TSFMAUDIT: DATA CONTAMINATION AUDITING IN FORECASTING TIME SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series foundation models (TSFMs) are increasingly pretrained on large corpora, raising concerns that evaluation datasets may have been exposed during pretraining and thus yield overly optimistic performance estimates. Auditing such contamination is challenging in time series because signals are continuous and heterogeneous, and often lack corpus documentation. We formalize pretraining source-exposure auditing for TSFMs and propose TSFMAUDIT, a method based on probe adaptation dynamics. Our key hypothesis is that prior exposure can support efficient adaptation: a short fine-tuning probe may achieve faster loss reduction with smaller backbone movement. Candidate-to-reference differences and ratios compare this behavior across heterogeneous forecasters. We evaluate TSFMAUDIT on six TSFMs and 137 datasets using documented training-source evidence, with a separate evaluation on 50 TIME datasets. Against documentation-derived labels, combining reference-relative and candidate features yields candidate-averaged MCC 0.317, compared with 0.050 for the strongest tested scalar baseline and 0.169 for candidate-only dynamics. We further validate TSFMAUDIT under ground-truth membership by pretraining Chronos-T5-mini and Moirai-1.0-R-Small from random initialization on controlled dataset manifests, obtaining MCCs of 0.356 and 0.618, respectively.

1 INTRODUCTION

Time series foundation models (TSFMs) have recently attracted growing attention because of their strong cross domain capabilities across applications such as finance, transportation, and health (Li et al., 2025b;a; Rahimikia et al., 2025; Pulido & Rodrigues, 2026). Although there have been promising attempts to train TSFMs on purely synthetic data, current state-of-the-art models still rely heavily on large scale real world corpora collected from diverse sources (Goswami et al., 2024; Xie et al., 2026). At the same time, widely used evaluation benchmarks are themselves assembled from multiple public sources (Aksu et al., 2024; Qiao et al., 2026). This shared multi-source pretraining evaluation pipeline creates a substantial risk of contamination: data used for evaluation may already have been exposed during pretraining (Meyer et al., 2026).

Such exposure can artificially inflate downstream performance, and thereby bias our assessment of TSFM generalization (Meyer et al., 2026; Aksu et al., 2024; Dong et al., 2024). More importantly, contamination may arise unintentionally (Meyer et al., 2026; Xu et al., 2024). In time series data, the same underlying signal can reappear under rescaling, renaming, rewindowing, or other preprocessing changes while preserving nearly identical content (Meyer et al., 2026; Godahewa et al., 2021). For example, the Monash archive's Elecdemand dataset is a $1/1000$-scaled subset of Australian Electricity Demand (Meyer et al., 2026). To ensure the fairness and validity of TSFM benchmarking, we therefore need efficient methods for contamination auditing (Meyer et al., 2026; Aksu et al., 2024; Dekoninck et al., 2024; Choi et al., 2025).

Contamination auditing has been studied extensively in large language models (LLMs) and related domains. Popular families of methods include overlap detection, which searches for direct or near-duplicate benchmark content, memorization extraction, which tests whether a model can reproduce training examples, perplexity-based tests, which flag unusually low model loss, and membership inference attacks, which calibrate a score for deciding whether an example was used in training, often using shadow models, i.e., auxiliary models trained on controlled data splits (Yang et al., 2023; Shi

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.

et al., 2024; Carlini et al., 2021; Shokri et al., 2017; Carlini et al., 2022). However, these methods do not transfer naturally to TSFMs because time series data are continuous, heterogeneous, and often only partially documented at the corpus level (Woo et al., 2024; Das et al., 2024; Qiu et al., 2024; Garza et al., 2024; Goswami et al., 2024). For instance, leakage can indeed lead to abnormally low loss, and performance based tests such as ConStat compare primary and reference benchmarks relative to reference models (Dekoninck et al., 2024); however, a smooth seasonal series may also yield low loss even when the model has never seen it before, making static loss or perplexity style criteria ambiguous (Liu et al., 2022; Jiang et al., 2024). Likewise, membership inference attacks typically require multiple shadow models to calibrate membership scores, but reproducing TSFM pretraining at that scale is expensive (Shokri et al., 2017; Carlini et al., 2022; Hayes et al., 2025). These limitations call for a contamination auditing method designed specifically for TSFMs.

We make the following empirical observation: in TiRex probes across all 137 datasets, documented-positive datasets exhibit faster relative loss reduction with smaller backbone movement than documented-negative datasets (Figure 1). This observation motivates adaptation efficiency as an audit signal; Table 4 evaluates its predictive contribution, and Section 5.6 tests the signal under manifest-defined membership. We therefore propose TSFMAUDIT, a contamination auditing framework for TSFMs based on learning dynamics. TSFMAUDIT extracts probe time signals such as loss reduction and parameter displacement to infer contamination risk. We summarize our contributions as follows:

- To the best of our knowledge, this is the first work to study pretraining contamination auditing for forecasting TSFMs. We formalize the dataset level audit objective for contamination with documentation derived supervision for public candidates and manifest defined membership for controlled models.
- We propose TSFMAUDIT[1], a contamination auditing framework for TSFMs that infers contamination risk from probe-time adaptation dynamics relative to heterogeneous reference forecasters.
- We evaluate TSFMAUDIT on six TSFMs using their reported pretraining corpora, GIFT-Eval (Aksu et al., 2024), and the TIME (Qiao et al., 2026) benchmark, covering 187 datasets in total. The main comparison includes ten scalar and dynamic baselines. Controlled pre-training, complementary checkpoints, and candidate-held-out evaluation provide additional tests of membership and transfer.

2 RELATED WORK

TSFMs and Benchmarks Contamination Concerns. Recent time series forecasting research has moved from small dataset-specific evaluations toward standardized large-scale benchmarks such as M4, LSF, and the Monash archive and, with the rise of TSFMs, broad zero-shot suites such as GIFT-Eval, TSFM-Bench, and TIME (Makridakis et al., 2020; Godahewa et al., 2021; Wu et al., 2021; Aksu et al., 2024; Li et al., 2025b; Qiao et al., 2026). In parallel, TSFMs themselves are increasingly trained on large heterogeneous corpora collected from diverse real-world and synthetic sources, reflecting the field's ambition to build universal forecasters across domains (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Auer et al., 2025). However, this development has also raised a new concern about benchmark validity: many modern benchmarks are assembled by reusing public datasets, evaluation data may overlap with, or remain implicitly connected to, pretraining corpora, leading to hidden information leakage and overly optimistic estimates of zero-shot generalization (Aksu et al., 2024; Meyer et al., 2026). This problem is particularly acute in time series, where fragmented naming and versioning, weak semantic transparency, and strong temporal dependence make contamination harder to detect than in other modalities (Meyer et al., 2026). These concerns motivate benchmark designs using fresher data sources and stricter evaluation protocols to reduce leakage and improve TSFM comparison (Aksu et al., 2024; Qiao et al., 2026).

Contamination Auditing in LLMs. Large language models have motivated a broad literature on contamination auditing. Existing methods mainly fall into four families: corpus-side overlap detection based on n-gram or token-level matching, memorization or extraction attacks that test whether benchmark content can be reproduced, perplexity-based methods that treat unusually low

[1]Code is available at: `https://anonymous.4open.science/r/TSFMAudit-FCC5`.

perplexity or high likelihood as evidence of prior exposure, and membership inference attacks (MIA) such as LiRA that estimate whether a sample was included in training (Brown et al., 2020; OpenAI, 2023; Carlini et al., 2021; 2023; Dekoninck et al., 2024; Shi et al., 2024; Oren et al., 2024; Carlini et al., 2022).

Despite the rich literature on contamination auditing in LLMs, these methods do not transfer directly to TSFMs. Overlap-based and extraction-based approaches rely on discrete tokens or verbatim matching, whereas time series are continuous values and the same underlying signal may reappear after rescaling, resampling, rewindowing, or renaming, making exact matching both brittle and incomplete (Yang et al., 2023; Jiang et al., 2024; Meyer et al., 2026). Perplexity- or loss-based criteria are also especially unreliable in time series, because many datasets are intrinsically easy to predict: smooth trends or strong seasonality can produce low error even without prior exposure, so static low loss alone is not a valid proxy for contamination (Dekoninck et al., 2024; Jiang et al., 2024; Sainz et al., 2023; Dong et al., 2024). Membership inference methods face a different mismatch: they are typically designed for sample-level membership and often require shadow model calibration, while our audit aggregates source series exposure at the dataset level and repeating full TSFM pretraining for each audit would be expensive (Shokri et al., 2017; Carlini et al., 2022).

Our Positioning. Our work addresses this gap by formulating contamination auditing for TSFMs in a realistic setting where the auditor can probe model behavior and run limited fine tuning, but cannot inspect the full pretraining corpus or reproduce pretraining at scale. Our method TSFMAUDIT uses adaptation dynamics, rather than static loss alone, as the primary signal, following the broader observation that training dynamics can reveal dataset and model behavior (Swayamdipta et al., 2020; Frankle et al., 2020; Zhang et al., 2017). ConStat compares a primary and a reference benchmark relative to reference models (Dekoninck et al., 2024); KDS measures changes in sample-embedding kernel similarities after fine-tuning (Choi et al., 2025); loss-trajectory attacks use changes in prediction loss (Liu et al., 2022). TSFMAUDIT instead measures forecasting-loss improvement relative to backbone movement and compares the resulting traces on the same dataset across heterogeneous references. Its audit stage does not require training shadow TSFMs from scratch.

3 PROBLEM FORMULATION

In this section, we formalize the contamination auditing problem for multivariate TSFMs. We introduce the forecasting setup, a notion of pretraining contamination, and the auditing objective.

Setup We consider a time series forecasting setting with lookback length L, forecast horizon H, and number of variables (channels) P. Let $f_{\boldsymbol{\theta}} : \mathbb{R}^{L\times P} \to \mathbb{R}^{H\times P}$ denote a candidate TSFM with parameter $\boldsymbol{\theta}$. For sample i, let $\mathbf{H}_i \in \mathbb{R}^{L\times P}$ be the historical input and $\mathbf{Y}_i \in \mathbb{R}^{H\times P}$ be the future target. Following the supervised forecasting formulation, we define the i-th sample and an audited dataset with n samples by

$$\mathbf{Z}_i := (\mathbf{H}_i; \mathbf{Y}_i) \in \mathbb{R}^{(L+H)\times P}, \qquad \mathcal{D} := \{\mathbf{Z}_i\}_{i=1}^{n}.$$

Contamination Let $\mathcal{D}^{\mathrm{pt}} = \bigcup_{j=1}^{J} \mathcal{D}_j^{\mathrm{pt}}$, denote the pretraining corpus associated with the candidate model $f_{\boldsymbol{\theta}}$, where J is the number of pretraining sources. Here, $\mathcal{D}^{\mathrm{pt}}$ is understood as the collection of pretraining source time series. For each audited sample $\mathbf{Z}_i$, let $\mathbf{X}_i$ denote the underlying source time series from which $\mathbf{Z}_i$ is extracted, for example through windowing or preprocessing. Accordingly, we define the sample-level contamination indicator for sample $\mathbf{Z}_i$ as

$$c_i = c\big(\mathbf{Z}_i; \mathcal{D}^{\mathrm{pt}}\big) := \begin{cases} 1, & \mathbf{X}_i \in \mathcal{D}^{\mathrm{pt}}, \\ 0, & \mathbf{X}_i \notin \mathcal{D}^{\mathrm{pt}}. \end{cases}$$

The corresponding dataset-level contamination ratio is defined as $C(\mathcal{D}, \mathcal{D}^{\mathrm{pt}}) := \frac{1}{n}\sum_{i=1}^{n} c_i \in [0, 1]$. Under this definition, $C(\mathcal{D}, \mathcal{D}^{\mathrm{pt}}) = 0$ means that no audited sample is derived from a source time series that appears in the pretraining corpus, while $C(\mathcal{D}, \mathcal{D}^{\mathrm{pt}}) = 1$ means that all audited samples are derived from source time series contained in the pretraining corpus.

Objective Given a candidate model $f_{\boldsymbol{\theta}}$ and an audited dataset $\mathcal{D}$, the goal of contamination auditing is to estimate whether $\mathcal{D}$ has been exposed during pretraining. More precisely, we seek a scoring function

$$S(f_{\boldsymbol{\theta}}, \mathcal{D}) \in [0, 1]$$

that quantifies the contamination risk of the pair $(f_{\boldsymbol{\theta}}, \mathcal{D})$, where larger values indicate stronger evidence of pretraining contamination. A binary decision can then be obtained by thresholding the score:

$$\widehat{S}(f_{\boldsymbol{\theta}}, \mathcal{D}) := \begin{cases} 1, & S(f_{\boldsymbol{\theta}}, \mathcal{D}) > \tau, \\ 0, & S(f_{\boldsymbol{\theta}}, \mathcal{D}) \leq \tau, \end{cases}$$

where $\tau \in [0, 1]$ is a decision threshold, and $\widehat{S} = 1$ denotes "contaminated". Under the idealized formulation, one may directly take $S(f_{\boldsymbol{\theta}}, \mathcal{D}) = C(\mathcal{D}, \mathcal{D}^{\mathrm{pt}})$. In practice, C is generally unobservable because the pretraining corpus is only partially documented. We therefore work with an observable proxy $\widetilde{C}(f_{\boldsymbol{\theta}}, \mathcal{D}) \in \{0, 1\}$ derived from official training-source documentation; see Appendix A.2 for the full definition and Appendix A.3 for transformed duplicate semantics.

4 TSFMAUDIT: DEBIASED ADAPTATION EFFICIENCY

We now describe TSFMAUDIT, which constructs the score $S\left(f_{\boldsymbol{\theta}}, \mathcal{D}\right)$ by evaluating how efficiently a candidate model adapts to $\mathcal{D}$ during fine tuning. Figure 2 provides an overview of the full pipeline. The design of TSFMAUDIT is guided by two core observations.

Observation 1: Contamination can appear as unusually efficient adaptation. If a model has already encoded predictive structure specific to $\mathcal{D}$ during pretraining, it may require less adaptation when subsequently fine tuning on $\mathcal{D}$. In such cases, the model can exhibit faster loss reduction with relatively little backbone movement (as shown in Figure 1). We treat this contrast as behavioral evidence of possible contamination. This interpretation is motivated by prior work on neural network memorization (Zhang et al., 2017; Feldman & Zhang, 2020) and recent detection (audit) methods that use fine tuning induced change as evidence of leakage (Choi et al., 2025).

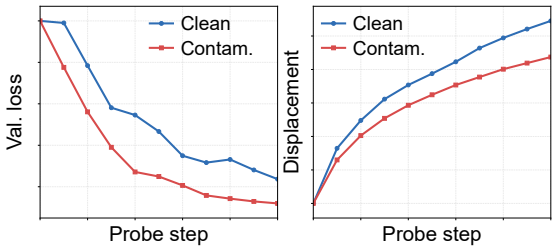

Figure 1: Measured TiRex traces over 137 datasets.

Observation 2: Dataset difficulty can bias adaptation behavior. Fast adaptation is not unique to contaminated models: a clean model may adapt just as quickly if $\mathcal{D}$ is easy. To compare difficulty-dependent responses, we probe heterogeneous reference forecasters on the same $\mathcal{D}$. ScratchCNN, ScratchTransformer, Stat, and VisionTS provide convolutional, attention-based, statistical, and image-pretrained views. These comparative responses reduce reliance on a single inductive bias; they are not architecture-matched counterfactual checkpoints. Their utility is tested empirically. This strategy follows the high-level motivation of reference-based tests such as ConStat (Dekoninck et al., 2024).

4.1 ADAPTATION TRACES FOR AUDITING

Probe Traces We construct the auditing signal by probing the candidate model $f_{\boldsymbol{\theta}}$ and a reference suite $\mathcal{R} = \{f_{\boldsymbol{\theta}^{(r)}}\}_{r=1}^{|\mathcal{R}|}$ under a shared fine tuning protocol on the audited dataset $\mathcal{D}$. The probing phase runs for T_{probe} gradient epochs fine-tuning $f_{\boldsymbol{\theta}}$ on $\mathcal{D}$. The resulting optimization trace is

$$\mathcal{T}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \{\ell_t,\ w_t\}_{t=0}^{T_{\mathrm{probe}}}, \qquad \ell_t := \frac{1}{nHP} \sum_{i=1}^{n} \|f_{\boldsymbol{\theta},t}(\mathbf{H}_i) - \mathbf{Y}_i\|_F^2, \qquad w_t := \left\|\boldsymbol{\theta}_t^{\mathrm{bb}} - \boldsymbol{\theta}_0^{\mathrm{bb}}\right\|_2,$$

where ℓ_t is the loss at epoch t, and w_t is the endpoint ℓ_2 displacement of the backbone parameters from their initial values. For each parameterized reference model $f_{\boldsymbol{\theta}^{(r)}} \in \mathcal{R}$, we define the corresponding

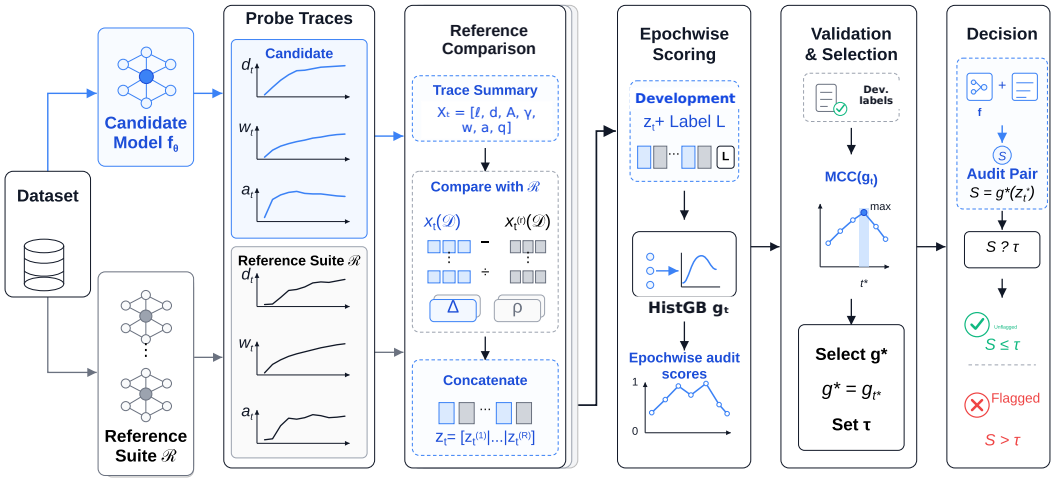

Figure 2: Overview of TSFMAUDIT. A candidate model and a reference suite are probed on $\mathcal{D}$ for T_{probe} gradient epochs under the same fine tuning protocol. The resulting traces are summarized into adaptation features, transformed into reference-based debiased features, and mapped to a scalar contamination-risk score $S(f_{\boldsymbol{\theta}}, \mathcal{D})$.

probe trace $\mathcal{T}^{(r)} = \{\ell_t^{(r)}, w_t^{(r)}\}_{t=0}^{T_{\text{probe}}}$ analogously, with

$$\ell_t^{(r)} := \frac{1}{nHP} \sum_{i=1}^{n} \left\| f_{\boldsymbol{\theta}_t^{(r)}}(\mathbf{H}_i) - \mathbf{Y}_i \right\|_F^2, \qquad w_t^{(r)} := \left\| \boldsymbol{\theta}_t^{(r)} - \boldsymbol{\theta}_0^{(r)} \right\|_2.$$

From the candidate trace, we compute at each epoch t the relative loss drop and the adaptation efficiency for candidate and reference models

$$d_t := \frac{\ell_0 - \ell_t}{\ell_0 + \varepsilon_\ell}, \qquad a_t := \frac{d_t}{w_t + \varepsilon_{\text{ae}}}, \qquad d_t^{(r)} := \frac{\ell_0^{(r)} - \ell_t^{(r)}}{\ell_0^{(r)} + \varepsilon_\ell}, \qquad a_t^{(r)} := \frac{d_t^{(r)}}{w_t^{(r)} + \varepsilon_{\text{ae}}},$$

where $\varepsilon_\ell > 0$ and $\varepsilon_{\text{ae}} > 0$ are numerical stabilizers. A large loss drop d_t together with a small backbone displacement w_t yields a high a_t, which we interpret as behavioral evidence of possible contamination.

At a fixed set of checkpoints, we summarize the core candidate and reference traces by

$$\mathbf{x}(\mathcal{D}) = [w_t,\ d_t,\ a_t]_{1 \le t \le T_{\text{probe}}}, \qquad \mathbf{x}^{(r)}(\mathcal{D}) = \left[w_t^{(r)},\ d_t^{(r)},\ a_t^{(r)}\right]_{1 \le t \le T_{\text{probe}}}.$$

For scoring, we augment these core quantities with loss, loss-curve area, gradient norm, and the gradient-to-loss ratio. The recorded summary is $\widetilde{\mathbf{x}}_t = [\ell_t, d_t, A_t, \gamma_t, w_t, a_t, \gamma_t/(\ell_t + 10^{-6})]$, where A_t summarizes the loss curve and γ_t is the gradient norm. The same construction is used for the reference summaries; Appendix B.3 specifies the main comparison features and ablations.

4.2 Debiasing with Reference Models

Reference Comparison Dataset specific factors, especially intrinsic difficulty, affect these summaries across models. To reduce bias, we compare the candidate with each reference under the same probing protocol. For each epoch t and reference r, we define the elementwise difference and ratio

$$\boldsymbol{\Delta}_t^{(r)}(\mathcal{D}) = \mathbf{x}_t(\mathcal{D}) - \mathbf{x}_t^{(r)}(\mathcal{D}) \in \mathbb{R}^3, \qquad \boldsymbol{\rho}_t^{(r)}(\mathcal{D}) = \mathbf{x}_t(\mathcal{D}) \oslash \left(\mathbf{x}_t^{(r)}(\mathcal{D}) + \varepsilon_{\text{ref}} \mathbf{1}\right) \in \mathbb{R}^3,$$

where $\oslash$ denotes elementwise division and $\varepsilon_{\text{ref}} > 0$ is a numerical stabilizer.

The implementation extends these comparisons to $\widetilde{\mathbf{x}}_t$, including normalized differences and a loss–drop interaction, to form a 13-feature block per reference. All Ref concatenates these blocks into $\mathbf{z}_t$; All Ref+Det additionally appends the seven candidate quantities. Feature construction is described in Appendix B.3, and scorer fitting and selection in Appendix B.1.

Epochwise Scoring At each probe epoch t, a histogram gradient-boosting scorer (HistGB), $g_t : \mathbb{R}^{d_z} \to [0,1]$, maps $\mathbf{z}_t$ to an audit score. For public candidates, g_t is fitted using documentation-derived labels $\widetilde{C}(f_{\boldsymbol{\theta}}, \mathcal{D})$. Its output measures relative adaptation evidence, not a calibrated exposure fraction C.

4.3 Calibration and Decision

Validation & Selection Among the epochwise scorers $\{g_t\}_{t=0}^{T_{\text{probe}}}$, we select the final scorer g^* by the highest development Matthews correlation coefficient (MCC) (Matthews, 1975). In the public-model evaluation, the same development partition is used for scorer fitting, epoch selection, and threshold calibration; the tie-breaking rules are given in Appendix B.1.

To convert this score into a binary decision, we set the threshold using negative development pairs. Let $\mathcal{V}_-$ denote this set; public-model negatives are documentation-derived. At the selected epoch t^*, we choose

$$\tau_0 = \max_{(f_{\boldsymbol{\theta}}, \mathcal{D}) \in \mathcal{V}_-} g_{t^*}(\mathbf{z}_{t^*}(\mathcal{D}))\,.$$

Decision With a strict inequality, this is the smallest threshold that flags none of these negative pairs. The final binary audit decision is

$$\widehat{S}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathbb{I}[S(f_{\boldsymbol{\theta}}, \mathcal{D}) > \tau_0]\,, \qquad S(f_{\boldsymbol{\theta}}, \mathcal{D}) = g_{t^*}(\mathbf{z}_{t^*}(\mathcal{D}))\,.$$

This rule enforces zero exceedances on the development negatives and instantiates the FP-0 protocol of Section B.1; test false-positive rates are evaluated separately.

5 Experiments

Our experiments are designed to answer four questions: (Q1) whether TSFMAudit identifies documented source exposure more effectively than the tested baselines; (Q2) how reference comparisons affect auditing; (Q3) which probe-time features contribute to performance; and (Q4) whether the auditor can be applied to a fresh external benchmark. Controlled experiments provide supplementary tests of membership, transfer, and forecasting effects.

5.1 Experimental Setup

Candidates and datasets. We audit six TSFMs: Chronos (Ansari et al., 2024), TiRex (Auer et al., 2025), Moirai1/2 (Woo et al., 2024), TimesFM2.0 (Das et al., 2024), and Kairos (Feng et al., 2025). The dataset pool combines benchmark datasets from GIFT-Eval and the GIFT-Eval Pretrain corpus (Aksu et al., 2024), covering both evaluation-side and pretraining-side catalogs used by recent TSFMs. We retain datasets with $\geq$30 series, each $\geq$152 samples, and subsample to $\leq$50 series per dataset; this yields 137 audited datasets. Appendix A.1 gives the filtering details. Since public-model membership is unavailable, we derive proxy labels from documented training sources and benchmark metadata; Table 1 summarizes their balance.

Table 1: Class distribution per candidate TSFM.

Candidate	Clean	Leaked	Total	Prevalence
Chronos	118	19	137	13.9%
TiRex	91	46	137	33.6%
TimesFM2.0	125	12	137	8.8%
Kairos	20	117	137	85.4%
Moirai2	20	117	137	85.4%
Moirai1	5	132	137	96.4%

Baselines. We compare TSFMAudit to static-loss and MIA-inspired baselines, all using the FP-k decision rule:

- **Raw Loss**: Candidate pre-probe loss ℓ_0.
- **Loss Drop Rate**: Candidate relative loss reduction after probing, $d_{T_{\text{probe}}} = \frac{\ell_0 - \ell_{T_{\text{probe}}}}{\ell_0 + \varepsilon_\ell}$.
- **LiRA Ratio**: LiRA-inspired (Carlini et al., 2022) zero-probe loss ratio $\ell_0 / \ell_0^{(r)}$, evaluated separately for each reference model r.

Table 2: Main effectiveness comparison, higher is better.

Family	Method	MCC ↑	Macro-F1 ↑	Bal.Acc ↑
Static baselines	Raw Loss (ℓ_0)	0.045 ± 0.090	0.335 ± 0.245	0.536 ± 0.065
	Loss Drop Rate ($d_{T_{\text{probe}}}$)	0.050 ± 0.052	0.394 ± 0.119	0.561 ± 0.062
LiRA-inspired	LiRA Ratio (ScratchCNN)	0.012 ± 0.049	0.332 ± 0.199	0.520 ± 0.040
	LiRA Ratio (ScratchTransformer)	-0.017 ± 0.016	0.299 ± 0.215	0.489 ± 0.012
	LiRA Ratio (Stat)	-0.001 ± 0.020	0.289 ± 0.231	0.498 ± 0.018
	LiRA Ratio (VisionTS)	0.010 ± 0.016	0.323 ± 0.187	0.501 ± 0.020
Frequency baseline	TS MinK FFT ($K = 10\%$)	0.013 ± 0.091	0.336 ± 0.241	0.544 ± 0.095
	TS MinK FFT ($K = 20\%$)	0.017 ± 0.083	0.334 ± 0.240	0.542 ± 0.091
	TS MinK FFT ($K = 30\%$)	0.013 ± 0.080	0.331 ± 0.240	0.534 ± 0.072
Dynamics baseline	Candidate-only dynamics (No Ref)	0.169 ± 0.083	0.474 ± 0.137	0.670 ± 0.029
TSFMAUDIT (ours)	+ ScratchCNN	0.173 ± 0.125	0.495 ± 0.128	0.675 ± 0.077
	+ ScratchTransformer	0.123 ± 0.044	0.463 ± 0.108	0.639 ± 0.042
	+ Stat	0.218 ± 0.225	0.497 ± 0.199	0.681 ± 0.117
	+ VisionTS	0.096 ± 0.076	0.460 ± 0.102	0.597 ± 0.059
	All Ref+Det	$\mathbf{0.317 \pm 0.211}$	$\mathbf{0.576 \pm 0.180}$	$\mathbf{0.761 \pm 0.084}$
	All Ref	0.233 ± 0.182	0.525 ± 0.155	0.723 ± 0.105

- **TS MinK FFT**: Min-K% (Shi et al., 2024) style frequency-domain score $S_{\text{FFT}}(f_{\boldsymbol{\theta}}, \mathcal{D})$ computed on forecast residuals across all six candidate TSFMs; see Appendix B.3.
- **Dyn no Ref**: Candidate-only dynamics baseline using the probe summary without reference comparisons.

The loss-ratio and residual-spectrum scores are forecasting adaptations, rather than full implementations of LiRA and Min-K%.

Reference models. To reduce bias induced by intrinsic dataset difficulty, we instantiate the reference suite $\mathcal{R} = \{f_{\boldsymbol{\theta}^{(r)}}\}_{r=1}^{|\mathcal{R}|}$ with four models: ScratchCNN, ScratchTransformer, Stat, and VisionTS. Their distinct initialization and pretraining histories provide several views of dataset difficulty and adaptation speed under downstream fine tuning. ScratchCNN and ScratchTransformer are lightweight forecasters initialized randomly and trained from scratch, with convolutional and self-attention inductive biases, respectively. Stat aggregates predictions from methods such as ETS (Hyndman & Athanasopoulos, 2021) and serves as a statistical difficulty anchor. VisionTS (Chen et al., 2025) is a TSFM whose representations are derived from a masked autoencoder trained on natural images, providing a pretrained reference with no time-series pretraining exposure. All Ref concatenates their reference relative blocks for HistGB; All Ref+Det also appends the seven candidate quantities. Probe settings and cost are given in Appendix B.2; feature definitions and scorer settings are in Appendix B.

Evaluation metrics. We report MCC (Matthews, 1975), Macro-F1, and balanced accuracy for thresholded decisions, and AUROC for ranking quality. Model-MACRO gives each candidate equal weight; Pair-MICRO pools test decisions. In the main comparison, the reported Model-MACRO SD is across six candidate means. Definitions and repeated-split aggregation are given in Appendix B.

5.2 Main Results: Can TSFMAudit Identify Contamination?

We begin by answering Q1. Table 2 reports the main results against documentation-derived labels.

The overall pattern is clear: Static-loss baselines and LiRA-inspired ratios provide weak discrimination. Raw loss and loss-drop rate yield only small positive MCCs, while the LiRA variants remain near zero or negative. The frequency baseline is similarly weak (MCC ≤ 0.017), with high variability. Probe dynamics provide a more useful signal: even without references, candidate-only dynamics raises MCC from 0.050 to 0.169 and Macro-F1 from 0.394 to 0.474 over the strongest scalar baseline. This supports our central intuition that exposure-related behavior appears in adaptation during fine tuning, not only in prediction before probing. Reference comparison improves the strongest configurations further. ScratchCNN and Stat improve both metrics over candidate-only dynamics, whereas ScratchTransformer and VisionTS do not. All Ref raises MCC from 0.169 to 0.233. All Ref+Det additionally retains candidate dynamics and reaches MCC 0.317, a gain of 0.148 over No Ref. Relative contrasts thus complement the candidate's own adaptation signal.

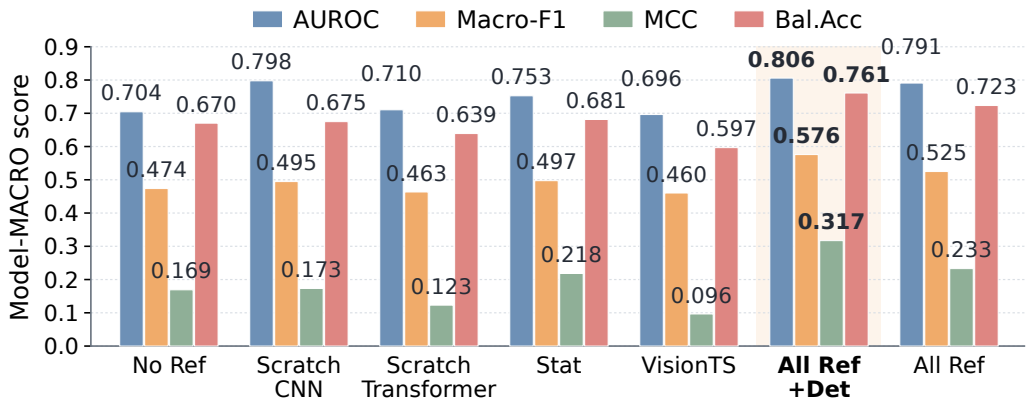

Figure 3: Reference model ablation. Higher is better.

Taken together, these results answer Q1 positively: TSFMAUDIT distinguishes documented-positive from documented-negative datasets more effectively than the tested scalar baselines. Section 5.6 examines known membership and unseen candidates separately.

5.3 REFERENCE-BASED DEBIASING

We next answer Q2. Figure 3 compares single-reference choices and all-reference combinations, while Table 3 ablates how candidate and reference traces are combined. First, reference comparison is useful, but a single reference does not improve every candidate. Stat yields the strongest single-reference Model-MACRO MCC, while ScratchCNN gives higher AUROC. The candidate-level results in Appendix C.2 show Stat leading for Chronos and TiRex, and ScratchCNN for Moirai2; TimesFM2.0 does not benefit from any displayed single reference. Second, relative and direct candidate information work best together in the aggregate comparison: All Ref+Det has the highest mean MCC, Macro-F1, balanced accuracy, and AUROC. Under FP-k calibration, ranking datasets well and making accurate thresholded decisions are related but different objectives.

Table 3 compares the same seven quantities, averaged over single-reference configurations. Differences give the highest MCC (0.201 versus 0.169 for candidate-only features), ahead of ratios (0.086) and interactions (0.122). Full concat yields the highest Macro-F1 (0.505), but lower MCC than differences (0.174). Thus, reference comparisons can help, but adding transformations does not uniformly improve discrimination. This 21-feature concatenation differs from the engineered 13-feature reference blocks in Table 2.

Table 3: Ablation of debiasing representations.

Representation	**MCC** ↑	**Macro-F1** ↑
X_c only (candidate only)	0.169 ± 0.083	0.474 ± 0.137
X_{diff} $(X_c - X_r)$	$\mathbf{0.201 \pm 0.128}$	0.503 ± 0.148
X_{ratio} (X_c / X_r)	0.086 ± 0.050	0.450 ± 0.084
X_{inter} $(X_c \odot X_r)$	0.122 ± 0.109	0.447 ± 0.141
Full concat (all)	0.174 ± 0.118	$\mathbf{0.505 \pm 0.118}$

5.4 FEATURE ABLATION: ARE THE PROPOSED FEATURES USEFUL?

We next answer Q3 by comparing individual feature families with the full All Ref+Det representation (Table 4). Each row refits the same HistGB scorer using the corresponding subset of candidate and reference features. The full dynamics representation is most effective, reaching MCC 0.317 and Macro-F1 0.576. Loss and loss drop are the strongest individual families: Loss gives higher MCC (0.169 versus 0.155), while drop gives slightly higher Macro-F1

Table 4: Feature ablation of TSFMAUDIT. Higher is better.

Feature	**MCC** ↑	**Macro-F1** ↑
Loss (incl. AULC)	0.169 ± 0.133	0.485 ± 0.131
Loss drop rate	0.155 ± 0.080	0.486 ± 0.110
Adaptation efficiency	0.066 ± 0.041	0.419 ± 0.111
Parameter displacement	0.096 ± 0.068	0.454 ± 0.100
Full dynamics	$\mathbf{0.317 \pm 0.211}$	$\mathbf{0.576 \pm 0.180}$

(0.486 versus 0.485). The Loss family includes cumulative loss information (AULC), rather than only raw pre-probe loss. Efficiency and displacement alone yield MCCs of 0.066 and 0.096. Combining the probe measurements raises MCC from 0.169 for the best single family to 0.317; no tested family, including adaptation efficiency, suffices alone.

The separate loss-based restriction in Appendix C.1 gives a different pattern for All Ref: full features improve pooled MCC, while restricted features improve candidate-averaged MCC. Feature utility therefore remains configuration- and aggregation-dependent.

5.5 Practical Deployment on the TIME Benchmark

We finally answer Q4 by applying the auditor without target-benchmark labels for fitting or threshold selection. TIME (Qiao et al., 2026) supplies 50 external datasets. Its later assembly motivates a freshness check, although source exposure depends on the underlying series' dates and provenance. The TIME deployment applies a fixed All Ref scorer and FP-0 threshold without TIME retraining or recalibration (Appendix C.3). Table 5 reports 292 of 300 pairs unflagged (97.33%), with candidate-level rates of 92–100%. These results reflect fixed-auditor deployment, not TIME membership ground truth. The binomial lower bounds are descriptive because datasets are reused across candidates. We test known exposure below.

Table 5: TIME Benchmark deployment check.

Candidate	**Unflagged**	**95% LB**
Chronos	96.0%	87.9%
Kairos	100.0%	94.2%
Moirai1	92.0%	82.6%
Moirai2	100.0%	94.2%
TiRex	96.0%	87.9%
TimesFM2.0	100.0%	94.2%
Pooled	97.33%	95.24%

5.6 Controlled Membership and Transfer Evaluation

We evaluate TSFMAudit against ground-truth dataset membership by controlling which datasets enter pretraining, rather than inferring exposure from model documentation.

Known training exposure. We partition the 137 datasets into 69 used for pretraining and 68 excluded. Chronos-T5-mini and Moirai-1.0-R-Small are pretrained from random initialization using only the 69 included datasets, then audited on both groups. Membership is therefore known from pretraining itself, independently of audit predictions. Using HistGB with the core trace features — loss reduction, backbone displacement, and adaptation efficiency (Appendix D), TSFMAudit obtains MCCs of 0.3563 and 0.6182, respectively, with test FPRs of 16.67% and 18.18%.

Paired model comparison. We separately pretrain two same-architecture Chronos models on complementary groups, so each dataset enters one model's pretraining but not the other's. Both are probed on the same data with identical preprocessing. The exposed model shows larger loss reduction on 108 of 137 datasets, smaller backbone displacement on 99, and higher adaptation efficiency on 108. These directions support the proposed adaptation pattern with the audited data fixed; the models' other pretraining sources also differ.

Transfer and few-window exposure. Under the same audit, leave-one-model-out evaluation across six released candidates yields MCC 0.1968; dataset identities may still occur with other candidates during fitting. A separate single-seed, 40-dataset continuation experiment gives AUROC 0.576 for distinguishing zero from 16 injected windows per dataset. The controlled-membership, candidate-held-out, and continuation protocols are detailed in Appendices D and E.

6 Conclusion

We formalize TSFM pretraining contamination auditing and introduce TSFMAudit, which uses reference-relative probe dynamics. In the public comparison, All Ref+Det obtains the highest mean thresholded metrics (MCC 0.317 vs. 0.169 without references), while reference and feature effects vary by configuration and aggregation. Separate controlled, candidate-held-out, and continuation experiments examine membership, transfer, and forecasting effects.

AI USE STATEMENT

We used generative AI to assist with manuscript editing, translation, LaTeX preparation, literature search, and interpretation of experimental summaries. The authors take responsibility for the final content and conclusions.

ETHICS STATEMENT

This work aims to support fair and transparent evaluation of time series foundation models. Audit flags are statistical indicators of possible training source exposure for candidate–dataset pairs. Public model labels reflect documented source use; attributing benchmark test reuse to a provider requires independent verification of source and evaluation-window provenance.

REPRODUCIBILITY STATEMENT

The appendix describes dataset filtering, public-label construction, probe settings, reference models, repeated splits, and FP-k thresholds. The controlled and continuation configurations and their evaluation units are specified separately in Appendices D and E. Code and reproducibility materials: https://anonymous.4open.science/r/TSFMAudit-FCC5.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23719.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5DSj3MfWrB.

Hyeong Kyu Choi, Maxim Khanov, Hongxin Wei, and Yixuan Li. How contaminated is your benchmark? measuring dataset leakage in large language models with kernel divergence. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=wVDR2qmE28.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Jasper Dekoninck, Mark Niklas Mueller, and Martin Vechev. Constat: Performance-based contamination detection in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ALISPmDPCq.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12039–12050, 2024. doi: 10.18653/v1/2024.findings-acl.716. URL https://aclanthology.org/2024.findings-acl.716/.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Lintao Ma, Xingyu Lu, and Kan Ren. Kairos: Towards adaptive and generalizable time series foundation models, 2025. URL https://arxiv.org/abs/2509.25826.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkl1iRNFwS.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1, 2024. URL https://arxiv.org/abs/2310.03589.

Rakshitha Wathsadini Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wEc1mgAjU-.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FVvf69a5rx.

Jamie Hayes, Ilia Shumailov, Christopher A. Choquette-Choo, Matthew Jagielski, George Kaissis, Milad Nasr, Sahra Ghalebikesabi, Meenatchi Sundaram Mutu Selva Annamalai, Niloofar Mireshghallah, Igor Shilov, Matthieu Meeus, Yves-Alexandre de Montjoye, Katherine Lee, Franziska Boenisch, Adam Dziedzic, and A. Feder Cooper. Exploring the limits of strong membership inference attacks on large language models. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=x0i7wvRLHK.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3 edition, 2021. URL https://otexts.com/fpp3/.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. Investigating data contamination for pre-training language models, 2024. URL https://arxiv.org/abs/2401.06059.

Hao Li, Bowen Deng, Chang Xu, ZhiYuan Feng, Viktor Schlegel, Yu-Hao Huang, Yizheng Sun, Jingyuan Sun, Kailai Yang, Yiyao Yu, and Jiang Bian. MIRA: Medical time series foundation model for real-world health data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=Auy2DmlJBO.

Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, et al. Tsfm-bench: A comprehensive and unified benchmark of foundation models for time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5595–5606, 2025b.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership inference attacks by exploiting loss trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2085–2098, 2022. doi: 10.1145/3548606.3560684. URL https://dblp.org/rec/conf/ccs/Liu00022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. doi: 10.1016/j.ijforecast.2019.04.014.

B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975. doi: 10.1016/0005-2795(75)90109-9. URL https://www.sciencedirect.com/science/article/abs/pii/0005279575901099.

Marcel Meyer, Sascha Kaltenpoth, Kevin Zalipski, and Oliver Müller. Rethinking evaluation in the era of time series foundation models: (un)known information leakage challenges, 2026. URL https://arxiv.org/abs/2510.13654.

OpenAI. GPT-4 technical report. Technical report, OpenAI, 2023. URL https://arxiv.org/abs/2303.08774.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KS8mIvetg2.

Javier Pulido and Filipe Rodrigues. Time series foundation models as strong baselines in transportation forecasting: A large-scale benchmark analysis. *arXiv preprint arXiv:2602.24238*, 2026.

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's time: Towards the next generation of time series forecasting benchmarks, 2026. URL https://arxiv.org/abs/2602.12147.

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. doi: 10.14778/3665844.3665863. URL https://www.vldb.org/pvldb/vol17/p2363-hu.pdf.

Eghbal Rahimikia, Hao Ni, and Weiguan Wang. Re(visiting) time series foundation models in finance. *arXiv preprint arXiv:2511.18578*, 2025.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, 2023. doi: 10.18653/v1/2023.findings-emnlp.722. URL https://aclanthology.org/2023.findings-emnlp.722/.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41. URL https://dblp.org/rec/conf/sp/ShokriSSS17.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020. doi: 10.18653/v1/2020.emnlp-main.746.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

Shifeng Xie, Vasilii Feofanov, Jianfeng Zhang, Themis Palpanas, and Ievgen Redko. Cauker: Classification time series foundation models can be pretrained on synthetic data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xBW2FIfswU.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. Benchmark data contamination of large language models: A survey, 2024. URL https://arxiv.org/abs/2406.04244.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples, 2023. URL https://arxiv.org/abs/2311.04850.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

A BENCHMARK SCOPE AND CONTAMINATION LABELS

This appendix supports the six-candidate, 137-dataset public-model benchmark used in the main comparison. Unless stated otherwise, the public-model tables, figures, and candidate–dataset proxy labels refer to this benchmark and to the HistGB-based TSFMAUDIT configuration defined in Section 4. The TIME deployment, manifest-controlled pretraining, candidate-held-out evaluation, and low-exposure continuation study use the experiment-specific cohorts stated in their respective subsections and are not pooled with the 137-dataset public-model results.

Throughout, we follow the notation of the main paper: $f_{\boldsymbol{\theta}}$ denotes the candidate TSFM, $\mathcal{D}$ the audited dataset, $\mathcal{D}^{\mathrm{pt}}$ the candidate's pretraining corpus, and $\mathcal{R}$ the reference suite. The basic audit unit is the candidate dataset pair $(f_{\boldsymbol{\theta}}, \mathcal{D})$.

A.1 DATASET POOL AND FILTERING

The public-model dataset pool combines (i) benchmark datasets from GIFT-Eval and (ii) datasets from GIFT-Eval Pretrain. After the uniform filtering and loading protocol described below, the public-model benchmark contains 137 datasets. These filtering rules define the source benchmark used in the main comparison; experiment-specific context/horizon settings for controlled pretraining and continuation are stated separately.

Filtering rules. A dataset is retained only if it satisfies all of the following conditions:

- it contains at least 30 valid series;
- each retained series contains at least $L + H = 128 + 24 = 152$ time steps;
- nearly constant series with standard deviation below 10^{-6} are removed;
- at most 50 series are used per dataset in order to bound probe cost.

Datasets are converted to Arrow format for unified loading. This conversion is only an implementation convenience and does not alter the underlying series values.

Family grouping. For data splitting, we retain family metadata that records which datasets belong to the same benchmark family. Under `group_split=family`, datasets from the same family are assigned to the same split. Under `group_split=none`, this grouping constraint is removed. We use the family split by default because it keeps closely related datasets on the same side of the split and reduces cross-split transfer through renamed or lightly transformed variants.

A.2 FROM IDEALIZED CONTAMINATION TO OBSERVABLE SUPERVISION

Section 3 defines the idealized contamination ratio

$$C\big(\mathcal{D}, \mathcal{D}^{\mathrm{pt}}\big) \in [0, 1],$$

which depends on the membership of the audited samples' source series in the pretraining corpus. In realistic TSFM settings, however, the pretraining corpus $\mathcal{D}^{\mathrm{pt}}$ is usually only partially documented. As a result, the ground truth quantity $C(\mathcal{D}, \mathcal{D}^{\mathrm{pt}})$ is generally unobservable. In addition, the available documentary evidence depends on the candidate model under consideration. The same audited dataset may therefore receive different observable labels for different candidates. We therefore define the observable proxy at the level of the audit pair $(f_{\boldsymbol{\theta}}, \mathcal{D})$, rather than at the level of the dataset alone.

For evaluation, we use the binary proxy label

$$\widetilde{C}(f_{\boldsymbol{\theta}}, \mathcal{D}) \in \{0, 1\},$$

defined for each audit pair $(f_{\boldsymbol{\theta}}, \mathcal{D})$. We set

$$\widetilde{C}(f_{\boldsymbol{\theta}}, \mathcal{D}) = 1$$

only when official sources associated with the candidate, including the paper, model card, dataset card, or official repository documentation, explicitly state that pretraining used the audited dataset. Otherwise, we set

$$\widetilde{C}(f_{\boldsymbol{\theta}}, \mathcal{D}) = 0.$$

A zero label should be interpreted as no documented evidence in official sources, rather than as evidence that overlap is absent.

This proxy label records documented exposure to the audited corpus. It should not be interpreted as direct evidence of benchmark test leakage. In particular, official documentation may establish shared provenance across renamed, rescaled, or repackaged variants even when exact sample overlap cannot be established from public documentation alone. Accordingly, the proxy records documented source use; it does not identify the specific evaluation windows encountered during pretraining. Appendix A.3 returns to this issue through a representative normalized-duplicate analysis.

To make this candidate dependence concrete, we briefly summarize how official sources describe the training data of each audited candidate:

- **Chronos.** Official sources describe training on large collections of public datasets together with synthetic data (Ansari et al., 2024).
- **TiRex.** The official model card lists `autogluon/chronos_datasets` and `Salesforce/GiftEvalPretrain` among the training sources (Auer et al., 2025).
- **Moirai1.** Official sources state that Moirai 1 is pretrained on LOTSA (Woo et al., 2024).
- **Moirai2.** The official model card states that Moirai 2 uses a subset of the GIFT-Eval Pretrain and Train datasets as non-leaking historical context, together with additional synthetic and internal operational data (Liu et al., 2025).
- **TimesFM2.0.** Official documentation states that TimesFM 2.0 uses the TimesFM 1.0 pretraining set together with a listed subset of LOTSA datasets (Das et al., 2024).
- **Kairos.** The official project page, repository, and paper describe Kairos as pretrained on the Predictability-Stratified Time Series corpus, which is stated to contain real-world observations from Chronos and Moirai together with synthetic data (Feng et al., 2025).

Candidate–dataset proxy labels. Full candidate dataset proxy labels are listed in Table 6. To reduce space, datasets with identical labels and contiguous naming patterns are merged into grouped entries. In grouped rows, the reported labels apply to all datasets in the indicated range. Here, "Y" denotes documented exposure in official sources, and "N" denotes no documented evidence in official sources.

Table 6: Full candidate–dataset proxy labels for the evaluated benchmark. Column abbreviations: Ch = Chronos, Ka = Kairos, M1 = Moirai1, M2 = Moirai2, TF = TimesFM2.0, Ti = TiRex-1.1-gifteval.

Dataset	Ch	Ka	M1	M2	TF	Ti	Dataset	Ch	Ka	M1	M2	TF	Ti
alibaba_cluster_trace_2018	N	Y	Y	Y	N	N	azure_vm_traces_2017	N	Y	Y	Y	N	Y
bdg-2_bear	N	Y	Y	Y	N	Y	bdg-2_fox	N	Y	Y	Y	N	Y
bdg-2_panther	N	Y	Y	Y	N	Y	bdg-2_rat	N	Y	Y	Y	N	Y
BEIJING_SUBWAY_30MIN	N	Y	Y	Y	N	Y	bitbrains_fast_storage/H	N	N	N	N	N	N
bitbrains_rnd/H	N	N	N	N	N	N	borg_cluster_data_2011	N	Y	Y	Y	N	Y
buildings_900k	N	Y	Y	Y	N	Y	bull	N	Y	Y	Y	N	N
cdc_fluview_ilinet	N	Y	Y	Y	N	Y	cdc_fluview_who_nrevss	N	Y	Y	Y	N	Y
china_air_quality	N	Y	Y	Y	N	Y	cmip6_1850–2010	N	Y	Y	Y	N	N
covid_deaths	Y	N	Y	N	N	N	covid_mobility	N	Y	Y	Y	N	N
electricity/H	Y	N	N	N	Y	N	electricity/W	Y	N	N	N	Y	N
era5_1989–2018	N	Y	Y	Y	N	N	extended_web_traffic_with_missing	N	Y	Y	Y	Y	N
favorita_sales	N	Y	Y	Y	Y	Y	favorita_transactions	N	Y	Y	Y	N	N
fred_md	Y	Y	Y	Y	N	Y	hierarchical_sales/D	N	N	Y	N	N	N
HZMETRO	N	Y	Y	Y	N	Y	ideal	N	Y	Y	Y	N	Y
kdd_cup_2018_with_missing/D	N	N	Y	N	N	N	kdd_cup_2018_with_missing/H	N	N	Y	N	N	N
kdd2022	N	Y	Y	Y	N	Y	largest_2017–2021	N	Y	Y	Y	N	Y
lcl	N	Y	Y	Y	N	Y	london_smart_meters_with_missing	N	Y	Y	Y	N	Y
LOOP_SEATTLE/H	N	N	Y	N	N	N	LOS_LOOP	N	Y	Y	Y	N	Y
M_DENSE/H	N	N	Y	N	N	N	m4_daily	Y	N	Y	N	Y	N
m4_hourly	Y	N	Y	N	Y	N	m4_monthly	Y	N	Y	N	Y	N
m4_quarterly	Y	N	Y	N	Y	N	m4_weekly	Y	N	Y	N	N	N
m4_yearly	Y	N	Y	N	Y	N	m5	Y	Y	Y	Y	N	Y
nn5_daily_with_missing	Y	Y	Y	Y	N	Y	pedestrian_counts	Y	Y	Y	Y	N	Y
PEMS_BAY	N	Y	Y	Y	N	Y	PEMS03	N	Y	Y	Y	N	Y
PEMS04	N	Y	Y	Y	N	Y	PEMS07	N	Y	Y	Y	N	Y
PEMS08	N	Y	Y	Y	N	Y	project_tycho	N	Y	Y	Y	N	N
Q-TRAFFIC	N	Y	Y	Y	Y	Y	residential_load_power	N	Y	Y	Y	N	Y
residential_pv_power	N	Y	Y	Y	N	Y	restaurant	N	N	Y	N	N	N
rideshare_with_missing	N	Y	Y	Y	N	Y	SHMETRO	N	Y	Y	Y	N	Y

Dataset	Ch	Ka	M1	M2	TF	Ti	Dataset	Ch	Ka	M1	M2	TF	Ti
solar/D	Y	N	N	N	N	N	subseasonal	N	Y	Y	Y	N	Y
subseasonal_precip	N	Y	Y	Y	N	Y	SZ_TAXI/H	N	N	Y	N	N	N
taxi_30min	Y	Y	Y	Y	N	Y	temperature_rain_with_missing	N	N	Y	N	N	N
tourism_monthly	Y	Y	Y	Y	N	Y	traffic_hourly	N	Y	Y	Y	Y	Y
uber_tlc_daily	Y	Y	Y	Y	N	Y	uber_tlc_hourly	Y	Y	Y	Y	N	Y
vehicle_trips_with_missing	N	Y	Y	Y	N	N	weather	Y	Y	Y	Y	Y	Y
wiki-rolling_nips	N	Y	Y	Y	N	N	wind_farms_with_missing	N	Y	Y	Y	N	Y

A.3 Contamination Semantics Beyond Exact Overlap

Section 3 defines exposure through source-series membership. It also notes that, in time series, contamination can be more subtle than literal reuse of identical values: the same underlying signal may reappear after rescaling, renaming, reindexing, or related preprocessing changes. We therefore examine a stricter semantic question: should contamination-relevant repetition be limited to exact overlap, or should transformed variants of the same underlying signal also be treated as leakage-relevant?

We examine this boundary by normalizing dataset values while retaining the public-model candidates, the reference suite $\mathcal{R} = \{\text{ScratchCNN}, \text{ScratchTransformer}, \text{Stat}, \text{VisionTS}\}$, the HistGB scorer, the repeated family-aware splits, and FP-0 calibration used in the main comparison. This tests whether agreement with the documentation-derived labels persists under a change in numerical representation.

Agreement persists after normalization (Table 7). Under the candidate-only baseline, the normalized setting still yields Pair-MICRO MCC 0.274 and Macro-F1 0.473. Reference-based variants remain clearly above this baseline. For example, ScratchCNN reaches Pair-MICRO MCC 0.389 and Macro-F1 0.628, while All Ref reaches Pair-MICRO MCC 0.415 and Macro-F1 0.642. The same pattern also appears at the Model-MACRO level: the no-reference baseline attains MCC 0.101 and Macro-F1 0.434, whereas All Ref reaches MCC 0.128 and Macro-F1 0.508.

These results show that agreement with the documentation-derived labels persists after this value normalization. The transformation changes the numerical representation while preserving the recorded source identity. This supports considering provenance alongside raw-value matching; it does not measure exact evaluation-window overlap.

Table 7: Representative results in the normalized-duplicate setting under the same FP-0 protocol and family-aware splits. Agreement with documentation-derived labels remains after value normalization.

Reference	Model-MACRO Macro-F1 ↑	Model-MACRO MCC ↑	Pair-MICRO Macro-F1 ↑	Pair-MICRO MCC ↑
None	0.434±0.175	0.101±0.124	0.473	0.274
ScratchCNN	0.492±0.095	0.101±0.108	0.628	0.389
Stat	0.481±0.130	0.123±0.092	0.559	0.337
All Ref	0.508±0.110	0.128±0.121	0.642	0.415

B Experimental Protocol and Implementation Details

B.1 Splits, Scoring, and Calibration

This subsection specifies the development/test splits, epoch selection, threshold calibration, and metric aggregation used for the public-model evaluation.

Development and test splits. The label-bearing unit is the candidate–dataset pair $(f_{\boldsymbol{\theta}}, \mathcal{D})$. The split generator uses a pool of 30 repeats with `group_split=family`; repeat r uses seed $10007r$. The development sampler prioritizes documented-negative pairs, with negative sampling fraction 0.8 and a target positive:negative ratio of 1:1. Generation is retried until both proxy-label classes are present. Methods draw from the common split pool for each candidate and setting. All TSFMAudit variants use HistGB as the scorer; baseline methods retain their native score definitions, and thresholds are calibrated separately for each method.

Scorer fitting and epoch selection. The public-model comparison uses HistGB with 64 boosting iterations, learning rate 0.05, maximum depth 2, maximum 4 leaves, minimum 3 samples per leaf, L2 regularization 10, balanced class weights, and no early stopping. The random seed is $1009t$ at epoch t. At each eligible epoch $0, \ldots, T_{\text{probe}}$, the scorer is fitted on the development partition. Selection maximizes development MCC, then TPR, then AUROC, then the negative threshold. This partition also supplies the FP-k threshold; the outer test partition is not used for fitting, selection, or calibration. Test labels and metrics are not used to select the reporting subset. HistGB is the scorer for every fitted TSFMAUDIT auditor reported in this appendix. Scalar baselines keep their native scores, and trace-only diagnostics such as the complementary-checkpoint comparison do not fit an auditor.

FP-k threshold calibration. Let $S(f_{\boldsymbol{\theta}}, \mathcal{D})$ denote the selected continuous score for an audit pair under the chosen epoch. The threshold is determined only from negative calibration pairs,

$$\mathcal{V}_- = \left\{ (f_{\boldsymbol{\theta}}, \mathcal{D}) \in \mathcal{V} : \widetilde{C}(f_{\boldsymbol{\theta}}, \mathcal{D}) = 0 \right\}.$$

Under FP-0, the threshold is set to the largest negative calibration score. Under FP-1, it is set to the second-largest negative calibration score. More generally, with at least $k+1$ negative calibration pairs, FP-k uses their $(k+1)$-st largest score as the threshold. The final binary decision is

$$\widehat{S}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathbb{I}[S(f_{\boldsymbol{\theta}}, \mathcal{D}) > \tau_k].$$

The main paper uses FP-0 unless otherwise stated.

Aggregation and uncertainty. Model-MACRO first averages the repeated-split metric within each candidate and then averages the six candidate means equally. In the main comparison, the reported SD is the sample SD of those six candidate-level means. Figure 6 instead reports within-candidate split SD. Pair-MICRO pools the recorded test occurrences, including repeated candidate–dataset pairs, before computing the metric. These SDs describe variability, not confidence intervals from independent pretraining repetitions.

B.2 Probe Settings and Reference Models

Table 8 summarizes the public-model probe settings. Controlled and continuation settings are described separately in Appendices D and E.

Reference suite. The six audited candidates are Chronos, TiRex, Moirai1, Moirai2, TimesFM2.0, and Kairos. The reference suite used throughout the main experiments is

$$\mathcal{R} = \{\text{ScratchCNN}, \text{ScratchTransformer}, \text{Stat}, \text{VisionTS}\}.$$

Appendix comparisons include the candidate-only baseline, the single-reference variants, and the all-reference variants defined from this reference suite. All Ref concatenates the engineered reference blocks defined for the public comparison; All Ref+Det also appends the seven-coordinate candidate summary. The representation and feature-family ablations are described in Appendix B.3.

Shared defaults. Unless otherwise stated, we use $T_{\text{probe}} = 10$ probe epochs, batch size 4, and the context–horizon pair $(L, H) = (128, 24)$ following GIFT-Eval. The default learning rate is 10^{-3}, while TimesFM2.0 uses 10^{-4} for stability.

Window-level preprocessing. Each context–target window $(\mathbf{H}_i, \mathbf{Y}_i)$ is normalized using the mean and standard deviation of the context segment $\mathbf{H}_i$. Concretely, the context is standardized by its own statistics, and the target is normalized using the same context statistics. This keeps the probe dynamics comparable across datasets with substantially different scales.

Optimization details. The probing procedure uses AdamW (Loshchilov & Hutter, 2019), mean squared forecasting loss, weight decay 0.01, and gradient clipping with maximum norm 1.0. The initial loss ℓ_0 is recorded before any parameter updates, and subsequent losses are recorded after each probe epoch. For a candidate TSFM, w_t is the ℓ_2 displacement of the backbone parameters from their state at probe start, matching Section 4. Before auditing a new dataset, the model is reset to the same initial checkpoint so that all candidate–dataset traces are measured from a common origin.

Reference trace definitions. For each reference model $f_{\boldsymbol{\theta}^{(r)}} \in \mathcal{R}$, we define the corresponding probe trace $\mathcal{T}^{(r)} = \{\ell_t^{(r)}, w_t^{(r)}\}_{t=0}^{T_{\text{probe}}}$ analogously, with

$$\ell_t^{(r)} := \frac{1}{nHP} \sum_{i=1}^{n} \left\| f_{\boldsymbol{\theta}_t^{(r)}}(\mathbf{H}_i) - \mathbf{Y}_i \right\|_F^2, \qquad w_t^{(r)} := \left\| \boldsymbol{\theta}_t^{(r)} - \boldsymbol{\theta}_0^{(r)} \right\|_2.$$

Trace summaries. At each probe epoch t, the implementation records the loss ℓ_t and parameter displacement w_t, and then computes the relative loss drop

$$d_t = \frac{\ell_0 - \ell_t}{\ell_0 + \varepsilon_\ell},$$

and the adaptation efficiency

$$a_t = \frac{d_t}{w_t + \varepsilon_{\text{ae}}}.$$

These quantities define the candidate and reference trace summaries used in the main method,

$$\mathbf{x}(\mathcal{D}) = [w_t, d_t, a_t]_{1 \le t \le T_{\text{probe}}}, \qquad \mathbf{x}^{(r)}(\mathcal{D}) = [w_t^{(r)}, d_t^{(r)}, a_t^{(r)}]_{1 \le t \le T_{\text{probe}}}.$$

The reference-based features are then constructed from the per-epoch differences and ratios defined in Section 4.2. For the public-model evaluations, these core trace variables are augmented with the recorded implementation features listed below, including loss-area and gradient-derived coordinates.

Probe cost. The reported six-model probe cost is approximately 5.5 A800 GPU-hours, including four references but excluding controlled pretraining. Probe-length effects depend on the configuration (Appendix C.2).

Table 8: Core probe defaults and HistGB settings for TSFMAUDIT. Experiment-specific feature representations and split protocols are documented separately.

Item	Value
Audit unit	candidate–dataset pair $(f_{\boldsymbol{\theta}}, \mathcal{D})$
Split-generation pool	30
Repeat seed schedule	$10007r$ for repeat r
Family-aware split	`group_split=family`
Calibration negative fraction	0.8
Calibration positive:negative ratio	1.0
Main threshold rule	FP-0
TSFMAUDIT scorer	HistGB
HistGB settings	64 iterations; depth 2; 4 leaves; learning rate 0.05
Probe length	$T_{\text{probe}} = 10$
Batch size	4
Default context / horizon	$(128, 24)$
Window normalization	context-wise z-score
Probe optimizer	AdamW
Weight decay	0.01
Gradient clipping	max norm 1.0
Per-dataset model reset	yes
Efficiency stabilizer ε_{ae}	10^{-6}
Reference ratio stabilizer ε_{ref}	10^{-6}

B.3 Feature Construction, Baselines, and Metrics

Main-comparison reference features. The main comparison in Table 2 extends the candidate–reference comparisons in Section 4.2 to the recorded probe quantities. At epoch t, ℓ_t, d_t, γ_t, w_t, and a_t denote loss, relative loss drop, recorded gradient norm, parameter displacement, and adaptation efficiency. The loss-area statistic averages the trapezoidal area over the probe prefix:

$$A_0 = \ell_0, \qquad A_t = \frac{1}{t} \sum_{u=1}^{t} \frac{\ell_{u-1} + \ell_u}{2}, \quad t \ge 1.$$

For reference r, write $\Delta v_t = v_t - v_t^{(r)}$ and set $\epsilon = 10^{-6}$. The nominal reference block is

$$\mathbf{z}_t^{(r)} = \begin{bmatrix} \Delta\ell_t,\ \Delta d_t,\ \Delta A_t,\ \Delta\gamma_t,\ \Delta w_t, \\ \dfrac{\ell_t}{\ell_t^{(r)}+\epsilon},\ \dfrac{\gamma_t}{\gamma_t^{(r)}+\epsilon},\ \Delta a_t,\ \Delta\ell_t\,\Delta d_t,\ \dfrac{d_t}{d_t^{(r)}+\epsilon}, \\ \dfrac{w_t}{w_t^{(r)}+\epsilon},\ \dfrac{\Delta\ell_t}{\ell_t+\ell_t^{(r)}+\epsilon},\ \dfrac{\Delta\gamma_t}{\gamma_t+\gamma_t^{(r)}+\epsilon} \end{bmatrix}.$$

This block contains six differences, four stabilized ratios, one loss–drop interaction, and two normalized differences. It is constructed at each eligible epoch; the auditor uses the block at its selected epoch rather than concatenating all epochs.

Before classification, a fixed availability-based finite-column mask removes unavailable coordinates. ScratchCNN, ScratchTransformer, and VisionTS retain all 13. For Stat, the recorded relative loss drop is undefined in the zero-initial-loss case, making the associated efficiency unavailable. The mask therefore removes Δd_t, Δa_t, $\Delta\ell_t\,\Delta d_t$, and $d_t/(d_t^{(r)}+\epsilon)$, leaving nine coordinates. The mask follows availability in the recorded traces; undefined values are not replaced by zero. All Ref thus retains $3 \times 13 + 9 = 48$ coordinates. All Ref+Det appends the seven candidate quantities $[\ell_t, d_t, A_t, \gamma_t, w_t, a_t, \gamma_t/(\ell_t+\epsilon)]$, giving 55 coordinates. These are the main-comparison inputs, not the separate 21-feature representation ablation below.

Representation ablation. Table 3 uses the same seven recorded quantities at each epoch, denoted X_c for the candidate and X_r for a reference. For each single reference, it evaluates $X_c - X_r$, $X_c/(X_r + 10^{-6})$, and $X_c \odot X_r$ elementwise. The relative rows average the single-reference configurations in $\mathcal{R}$; the candidate-only row does not depend on the reference. Full concat joins the three seven-coordinate transformations for one reference (21 features), rather than pooling different reference models. This ablation differs from the engineered 13-feature reference blocks in the main comparison.

Feature-family ablation. Table 4 restricts All Ref+Det to the named feature families and refits HistGB for each subset. The Loss family includes loss and AULC information; the other restricted rows use loss drop, adaptation efficiency, or displacement. Full dynamics retains all reference blocks and the seven candidate quantities, including the gradient-derived inputs and interactions. These single-family restrictions differ from the multi-family loss-based auditor described below.

Loss-based restricted representation. Table 9 retains candidate loss, drop, and AULC for No Ref. For each reference it retains differences of loss/drop/AULC, ratios of loss/drop, the product of loss and drop differences, and normalized loss difference, where defined in the feature mask. All Ref concatenates these retained reference blocks. Gradient, displacement, efficiency, and every derived input requiring them are excluded. The learner and calibration rule are unchanged.

Frequency-domain baseline. We additionally report a frequency-domain heuristic baseline, denoted `ts_mink_fft_*`. Given residuals $r_t = y_t - \hat{y}_t$, we compute the frequency energy spectrum $|\mathrm{FFT}(r)|^2$ and define a series-level score as the mean energy of the top-$K\%$ frequency components, with $K \in \{10, 20, 30\}$. The dataset-level score is obtained by averaging these series-level scores over the audited dataset. This baseline is included as a time-series analogue of Min-K% style (Shi et al., 2024) tests and is evaluated under the same six-candidate protocol as the main comparison.

Metric definitions. We report Matthews Correlation Coefficient (MCC) (Matthews, 1975), Macro-F1, AUROC, and balanced accuracy (Bal.Acc) to capture both ranking quality and thresholded decision performance under class imbalance. Denoting true/false positives and negatives as TP, TN, FP, FN:

$$\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}, \mathrm{Bal.Acc} = \tfrac{1}{2}\left(\tfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \tfrac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right).$$

Macro-F1 is the unweighted mean of per-class F1 scores; AUROC is the area under the receiver operating characteristic curve, which plots the true positive rate against the false positive rate over all decision thresholds. Higher is better for all metrics.

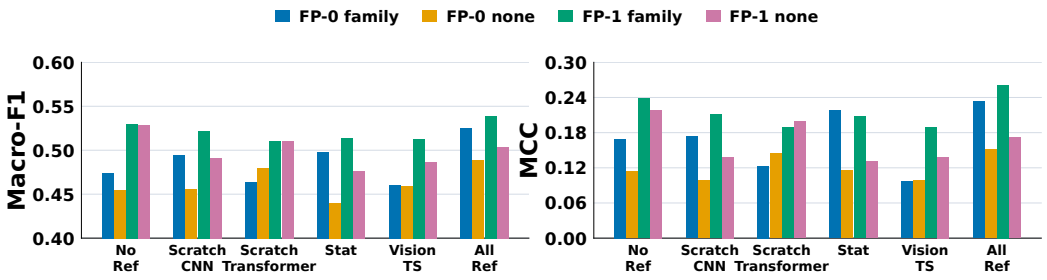

Figure 4: Sensitivity to the false-positive budget and split grouping on Model-MACRO aggregation. Bars report mean Macro-F1 and MCC over repeated splits. FP-0 sets the threshold to the largest negative calibration score; FP-1 uses the second-largest. All four conditions are recomputed using the fixed HistGB configuration.

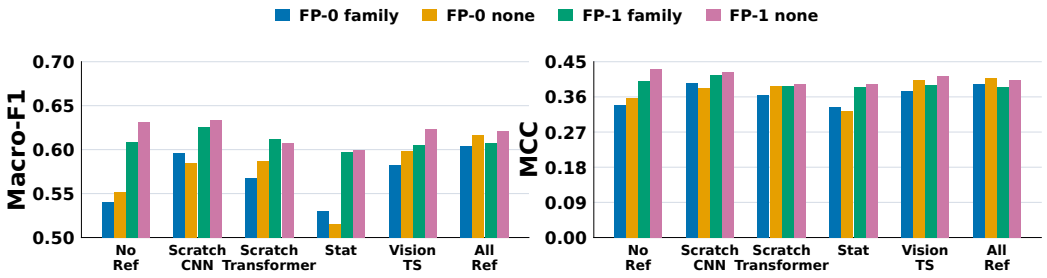

Figure 5: Sensitivity to the false-positive budget and split grouping on Pair-MICRO aggregation. Bars report Macro-F1 and MCC after pooling test occurrences. FP-0 sets the threshold to the largest negative calibration score; FP-1 uses the second-largest. All four conditions are recomputed using the fixed HistGB configuration.

C Additional Main Results

C.1 Calibration and Feature Sensitivity

Figures 4 and 5 examine sensitivity to two protocol choices: the false-positive budget, comparing FP-0 and FP-1, and the split grouping strategy, comparing `group_split=family` and `group_split=none`. The two figures report the same comparison under Model-MACRO and Pair-MICRO aggregation, respectively.

Calibration sensitivity. Each of the four FP-budget and grouping conditions is recomputed with the fixed HistGB configuration. For both grouping choices, the same 30-repeat seed schedule described in Appendix B.1 is used; `group_split=family` additionally constrains benchmark-family members to the same side of a split, whereas `group_split=none` removes that constraint. FP-1 changes both calibration threshold selection and epoch selection. Under family grouping, changing FP-0 to FP-1 raises All Ref Model-MACRO MCC from 0.233 to 0.261, whereas Pair-MICRO MCC changes from 0.392 to 0.385. The effect of the FP budget therefore depends on aggregation.

Loss-based restriction. We additionally evaluate a restricted variant of TSFMAudit that uses only loss-based trace quantities. This variant retains features derived from the loss trajectory and the relative loss drop, and removes parameter-displacement quantities and adaptation-efficiency

Table 9: Loss-based restricted variant under FP-0 auditing with family-aware splits. Each cell reports Macro-F1 / MCC. Both variants use fixed HistGB; the restricted variant excludes all gradient-, displacement-, and efficiency-dependent inputs.

Reference	MACRO (full)	MACRO (loss-based)	MICRO (full)	MICRO (loss-based)
No ref	0.474 / 0.169	0.386 / 0.064	0.540 / 0.338	0.430 / 0.261
ScratchCNN	0.495 / 0.173	0.516 / 0.250	0.596 / 0.395	0.546 / 0.355
ScratchTransformer	0.463 / 0.123	0.493 / 0.192	0.567 / 0.364	0.556 / 0.356
Stat	0.497 / 0.218	0.445 / 0.171	0.529 / 0.334	0.444 / 0.265
VisionTS	0.460 / 0.096	0.490 / 0.177	0.582 / 0.375	0.566 / 0.365
All Ref	0.525 / 0.233	0.538 / 0.273	0.604 / 0.392	0.569 / 0.365

quantities. Table 9 compares this restricted variant against the default full trace representation under the same FP-0 protocol and family-aware splits used in the main appendix tables.

Both variants use the fixed HistGB learner. For All Ref, loss-based restriction raises Model-MACRO MCC from 0.233 to 0.273 but lowers Pair-MICRO MCC from 0.392 to 0.365. Full features have higher pooled MCC in all six rows, whereas the restricted vector has higher Model-MACRO MCC in four. The contribution of non-loss features is therefore reference- and aggregation-dependent.

C.2 Candidate and Probe-Length Effects

Figure 6 reports candidate-specific MCC under FP-0 calibration with family-aware splits. The aggregate results reported elsewhere in the appendix average over substantial heterogeneity across candidates, and this figure makes that heterogeneity explicit.

Reference effects across candidates. Stat has the highest displayed mean MCC for Chronos (0.555) and TiRex (0.446), compared with 0.206 and 0.169 without a reference. ScratchCNN is strongest for Moirai2 (0.161). TimesFM2.0 has a higher candidate-only mean (0.312) than any displayed single-reference variant. The wide split SDs for Chronos and TiRex show that these mean differences coexist with considerable partition sensitivity.

Interpretation under label imbalance. This candidate-level heterogeneity should be interpreted together with proxy-label balance. Under strict FP-0 calibration, threshold selection depends only on negative calibration pairs, so candidates with very few documented-negative audit pairs can show higher across-split variance. This is particularly relevant for Moirai1 under the current proxy labels. For this reason, the candidate-level patterns in Figure 6 should be read together with the FP-0 and FP-1 sensitivity results in Figures 4 and 5.

Probe length. Figure 7 reports Model-MACRO MCC and Pair-MICRO MCC as functions of probe length $T_{\text{probe}} \in \{2, 4, 6, 8, 10\}$ under FP-0 calibration with family-aware splits. The goal is to determine whether useful audit signal already appears under short probing, and whether longer probes materially change the relative ranking of representative reference configurations.

For each budget, the epoch and FP-0 threshold are selected using only the training/calibration partition. Each point summarizes the corresponding budget-specific evaluation. All Ref has Model-MACRO MCC 0.255, 0.265, 0.258, 0.275, and 0.233 at budgets 2, 4, 6, 8, and 10, respectively; its Pair-MICRO MCC varies over a much narrower range. Longer probing is therefore not uniformly better.

C.3 TIME Deployment and Joint-Pool Evaluation

This subsection describes the deployment-oriented evaluation on the TIME Benchmark (Qiao et al., 2026) reported in Section 5.5. The TIME deployment reuses the source-benchmark All Ref HistGB auditor defined in the main method. Its feature construction, selected epoch, fitted scorer, and FP-0 threshold are fixed before application to TIME; no TIME data are used for fitting, selection, or calibration.

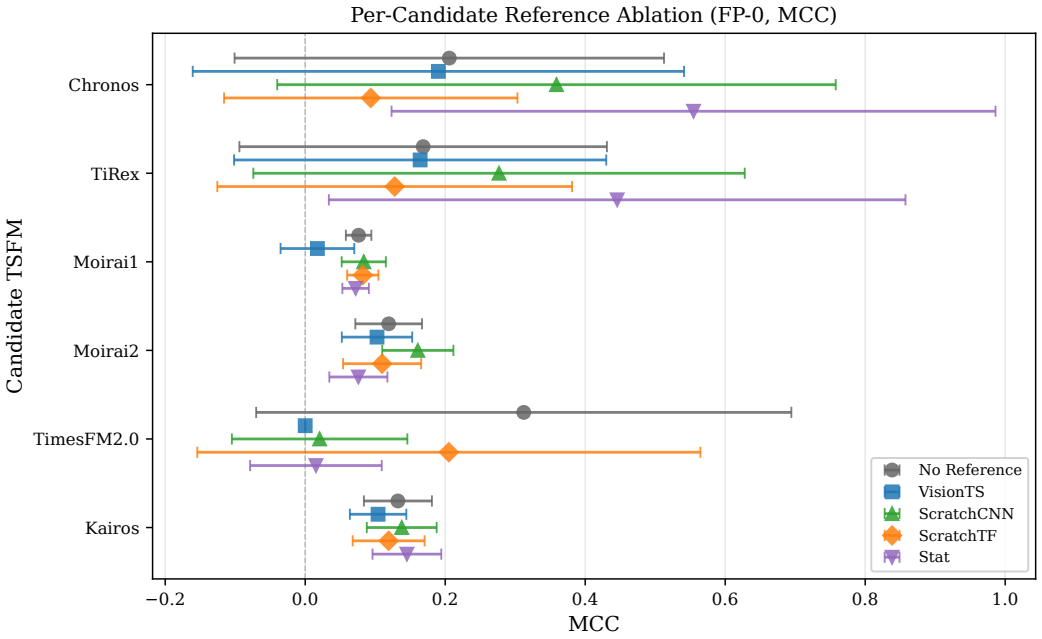

Figure 6: Per-candidate reference ablation under FP-0 calibration with family-aware splits. Points are candidate-specific means; error bars are within-candidate split SD, not confidence intervals.

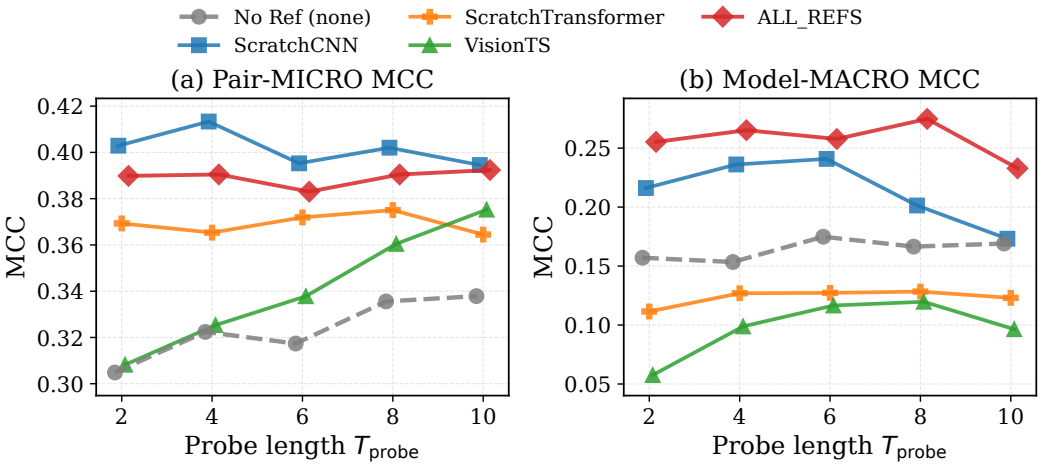

Figure 7: Probe length sensitivity under FP-0 calibration with family-aware splits. The combined figure reports Model-MACRO MCC and Pair-MICRO MCC as functions of T_{probe}. Each budget is evaluated separately, allowing epoch selection only within that budget.

Scorer and threshold reuse. The TIME experiment adopts the same All Ref feature configuration used in the public-model comparison: the engineered reference-relative blocks from Appendix B.3, without the seven direct candidate quantities added by All Ref+Det. At each eligible epoch, HistGB is fitted on the source-benchmark development partition, and the selected epoch and FP-0 threshold τ_0 are determined using Appendix B.1. The selected scorer g^*, its availability mask, feature ordering, and threshold are then applied unchanged to TIME.

Probing on TIME. For each candidate TSFM $f_{\boldsymbol{\theta}}$ and each TIME dataset $\mathcal{D}$, we run the same probing protocol used in the main experiment. Concretely, we use $T_{\text{probe}} = 10$ probe epochs, batch size 4, context–horizon pair $(L, H) = (128, 24)$, AdamW with the candidate-specific learning rate listed in Appendix B.2, context-wise z-score normalization, and per-dataset reset to the same initial checkpoint. The reference suite is the main suite $\mathcal{R} = \{\text{ScratchCNN}, \text{ScratchTransformer}, \text{Stat}, \text{VisionTS}\}$, and the same recorded quantities and All Ref feature mask are constructed before scoring. The fixed source-benchmark pair (g^*, τ_0) then produces one binary audit decision $\widehat{S}(f_{\boldsymbol{\theta}}, \mathcal{D})$ for each candidate–dataset pair.

Evaluation set and confidence bounds. Each of the six candidates is audited on 50 TIME datasets, yielding 300 audit pairs $(f_{\boldsymbol{\theta}}, \mathcal{D})$ in total. The per-candidate rates and one-sided 95% Clopper–Pearson lower bounds in Table 5 are computed under the standard binomial model on the 50 decisions per candidate; the pooled lower bound is computed analogously over the 300 pooled decisions. As noted in Section 5.5, the independence assumption underlying these bounds holds only approximately because the 50 TIME datasets share a common evaluation protocol, so the bounds should be read as descriptive rather than as a formal coverage guarantee.

Joint-pool evaluation. A separate evaluation combines the 137 source-benchmark datasets with 50 TIME datasets before development/test splitting. Across all 180 model-split results, the main All Ref+Det representation defined in Appendix B.3 obtains MCC 0.2408, Macro-F1 0.5790, balanced accuracy 0.6637 and FPR 10.08% against the pooled evaluation labels. These are whole-pool metrics. This protocol fits within the joint pool, whereas Section 5.5 applies the source-benchmark-fitted HistGB scorer and threshold without using TIME for fitting or calibration. Neither protocol supplies direct ground truth about all TIME source memberships.

D Controlled Membership and Candidate Transfer

These experiments supplement the released-model evaluation with known pretraining membership and candidate-held-out transfer. Whenever an auditor is fitted, TSFMAudit uses HistGB. The manifest-controlled and leave-one-model-out evaluations use the core loss-drop, backbone-displacement, and adaptation-efficiency representation from the main method together with the main reference suite; the complementary-checkpoint subsection is a trace-level diagnostic and does not fit a scorer. The controlled models are pretrained independently from random initialization and therefore use checkpoints distinct from the released candidates.

D.1 Manifest-Controlled Membership

We partition the 137 datasets into 69 used for pretraining and 68 excluded, balancing domain, sampling frequency, and benchmark family as far as possible. Chronos-T5-mini and Moirai-1.0-R-Small are pretrained from random initialization using only the included group, then audited on both groups. We call these groups manifest-IN and manifest-OUT, respectively. Their labels follow inclusion in the controlled pretraining corpus, independently of model documentation or audit predictions. Table 6, by contrast, lists documentation-derived proxy labels for the released models; it is not the controlled pretraining manifest.

The controlled-membership evaluation uses the core trace features defined in Section 4—loss drop, backbone displacement, and adaptation efficiency—together with their reference-relative features computed against $\mathcal{R} = \{\text{ScratchCNN}, \text{ScratchTransformer}, \text{Stat}, \text{VisionTS}\}$. HistGB scoring, epoch selection, and the FP-0 decision rule follow Appendix B.1.

Table 10: TSFMAudit under manifest-defined membership using the core trace features and HistGB scorer. FPR is evaluated on manifest-OUT test pairs.

Pretrained model	MCC	Macro-F1	Bal.Acc	FPR (%)
Chronos-T5-mini	0.3563	0.6645	0.6667	16.67
Moirai-1.0-R-Small	0.6182	0.8091	0.8091	18.18

Table 10 reports the model-specific results. Both controlled models yield positive membership discrimination. These tests complement the released-model evaluation by replacing documentation-derived supervision with membership defined directly by the controlled pretraining manifests.

Reproducibility materials. We will provide the controlled pretraining and audit materials in the anonymous repository.[2] The manifest lists all 137 datasets, their identifiers and versions, the subsets used, and their pretraining inclusion/exclusion assignments. Training configurations and run records specify the architecture, preprocessing and sampling, objective, optimizer and learning-rate schedule, batch size, training budget, and random seeds. The checkpoint index links each audited checkpoint to its training run and step and records the selection rule. Audit configurations, execution commands, and per-dataset result exports connect these checkpoints to the results reported in Section 5.6.

D.2 Complementary Checkpoints

Two Chronos checkpoints use the same architecture and complementary pretraining manifests. Each audited dataset is IN for one checkpoint and OUT for the other. For a trace statistic $u \in \{d, w, a\}$, define $\Delta u_i = u(f_i^{\mathrm{IN}}, \mathcal{D}_i) - u(f_i^{\mathrm{OUT}}, \mathcal{D}_i)$. The directions $\Delta d_i > 0$, $\Delta w_i < 0$, and $\Delta a_i > 0$ occur on 108, 99, and 108 of 137 datasets, respectively. The input dataset and preprocessing are identical within each pair, so its trend and seasonal structure are shared. The comparison shows how adaptation differs with membership across these complementary corpora; the checkpoints also differ in their other pretraining sources.

D.3 Leave-One-Model-Out Evaluation

Each released candidate is held out from auditor fitting, epoch selection, and threshold calibration. The same core feature representation, main reference suite, HistGB scorer, and FP-0 calibration rule used in the controlled analysis above are retained; the remaining candidates determine the fitted scorer and selection choices before held-out evaluation. Test metrics are used only for reporting. The result is MCC 0.1968, Macro-F1 0.6095, balanced accuracy 0.6893, and FPR 13.06%. This evaluates transfer to a candidate not used to construct the auditor. Dataset identities can occur with other candidates during fitting, so candidate transfer and source transfer remain distinct evaluations.

E Low-Exposure Continuation

Intervention and forecasting. Chronos-T5-mini is evaluated on 40 datasets in two complementary directions, with 20 target datasets and a separate base checkpoint per direction. Each branch injects $k \in \{0, 1, 2, 4, 8, 16\}$ distinct windows per target dataset, presenting each once. Within a direction, branches share the initial checkpoint, 1,000-step continuation budget, batch size 512 and background sampling stream. Injected windows replace background examples. The experiment uses one training seed. Exposure, adaptation and evaluation windows are disjoint; the audit uses 40 adaptation and 40 evaluation windows per dataset, with lookback/horizon $512/64$.

The intervention jointly changes exposure for 20 targets per direction. The primary forecasting summary averages within-dataset relative changes in MAE from $k = 0$. Table 11 lists all doses. The 90% dataset-cluster bootstrap intervals condition on the trained checkpoints. For random $k = 1$, the interval lies inside the $\pm 1\%$ equivalence criterion used in this analysis. This finding concerns the average MAE effect of this intervention. Difficulty-selected windows and other metrics are evaluated separately.

[2] https://anonymous.4open.science/r/TSFMAudit-FCC5

Table 11: Random-window continuation on 40 datasets. MAE changes are means of within-dataset relative changes from $k = 0$; intervals are 90% dataset-cluster bootstrap intervals. Audit scores use dataset-held-out predictions. One training seed is used, so the intervals condition on the trained checkpoints.

Distinct windows k	MAE change (%)	90% interval (%)	Mean audit score
0	0.00	—	0.481
1	-0.47	[-0.88, -0.07]	0.510
2	-1.02	[-1.64, -0.44]	0.496
4	-1.08	[-1.73, -0.48]	0.520
8	-1.43	[-2.37, -0.58]	0.519
16	-3.10	[-5.06, -1.38]	0.548

Auditor evaluation. Consistent with the main method, the continuation audit uses a class-balanced HistGB scorer on the core probe dynamics and their reference-relative features. An outer leave-one-dataset-out evaluation with inner grouped selection uses the $k = 0$ and $k = 16$ endpoints for fitting and selection; intermediate doses are not used for fitting or model selection. The endpoint AUROC is 0.575625 (reported as 0.576 in the main text). Mean score increases from 0.481 at $k = 0$ to 0.548 at $k = 16$. Spearman correlation between dose and score change from $k = 0$ is 0.222 (95% dataset-cluster interval $[0.084, 0.358]$); for intermediate doses it is 0.093 ($[-0.016, 0.220]$). Because this continuation experiment uses a different intervention and context–horizon setting from the public-model benchmark, these ranking and dose-response statistics are reported separately rather than pooled with the main MCC results.

Secondary outcomes. Tables 12 and 13 retain the other reported forecast outcomes. The outcome named relative-MAE is distinct from the within-dataset relative change of primary MAE: for random $k = 1$, its mean change is 89.79%, median change 0.054%, and 90% interval $[2.52\%, 255.61\%]$. Thus, the primary MAE equivalence result does not extend to every reported metric. The high-NLL window selection uses a 20-dataset subset; at $k = 1$, mean primary MAE changes by 0.440% with interval $[-0.051\%, 0.923\%]$.

Table 12: Random-window continuation: full primary and secondary outcomes across 40 datasets. All effects and intervals are percentages relative to the zero-exposure condition. Intervals use dataset clusters, conditional on one training seed.

k	Metric	Mean change	Median change	90% interval
1	MAE	-0.466	-0.141	[-0.885, -0.065]
2	MAE	-1.017	-0.357	[-1.643, -0.439]
4	MAE	-1.076	-0.481	[-1.726, -0.483]
8	MAE	-1.435	-1.169	[-2.367, -0.583]
16	MAE	-3.096	-1.748	[-5.063, -1.382]
1	RMSE	-0.168	-0.117	[-0.484, +0.137]
2	RMSE	-0.580	-0.156	[-1.051, -0.162]
4	RMSE	-0.733	-0.557	[-1.224, -0.274]
8	RMSE	-0.590	-0.684	[-1.260, +0.015]
16	RMSE	-1.267	-0.568	[-2.450, -0.219]
1	relative_mae	+89.793	+0.054	[+2.520, +255.609]
2	relative_mae	+0.296	-0.441	[-4.181, +4.644]
4	relative_mae	+1.792	-0.331	[-4.429, +8.502]
8	relative_mae	+8.685	-0.272	[-4.244, +24.591]
16	relative_mae	-4.376	-2.776	[-15.584, +7.705]
1	WQL	-0.316	-0.123	[-0.771, +0.136]
2	WQL	-0.751	-0.395	[-1.584, +0.019]
4	WQL	-0.810	-0.041	[-1.697, -0.055]
8	WQL	-1.676	-0.019	[-3.130, -0.409]
16	WQL	-2.371	+0.412	[-4.992, -0.057]

Table 13: Difficulty-selected continuation: full outcomes across the 20 reported datasets. All effects and intervals are percentages relative to the zero-exposure condition. Intervals use dataset clusters, conditional on one training seed.

k	Metric	Mean change	Median change	90% interval
1	MAE	+0.440	+0.470	[-0.051, +0.923]
4	MAE	+0.598	-0.077	[-0.633, +2.450]
1	RMSE	+0.302	+0.362	[-0.150, +0.757]
4	RMSE	+0.183	+0.041	[-0.457, +0.938]
1	relative_mae	+1.749	+0.006	[-4.177, +7.826]
4	relative_mae	+39.750	+0.519	[+4.910, +86.294]
1	WQL	+0.340	+0.764	[-0.170, +0.819]
4	WQL	+1.323	+0.560	[+0.197, +2.857]