FAITHFULNESS IS CONDITIONAL: AUDITING PERTURBATION-BASED EXPLANATION EVALUATION IN TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning plays an important role in time-series classification (TSC), but its black-box nature motivates explanations that help users understand and inspect predictions. *Faithfulness* is a central criterion for evaluating these explanations, yet the suitability of the evaluator itself also requires examination. In particular, dataset-level results do not establish whether an evaluator remains informative within each class. We propose a *simple class-conditional audit that compares informative reference attributions with random rankings*. Across controlled synthetic datasets and an ECG benchmark, we find that *an evaluator can distinguish these attributions overall while failing the same comparison within particular classes*. Motivated by these findings and studies of animal perceptual decision-making, we examine perturbation baseline predictions and representations, identifying baseline class preference as a possible contributor to these failures. We introduce a *NULL response* that changes how the model evaluates baselines by training an additional output while keeping the original representation and real-class logits frozen. We also investigate *input-dependent counterfactual (CF)* as baselines that change the replacement input itself, using a transductive, label-informed construction. Both strategies improve meaningful–random separation in several settings, but neither passes the audit across all tested configurations. We therefore treat NULL and CF as two options whose use is guided by the class-conditional audit. *This audit acts as an initial gate for selecting evaluation configurations* for each trained model, dataset, and intended class scope before their scores are used to compare explanation methods. Although the audit does not certify explanation correctness or replace more comprehensive evaluation, this audit provides evidence of simple but mandatory meaningful–random explanation separation.

1 INTRODUCTION

Time-series classification (TSC) is an important task in domains such as healthcare and manufacturing, where predictions from sequential measurements can support clinical assessment and fault detection. Deep learning has become increasingly important for these tasks because of its ability to learn complex patterns from time-series data. For example, a large-scale ECG study showed that a deep neural network (DNN) could achieve performance comparable to cardiologists in classifying 12 cardiac rhythm classes (Hannun et al., 2019). In manufacturing, deep-learning-based TSC has also supported fault detection in settings with limited and imbalanced data (Lee et al., 2020). These results illustrate the potential of deep learning, but strong predictive performance does not by itself explain how a model reaches a particular decision. DNNs often remain difficult to understand because of their complexity and the mismatch between high-dimensional mathematical operations and human reasoning (Burrell, 2016). Addressing this opacity may require a combination of measures, including transparency, auditing, and regulation (Burrell, 2016). Regulatory developments such as the EU AI Act further emphasize transparency and human oversight for AI systems (Smuha, 2025). For practitioners, the need to understand a prediction becomes particularly apparent when it conflicts with their own assessment. Interviews with ICU and emergency-department clinicians show that they want to understand the factors underlying model predictions, especially when these disagree with clinical judgment or established protocols (Tonekaboni et al., 2019). This practical

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.

need motivates explainable AI (XAI), which aims to help users understand model predictions and examine the evidence behind them. Such explanations can support practitioners in understanding model behavior and deciding how to act on its predictions.

TSC explanation methods can be broadly divided into post-hoc approaches, which explain a trained model without modifying it (Ribeiro et al., 2016; Sundararajan et al., 2017; Crabbe & van der Schaar, 2021; Sivill & Flach, 2022; Huang et al., 2025), and ante-hoc approaches, which incorporate an explanatory mechanism into the model by design (Vinayavekhin et al., 2018; Hsu et al., 2019). Many of these methods produce attribution or importance maps that assign relevance scores to individual timesteps (Queen et al., 2023; Liu et al., 2024a) or temporal segments (Sivill & Flach, 2022; Tran et al., 2025). These maps indicate which parts of an input are considered important for a prediction, but an understandable explanation is not necessarily a faithful one. Faithfulness concerns whether an explanation reflects the evidence used by the model, whereas plausibility concerns whether it appears reasonable to humans (Jacovi & Goldberg, 2020). This distinction matters for both post-hoc and ante-hoc approaches, since incorporating an explanatory mechanism into a model does not remove the need to evaluate it. Several faithfulness evaluations have therefore been proposed for TSC explanations (Turbé et al., 2022; Nguyen et al., 2023; Zheng et al., 2025). Such evaluations help assess whether explanations can support model diagnosis, reveal reasoning that conflicts with human expectations, or support practical decisions. An unreliable faithfulness evaluation may favour misleading explanations that conceal problematic model behaviour and lead users to trust a model for the wrong reasons (Lakkaraju & Bastani, 2019).

Faithfulness is commonly estimated by perturbing input features according to their attributed importance and measuring the resulting change in model behavior. For example, Least Relevant Feature (LeRF) removes less important features first and evaluates how well the prediction is preserved, whereas Most Relevant Feature (MoRF) removes the most important features first and measures how rapidly the prediction degrades. Depending on the measured response, fidelity can be expressed through changes in prediction confidence or predictive accuracy (Zheng et al., 2025). Using these scores to compare explanations also requires examining the reliability of the evaluator itself. Meta-evaluation frameworks such as MetaQuantus (Hedström et al., 2023) address this need by assessing how explanation-quality estimators behave under controlled perturbations. However, an evaluator's ability to distinguish informative attributions from random attributions at the dataset level does not mean that it is consistent for every class. We therefore ask: *can a perturbation-based evaluator distinguish informative from random attributions within each class, even when it separates them at the dataset level?* To address this question, we make the following contributions.

Contributions. First, we introduce a simple **class-conditional audit** for perturbation-based faithfulness and evaluate it across controlled synthetic and real-world time-series datasets. The audit asks whether an evaluator that distinguishes informative from random attributions at the dataset level can still do so within each class. We find that aggregate separation can conceal class-conditional failures, showing why dataset-level results alone are insufficient to assess an evaluator's suitability. Motivated by perceptual-decision studies suggesting that the absence of sensory evidence need not produce a neutral decision (Findling et al., 2025), we investigate the **perturbation baseline** as a possible contributor to these failures.

Second, we investigate two complementary strategies for addressing these failures: **NULL responses and input-dependent counterfactual (CF) references**. For NULL, we add an output to the classifier while keeping its learned representation and real-class logits frozen. Inspired by monkey opt-out experiments (Komura et al., 2013), we hypothesize that frozen representations may contain information that distinguishes baseline-like inputs from real task inputs, even though the original classifier has no response for learning this distinction. Training the additional NULL output tests whether this information can support baseline recognition. When evaluating, the augmented classifier is used only during faithfulness evaluation; attribution generation and deployment retain the original classifier. CF references address a different limitation: the replacement input itself may remove useful shared structure or introduce unwanted patterns. Thus, NULL changes how the model responds to a baseline, while CF changes the input used as the replacement.

Finally, we combine these strategies with an **audit gate that guides their use for each case**. Our experiments show that neither strategy passes across all tested settings: a configuration that improves reference–random separation in one setting may fail in another. We therefore use the class-conditional audit to assess candidate configurations, including NULL and CF, for each trained

model, dataset, and set of application-relevant classes. Passing this gate provides an initial diagnostic of evaluator suitability within the stated scope; it does not certify explanation correctness or replace more comprehensive evaluation. This approach is consistent with the broader view in the literature that understanding opaque models requires a combination of analyses and auditing mechanisms (Burrell, 2016).

2 Aggregate Faithfulness Scores Can Conceal Class-Conditional Failures

2.1 Perturbation-Based Faithfulness

Time-series classification and attribution map. Let $\mathbf{x} \in \mathbb{R}^{T\times C}$ denote a time series with T timesteps and C channels, and let f be a trained K-class classifier. For a target class $c \in \{0, \ldots, K-1\}$, $f_c(\mathbf{x})$ denotes the predicted class probability of c on $\mathbf{x}$ via f. An attribution method a produces an attribution map $attr_{f,\mathbf{x}} = a(f, \mathbf{x}) \in \mathbb{R}^{T\times C}$, assigning importance scores to all elements of $\mathbf{x}$. The ranking $\boldsymbol{\pi}_{f,\mathbf{x}} = \mathrm{rank}(attr_{f,\mathbf{x}})$ orders these elements from most to least important. We skip f and $\mathbf{x}$ whenever it is clear from the context.

For an input time series $\mathbf{x}_i$ and retention ratio $p \in [0, 1]$, a binary mask $\mathbf{M}^d_{i,p,\mathrm{attr}}$ retains the top-p fraction of ranked elements for LeRF and the bottom-p fraction for MoRF based on attribution map attr. Retained elements have mask value one; the remaining elements are replaced by the corresponding baseline $\tilde{\mathbf{x}}_i \in \mathbb{R}^{T\times C}$:

$$\hat{\mathbf{x}}^d_{i,p,\mathrm{attr}} = \mathbf{M}^d_{i,p,\mathrm{attr}} \odot \mathbf{x}_i + \left(\mathbf{1} - \mathbf{M}^d_{i,p,\mathrm{attr}}\right) \odot \tilde{\mathbf{x}}_i, \tag{1}$$

where $d \in \{\mathrm{LeRF}, \mathrm{MoRF}\}$ and $\odot$ denotes element-wise multiplication. As p decreases, LeRF removes the least important elements first and MoRF removes the most important elements first.

Aggregate faithfulness. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i, c_i)\}_{i=1}^N$ denote the evaluation dataset, where y_i is the ground-truth label and c_i is the explained target class. The target c_i may differ from y_i, for example when explaining the original prediction $c_i = \arg\max_k f_k(\mathbf{x}_i)$, but remains fixed throughout perturbation. For direction d and retention ratio p, let $\hat{\mathcal{D}}^d_{p,\mathrm{attr}} = \{(\hat{\mathbf{x}}^d_{i,p,\mathrm{attr}}, y_i, c_i)\}_{i=1}^N$ denote the perturbed dataset. A metric $\mathcal{M}$, such as mean target-class confidence, accuracy, or ROC-AUC, gives the response $R^d_{\mathcal{M},\mathrm{attr}}(p; \mathcal{D}) = \mathcal{M}(f, \hat{\mathcal{D}}^d_{p,\mathrm{attr}})$. We average this response over m retention ratios $\mathcal{P} = \{p_1, \ldots, p_m\}$ to obtain a target class confidence:

$$S^{\mathcal{M}}_{\mathrm{attr},d}(\mathcal{D}) = \frac{1}{m} \sum_{p\in\mathcal{P}} R^d_{\mathcal{M},\mathrm{attr}}(p; \mathcal{D}). \tag{2}$$

For target-class confidence, higher scores are preferred for LeRF because important evidence should be preserved for longer, whereas lower scores are preferred for MoRF because important evidence is removed first.

Comparison against random attribution. A faithfulness score alone does not indicate whether the evaluator can distinguish an informative attribution ranking from a trivial one. We therefore construct a random ranking using the same perturbation protocol and denote its aggregate score by $S^{\mathcal{M}}_{\mathrm{rand},d}(\mathcal{D})$. We define the direction-corrected separation between the attribution and random rankings as

$$\Delta^{\mathcal{M}}_d(\mathcal{D}) = \begin{cases} S^{\mathcal{M}}_{\mathrm{attr},d}(\mathcal{D}) - S^{\mathcal{M}}_{\mathrm{rand},d}(\mathcal{D}), & d = \mathrm{LeRF}, \\ S^{\mathcal{M}}_{\mathrm{rand},d}(\mathcal{D}) - S^{\mathcal{M}}_{\mathrm{attr},d}(\mathcal{D}), & d = \mathrm{MoRF}. \end{cases} \tag{3}$$

Using a tolerance threshold ϵ, we categorize an evaluation as **Healthy** if $\Delta^{\mathcal{M}}_d(\mathcal{D}) \geq \epsilon$, **Blind** if $-\epsilon < \Delta^{\mathcal{M}}_d(\mathcal{D}) < \epsilon$, and **Inverted** if $\Delta^{\mathcal{M}}_d(\mathcal{D}) \leq -\epsilon$. Throughout this work, we use $\epsilon = 0.005$ as a descriptive effect-size threshold rather than a statistical-significance criterion. A healthy evaluation favors the attribution ranking over random, a blind evaluation cannot meaningfully distinguish them, and an inverted evaluation favors the random ranking.

Class-conditional faithfulness. An aggregate faithfulness score can conceal differences in evaluator behavior across classes. We therefore ask: *does an evaluator remain informative within every class,*

Table 1: Baseline predictions and faithfulness on **FPH**. **(a)** Class 0 confidence on background inputs across three architectures and five seeds. **(b)** Aggregate and class-conditional scores for ground-truth (GT) and random attributions (seed 0). Bold pairs indicate comparisons favoring GT over Random with gap larger than threshold ϵ: higher scores for LeRF and lower scores for MoRF.

(a) Noise-baseline prediction

Seed	Confidence to Class 0		
	Linear	CNN	RNN
0	1.0	1.0	1.0
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0
4	1.0	1.0	1.0

(b) Faithfulness scores (seed 0)

Model	Dir.	Overall		Class 0		Class 1	
		GT	Rand.	GT	Rand.	GT	Rand.
Linear	LeRF	**0.9545**	**0.6459**	0.9999	1.0000	**0.9091**	**0.2918**
	MoRF	**0.5454**	**0.6449**	1.0000	1.0000	**0.0909**	**0.2899**
CNN	LeRF	**0.9505**	**0.6489**	0.9949	0.9991	**0.9061**	**0.2986**
	MoRF	**0.5451**	**0.6479**	0.9995	0.9991	**0.0906**	**0.2967**
RNN	LeRF	**0.9545**	**0.9007**	1.0000	1.0000	**0.9091**	**0.8014**
	MoRF	**0.5455**	**0.9039**	1.0000	1.0000	**0.0909**	**0.8079**

even when its aggregate comparison appears healthy? Since ROC-AUC is not directly defined on a subset containing only one ground-truth class, we use mean target-class confidence as the primary class-conditional metric. We denote this metric by conf, which averages the probability assigned to each sample's fixed target class c_i. For class c, let $\mathcal{D}_c = \{(\mathbf{x}_i, y_i, c_i) \in \mathcal{D} : y_i = c\}$, where membership is determined by the true label y_i, while c_i remains the explained target throughout perturbation. We apply the score and gap definitions above with $\mathcal{M} = \mathrm{conf}$ and $\mathcal{D}$ replaced by $\mathcal{D}_c$, yielding the class-conditional score $S^{\mathrm{conf}}_{\mathrm{attr},d}(\mathcal{D}_c)$ and direction-corrected attribution–random gap $\Delta^{\mathrm{conf}}_d(\mathcal{D}_c)$. The same tolerance ϵ determines whether the evaluation is **Healthy**, **Blind**, or **Inverted**.

2.2 From Aggregate Separation to Class-Conditional Failure

We find that a faithfulness evaluator can separate ground-truth (GT) from random attributions overall while showing little separation within a particular class. We illustrate this on a synthetic dataset Fixed-Position Peak Height (FPH), where the height of a fixed-position peak determines the class (Section 5.1). Using the Gaussian noise background as the baseline, both MoRF and LeRF favor GT over Random overall and for class 1 across Linear, CNN, and RNN models trained with seed 0 (Table 1(b)). For class 0, however, the scores are nearly identical. For example, CNN LeRF scores are 0.9505 for GT and 0.6489 for Random overall, compared with 0.9949 and 0.9991 within class 0. All six class 0 evaluations satisfy $|\Delta^{\mathrm{conf}}_d(\mathcal{D}_0)| < \epsilon$ and are therefore classified as **Blind**. This pattern persists across five training seeds for all three studied architectures (Linear, CNN, RNN) and both directions (LeRF, MoRF); the full heatmap appears in Appendix C. This discrepancy motivates examining how the model responds to the baseline.

Why Can This Happen? Baseline Prediction Collapse. To investigate the results in Table 1, we examine the original inputs and background baselines for seed 0 in Figure 1. The background samples appear separate from both real classes in the t-SNE projections, yet all three models assign the highlighted background to class 0 with confidence reported as 1.0. The linear model provides a more direct view: background samples have higher class 0 logits and lower class 1 logits than the original class 0 samples (Figure 1(c, d)). For this model, replacing a class 0 input with its peakless background can therefore strengthen its preference for that class. Although the background removes the annotated peak, the model does not treat it as a neutral thing. This may help explain why GT and random rankings receive similar scores for class 0: removing informative features may leave strong support for the same class. Baseline predictions alone do not establish what happens throughout the perturbation path, but they identify a possible contributor to the observed class-conditional failure. This phenomenon is consistent in visualization in several datasets shown in Appendix E and F.

3 Rethinking How We Treat Perturbation Baselines

A NULL Response on Frozen Representations. In Figure 1, background samples appear distinct from real inputs in the projected feature spaces, yet the classifier assigns them to a real class. This raises a simple question: can an additional response distinguish baseline-like inputs while keep-

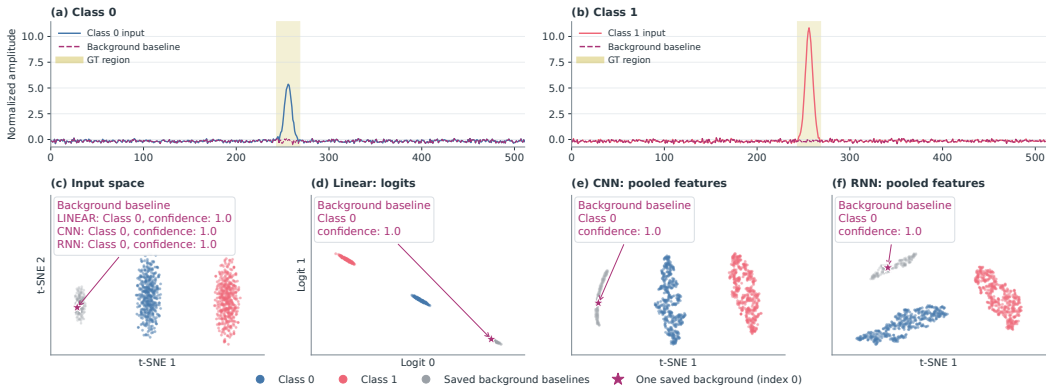

Figure 1: Background-baseline analysis on **FPH** (seed 0). **(a–b)** Class 0 and class 1 inputs, peakless backgrounds, and ground-truth (GT) regions. **(c)** t-SNE of inputs. **(d)** Linear-model logits. **(e–f)** t-SNE of pooled CNN and RNN features, respectively. Backgrounds appear distinct from both classes in visualization (c–f), yet all three models assign the highlighted background to class 0 with confidence reported as 1.0.

ing the learned representation unchanged? A related idea appears in monkey opt-out experiments, where an additional response offered a small guaranteed reward and was chosen more frequently as the stimulus became ambiguous (Komura et al., 2013). The study also linked opt-out behavior to neural activity associated with perceptual confidence. Taking this additional decision option as conceptual inspiration, we hypothesize that frozen representations may support a *NULL response* for perturbation baselines when given explicit supervision.

Let f_θ denote the original K-class classifier, with representation $h_\theta(\mathbf{x})$ and logits $z_{\theta,k}(\mathbf{x})$ for $k \in \{0, \dots, K-1\}$. Given labelled training data $\mathcal{D}_{\text{train}}$, we train the classifier using cross-entropy:

$$\mathcal{L}_{\text{cls}}(\theta) = -\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \log \frac{\exp(z_{\theta,y}(\mathbf{x}))}{\sum_{k=0}^{K-1} \exp(z_{\theta,k}(\mathbf{x}))}. \tag{4}$$

This training setup distinguishes the original classes but provides neither a separate output nor supervision for recognizing perturbation baselines.

After training, we freeze θ and add a fully connected output with a single NULL logit, $z_{\text{null},\phi}(\mathbf{x}) = \mathbf{w}_{\text{null}}^\top h_\theta(\mathbf{x}) + b_{\text{null}}$, where $\phi = (\mathbf{w}_{\text{null}}, b_{\text{null}})$ contains the only trainable parameters. The augmented classifier $f_{\theta,\phi}^{\text{NULL}}$ applies softmax over the unchanged K real-class logits and the additional NULL logit. Given training baselines $\mathcal{B}_{\text{train}}$ (Appendix B.2), we assign them label K and form $\mathcal{D}_{\text{train}}^+ = \mathcal{D}_{\text{train}} \cup \{(\tilde{\mathbf{x}}, K) : \tilde{\mathbf{x}} \in \mathcal{B}_{\text{train}}\}$. Real inputs retain their original labels, and we optimize only ϕ:

$$\mathcal{L}_{\text{NULL}}(\phi;\theta) = -\frac{1}{|\mathcal{D}_{\text{train}}^+|} \sum_{(\mathbf{u},v)\in\mathcal{D}_{\text{train}}^+} \log f_{\theta,\phi,v}^{\text{NULL}}(\mathbf{u}), \qquad \theta \text{ fixed.} \tag{5}$$

This trains the additional output to recognize baselines using the frozen representation. More details are in Appendix B.2. For an original class $c \in \{0, \dots, K-1\}$, the resulting probability is

$$f_{\theta,\phi,c}^{\text{NULL}}(\mathbf{x}) = \frac{\exp(z_{\theta,c}(\mathbf{x}))}{\sum_{k=0}^{K-1} \exp(z_{\theta,k}(\mathbf{x})) + \exp(z_{\text{null},\phi}(\mathbf{x}))}. \tag{6}$$

Using the NULL Response. During faithfulness evaluation, we measure $f_{\theta,\phi,c_i}^{\text{NULL}}(\hat{\mathbf{x}}_{i,p,\text{attr}}^d)$ while keeping the original target class c_i fixed. Probability mass can now shift toward NULL as the input is perturbed. The real-class logits and their ordering remain unchanged, but their probabilities and the prediction over all $K+1$ classes can change. The augmented classifier is used only for faithfulness evaluation. Deployment and attribution generation continue to use the original classifier f_θ, preserving the deployed predictions and attribution maps being evaluated. Additional checks of baseline recognition and classification performance on real inputs are reported in the Appendix. Recognizing baselines alone does not guarantee an informative evaluation, so we still audit whether the perturbation responses distinguish informative from random attributions within each class.

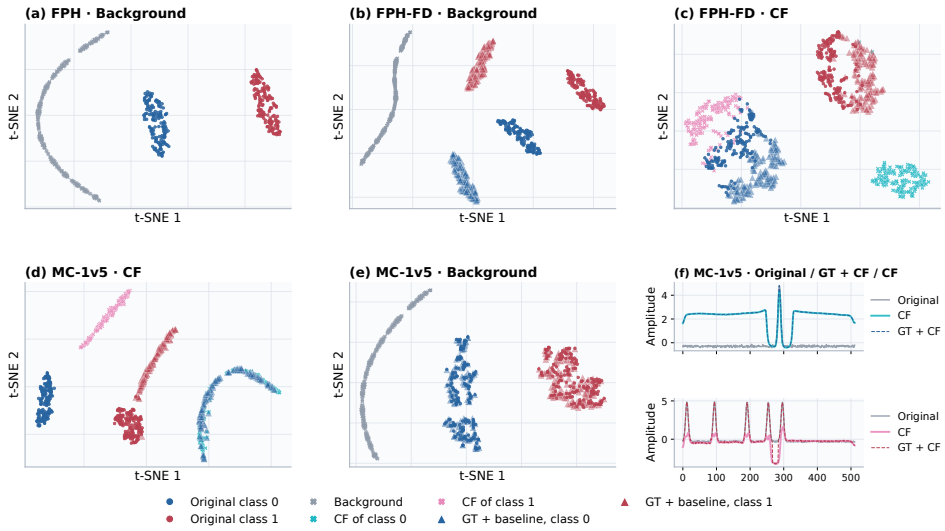

Figure 2: Background and counterfactual (CF) references across synthetic tasks, shown through t-SNE projections of CNN penultimate features in **(a–e)**. **(a)** On FPH, background samples appear separate from the original classes, motivating a NULL response. **(b–c)** On FPH-FD, GT-preserving perturbations (c) overlap more with the original class groups when using CF rather than background references (b). **(d–e)** On MC-1v5, background references instead yield greater visual overlap. **(f)** Example signals illustrate how CF can introduce structure outside the retained GT regions.

4 Counterfactual References for Perturbation-Based Faithfulness

Constructing Counterfactual References. NULL changes how the model responds to a baseline, but the replacement input may still remove useful shared structure. We therefore construct input-dependent references $\mathbf{x}_i^r = \Phi_\psi(\mathbf{x}_i)$ that seek to move toward other-label examples while limiting changes to the original signal. The generator produces references with the same shape as the input and is fitted on all inputs in $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, which also form the reference pool. True test labels guide reference selection, making this a *transductive, label-informed* construction; it does not evaluate generalization to unseen inputs.

For each input $\mathbf{x}_i$, we select its k nearest same-label neighbors, excluding the input itself, and its k nearest different-label neighbors using ℓ_1 distance. Let $\mathcal{N}_i^{\text{same}}$ and $\mathcal{N}_i^{\text{other}}$ denote their respective index sets. We average the selected neighbors to obtain two reference signals:

$$\mathbf{r}_i^{\text{same}} = \frac{1}{k} \sum_{j \in \mathcal{N}_i^{\text{same}}} \mathbf{x}_j, \qquad \mathbf{r}_i^{\text{other}} = \frac{1}{k} \sum_{j \in \mathcal{N}_i^{\text{other}}} \mathbf{x}_j. \tag{7}$$

These references are computed once from the original inputs and remain fixed throughout training. With the default $k = 1$, each reference is an individual neighboring sample. For multiclass tasks, different-label neighbors are selected from all remaining classes, without specifying a target class.

We train the generator using a triplet-style objective and an input-change penalty. Let $D_1(\mathbf{u}, \mathbf{v}) = \|\operatorname{vec}(\mathbf{u} - \mathbf{v})\|_1/(TC)$ denote the mean absolute difference, with $C = 1$ for the univariate series used here (vec is a flatten operator). The objective is

$$\mathcal{L}_{\text{CF}}(\psi) = \frac{1}{N} \sum_{i=1}^{N} \left[\beta \max\big(0,\, D_1(\mathbf{x}_i^r, \mathbf{r}_i^{\text{other}}) - D_1(\mathbf{x}_i^r, \mathbf{r}_i^{\text{same}}) + \gamma\big) + \lambda D_1(\mathbf{x}_i^r, \mathbf{x}_i)\right]. \tag{8}$$

The first term encourages the generated signal to be closer to the different-label reference than to the same-label reference by a margin γ as a hyperparameter, we set to 0.2 by default. The second penalizes changes to the original signal, with β and λ balancing the objectives. Only the generator parameters ψ are optimized. The classifier remains frozen, so the predictions of it are unchanged.

After fitting, we use $\tilde{\mathbf{x}}_i = \mathbf{x}_i^r$ as the perturbation reference for each test input. The generator architecture and optimization details are provided in the Appendix B.1.

Replacing Artificial Baselines with Adaptive References. NULL responses and CF references address complementary limitations of perturbation baselines, as illustrated by the CNN feature projections in Figure 2. On FPH, background samples appear distinct from both classes, while GT-preserving perturbations overlap with their original class groups (panel (a)), motivating a NULL response. FPH-FD, a synthetic data, extends FPH with a tall end peak shared by both classes but excluded from the GT mask. Background replacement therefore removes this peak, producing samples that appear separate from the original classes (panel (b)). NULL cannot restore this missing structure, whereas CF changes the replacement input and yields greater visual overlap in this example (panel (c)). The pattern reverses on MC-1v5: CF-based perturbations appear separate from the original classes, particularly class 0 (panel (d)), possibly because CF introduces structure outside the retained GT region (panel (f)). Background-based perturbations instead overlap more closely with the original classes, while the background remains visually distinct (panel (e)), motivating background-based NULL evaluation. These contrasting patterns support considering both approaches, with their suitability checked through the class-conditional audit for each model and dataset.

5 Experiments

5.1 Experimental Setup

Datasets. We use six controlled synthetic datasets to examine variations in peak height, position, motif count, and shared distractors. In Fixed-Position Peak Height (FPH) and Random-Position Peak Height (RPH), peak height determines the class, while the peak position is fixed or random, respectively. Peak Position (PP) instead determines the class by which of two predefined positions contains the peak and peak heights. FPH-FD and RPH-FD extend FPH and RPH with a Fixed Distractor (FD), an identical end peak shared across classes that leaves the original class-defining rule unchanged. Motif Count (1 vs. 5) (MC-1v5) distinguishes series containing one or five peaks at random positions. We additionally evaluate on the SHAPEX synthetic datasets (including MCCE, MCCH, MTCE, MTCH) (Huang et al., 2025). For real-world evaluation, we use MIT-ECG with cardiologist-annotated beat labels. QRS intervals serve as reference explanation regions described in (Queen et al., 2023). Further details on the datasets is in the Appendix C.1. Visualization for these datasets is in Appendix E and F. More experiments on UCR benchmarks are in the Appendix D.

Architectures. We evaluate perturbation-based faithfulness across three classifier architectures for synthetic datasets: a convolutional neural network (CNN), a linear classifier (Linear), and a recurrent neural network (RNN) (Appendix C.2). For UCR datasets, we fully finetune Mantis time-series foundation model (Feofanov et al., 2026) and HYDRA (Dempster et al., 2022) (Appendix D.1).

Faithfulness Evaluation. We evaluate faithfulness across multiple retention ratios using LeRF, following previous TSC explanation studies (Huang et al., 2025; Liu et al., 2024a; Queen et al., 2023), and MoRF. We also include F-Fidelity (Zheng et al., 2025) as an additional evaluation method. We assess these configurations through aggregate and class-conditional comparisons between reference attributions and random rankings, following the audit defined in Section 2.1.

5.2 Results

How Widespread Are Class-Conditional Failures? The top rows of Figures 3 and 4 show that the class-conditional failures illustrated on FPH in Section 2 also occur on other synthetic datasets and MIT-ECG. For example, on MCCE and MCCH, all three architectures pass overall and for class 1 under both MoRF and LeRF with either zero or noise baselines, yet none passes for class 0. These results show that aggregate reference–random separation can conceal class-conditional failures across the tested datasets and evaluation settings. F-Fidelity (Zheng et al., 2025) also fails the aggregate audit in some configurations, despite using explanation-agnostic fine-tuning on randomly masked inputs and controlled stochastic removal to address distribution shift. One possible contributor is the interaction between masking and localized task evidence. For example, the GT region in FPH occupies only $25/512 \approx 4.9\%$ of the series. The effect of masking therefore depends on its overlap with the class-defining peak, while fine-tuning may change the model's sensitivity to partially removed evidence. Whether these effects explain the observed failures requires further investigation.

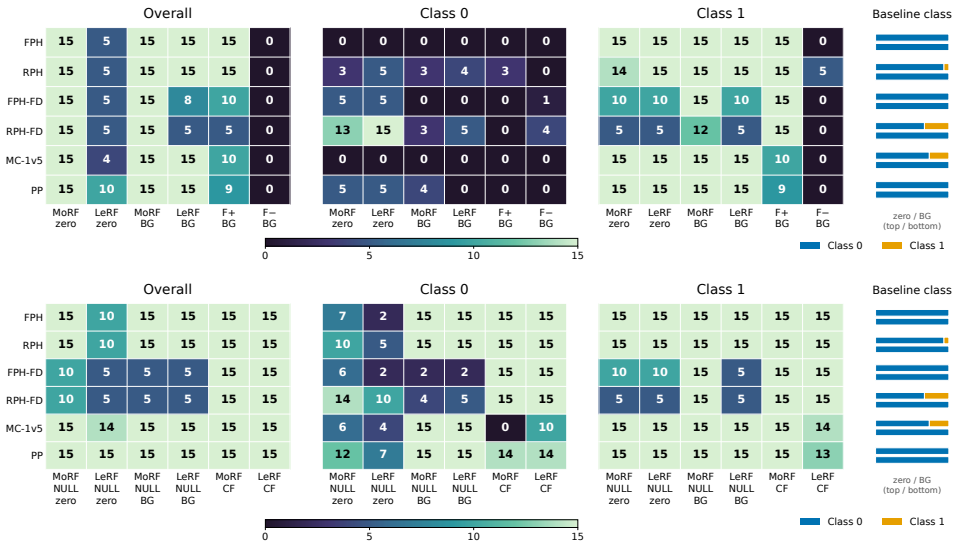

Figure 3: Faithfulness audit across six synthetic datasets. Cells count **healthy** evaluations among 15 model–seed combinations (three architectures, five seeds). **Top:** Zero/background (BG) baselines and F-Fidelity variants. **Bottom:** NULL responses and counterfactual (CF) references. **Right:** Baseline prediction proportions for zero (upper bar) and BG (lower bar), illustrating near-collapse to a single class. Aggregate separation can conceal class-conditional failures; NULL and CF improve separation in several settings, but neither passes all tested configurations.

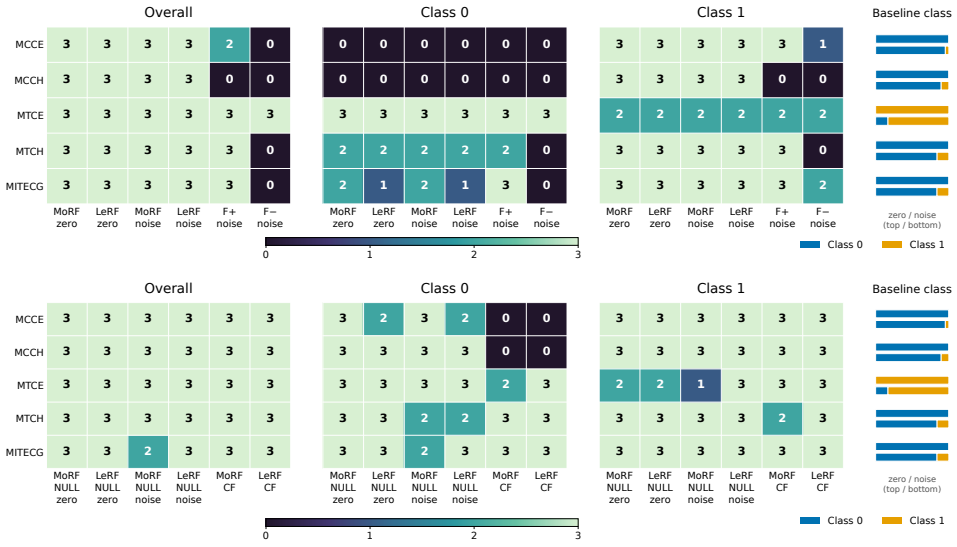

Figure 4: Faithfulness audit across the annotated time-series benchmarks. Cells count **healthy** evaluations across three architectures. **Top:** Zero/noise baselines and F-Fidelity variants. **Bottom:** NULL responses and counterfactual (CF) references. **Right:** Baseline prediction proportions for zero (upper bar) and noise (lower bar), illustrating near-collapse to a single class. Aggregate separation can conceal class-conditional failures; NULL and CF improve separation in several settings, but neither passes all tested configurations.

Do NULL Responses Improve Class-Conditional Separation? We next examine whether adding a NULL response improves GT–Random separation within each class for our and SHAPEX synthetic dataset and MIT-ECG (lower row in Figures 3 and 4). On FPH and MC-1v5, background-based NULL increases **healthy** class 0 evaluations from $0/15$ to $15/15$ under both MoRF and LeRF, while preserving **healthy** overall and class 1 results. However, on FPH-FD with the background

baseline, only $2/15$ class 0 evaluations pass in either direction. NULL therefore restores separation in several settings but leaves failures in others, motivating comparison with CF references.

Do Counterfactual References Improve Class-Conditional Separation? We next examine whether CF improves separation where fixed baselines and NULL responses remain insufficient. On FPH-FD and RPH-FD, all $15/15$ model–seed combinations pass the audit overall and within each class under both MoRF and LeRF (Figure 3). CF also passes across all three architectures, both classes, and both directions on MIT-ECG (Figure 4). However, CF is not consistently better. On MC-1v5, all aggregate evaluations pass, but class 0 has $0/15$ **healthy** evaluations under MoRF and $10/15$ under LeRF, compared with $15/15$ in both directions for background-based NULL. MCCE and MCCH, also motif-count tasks (Huang et al., 2025), show similar class 0 failures. This pattern suggests that CF may alter count-related evidence outside the retained GT regions, although further analysis is needed to confirm this explanation. CF can therefore improve separation where NULL struggles, while NULL performs better in others, reinforcing the need to audit each configuration.

6 A Practical Audit for Perturbation-Based Faithfulness

Our results show that NULL and CF improve separation in different settings, but neither passes the audit in every case. We therefore recommend an audit system to guide evaluator selection. The first step is to list candidate faithfulness metrics and baseline configurations and define which classes require explanations and which fall outside the application's scope. The audit then checks whether each candidate separates informative reference attributions from random rankings within every required class, using a predefined tolerance. Configuration selection should use a separate audit or validation set where possible, and all class-level outcomes should be reported. Only configurations that pass this gate should be used within the specified scope, although passing does not certify explanation correctness. If none passes, the scores should not be treated as reliable evidence for ranking explainers. The audit should be repeated whenever the training and evaluation setup changes.

7 Related Work

Faithfulness evaluation and evaluator reliability. Existing work evaluates time-series explanations across perturbation strategies and classifiers, including comparisons with random attributions (Nguyen et al., 2023). Other approaches address distribution shifts through training or fine-tuning with perturbed inputs (Turbé et al., 2022; Zheng et al., 2025). MetaQuantus examines evaluator reliability through resilience to noise and reactivity to randomness (Hedström et al., 2023). Our audit asks whether informative–random separation observed overall persists within each class of a trained model. More extensive related work is in Appendix Section A.

Perturbation baselines and adaptive references. Time-series explanation methods develop realistic or dynamic perturbations (Sivill & Flach, 2022; Crabbe & van der Schaar, 2021), learn replacements alongside attribution masks (Enguehard, 2023), or construct counterfactual perturbations (Liu et al., 2024b). SpectralX also considers class-specific explanations in the time-frequency domain (Chung et al., 2024). These approaches primarily use perturbations to construct explanations while we investigate CF references for evaluating existing attributions.

8 Conclusion

In this paper, we introduce a class-conditional audit for perturbation-based faithfulness evaluation in time series. Across synthetic and real-world datasets, we found that evaluators can distinguish informative from random attributions overall yet fail within individual classes. Our analysis suggests that perturbation baselines can contribute to these failures. Therefore, we introduce a NULL response on frozen representations and investigated input-dependent counterfactual references as another approache. Both improve separation in some settings, but neither passes the audit across all tested configurations. We therefore recommend auditing each configuration for the trained model, dataset, and application-relevant classes before comparing explanation methods. This audit provides an initial check of evaluator suitability but does not certify explanation correctness or replace more comprehensive evaluation.

AI USE STATEMENT

We used generative AI tools to improve the language and readability of paragraphs drafted by the authors. The authors reviewed all suggested changes and revised them where necessary to preserve the intended meaning and research content. The authors take full responsibility for the contents of this submission.

ETHICS STATEMENT

This work aims to support responsible use of time-series explanations by identifying failures in faithfulness evaluation. Our experiments use synthetic datasets and existing time-series benchmarks, including ECG data. Misleading evaluation scores may encourage unjustified trust in explanations, particularly when aggregate results conceal failures within classes relevant to an application. Our class-conditional audit helps identify such failures, but passing it does not establish explanation correctness or guarantee suitability for a particular application. We therefore recommend reporting all class-level outcomes and defining the intended evaluation scope before selecting a configuration. The ECG experiments concern explanation evaluation, only use published ECG dataset, and should not be interpreted as validation for clinical deployment.

REPRODUCIBILITY STATEMENT

The main text defines the perturbation procedure, faithfulness scores, and class-conditional audit. Appendices C.1 and C.2 describe the datasets, reference explanation regions, and classifier architectures. Appendix D documents the additional UCR experiments, including the Mantis and HYDRA models, reference attribution methods, and attribution generation using original and NULL-augmented models. Appendix B.1 describes the counterfactual generator, reference construction, and optimization settings, including the use of test inputs and labels. Appendix B.2 specifies the NULL-training baselines, frozen model components, optimization settings, validation procedure, and reproducible noise generation. Detailed audit results in Appendices C and D support inspection of the reported aggregate and class-conditional outcomes.

REFERENCES

Emre Cihan Ates, Burak Aksar, Vitus J. Leung, and Ayse Kivilcim Coskun. Counterfactual explanations for multivariate time series. *ICAPAI*, pp. 1–8, 2020. URL https://api.semanticscholar.org/CorpusID:235739487.

Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data Soc.*, 3, 2016. URL https://api.semanticscholar.org/CorpusID:61330970.

Hyunseung Chung, Sumin Jo, Yeonsu Kwon, and Edward Choi. Time is not enough: Time-frequency based explanation for time-series black-box models. In *CIKM*, pp. 394–403, 2024.

Jonathan Crabbe and Mihaela van der Schaar. Explaining time series predictions with dynamic masks. In *ICML*, 2021.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Eoin Delaney, Derek Greene, and Mark T. Keane. Instance-based counterfactual explanations for time series classification. *ArXiv*, abs/2009.13211, 2020. URL https://api.semanticscholar.org/CorpusID:221970060.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. Hydra: competing convolutional kernels for fast and accurate time series classification. *Data Min. Knowl. Discov.*, 37:1779–1805, 2022. URL https://api.semanticscholar.org/CorpusID:247748968.

Joseph Enguehard. Learning perturbations to explain time series predictions. In *International Conference on Machine Learning*, pp. 9329–9342. PMLR, 2023.

Vasilii Feofanov, Songkang Wen, Shifeng Xie, Simon Roschmann, Marius Alonso, Hongbo Guo, Romain Ilbert, Malik Tiomoko, Quentin Bouniot, Zeynep Akata, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight foundation model for time series classification. In *ICML*, 2026. URL https://openreview.net/forum?id=gbJMAjXLZ4.

Charles Findling, Félix Hubert, Luigi Acerbi, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Matteo Carandini, Joana A Catarino, Gaëlle A. Chapuis, Anne K. Churchland, Yang Dan, Eric E. J. DeWitt, Tatiana A. Engel, Michele Fabbri, Mayo Faulkner, Ila Rani Fiete, Laura Freitas-Silva, Berk Gerçek, Kenneth D. Harris, Michael Häusser, Sonja B. Hofer, Fei Hu, Julia M. Huntenburg, Anup Khanal, Christopher S. Krasniak, Christopher Langdon, Peter E. Latham, Petrina Y. P. Lau, Zachary F. Mainen, Guido T. Meijer, Nathaniel J. Miska, Thomas D. Mrsic-Flogel, Jean-Paul Noel, Kai Nylund, Alejandro Pan-Vazquez, Liam Paninski, Jonathan W. Pillow, Cyrille Rossant, Noam Roth, Rylan Schaeffer, Michael M Schartner, Yan Shi, Karolina Z. Socha, Nicholas A. Steinmetz, Karel Svoboda, Charline Tessereau, Anne E. Urai, Miles J. Wells, Steven Jon West, Matthew R Whiteway, Olivier Winter, Ilana B. Witten, Anthony M. Zador, Peter Dayan, and Alexandre Pouget. Brain-wide representations of prior information in mouse decision-making. *Nature*, 645:192 – 200, 2025. URL https://api.semanticscholar.org/CorpusID:259361846.

Alan H. Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. Explaining deep classification of time-series data with learned prototypes. *CEUR workshop proceedings*, 2429:15 – 22, 2019. URL https://api.semanticscholar.org/CorpusID:125974424.

Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. *KDD*, 2014. URL https://api.semanticscholar.org/CorpusID:207214410.

Awni Y. Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H. Tison, Codie Bourn, Mintu P. Turakhia, and A. Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25:65 – 69, 2019. URL https://api.semanticscholar.org/CorpusID:57574627.

Anna Hedström, Philine Lou Bommer, Kristoffer Wickstrøm, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. The meta-evaluation problem in explainable ai: Identifying reliable estimators with metaquantus. *ArXiv*, abs/2302.07265, 2023. URL https://api.semanticscholar.org/CorpusID:256846994.

Tsung-Yu Hsieh, Suhang Wang, Yiwei Sun, and Vasant Honavar. Explainable multivariate time series classification: a deep neural network which learns to attend to important variables as well as time intervals. In *WSDM*, pp. 607–615, 2021.

En-Yu Hsu, Chien-Liang Liu, and Vincent S. Tseng. Multivariate time series early classification with interpretability using deep learning and attention mechanism. In *PAKDD*, 2019. URL https://api.semanticscholar.org/CorpusID:92996201.

Bosong Huang, Ming Jin, Yuxuan Liang, Johan Barthelemy, Debo Cheng, Qingsong Wen, Chenghao Liu, and Shirui Pan. Shapex: Shapelet-driven post hoc explanations for time series classification models. In *NeurIPS*, 2025.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *ACL*, 2020. URL https://api.semanticscholar.org/CorpusID:215416110.

Hyeongwon Jang, Changhun Kim, and Eunho Yang. Timing: Temporality-aware integrated gradients for time series explanation. In *ICML*, 2025.

Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and A. Gionis. Locally and globally explainable time series tweaking. *Knowl. Inf. Syst.*, 62:1671 – 1700, 2019. URL https://api.semanticscholar.org/CorpusID:201671223.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch. *ArXiv*, abs/2009.07896, 2020. URL https://api.semanticscholar.org/CorpusID:221761135.

Yutaka Komura, Akihiko Nikkuni, Noriko Hirashima, Teppei Uetake, and Aki Miyamoto. Responses of pulvinar neurons reflect a subject's confidence in visual categorization. *Nature neuroscience*, 16(6):749–755, 2013.

Jokin Labaien, Ekhi Zugasti, and Xabier De Carlos. Contrastive explanations for a deep learning model on time-series data. In *International Conference on Data Warehousing and Knowledge Discovery*, 2020. URL https://api.semanticscholar.org/CorpusID:221667581.

Himabindu Lakkaraju and Osbert Bastani. "how do i fool you?": Manipulating user trust via misleading black box explanations. *AIES*, 2019. URL https://api.semanticscholar.org/CorpusID:208077044.

Jeongsu Lee, Young Chul Lee, and Jeong Tae Kim. Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database. *J. Manuf. Syst.*, 57:357–366, 2020. URL https://api.semanticscholar.org/CorpusID:229028398.

Jason Lines, Luke M. Davis, Jon Hills, and A. Bagnall. A shapelet transform for time series classification. In *KDD*, 2012. URL https://api.semanticscholar.org/CorpusID:2505904.

Zichuan Liu, Tianchun Wang, Jimeng Shi, Xu Zheng, Zhuomin Chen, Lei Song, Wenqian Dong, Jayantha Obeysekera, Farhad Shirani, and Dongsheng Luo. Timex++: Learning time-series explanations with information bottleneck. *arXiv preprint arXiv:2405.09308*, 2024a.

Zichuan Liu, Yingying Zhang, Tianchun Wang, Zefan Wang, Dongsheng Luo, Mengnan Du, Min Wu, Yi Wang, Chunlin Chen, Lunting Fan, and Qingsong Wen. Explaining time series via contrastive and locally sparse perturbations. *ArXiv*, abs/2401.08552, 2024b. URL https://api.semanticscholar.org/CorpusID:267027677.

Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. *KDD*, 2019. URL https://api.semanticscholar.org/CorpusID:189805778.

Abdullah Mueen, Eamonn J. Keogh, and N. Young. Logical-shapelets: an expressive primitive for time series classification. In *KDD*, 2011. URL https://api.semanticscholar.org/CorpusID:8214415.

Thu Trang Nguyen, Thach Le Nguyen, and Georgiana Ifrim. Robust explainer recommendation for time series classification. *Data Min. Knowl. Discov.*, 38:3372 – 3413, 2023. URL https://api.semanticscholar.org/CorpusID:263882730.

Owen Queen, Tom Hartvigsen, Teddy Koker, Huan He, Theodoros Tsiligkaridis, and Marinka Zitnik. Encoding time-series explanations through self-supervised model behavior consistency. *NeurIPS*, 36:32129–32159, 2023.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. *KDD*, 2016.

Pavel Senin and Sergey Malinchik. Sax-vsm: Interpretable time series classification using sax and vector space model. *ICDM*, pp. 1175–1180, 2013. URL https://api.semanticscholar.org/CorpusID:15401368.

Vera Shalaeva, Sami Alkhoury, Julien Marinescu, Cécile Amblard, and Gilles Bisson. Multi-operator decision trees for explainable time-series classification. In *International Conference on Information Processing and Management of Uncertainty*, 2018. URL https://api.semanticscholar.org/CorpusID:44162670.

Torty Sivill and Peter A. Flach. Limesegment: Meaningful, realistic time series explanations. In *AISTATS*, 2022.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Nathalie A Smuha. Regulation 2024/1689 of the eur. parl. & council of june 13, 2024 (eu artificial intelligence act). *ILM*, 64(5):1234–1381, 2025.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, pp. 3319–3328. PMLR, 2017.

Wensi Tang, Lu Liu, and Guodong Long. Interpretable time-series classification on few-shot samples. *IJCNN*, pp. 1–8, 2020. URL https://api.semanticscholar.org/CorpusID:219260916.

Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. *ArXiv*, abs/1905.05134, 2019. URL https://api.semanticscholar.org/CorpusID:152282619.

Viet-Hung Tran, Zichi Zhang, Tuan Dung Pham, Ngoc Phu Doan, Anh-Tuan Hoang, Peixin Li, Hans Vandierendonck, Ira Assent, and Son Thai Mai. Intedisux: Intepretation-guided discriminative user-centric explanation for time series. In *AAAI*, 2025.

Viet-Hung Tran, Ngoc Phu Doan, Zichi Zhang, Tuan Dung Pham, Phi Hung Nguyen, Xuan Hoang Nguyen, Hans Vandierendonck, Ira Assent, and Son T. Mai. MIX: A multi-view time-frequency interactive explanation framework for time series classification. In *NeurIPS*, 2026.

Hugues Turbé, Mina Bjelogrlic, Christian Lovis, and Gianmarco Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nat. Mach. Intell.*, 5:250–260, 2022. URL https://api.semanticscholar.org/CorpusID:257528824.

Phongtharin Vinayavekhin, Subhajit Chaudhury, Asim Munawar, Don Joven Agravante, Giovanni De Magistris, Daiki Kimura, and Ryuki Tachibana. Focusing on what is relevant: Time-series learning and understanding using attention. *ICPR*, pp. 2624–2629, 2018. URL https://api.semanticscholar.org/CorpusID:49396831.

Yi Yang, Qilin Deng, Shen Furao, Jinxi Zhao, and C. Luo. A shapelet learning method for time series classification. *ICTAI*, pp. 423–430, 2016. URL https://api.semanticscholar.org/CorpusID:17937218.

Lexiang Ye and Eamonn J. Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Min. Knowl. Discov.*, 22:149–182, 2010. URL https://api.semanticscholar.org/CorpusID:24632217.

Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *AAAI*, 2020. URL https://api.semanticscholar.org/CorpusID:210703726.

Xu Zheng, Farhad Shirani, Zhuomin Chen, Chaohao Lin, Wei Cheng, Wenbo Guo, and Dongsheng Luo. F-fidelity: A robust framework for faithfulness evaluation of explainable ai. In *ICLR*, volume 2025, pp. 12772–12804, 2025.

APPENDIX

We first provide a broader discussion of related work in Section A. Section C presents additional results on class-conditional failures and examines how NULL responses and counterfactual references can improve separation. We then extend the evaluation to eight UCR time-series datasets in Section D to assess whether our findings hold across additional tasks. Section E and F provides visualizations of inputs, linear-model logits, and CNN and RNN representations to further examine the distinction between baselines and real inputs that motivates the NULL response. Finally, Section G discusses the limitations of our analysis and recommendations.

A EXTENDED RELATED WORK

Post-hoc Explanations for Time-Series Classification. Post-hoc approaches explain trained TSC models without modifying their learned parameters. Many produce attribution maps that assign importance scores to individual timesteps or input features. These include general-purpose gradient methods (Sundararajan et al., 2017; Smilkov et al., 2017) and approaches developed specifically for temporal data (Queen et al., 2023; Liu et al., 2024a; Jang et al., 2025; Crabbe & van der Schaar, 2021; Enguehard, 2023; Liu et al., 2024b). Other methods explain predictions through temporal segments or discriminative subsequences (Sivill & Flach, 2022; Huang et al., 2025; Tran et al., 2025). Time-frequency approaches additionally examine importance across temporal and frequency representations (Chung et al., 2024; Tran et al., 2026). Beyond attribution maps, counterfactual and contrastive approaches examine alternative inputs or input changes that help explain a model's prediction (Delaney et al., 2020; Karlsson et al., 2019; Ates et al., 2020; Labaien et al., 2020).

Ante-hoc Explanations for Time-Series Classification. Ante-hoc approaches incorporate interpretable structures or explanatory mechanisms into the model design. Examples include shapelet representations (Ye & Keogh, 2010; Lines et al., 2012; Mueen et al., 2011; Grabocka et al., 2014; Yang et al., 2016), attention mechanisms (Vinayavekhin et al., 2018; Hsu et al., 2019; Hsieh et al., 2021), and learned prototypes (Tang et al., 2020; Zhang et al., 2020; Ming et al., 2019; Gee et al., 2019). Other approaches use symbolic representations (Senin & Malinchik, 2013) or decision trees designed for temporal data (Shalaeva et al., 2018). However, incorporating an explanatory mechanism does not automatically establish that the resulting explanation faithfully reflects the model's prediction process. Work in NLP highlights this distinction and questions whether mechanisms such as attention can be assumed faithful solely because they are part of the model (Jacovi & Goldberg, 2020). These observations motivate evaluating faithfulness in ante-hoc TSC explanations as well, without assuming that the same failures necessarily occur across domains.

Explanation Evaluation. Explanation methods are evaluated through agreement with reference annotations, faithfulness to model behaviour, and robustness. When explanatory regions are known from synthetic data construction or annotated by domain experts, attribution maps can be compared with these regions. Common measures include the area under the precision–recall curve (AUPRC), areas under the precision and recall curves (AUP and AUR), and overlap measures such as intersection over union (IoU) (Queen et al., 2023; Liu et al., 2024a). Agreement with these annotations measures recovery of the reference evidence, but does not by itself establish that an explanation reflects the evidence used by a particular trained model. Faithfulness evaluation instead examines consistency with model behaviour, commonly through prediction changes under attribution-guided perturbations, and can be applied without explanation annotations. Several evaluation frameworks address limitations such as perturbation-induced distribution shifts and sensitivity to the choice of perturbations or evaluating classifiers (Turbé et al., 2022; Nguyen et al., 2023; Zheng et al., 2025). Robustness studies provide a complementary perspective by measuring how explanations change under small input perturbations, including added noise (Sivill & Flach, 2022; Tran et al., 2025; Chung et al., 2024).

Evaluator Auditing. The reliability of explanation evaluators also requires examination before their scores are used to compare explanation methods. MetaQuantus addresses this meta-evaluation problem by assessing an evaluator's resilience to noise and reactivity to randomness (Hedström et al., 2023). Controlled comparisons also appear in evaluation frameworks: AMEE includes random and oracle explanations when assessing task informativeness (Nguyen et al., 2023), while F-Fidelity tests whether its scores distinguish explainers from degraded versions (Zheng et al., 2025). Our

work complements these efforts by checking whether informative–random separation observed at the dataset level persists within each class of a particular trained model. This provides a focused diagnostic for class-conditional failures alongside broader analyses of evaluator reliability.

B NULL AND COUNTERFACTUAL IMPLEMENTATION

B.1 COUNTERFACTUAL GENERATOR IMPLEMENTATION

Architecture. The generator consists of a single-layer bidirectional GRU with 64 hidden units per direction. The generator directly produces a univariate reference with the same length as the input. We use no dropout, residual connection, or output activation.

Reference construction. All test inputs serve as both queries and the reference pool. For each input, we select its nearest same-label neighbor, excluding itself, and its nearest different-label neighbor using ℓ_1 distance. Thus, the default neighborhood size is $k = 1$. For multiclass datasets, different-label candidates include all remaining classes. The references are computed once and remain fixed throughout optimization. Because reference selection uses true test labels, this is a label-informed construction.

Optimization. We train the generator for 100 epochs using Adam with a learning rate of 10^{-2}, batch size 512, and zero weight decay. The training order is randomly shuffled at each epoch. The loss uses triplet weight $\beta = 1$, input-change weight $\lambda = 0.1$, and margin $\gamma = 0.2$. Distances in the loss are mean absolute differences across input elements. We use the final-epoch generator, without learning-rate scheduling or early stopping. The classifier remains frozen and it is only used afterward to measure prediction changes.

B.2 NULL-TRAINING BASELINES AND OPTIMIZATION

Constructing the baseline collection. For each training input $\mathbf{x}_i$, we construct a corresponding baseline $\mathbf{b}_i$ with the same shape. The collection $\mathcal{B}_{\text{train}} = \{\mathbf{b}_i\}_{i=1}^{N_{\text{train}}}$ therefore contains one baseline example per real training example. Its construction follows the replacement strategy used during faithfulness evaluation.

For the *zero baseline*, we set $\mathbf{b}_i = \mathbf{0}$ in the model's input space. For the *noise baseline*, each input element is sampled independently as

$$b_{i,t,c} \sim \mathcal{N}(0, \sigma^2_{\text{noise}}).$$

The supplied Mantis and HYDRA implementation uses $\sigma_{\text{noise}} = 1$ by default. Noise is generated directly in the model-input space and sampled once per split, remaining fixed throughout NULL-head training.

For the *background baseline* on controlled synthetic datasets, $\mathbf{b}_i$ is the saved background signal associated with $\mathbf{x}_i$. This preserves the background realization used to generate that particular series while excluding the inserted synthetic patterns. The saved background undergoes the same preprocessing as the corresponding input. For example, on FPH, the raw background consists of independent Gaussian noise with standard deviation 0.1. The input and its background are then transformed using the same global Z-score normalization fitted on the training split.

Training the NULL head. We combine the real training inputs, with their original labels, and an equal number of baseline inputs assigned to the additional NULL class. The original representation and real-class logits remain fixed. We cache their values for real and baseline inputs without computing gradients, and optimize only the linear NULL head using cross-entropy over the augmented class outputs.

The default configuration uses Adam with a learning rate of 10^{-3}, weight decay 10^{-3}, batch size 256, and 100 epochs. The combined training pool is shuffled at each epoch. The minibatch size refers to the pooled real and baseline examples.

Validation and noise generation. We select the NULL head with the lowest validation loss across the training epochs. The selection criterion is the mean of the cross-entropy loss on real validation inputs and the loss on validation baselines labelled NULL. Test inputs and test baselines are not

used for optimization or checkpoint selection. For Gaussian baselines, separate reproducible random streams are used for training, validation, and testing.

C Experiments on Synthetic and MIT-ECG Datasets

C.1 Experimental Datasets

Controlled synthetic datasets. We use six univariate synthetic datasets to examine how perturbation-based faithfulness changes with the class-defining signal and the presence of shared structure. These datasets cover peak height, peak position, motif count, and fixed distractors, with task-defined regions serving as reference explanations for the audit.

Fixed-Position Peak Height (FPH). Each series contains a single peak at a predefined position on a low-amplitude Gaussian-noise background. The peak height determines the class, and the peak region forms the **ground-truth (GT)** explanation mask. The background is saved separately and used as a perturbation baseline, allowing the peak to be removed while retaining the original background realization.

Random-Position Peak Height (RPH). RPH follows the same height-based classification rule as FPH, but the peak position varies randomly between inputs. The class-defining evidence therefore occurs at different temporal locations, allowing us to examine whether the evaluation remains informative when the relevant region is not fixed.

Peak Position (PP). In PP, a single peak appears at one fixed position for class 0 and another for class 1. Peak heights are similar among samples from the same class but differ between the two classes. Both the height of the peak and where it appears therefore help distinguish the classes.

Fixed-distractor variants (FPH-FD and RPH-FD). These datasets extend FPH and RPH by adding a peak of height 28 near the end of every series. The added peak has the same height and position across classes, while the height of the original peak continues to determine the label. The class-defining peak remains fixed in FPH-FD and varies in position in RPH-FD. The shared end peak is excluded from the GT mask. Consequently, retaining only the GT region on a peakless background also removes this shared structure, allowing us to examine the effect of changes outside the annotated discriminative region.

Motif Count (1 vs. 5) (MC-1v5). The two classes contain one or five peaks at randomly selected positions. The classification rule therefore depends on the number of motifs rather than their occurrence at a fixed location. This setting allows us to examine perturbations that may remove existing motifs or introduce additional structure outside the retained reference regions.

SHAPEX synthetic datasets. We also use the four motif-based datasets introduced in (Huang et al., 2025): MCCE, MCCH, MTCE, and MTCH. MCCE and MCCH define the class through motif count, whereas MTCE and MTCH use motif type. The E variants use motifs with amplitudes comparable to the surrounding waveform, while the H variants use motifs with higher amplitudes. These variants distinguish temporal patterns that blend into the surrounding signal from patterns that are more prominent in amplitude. The supplied temporal annotations provide the reference explanation regions used in our audit.

MIT-ECG. For real-world evaluation, we use ECG inputs with cardiologist-annotated beat labels. We obtain QRS intervals using the NeuroKit-based extraction procedure described in (Queen et al., 2023) and use these intervals as reference explanation regions. The beat annotations define the classification labels, while the extracted intervals define the regions used for explanation evaluation. These reference regions do not establish that every trained model bases its predictions exclusively on the QRS interval.

C.2 Experimental Architectures

We use a linear classifier, a CNN, and an RNN. The linear classifier maps the flattened input directly to class logits. The CNN contains one one-dimensional convolutional layer with 8 filters of kernel size 21, followed by ReLU and global average pooling over time. The RNN uses a single-layer GRU

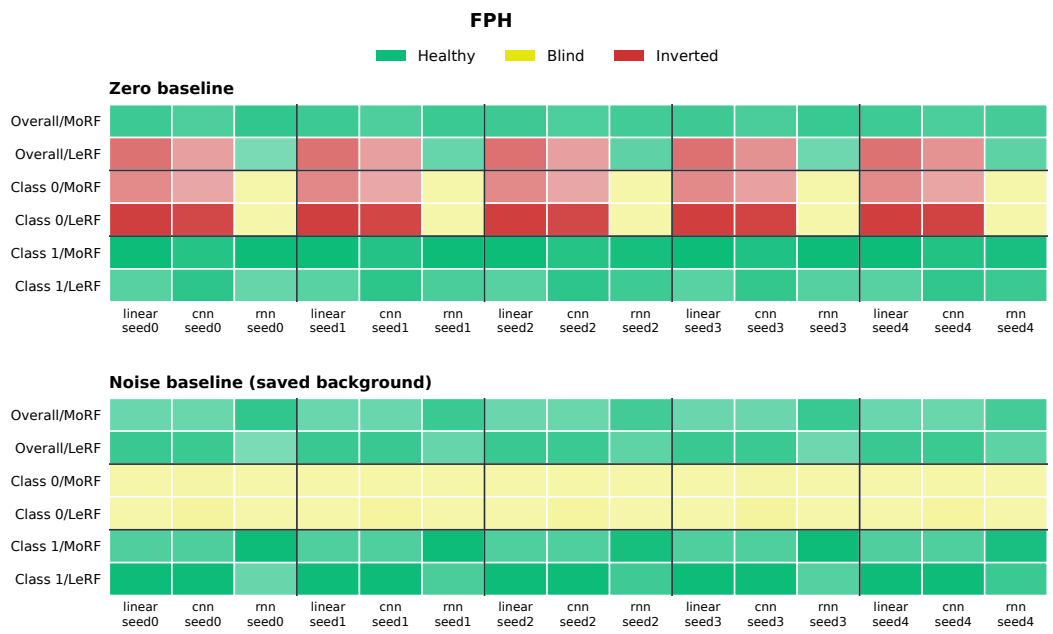

Figure 5: Faithfulness audit on **FPH** across three architectures and five training seeds. **Top:** Zero baseline. **Bottom:** Saved noise backgrounds used as perturbation baselines. Columns represent model–seed combinations, and rows show overall and class-conditional evaluations under MoRF and LeRF. Green, yellow, and red indicate **healthy**, **blind**, and **inverted** GT–Random separation, respectively, with audit tolerance $\epsilon = 0.005$. With background baselines, all combinations are **healthy** overall and for class 1 but blind for class 0 in both directions. With zero baselines, class 0 evaluations are inverted for Linear and CNN and blind for RNN; aggregate LeRF evaluations are also inverted for Linear and CNN.

with 64 hidden units and averages its hidden states across timesteps. Both pooled representations are passed to a linear classification layer.

C.3 Results on Synthetic Datasets

Figures 5, 6, 7, 8, and 9 report **Healthy**, **Blind**, and **Inverted** audit outcomes across five synthetic datasets. Across all five datasets, some configurations are **healthy** overall but blind or inverted within an individual class.

D Additional Experiments for UCR Datasets

D.1 Experimental Setup

Datasets. We extend the evaluation to seven binary UCR datasets in (Dau et al., 2019): FordA, Wafer, SemgHandGenderCh2, HouseTwenty, ECG200, HandOutlines, and GunPointMaleVersus-Female. We also include the four-class TwoPatterns dataset to examine class-conditional evaluation in a multiclass setting. The corresponding audit results are reported in Figures 12 and 13.

Architectures. For these experiments, we use a fully fine-tuned Mantis time-series foundation model (Feofanov et al., 2026) with a fully connected head adapted to the downstream task and HYDRA (Dempster et al., 2022). HYDRA uses fixed random convolutional kernels to construct count-based features, followed by a trained linear classifier. These models extend our evaluation to both a fine-tuned foundation model and a classifier built on fixed convolutional features.

Reference attributions. These UCR datasets do not provide ground-truth explanation annotations. We therefore use Integrated Gradients (IG) (Sundararajan et al., 2017) and feature ablation from

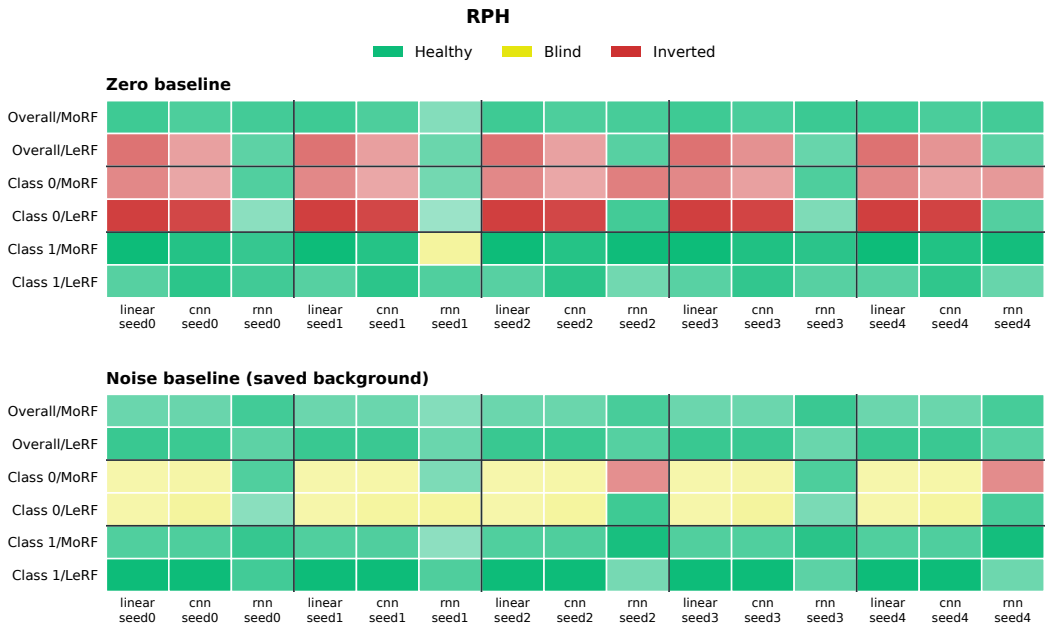

Figure 6: Faithfulness audit on **RPH** across three architectures and five training seeds. **Top:** Zero baseline. **Bottom:** Saved noise backgrounds used as perturbation baselines. Columns represent model–seed combinations, and rows show overall and class-conditional evaluations under MoRF and LeRF. Green, yellow, and red indicate **healthy**, **blind**, and **inverted** GT–Random separation, respectively, with audit tolerance $\epsilon = 0.005$. With background baselines, all combinations are **healthy** overall and for class 1, while class 0 is blind for Linear and CNN and varies across seeds and perturbation directions for RNN. With zero baselines, Linear and CNN produce inverted class 0 evaluations in both directions, despite **healthy** aggregate MoRF scores.

Captum (Kokhlikyan et al., 2020) to generate reference attribution maps, whose rankings are compared with random rankings. Because these references are estimated, a failure to separate them from random may reflect limitations of the attribution method, the evaluator, or both.

D.2 How We Treat Baselines in Explanation Generation

Experimental Setup. Alongside the conventional setting, where IG and feature ablation explain the original classifier, we consider an additional setting in which they explain the NULL-augmented model. For this comparison, we use real-class softmax probabilities as the attribution targets. This choice allows us to examine the effect of NULL, since adding the output changes these probabilities while leaving the real-class logit functions unchanged.

Motivation for Studying Attributions on NULL Models. Figures 10 and 11 motivate examining whether NULL also affects attribution generation. For IG applied to class probabilities, the signed attributions sum to the input–baseline probability difference (Sundararajan et al., 2017). Feature ablation likewise measures changes in the selected output when features are replaced (Kokhlikyan et al., 2020). Our FordA results show that adding NULL to Mantis can increase the input–baseline probability difference while preserving its representation and real-class logits. This motivates comparing probability-based attributions from the original and NULL-augmented models. However, the endpoint plots alone do not establish better attribution rankings: a small signed IG sum can reflect cancellation, and feature ablation depends on responses to partial replacements. This comparison therefore examines an additional use of NULL for explanation generation, beyond its role in evaluating existing attribution maps.

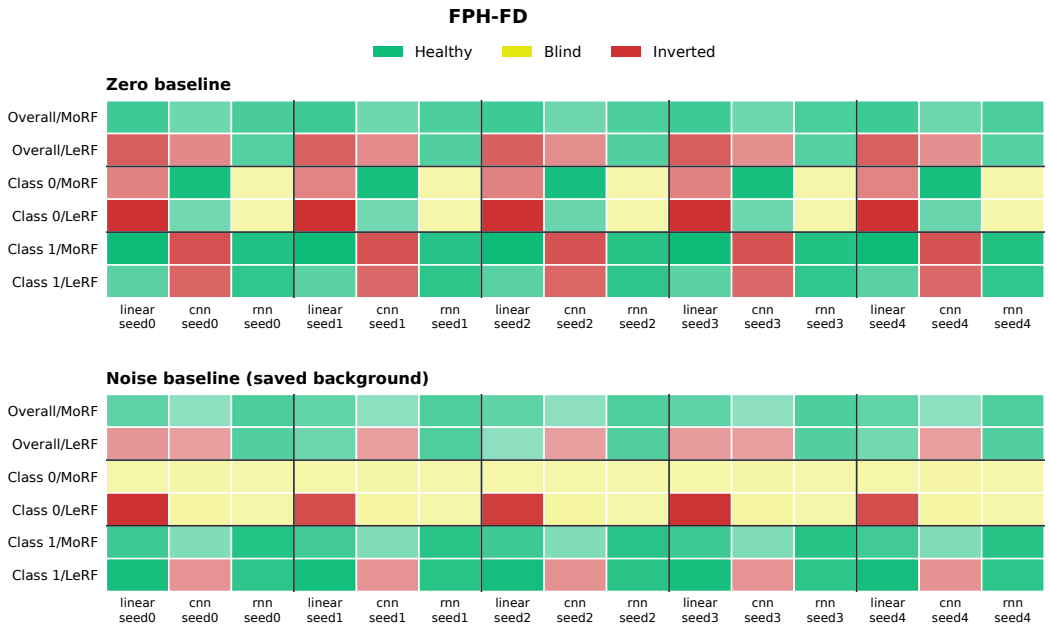

Figure 7: Faithfulness audit on **FPH-FD** across three architectures and five training seeds. **Top:** Zero baseline. **Bottom:** Saved noise backgrounds used as perturbation baselines. Columns represent model–seed combinations, and rows show overall and class-conditional evaluations under MoRF and LeRF. Green, yellow, and red indicate **healthy**, **blind**, and **inverted** GT–Random separation, respectively, with audit tolerance $\epsilon = 0.005$. With background baselines, all combinations are **healthy** under aggregate MoRF evaluation, yet class 0 is blind under MoRF and either blind or inverted under LeRF. With zero baselines, both directions yield inverted evaluations for Linear on class 0 and CNN on class 1, while RNN is blind for class 0.

D.3 Results on UCR Datasets

Results on TwoPatterns. Figure 12 shows that aggregate faithfulness can also conceal class-conditional failures in a multiclass setting. All tested configurations pass the aggregate audit for both Mantis and HYDRA, but their class-level outcomes vary. With zero or noise baselines, only one of the two models passes for class 1 under either MoRF or LeRF. Adding a noise-based NULL response yields **healthy** evaluations for both models across all four classes and both directions. CF references, however, yield only one **healthy** class 1 evaluation in each direction and none for class 2 under LeRF, despite passing every aggregate evaluation. These results extend the observed discrepancy to multiclass classification and reinforce the need to check reference–random separation within each class before selecting an evaluation configuration.

Results on Other UCR Datasets. The seven binary UCR datasets show similar differences between aggregate and class-level audit outcomes (Figure 13). For example, on ECG200, both models pass every aggregate evaluation with zero and noise baselines, yet class 0 has no **healthy** evaluations under MoRF and only one under LeRF. NULL responses and CF references improve separation in some settings. Noise-based NULL increases **healthy** class 0 evaluations on ECG200 from $1/2$ to $2/2$ under LeRF, while CF passes for both models, both classes, and both directions on SemgHandGenderCh2. However, these improvements remain configuration dependent. On HouseTwenty, class 1 still passes for only one model under every NULL configuration, while CF yields no **healthy** class 0 evaluations on ECG200 under MoRF. These results reinforce the need to assess reference–random separation for each model, class, and perturbation configuration, even when dataset-level results appear **healthy**.

Results on TwoPatterns with NULL Models. Generating attributions from NULL-augmented models does not by itself eliminate class-conditional failures. Figure 15 shows two **healthy** aggregate evaluations for every configuration, while class-level outcomes vary. Without a NULL response

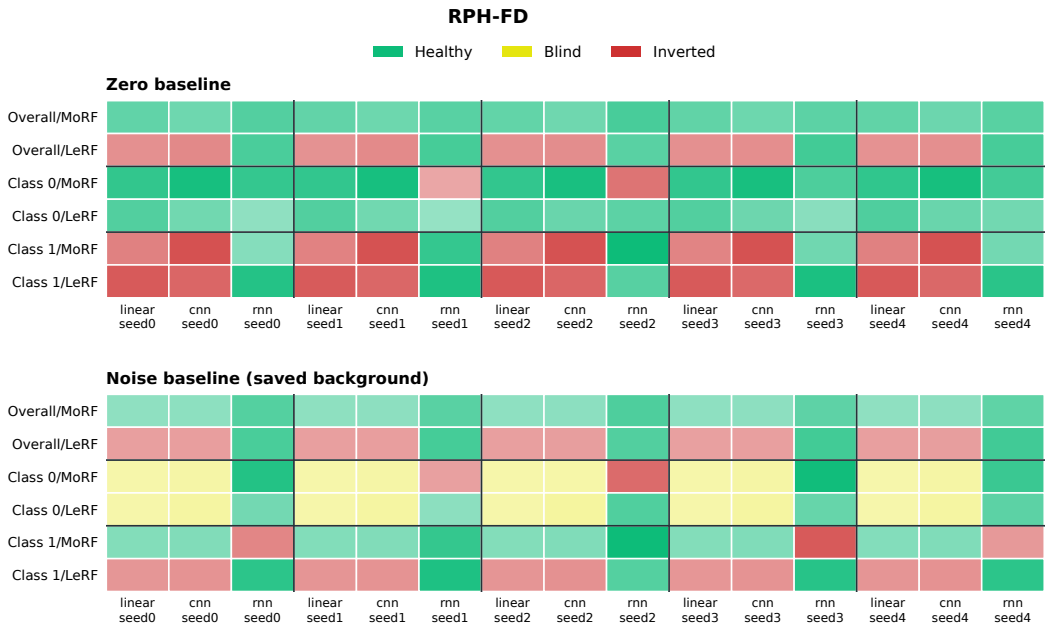

Figure 8: Faithfulness audit on **RPH-FD** across three architectures and five training seeds. **Top:** Zero baseline. **Bottom:** Saved noise backgrounds used as perturbation baselines. Columns represent model–seed combinations, and rows show overall and class-conditional evaluations under MoRF and LeRF. Green, yellow, and red indicate **healthy**, **blind**, and **inverted** GT–Random separation, respectively, with audit tolerance $\epsilon = 0.005$. Aggregate MoRF evaluations are healthy across all combinations with either baseline, despite class-conditional failures. With zero baselines, Linear and CNN are **healthy** for class 0 but inverted for class 1 in both directions. With background baselines, these models are blind for class 0 in both directions and inverted for class 1 under LeRF, while RNN outcomes vary across seeds and directions.

during evaluation, class 1 passes only one of the two evaluations under both MoRF and LeRF with either baseline. Noise-based NULL evaluation increases this count to two and passes for every class in both directions. However, zero-based NULL evaluation still yields only one **healthy** evaluation for classes 1, 2, and 3 under LeRF. CF references also leave failures: class 1 passes only one evaluation in each direction, while class 2 passes one under MoRF and none under LeRF. These results show that the evaluation configuration still requires class-conditional auditing when the attribution maps themselves are generated from NULL-augmented models.

Results on Other UCR Datasets with NULL Models. The seven binary UCR datasets show similar dependence on the evaluation configuration (Figure 14). On ECG200, zero/noise baselines without a NULL response yield two **healthy** aggregate evaluations, but class 0 passes neither evaluation under MoRF and only one under LeRF. Noise-based NULL evaluation increases the class 0 count to two in both directions, while preserving **healthy** overall and class 1 results. However, improvements are not universal. On GunPointMaleVersusFemale, noise-based NULL evaluation under LeRF yields two **healthy** evaluations both overall and within each class. CF references also leave class-conditional failures: on Wafer, both aggregate evaluations pass under LeRF, while only one passes for class 0. These results show that NULL-generated attribution maps can still receive misleading aggregate assessments. The choice of evaluation response and perturbation reference therefore requires class-conditional checking for each dataset and model.

E Sample and Representation Visualizations on Synthetic Datasets

We visualize the six synthetic datasets using representative samples from each class, input-space projections, linear-model logits, and t-SNE projections of CNN and RNN representations. The main

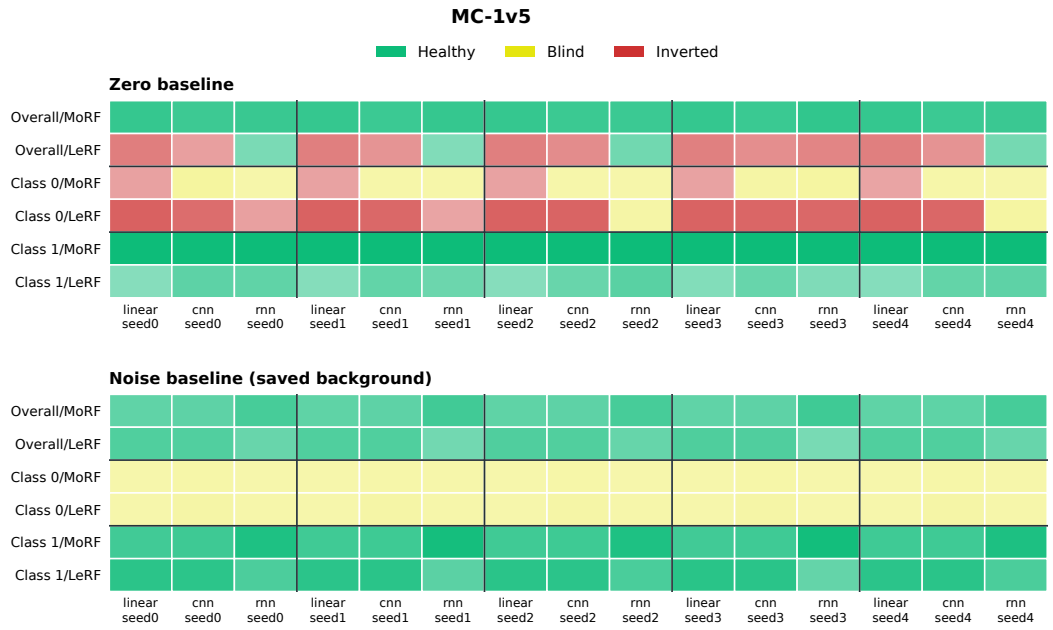

Figure 9: Faithfulness audit on **MC-1v5** across three architectures and five training seeds. **Top:** Zero baseline. **Bottom:** Saved noise backgrounds used as perturbation baselines. Columns represent model–seed combinations, and rows show overall and class-conditional evaluations under MoRF and LeRF. Green, yellow, and red indicate **healthy**, **blind**, and **inverted** GT–Random separation, respectively, with audit tolerance $\epsilon = 0.005$. With background baselines, all combinations are **healthy** overall and for class 1 but blind for class 0 in both directions. With zero baselines, class 0 is blind or inverted across all combinations, while aggregate MoRF and class 1 evaluations remain healthy.

paper presents FPH in Figure 1. The Appendix provides corresponding visualizations for RPH in Figure 17, PP in Figure 16, FPH-FD in Figure 18, RPH-FD in Figure 19, and MC-1v5 in Figure 20. These visualizations show a recurring pattern: background samples appear distinct from the real-class groups in representation space, yet the models predominantly assign them to a single real class.

F Sample and Representation Visualizations on SHAPEX Synthetic Datasets and MIT-ECG

We extend the FPH analysis in the main paper to the four SHAPEX synthetic datasets (MCCE, MCCH, MTCE, and MTCH) and MIT-ECG, using t-SNE projections of inputs and pooled features, alongside logit-space plots for the Linear model. On MCCE, noise baselines are not clearly separated from real inputs in the input-space projection and overlap with class 0 samples in the Linear model's logit space (Figure 21). In the CNN and RNN feature projections, however, they appear distinct from both classes despite receiving high-confident class 0 predictions with baseline input. This observation motivates testing whether an additional NULL output can use the frozen representation to recognize baseline inputs. MTCE shows similar visual separation in the CNN and RNN feature projections, but the mean probability assigned to class 1 differs across models: approximately 0.54 for Linear, 0.63 for CNN, and 0.96 for RNN (Figure 23). The lower mean probabilities in the first two models may help explain the relatively frequent **healthy** audit outcomes on MTCE in Figure 4, although baseline predictions alone do not determine the full perturbation response. MCCH follows the MCCE pattern, with the Linear model assigning a lower mean probability to class 0 than the CNN and RNN (Figure 22). On MTCH and MIT-ECG, the CNN and RNN likewise assign almost all noise baselines to class 0 with high confidence, whereas the Linear model's baseline predictions are approximately evenly split between the two classes (Figures 24 and 25).

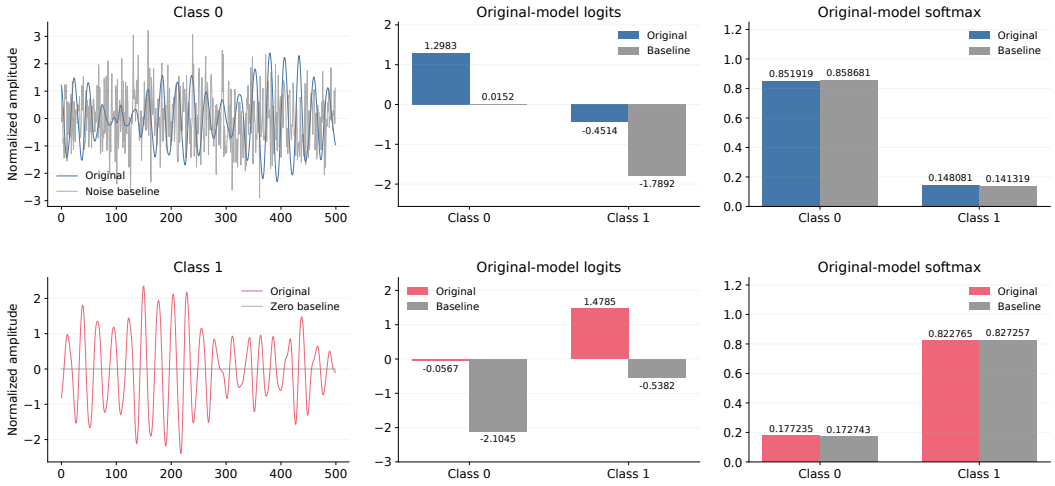

Figure 10: Input–baseline comparisons for Mantis on FordA. Rows show a class 0 input with a noise baseline and a class 1 input with a zero baseline. Columns show the signals, original-model logits, and corresponding softmax probabilities. Replacing each input with its baseline lowers both logits by similar amounts, leaving their difference nearly unchanged. Consequently, the probability assigned to the original class remains approximately 0.85 and 0.82, respectively. These examples illustrate how a baseline can retain high class confidence despite substantial changes in the input and logits.

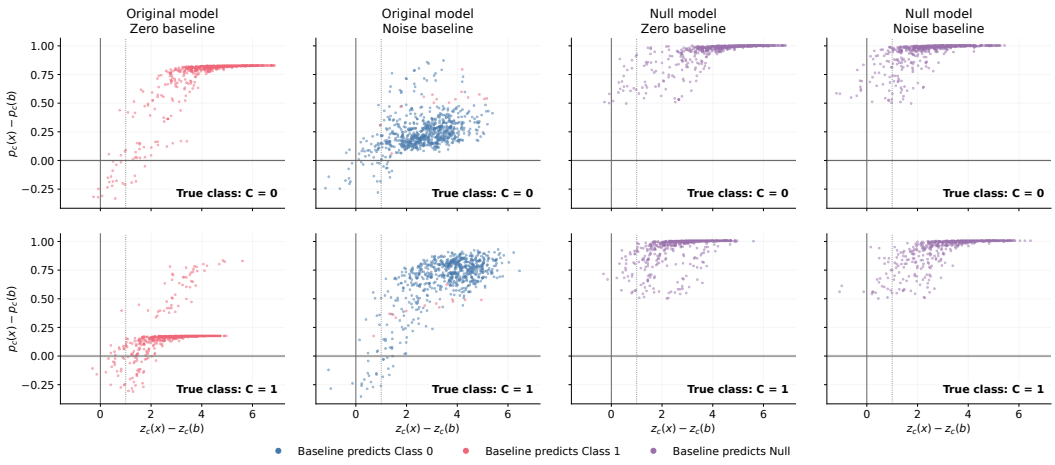

Figure 11: Input–baseline logit and probability differences on FordA before and after augmenting Mantis with a NULL response. Each point compares an input $\mathbf{x}$ with its baseline $\mathbf{b}$ using the evaluated class's logit difference, $z_c(\mathbf{x}) - z_c(\mathbf{b})$, and probability difference, $p_c(\mathbf{x}) - p_c(\mathbf{b})$. Rows separate the two true classes; columns compare zero and noise baselines under the original and NULL-augmented models. Colours indicate the baseline's predicted class. Positive logit differences can accompany small or negative probability differences in the original model. The NULL panels show baseline predictions assigned to NULL and larger positive probability differences, while the real-class logits remain unchanged.

G LIMITATIONS

Our study focuses on a basic failure mode of perturbation-based faithfulness evaluation for time-series explanations: an evaluator can distinguish informative from random attributions overall while failing within individual classes. We argue that, when informative reference attributions are available, this is a basic check that an evaluator should pass within its intended scope before undergoing

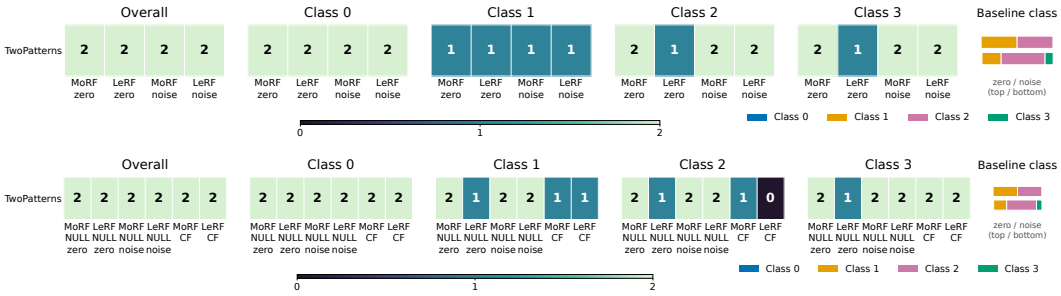

Figure 12: Faithfulness audit on the four-class **TwoPatterns** dataset. Each cell counts **healthy** evaluations overall or within an individual class. **Top:** Zero and noise baselines. **Bottom:** NULL responses with zero/noise baselines and counterfactual (CF) references. **Right:** Baseline prediction proportions for zero (upper bar) and noise (lower bar). Every configuration yields two **healthy** aggregate evaluations, but class-level counts vary. Noise-based NULL yields two **healthy** evaluations in every class under both MoRF and LeRF. CF yields only one **healthy** class 1 evaluation in each direction and none for class 2 under LeRF, showing that aggregate separation can still conceal class-conditional failures in this multiclass setting.

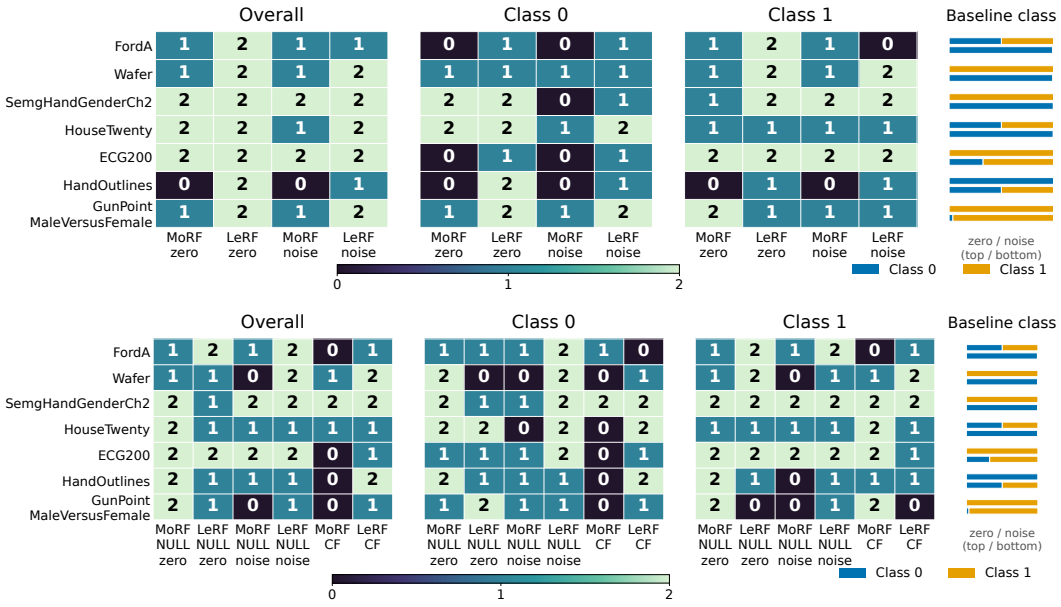

Figure 13: Faithfulness audit across seven binary UCR datasets. Each cell counts (over two setups Mantis and Hydra) **healthy** evaluations overall or within an individual class. **Top:** Zero and noise baselines. **Bottom:** NULL responses with zero/noise baselines and counterfactual (CF) references. **Right:** Baseline prediction proportions for zero (upper bar) and noise (lower bar). Aggregate separation can conceal class-conditional failures: on ECG200, all original configurations yield two **healthy** aggregate evaluations but only zero or one for class 0. NULL and CF can increase or decrease **healthy** counts, depending on the dataset, class, and perturbation direction, reinforcing the need to audit each configuration.

more comprehensive auditing. However, passing this audit does not establish that an evaluator is suitable for a particular application or that the explanations it favours are correct. Our audit therefore provides an initial diagnostic that requires further evaluation. This perspective is consistent with the view that addressing the opacity of machine-learning models requires a combination of analyses and auditing mechanisms (Burrell, 2016).

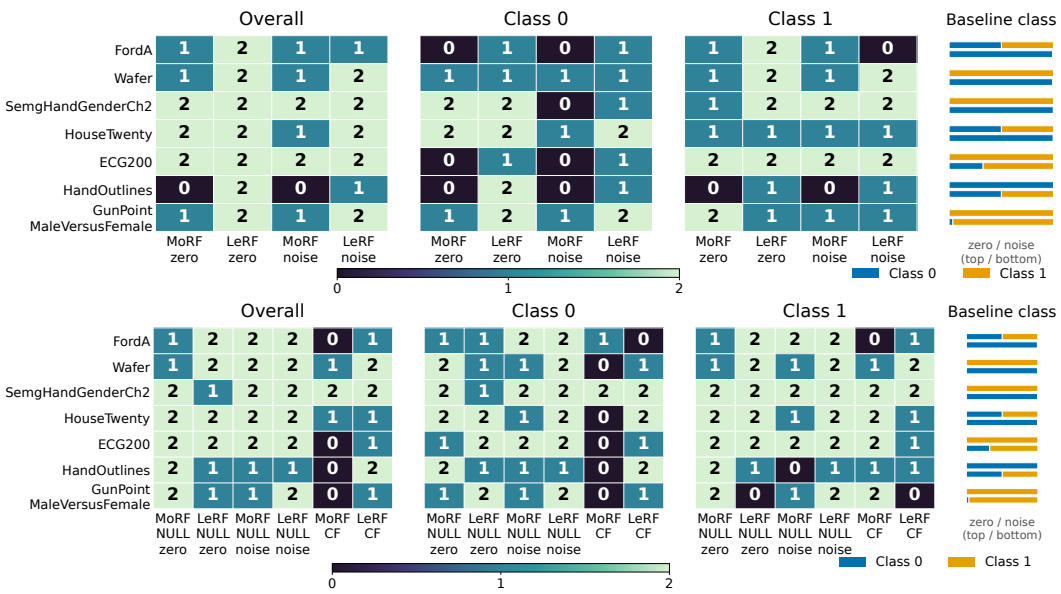

Figure 14: Confidence-based faithfulness audit of attributions generated from NULL-augmented models on seven binary UCR datasets. IG and feature-ablation rankings are compared with random rankings under MoRF and LeRF. Each cell counts **healthy** evaluations. **Top:** Evaluation with zero/noise baselines without a NULL response. **Bottom:** Evaluation with the same baseline types and a NULL response, alongside counterfactual (CF) references. **Right:** Baseline prediction proportions for zero (upper bar) and noise (lower bar) inputs.

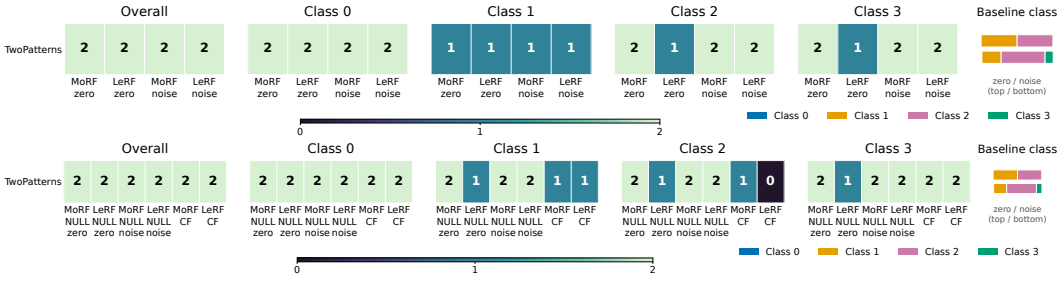

Figure 15: Confidence-based faithfulness audit of attributions generated from NULL-augmented models on the four-class TwoPatterns dataset. IG and feature-ablation rankings are compared with random rankings under MoRF and LeRF. Each cell counts **healthy** evaluations (0–2), overall or within each class. **Top:** Evaluation with zero/noise baselines without a NULL response. **Bottom:** Evaluation with the same baseline types and a NULL response, alongside counterfactual (CF) references. **Right:** Baseline prediction proportions for zero (upper bar) and noise (lower bar) inputs.

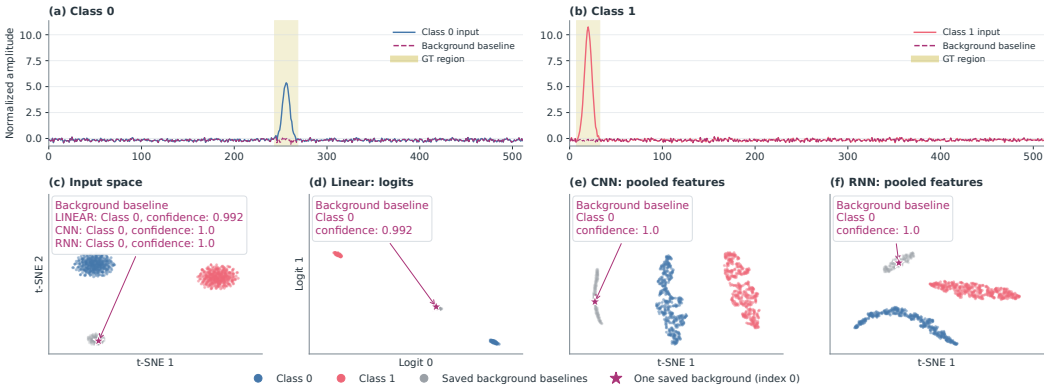

Figure 16: Input and representation-space analysis of background baselines on **PP** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one background baseline (index 0). Annotations summarize predictions across the background baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

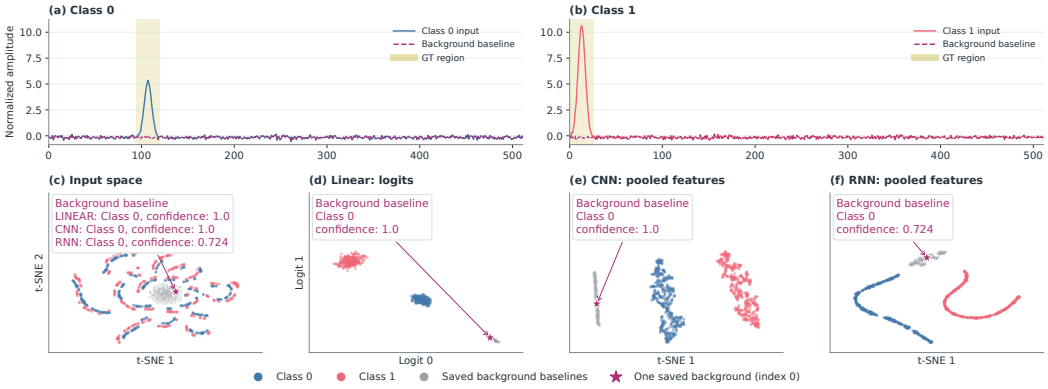

Figure 17: Input and representation-space analysis of background baselines on **RPH** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one background baseline (index 0). Annotations summarize predictions across the background baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

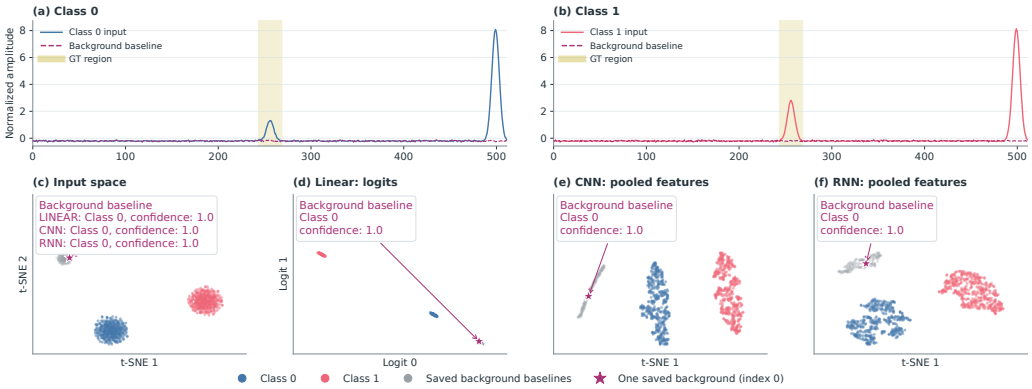

Figure 18: Input and representation-space analysis of background baselines on **FPH-FD** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one background baseline (index 0). Annotations summarize predictions across the background baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

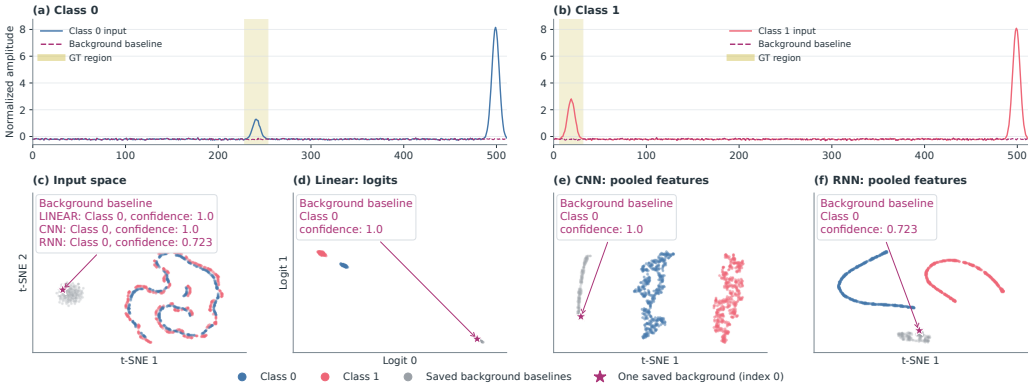

Figure 19: Input and representation-space analysis of background baselines on **RPH-FD** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one background baseline (index 0). Annotations summarize predictions across the background baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

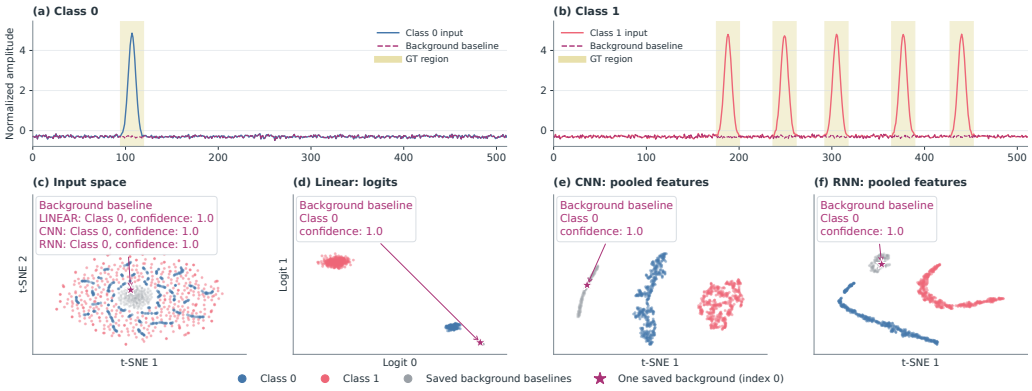

Figure 20: Input and representation-space analysis of background baselines on **MC-1v5** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one background baseline (index 0). Annotations summarize predictions across the background baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

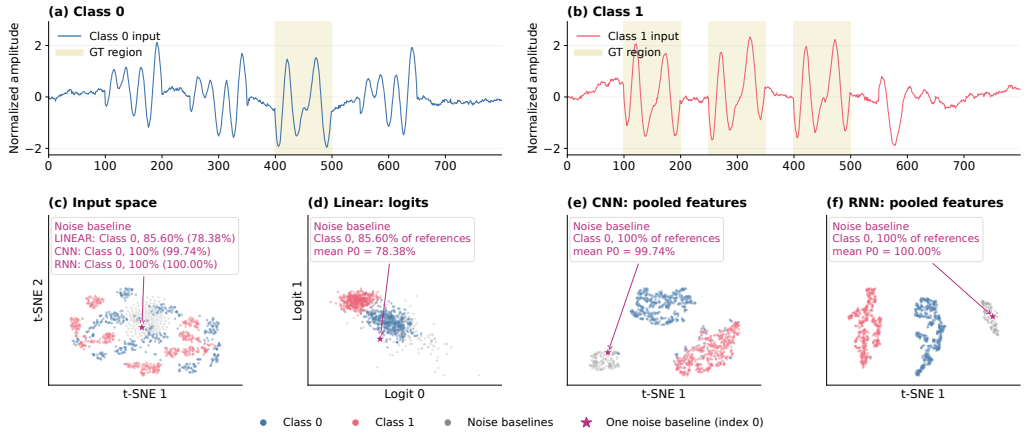

Figure 21: Input and representation-space analysis of noise baselines on **MCCE** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one noise baseline (index 0). Annotations summarize predictions across the noise baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all noise baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

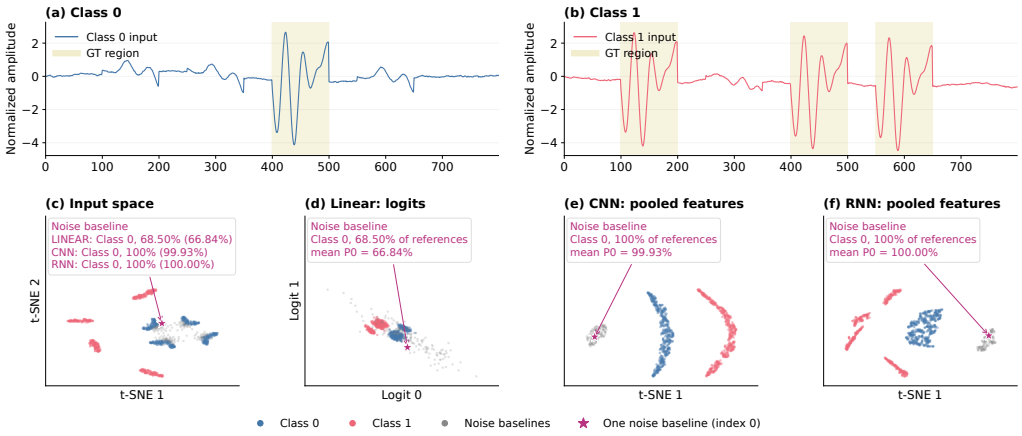

Figure 22: Input and representation-space analysis of noise baselines on **MCCH** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one noise baseline (index 0). Annotations summarize predictions across the noise baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign all noise baselines to class 0, although these baselines appear visually distinct from both real classes in the feature projections.

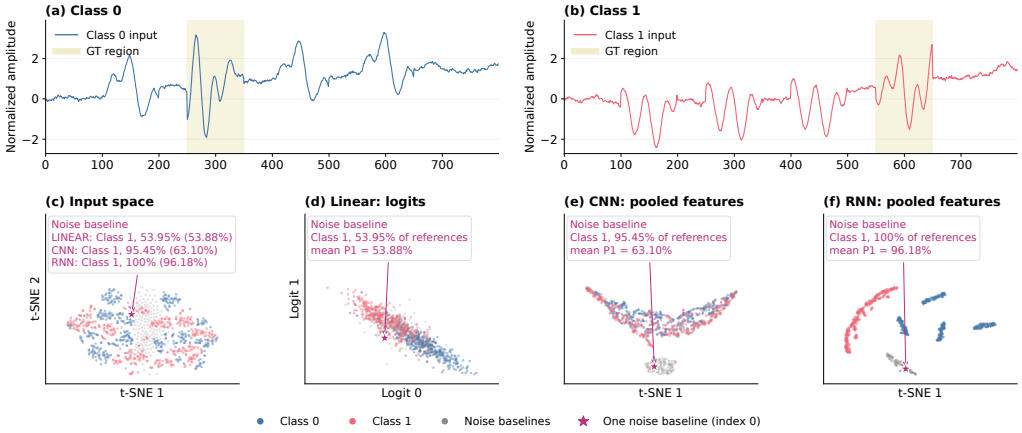

Figure 23: Input and representation-space analysis of noise baselines on **MTCE** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one noise baseline (index 0). Annotations summarize predictions across the noise baselines, reporting the proportion assigned to class 1 and the mean class 1 probability (shown in parentheses in panel (c)). Noise baselines appear visually distinct from real inputs in the CNN and RNN feature projections, yet these models assign most or all baselines to class 1. Linear predictions are more evenly divided between the two classes.

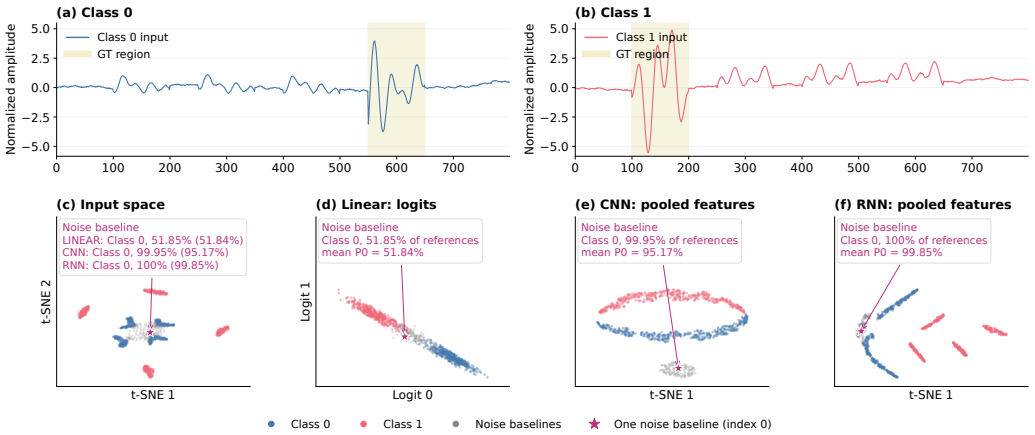

Figure 24: Input and representation-space analysis of noise baselines on **MTCH** (seed 0). **(a–b)** Representative class 0 and class 1 inputs with ground-truth (GT) regions. **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one noise baseline (index 0). Annotations summarize predictions across the noise baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). The CNN and RNN assign nearly all or all noise baselines to class 0, whereas Linear predictions are almost evenly split between classes. Noise baselines form a visually distinct group in the CNN projection but appear near class 0 samples in the RNN projection.

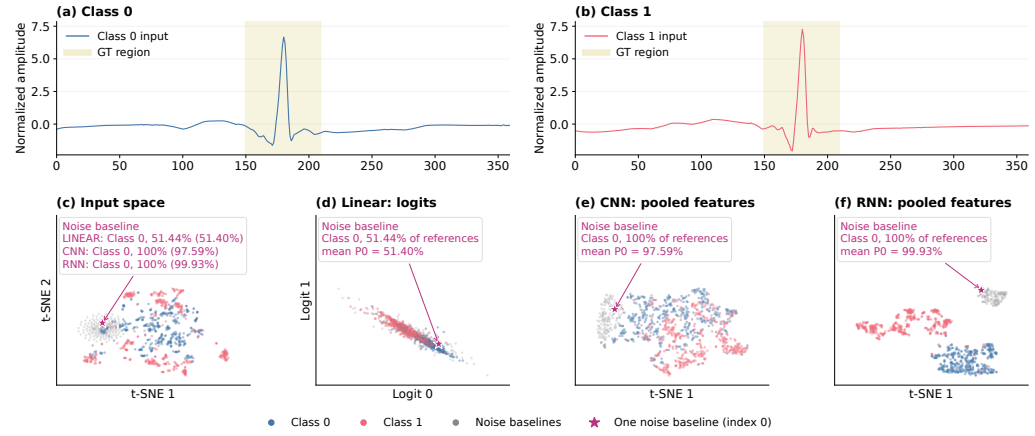

Figure 25: Input and representation-space analysis of noise baselines on **MIT-ECG** (seed 0). **(a–b)** Representative class 0 and class 1 ECG inputs with highlighted reference explanation regions (GT). **(c)** t-SNE projection of inputs. **(d)** Linear-model logit space. **(e–f)** t-SNE projections of pooled CNN and RNN features, respectively. Stars highlight one noise baseline (index 0). Annotations summarize predictions across the noise baselines, reporting the proportion assigned to class 0 and the mean class 0 probability (shown in parentheses in panel (c)). Noise baselines appear visually distinct from both real classes in the RNN projection, yet all are assigned to class 0. The CNN also assigns all baselines to class 0, whereas Linear predictions are almost evenly split between classes.