTOANCHOR	0.3785	+0.0777	**0.3657**	+0.0051

Table 9: Temporal-information tests for AUTOANCHOR at the standard windows. Original MSE is the unperturbed error; each Δ is the MSE change under the corresponding perturbation. Positive values indicate degradation. Lower sensitivity does not necessarily imply better predictive performance.

Dataset	Original MSE	Shuffle Δ	Swap Δ	Last-only Δ	Gaps-only Δ
P12	0.3020	+0.0007	+0.2240	+0.0575	+0.4997
MIMIC	0.3785	+0.0102	+0.3756	+0.0076	+0.6575
USHCN	0.1565	+0.3125	+0.6718	+0.2075	+0.3362
HumanActivity	0.0420	+0.0000	+0.0611	+0.0185	+4.2071

Table 10: Paired entity-level comparison of AUTOANCHOR and NaiveAnchor. Deltas are AUTOANCHOR minus NaiveAnchor, so negative values favor AUTOANCHOR. Intervals use paired entity bootstrap; p-values use paired entity-level sign flips. This comparison measures improvement over persistence within the Anchor family and is not a paired test against neural models.

Dataset	Metric	Delta (95% CI)	p	Entities
P12	MAE	-0.0366 [-0.0436, -0.0299]	$< 10^{-4}$	1,181
	MSE	-0.0972 [-0.1125, -0.0820]	$< 10^{-4}$	1,181
MIMIC	MAE	-0.0051 [-0.0126, +0.0021]	0.1714	1,534
	MSE	-0.0777 [-0.1010, -0.0545]	$< 10^{-4}$	1,534
USHCN	MAE	-0.1104 [-0.1562, -0.0693]	$< 10^{-4}$	112
	MSE	-0.2074 [-0.3165, -0.1149]	$< 10^{-4}$	112
HumanActivity	MAE	-0.0078 [-0.0130, -0.0027]	0.0041	218
	MSE	-0.0185 [-0.0273, -0.0107]	$< 10^{-4}$	218

Table 11: AUTOANCHOR and NaiveAnchor errors stratified by recent-history volatility. Positive MSE reduction favors AUTOANCHOR; strata are dataset-specific target-level tertiles.

Dataset	Stratum	Auto MAE	Naive MAE	Auto MSE	Naive MSE	MSE red. (%)
P12	low	**0.1878**	0.1882	**0.1280**	0.1396	+8.3
	mid	**0.3860**	0.4240	**0.2851**	0.3641	+21.7
	high	**0.5018**	0.5733	**0.4930**	0.6939	+29.0
MIMIC	low	0.2376	**0.2136**	**0.2123**	0.2170	+2.1
	mid	**0.3607**	0.3646	**0.2870**	0.3290	+12.8
	high	**0.4987**	0.5342	**0.6361**	0.8224	+22.7
USHCN	low	**0.0295**	0.0320	**0.0075**	0.0122	+38.3
	mid	**0.3400**	0.4622	**0.2495**	0.4717	+47.1
	high	**0.2875**	0.4943	**0.2133**	0.6088	+65.0
HumanActivity	low	**0.0470**	0.0535	**0.0059**	0.0072	+18.0
	mid	**0.1053**	0.1125	**0.0292**	0.0337	+13.4
	high	**0.1921**	0.2015	**0.0908**	0.1406	+35.4

Table 12: AUTOANCHOR and NaiveAnchor errors stratified by observation density. Positive MSE reduction favors AUTOANCHOR; strata are dataset-specific target-level tertiles.

Dataset	Stratum	Auto MAE	Naive MAE	Auto MSE	Naive MSE	MSE red. (%)
P12	low	**0.3544**	0.3753	**0.2985**	0.3778	+21.0
	mid	**0.3760**	0.4170	**0.3066**	0.4138	+25.9
	high	**0.3452**	0.3930	**0.3009**	0.4060	+25.9
MIMIC	low	**0.3642**	0.3792	**0.3207**	0.4034	+20.5
	mid	**0.3822**	0.3863	**0.3786**	0.4418	+14.3
	high	0.3506	**0.3470**	**0.4361**	0.5232	+16.6
USHCN	low	**0.2565**	0.3914	**0.1781**	0.4589	+61.2
	mid	**0.2236**	0.2952	**0.1832**	0.3221	+43.1
	high	**0.1760**	0.3005	**0.1084**	0.3105	+65.1
HumanActivity	low	**0.1340**	0.1392	**0.0510**	0.0757	+32.6
	mid	**0.1111**	0.1174	**0.0391**	0.0527	+25.8
	high	**0.0992**	0.1110	**0.0357**	0.0531	+32.7