From Post-Hoc to Ante-Hoc: Consistently Explainable Semi-Supervised Time Series Classification

Viet-Hung Tran[1], **Zichi Zhang**[1], **Ngoc Phu Doan**[1], **Xuan Hoang Nguyen**[1], **Phi Hung Nguyen**[1], **Yimeng An**[2], **Peixin Li**[1], **Huynh Thi Khanh Chi**[1], **Hans Vandierendonck**[1], **Ira Assent**[3], **Thai Son Mai**[1]

Abstract

Semi-Supervised Time Series Classification (SS-TSC) with Deep Neural Networks (DNNs) achieves strong accuracy by leveraging unlabeled data, yet the resulting models remain black boxes that offer no insight into which features drive each prediction. Existing post-hoc explanation methods for TSC can identify important features but provide no guarantee that the model genuinely relies on them, while ante-hoc approaches impose architectural constraints that limit flexibility in existing model usage. Neither setting explores the semi-supervised regime, where limited labels can affect explainability. We propose **Sensory Gating**, the first SS-TSC framework that simultaneously bridges post-hoc attribution and ante-hoc explanation through a four-stage pipeline: (i) a base classifier is trained with our proposed Forecasting Joint-Embedding Predictive Architecture (F-JEPA) auxiliary objective; (ii) post-hoc attribution maps are converted into per-instance binary masks via greedy selection of sufficient features; (iii) a lightweight **Sensory Gate** amortizes these masks through Binary Cross Entropy (BCE) supervision with data-driven threshold calibration; and (iv) a knowledge distillation stage provides an explainable SS-TSC student model that operates on gated (masked) input, while consistently reproducing the predictions of the original black-box teacher (F-JEPA). We validate our framework on eight datasets across four label ratios with five seeds per configuration. F-JEPA achieves better performance than recent state-of-the-art (SOTA) semi-supervised TSC methods, while the sensory gate reduces the fraction of features the classifier requires to 20–45% with competitive accuracy. On MIT-BIH, where ground-truth saliency annotations are available, the gate's selected features align with domain-relevant regions while classification performance remains competitive with the ungated baseline at 10–20% retention ratio.

1 Introduction

Time series classification (TSC) is a fundamental task in various domains such as healthcare [38] and industrial monitoring [28]. Deep neural networks (DNNs) achieve strong performance on TSC by capturing complex temporal patterns, but require large amounts of labeled data for training. In many settings, acquiring labels is expensive: ECG and EEG annotations require trained clinicians, and fault labels in manufacturing demand domain expertise and post-incident investigation. Semi-supervised learning addresses this constraint by leveraging abundant unlabeled data alongside limited labels, and SOTA methods such as f-VAT [35], SemiTime [11], and TS-TCC [10] achieve substantial accuracy gains over supervised baselines. However, these methods optimize exclusively for predictive performance, while the DNNs they train remain black boxes that offer no insight into which temporal features drove each prediction. This is not merely a practical inconvenience: the EU AI Act requires that high-risk AI systems provide information sufficient to explain their outputs to deployers [45]. In

[1]Queen's University Belfast [2]North China University of Water Resources and Electric Power [3]Aarhus University

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

clinical settings, for instance, the disconnect between a model's internal processing and a clinician's diagnostic reasoning hinders legally accountable decision-making [47]. Nonetheless, eXplainable AI (XAI) methods for DNN-based TSC, which aim to provide insights into these DNNs' behaviors, have emerged, such as [18, 48, 49, 43, 25, 37, 46, 49, 8, 36]; however, all of them focus on the fully supervised setting, whereas the semi-supervised setting remains unexplored to our knowledge.

Generally, explainability methods for TSC fall into two paradigms: post-hoc and ante-hoc. Post-hoc methods generate attribution maps after training, assigning importance scores to input features without modifying the studied models [18, 48, 49, 43, 25]. Ante-hoc methods such as [22, 51] design specific DNN architectures so that explanations emerge during inference. Though post-hoc explanation is common and highly flexible (i.e., it can be employed for any DNN TSC model [18, 43, 25, 46, 8, 36]), it shares a fundamental limitation with human introspection. Research in psychology [34, 54] shows that people rely on implicit causal theories rather than direct access to their own cognitive processes, and that acting on such explanations leads to worse decisions. The parallel to post-hoc XAI is direct: an attribution map identifies features that a method deems important, but without verification, there is no guarantee that the model's prediction actually depends on them. Currently, post-hoc explanations can be partially verified through faithfulness metrics evaluated at multiple masking ratios [36], since unlike the human brain, DNN weights are fully accessible for inspection. However, this verification protocol is limited in three ways: it is computationally expensive, requiring repeated forward passes at each ratio [36, 25]; it assumes a uniform feature budget across all instances, ignoring that different predictions may depend on different numbers of features [36, 25]; and the explanation never enters the actual inference pipeline, so faithfulness is assessed by proxy rather than by the model operating on its own explanation [39]. Ante-hoc methods appear to resolve the post-hoc verification gap by embedding explanations directly into their inference mechanisms [22, 51]. However, ante-hoc explanation operates at the architectural level: interpretability is enforced by designing neural components that constrain how the model processes input. This has no analogue in human cognition, where one cannot redesign individual neurons or manipulate internal representations to conform to a given explanation. Moreover, these methods exchange explainability with the cost of architectural flexibility, i.e., they cannot be used to explain other TSC DNNs except their owns.

Contributions. In this paper, we introduce Sensory Gating, the first post-hoc explanation framework for SS-TSC that has explicit verification ability of ante-hoc explanation, thus offering a new practical path for TSC XAI: preserving unconstrained model design while ensuring that explanations faithfully reflect the model's decision process, while overcoming above-mentioned limitations of traditional faithfulness protocols. Sensory Gating is not model-specific. With a gradient-free attribution method such as Feature Ablation, it does not need input gradients from the classifier, so it also works with non-differentiable classifiers such as HYDRA. The framework is complicated and need computational cost in training stages, but explanation inference is fast and faithful at deployment. Our contributions are detailed as follows.

First, we introduce a new design perspective for explainable time series classification grounded in a natural selection principle: models must satisfy fitness criteria along both utility and explainability axes. Inspired by sensory gating in the thalamus [27, 26], where irrelevant stimuli are filtered out (gating) while allowing important information flows to the cortex to prevent overstimulation, we develop a lightweight module called **sensory gate** that bridges post-hoc attribution and ante-hoc explanation. The gate amortizes per-instance post-hoc explanation attribution maps such as TIMING [18] into **binary masks** through a three-step process: greedy selection of sufficient features, BCE-supervised gate training, and data-driven threshold calibration producing faithful explanations without sparsity losses, Gumbel-Softmax, or straight-through estimators. **Second**, we propose a built-in refinement and verification stage like ante-hoc explanation based on knowledge distillation: a student encoder trained on gated (masked) input must reproduce the original black box teacher's predictions on all data, closing the post-hoc→ante-hoc→verification loop and providing a self-consistent faithfulness measure. **Third**, we show that explanation difficulty varies per instance. The greedy mask selection discovers that different samples require different fractions of input features to preserve the model's prediction. This per-instance variable sparsity contrasts with the fixed masking ratios assumed by prior evaluation protocols and provides an additional diagnostic signal for prediction reliability. **Fourth**, we introduce Forecasting-JEPA (F-JEPA), an auxiliary temporal prediction objective that demonstrates predictive tasks, inherently aligned with sequential data, can serve as effective semi-supervised regularization for SS-TSC, producing encoders competitive with

dedicated SSL methods such as f-VAT [35] while providing the representation quality needed for faithful downstream explanation. Though we employ F-JEPA as the central SS-TSC method for our Sensory Gating framework, it worths to note that Sensory Gating can be used with any TSC methods. **Our main contribution is Sensory Gating, an explanation framework for SS-TSC**. It works by selecting input features, and the model delivered to practitioners is the gated student. F-JEPA is the base classifier that supports the framework by improving generalization when labels are scarce.

We validate these contributions through comprehensive experiments on eight real-world datasets across four label ratios with five seeds per configuration. F-JEPA achieves better performance compared to SOTA semi-supervised TSC methods, while the sensory gate reduces the fraction of features the classifier requires to 20–45% with competitive accuracy. On MIT-BIH, where ground-truth saliency annotations are available, the gate's selected features align with domain-relevant regions while classification performance remains competitive with the ungated baseline at 10–20% kept ratio.

2 Preliminary

2.1 Problem Settings and Notations

We consider the semi-supervised time series classification (TSC) setting. Let $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ denote the full dataset, where $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ is a small labeled set with $\mathbf{x}_i \in \mathbb{R}^{T\times C}$ and $y_i \in \{1, \ldots, K\}$, K is the number of classes, and $\mathcal{D}_u = \{\mathbf{x}_j\}_{j=1}^{n_u}$ is a large unlabeled set, where $n_u \gg n_l$. The label ratio $\alpha = n_l/|\mathcal{D}|$ controls the degree of supervision. A classifier $F_\theta : \mathbb{R}^{T\times C} \to \mathbb{R}^K$ maps an input time series $\mathbf{x}$ to class logits $F_\theta(\mathbf{x})$, producing predicted class $\hat{y} = \arg\max_k [F_\theta(\mathbf{x})]_k$. Training combines a supervised cross-entropy loss $\mathcal{L}_{\text{CE}}$ on $\mathcal{D}_l$ with an unsupervised auxiliary loss $\mathcal{L}_{\text{aux}}$ on $\mathcal{D}$:

$$\mathcal{L}(\mathcal{D}; \theta) = \mathcal{L}_{\text{CE}}(\mathcal{D}_l; \theta) + \lambda \cdot \mathcal{L}_{\text{aux}}(\mathcal{D}; \theta), \tag{1}$$

where the form of $\mathcal{L}_{\text{aux}}$ varies across methods, for example, consistency regularization [35], contrastive objectives [10], or predictive losses.

Given a trained classifier F_θ, the target of an attribution method is to produce a per-instance map $\mathbf{a} \in \mathbb{R}^{T\times C}$, where each entry $a_{t,c}$ quantifies the contribution of input feature $x_{t,c}$ to the prediction $F_\theta(\mathbf{x})$. From the attribution map, a binary mask $\mathbf{m} \in \{0, 1\}^{T\times C}$ can be derived to identify a sufficient subset of features. Given a mask $\mathbf{m}$ and a perturbation baseline $\mathbf{b} \in \mathbb{R}^{T\times C}$, the perturbed input is defined as

$$\tilde{\mathbf{x}} = \mathbf{x} \odot \mathbf{m} + \mathbf{b} \odot (\mathbf{1} - \mathbf{m}), \tag{2}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{b}$ replaces masked positions with an uninformative reference value; common choices include zeros, Gaussian noise, or the per-channel training set mean. The mask should satisfy two desiderata: *sufficiency*—the prediction on the perturbed input preserves both the predicted class and the confidence of the original prediction, i.e., $\arg\max_k[\sigma((\tilde{\mathbf{x}}))]_k = \hat{y}$ and $\sigma(F_\theta(\tilde{\mathbf{x}}))_{\hat{y}} \approx \sigma(F_\theta(\mathbf{x}))_{\hat{y}}$, where σ denotes the softmax function; and *sparsity*—the fraction of selected features $\|\mathbf{m}\|_0/(T \cdot C)$ should be as small as possible while maintaining sufficiency (where $||\cdot||_0$ indicates L_0 norm operator).

2.2 TIMING: Temporality-Aware Integrated Gradients

Integrated Gradients (IG) [46] computes per-feature attributions by integrating gradients along a straight-line path from a baseline $\mathbf{x}'$ to the input $\mathbf{x}$:

$$\text{IG}_{t,d}(\mathbf{x}) = (x_{t,d} - x'_{t,d}) \int_0^1 \frac{\partial F_{\hat{y}}(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial x_{t,d}}\, d\alpha. \tag{3}$$

When the baseline scales all features uniformly, intermediate points may fall in out-of-distribution regions, producing unreliable gradients, and the path fails to capture the effect of disrupting temporal relationships [18]. TIMING [18] addresses both limitations by partially retaining certain features at their original values while scaling the rest. Given a binary mask $M \in \{0, 1\}^{T\times D}$ indicating which features are scaled ($M_{t,d} = 1$) versus retained ($M_{t,d} = 0$), MaskingIG is defined as

$$\text{MaskingIG}_{t,d}(\mathbf{x}, M) = x_{t,d} M_{t,d} \times \int_{\alpha=0}^1 \frac{\partial F_{\hat{y}}(\alpha(M \odot \mathbf{x}) + (\mathbf{1} - M) \odot \mathbf{x})}{\partial x_{t,d}}\, d\alpha. \tag{4}$$

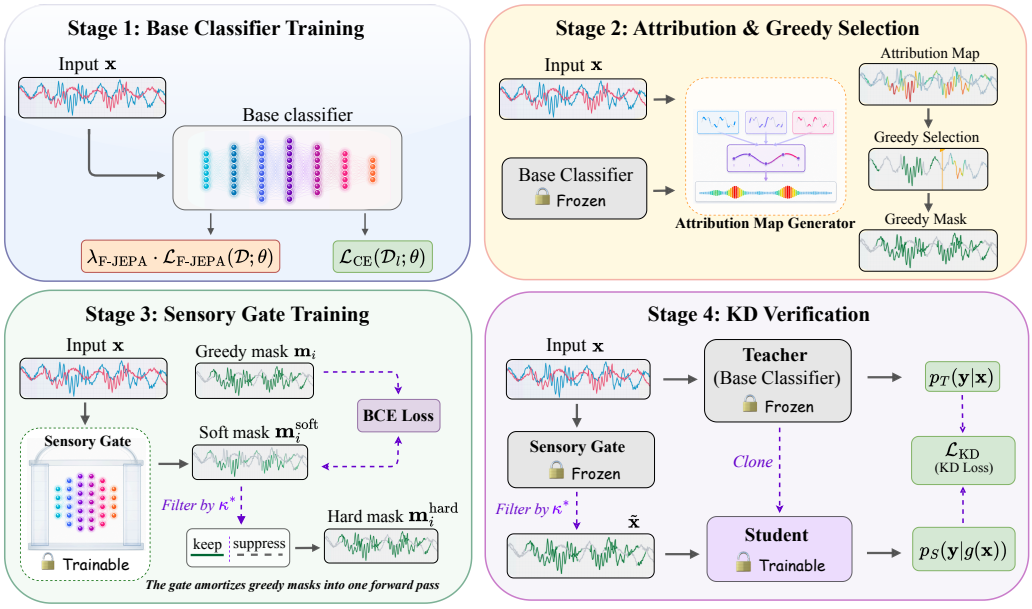

Figure 1: **Overview of the Sensory Gating framework. Stage 1:** The base classifier (encoder + classification head) is trained end-to-end with cross-entropy on labeled data $\mathcal{D}_l$, augmented by an F-JEPA auxiliary objective on all data $\mathcal{D}$. **Stage 2:** TIMING (or any other post-hoc methods) computes per-timestep attributions; greedy selection converts them into per-instance binary masks. **Stage 3:** A lightweight sensory gate is trained via BCE against the greedy masks, with threshold κ^* calibrated on the validation set. **Stage 4:** The frozen teacher receives original input $\mathbf{x}$; the student (cloned from the teacher) is fine-tuned behind the frozen gate on gated input $\tilde{\mathbf{x}} = \mathbf{m}^{\text{hard}} \odot \mathbf{x} + (1 - \mathbf{m}^{\text{hard}}) \odot \bar{\mathbf{x}}$, minimizing $\mathcal{L}_{\text{KD}}$. The resulting student achieves classification performance competitive with the Stage 1 teacher while operating only on the subset of features selected by the sensory gate.

Taking the expectation of MaskingIG over independent Bernoulli masks yields RandIG, but independent per-point masking may not align with meaningful temporal structures. TIMING therefore adopts segment-based masking: a mask generator $\mathcal{G}(n, s_{\min}, s_{\max})$ selects n contiguous segments of lengths drawn from $[s_{\min}, s_{\max}]$ and retains the selected positions, yielding

$$\text{TIMING}_{t,d}(\mathbf{x};\, n, s_{\min}, s_{\max}) = \mathbb{E}_{M \sim \mathcal{G}(n,\, s_{\min},\, s_{\max})} \left[\text{MaskingIG}_{t,d}(\mathbf{x}, M) \;\middle|\; M_{t,d} = 1 \right]. \quad (5)$$

The expectation over masks can be absorbed into the integration path [18], enabling efficient single-sample approximation. TIMING retains the sensitivity and implementation invariance properties of IG, though it does not satisfy the completeness axiom due to the use of multiple stochastic baselines.

3 The Sensory Gating Framework

Our Sensory Gating framework, illustrated in Figure 1, converts post-hoc attributions into an ante-hoc binary gate and verifies the conversion through knowledge distillation, proceeding in four stages. **Stage 1** trains the base classifier F_θ with cross-entropy on $\mathcal{D}_l$ and our proposed F-JEPA auxiliary objective on $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ (see Section 3.1). **Stage 2** computes per-instance post-hoc explanation attribution maps $\mathbf{a}$ via TIMING [18] (or any other selected attribution methods such as TimeX [36]) on the frozen classifier, scoring each time point across all channels, then applies greedy selection to produce a per-instance binary mask $\mathbf{m}$ (see Section 3.2). **Stage 3** trains a small sensory gate to predict $\mathbf{m}$ via BCE, with threshold κ^* calibrated on the validation set (see Section 3.3). Hence, binary masks arise by construction: greedy selection followed by BCE supervision. **Stage 4** clones the frozen classifier into a student and fine-tunes it on gated input $g(\mathbf{x})$, distilling from the teacher's ungated predictions on $\mathcal{D}_l \cup \mathcal{D}_u$, thus creating a new equivalent classifier with ante-hoc explanation ability via the sensory gate model provided in Stage 3 (see Section 3.4). The framework draws a

structural analogy from the thalamic reticular nucleus, which selects relevant sensory input before cortical processing [27]. Each stage is detailed in the following subsections.

3.1 Base Classifier Training with F-JEPA

Given a classifier F_θ with an encoder backbone f_θ, our base semi-supervised classifier training minimizes the join objective function of the conventional cross-entropy loss $\mathcal{L}_{\text{CE}}$ and our proposed F-JEPA loss $\mathcal{L}_{\text{F-JEPA}}$ as

$$\mathcal{L}(\mathcal{D};\theta) \;=\; \mathcal{L}_{\text{CE}}(\mathcal{D}_l;\,\theta) \;+\; \lambda_{\text{F-JEPA}}\,\mathcal{L}_{\text{F-JEPA}}(\mathcal{D};\,\theta,\xi), \tag{6}$$

where $\lambda_{\text{F-JEPA}}$ is a regularization parameter. The cross-entropy term $\mathcal{L}_{\text{CE}}$ operates on labeled samples only. For our proposed Forecasting Joint-Embedding Predictive Architecture (F-JEPA) loss term, the temporal axis of each input $\mathbf{x}$ is split into a context window (past) and a target window (future). The context window is encoded by f_θ, pooled, and mapped through a linear predictor h to produce a predicted embedding $\hat{\mathbf{z}}$. The target window is encoded by an exponential moving average (EMA) encoder f_ξ, pooled to produce the target embedding $\mathbf{z}$. Both $\hat{\mathbf{z}}$ and $\mathbf{z}$ are ℓ_2-normalized, and the loss is their cosine distance, $1 - \hat{\mathbf{z}}^\top \mathbf{z}$. While the original JEPA [2] optimizes MSE across token-level representations produced by a Transformer, our proposed F-JEPA instead formulates the objective as temporal forecasting, where each window is reduced to a single pooled vector. In this setting, MSE is susceptible to representation collapse, and cosine distance provides the necessary normalization constraint. The EMA encoder is updated as $\xi \leftarrow \tau \cdot \xi + (1-\tau)\cdot\theta$ after each step as BYOL [13], where τ is the decay rate. Please refer to Appendix A for detail information.

Why F-JEPA? Temporal prediction is a natural self-supervised objective for sequential data: it requires the encoder to capture temporal dynamics without additional labels. Our framework is encoder-agnostic by design as described above. Moreover, in Computer Vision and NLP, token-based architectures dominate, and self-supervised objectives can operate on individual token representations. Many TSC encoders, such as TCN [35], process the full sequence without tokenization, producing only sequence-level features; the auxiliary objective must accommodate this difference rather than constrain encoder choice. Our proposed F-JEPA demonstrates that a forecasting-based predictive objective suffices for semi-supervised TSC as shown in Section 4.2.

3.2 Attribution and Greedy Mask Selection

Given the frozen classifier from Stage 1, we compute a signed attribution map $\mathbf{a}_i \in \mathbb{R}^{T\times C}$ for each sample $\mathbf{x}_i \in \mathcal{D}$ via TIMING (see Section 2.2). The baseline is set to the training-set mean $\bar{\mathbf{x}}$, which matches the perturbation applied by the gate in Stage 3 in Section 3.3 below: masked time points are replaced with $\bar{\mathbf{x}}$ in both stages, ensuring that the attribution scores features under the same perturbation semantics used during gating. Attribution is computed once and stored offline; it does not enter the training loop of any subsequent stage, thus reducing computational overhead.

Having all attribution maps, for each sample $\mathbf{x}_i$, we find the smallest binary mask $\mathbf{m}_i \in \{0,1\}^{T\times C}$ such that the classifier's prediction on the gated input $\tilde{\mathbf{x}}_i = \mathbf{m}_i \odot \mathbf{x}_i + (1-\mathbf{m}_i)\odot\bar{\mathbf{x}}$ matches its prediction on the full input $\mathbf{x}_i$. To solve this optimization problem, we propose a greedy strategy: we sort the entries of $\mathbf{a}_i$ in descending order of their importance and grow $\mathbf{m}_i$ incrementally, adding the group of each 5% highest-attributed time-channel features at each step to avoid huge computational cost. The selection process stops when the prediction on $\tilde{\mathbf{x}}_i$ satisfies a sufficiency criterion: the predicted class matches $\arg\max F_\theta(\mathbf{x}_i)$ and the softmax confidence equal or large than that with full features. The output is a set of pairs $\{(\mathbf{x}_i, \mathbf{m}_i)\}$with per-instance sparsity $\|\mathbf{m}_i\|_0/(T\cdot C)$.

A key property of our greedy selection is that different instances converge at different kept ratios. Intuitively, samples whose predictions rely on a few localized features are explained with sparse masks, while samples near decision boundaries or with diffuse attribution may require denser masks before sufficiency is met. However, this per-instance variability is ignored in all existing works by imposing a uniform budge across instances [36, 25, 48] (see Section 4.3 for an analysis).

3.3 Sensory Gate Training

Our sensory gate g_ω is a lightweight temporal convolutional network parameterized by ω that maps each input $\mathbf{x}_i$ to a soft mask $\mathbf{m}_i^{\text{soft}} = g_\omega(\mathbf{x}_i) \in [0,1]^{T\times C}$ via a sigmoid output layer. Its parameter count is small relative to the classifier F_θ, and it introduces negligible overhead at inference.

The gate is trained by minimizing binary cross-entropy (BCE) against the greedy masks from Stage 2:

$$\min_{\omega} \mathcal{L}_{\text{gate}} = \text{BCE}(g_\omega(\mathbf{x}), \mathbf{m}), \tag{7}$$

with early stopping on validation BCE. The gate learns to reproduce the per-instance selection that greedy search discovered, not to optimize a sparsity-accuracy trade-off directly. This separation is intentional: the gate amortizes an already-solved combinatorial problem into a single forward pass, rather than jointly learning what to mask and how sparse to be.

At inference, soft outputs are binarized as $\mathbf{m}_i^{\text{hard}} = \mathbf{1}[g_\omega(\mathbf{x}_i) > \kappa^*]$. The threshold κ^* is calibrated by sweeping from high to low ($0.9 \to 0.1$) on the validation set and selecting the largest κ whose average kept ratio $\frac{1}{N}\sum_i \|\mathbf{m}_i^{\text{hard}}\|_0/(T \cdot C)$ does not exceed a target sparsity budget $s_{\max}$, subject to maximizing validation accuracy. The result is a deterministic binary mask produced in a single forward pass, with no stochastic sampling at test time.

This design avoids the relaxation machinery common in learnable-mask approaches. When a gate is trained end-to-end with the classifier under a classification loss, uniform soft masks $g_\omega(\mathbf{x}) \approx \mathbf{1}$ are a trivial optimum: passing all features through minimizes classification error. Methods such as TimeX [36] counter this with a sparsity penalty on $\|\mathbf{m}\|_1$ and Gumbel-Softmax or straight-through estimators to maintain gradient flow through discrete sampling. Our approach sidesteps this tension entirely: greedy selection produces binary masks without gradient-based optimization; BCE supervision transfers them to the gate without a sparsity hyperparameter; and κ^* makes the mask become binary while controlling the sparsity-utility trade-off post hoc. No component of the gate training requires differentiating through a discrete decision.

3.4 Knowledge Distillation Refinement and Verification

The teacher is the frozen classifier F_θ from Stage 1, receiving ungated input $\mathbf{x}$. The student F_ϕ is initialized as a clone of F_θ and fine-tuned on gated input:

$$\tilde{\mathbf{x}} = \mathbf{m}^{\text{hard}} \odot \mathbf{x} + (1 - \mathbf{m}^{\text{hard}}) \odot \bar{\mathbf{x}}, \tag{8}$$

where $\mathbf{m}^{\text{hard}} = \mathbf{1}[g_\omega(\mathbf{x}) > \kappa^*]$ is the binarized output of the frozen gate from Stage 3 and $\bar{\mathbf{x}}$ is the training-set mean. The student is optimized by minimizing

$$\min_{\phi} \mathcal{L}_{\text{KD}} = \lambda_{\text{kd}}\, \rho^2\, \text{KL}\Big(\sigma(\mathbf{y}_T/\rho) \,\Big\|\, \sigma(\mathbf{y}_S/\rho)\Big) + \lambda_{\text{hard}}\, \mathcal{L}_{\text{CE}}(\mathcal{D}_l;\, \phi), \tag{9}$$

where $\mathbf{y}_T = F_\theta(\mathbf{x})$ and $\mathbf{y}_S = F_\phi(\tilde{\mathbf{x}})$ are teacher and student logits respectively, $\sigma(\cdot/\rho)$ is the temperature-scaled softmax with temperature ρ, and ρ^2 rescales the KL gradient magnitude. The KL term is computed on all data $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, since it requires no labels. The hard cross-entropy term operates on labeled samples only and is opt-in ($\lambda_{\text{hard}} = 0$ by default). Knowledge distillation is the natural choice of supervision for this stage, since both the attribution maps and the greedy masks (Stage 2) are constructed to preserve the original classifier's predictions, not hard labels. The teacher's output distribution is therefore the correct target for the student, and distillation on the full dataset $\mathcal{D}$ exploits gated inputs from all samples, avoiding the data scarcity that would result from restricting fine-tuning to $\mathcal{D}_l$ alone.

If the student trained on gated input matches the teacher trained on full input, the gate has selected a sufficient subset of features for the classifier's predictions. Teacher-student agreement thus serves as a built-in faithfulness measure, replacing external metrics that require separate perturbation protocols. This stage closes the full cycle: a chosen post-hoc explanation method such as TIMING provides post-hoc attributions (Stage 2), the gate amortizes them into an ante-hoc mask (Stage 3), and knowledge distillation trains a student on the gated input to reproduce the teacher's predictions, helping model achieve utility comparable to the original black-box model and with ante-hoc explainability (Stage 4).

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our proposed method Sensory Gating on nine datasets spanning over four domains. Five are multivariate TSC benchmarks from the UCR/UEA archive [9, 4]: CricketX,

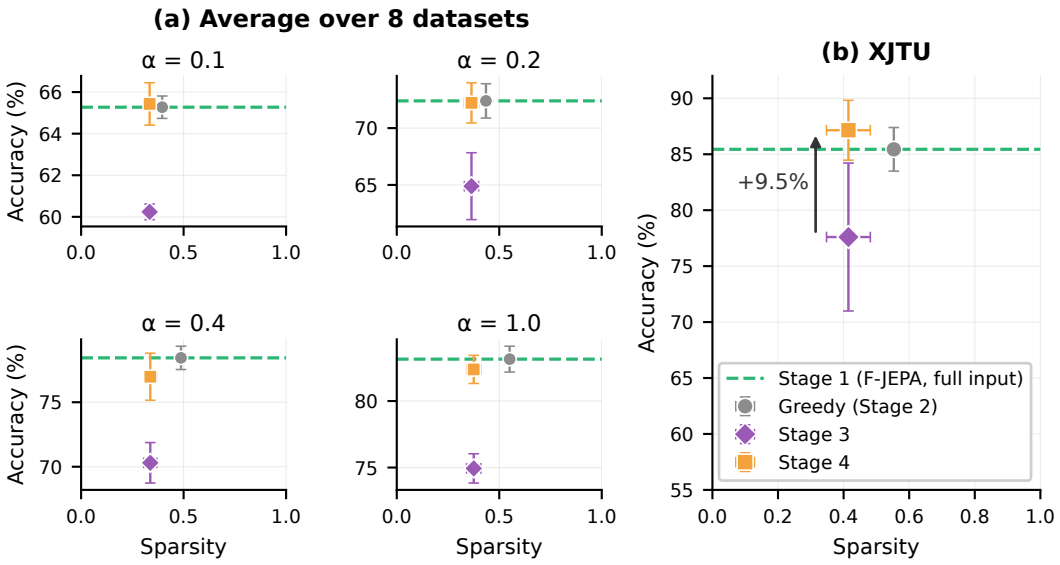

Figure 2: **Accuracy vs. sparsity averaged over eight datasets at each label ratio α.** Figure (b) shows XJTU at $\alpha = 0.4$ as an example. Each point shows mean $\pm$ standard error over 5 seeds. The full-input method (F-JEPA) appears as a dashed horizontal line; Stages 3 and 4 operate on gated input at reduced sparsity. Stage 4 (KD refinement and verification) consistently recovers accuracy toward the full-input baseline while retaining 20–45% of the input.

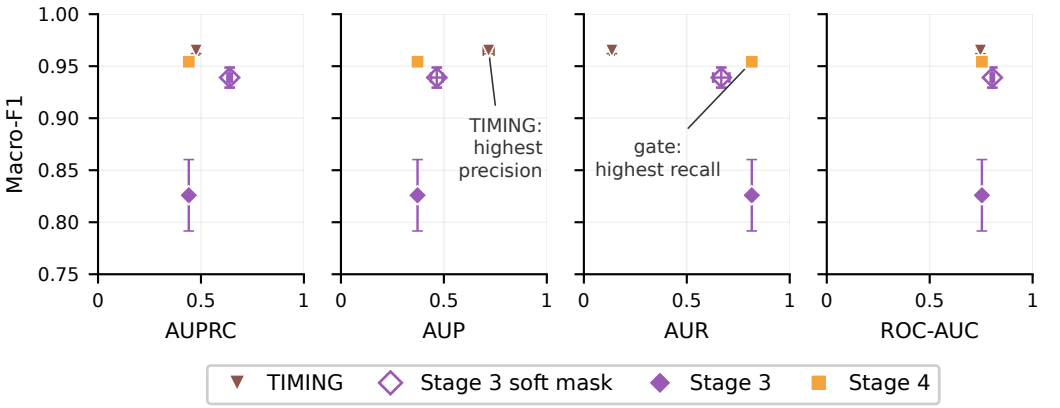

Figure 3: **Explanation faithfulness vs. macro-F1 on MIT-BIH at $\alpha = 0.4$.** Each figure plots a metric (AUPRC, AUP, AUR, ROC-AUC) against macro-F1, with mean $\pm$ standard error over 5 seeds. Stage 4 (orange) recovers macro-F1 to a be comparable with the ungated baseline while matching or exceeding the raw TIMING attributions on AUR and ROC-AUC. Stage 3 soft mask (purple) surpasses TIMING on AUPRC, indicating that BCE training denoises the raw attribution map.

Heartbeat, InsectWingbeatSound, NATOPS, and UWaveGestureLibraryAll. Three are from the SemiTime benchmark [11]: MFPT and XJTU (industrial fault diagnosis) and EpilepticSeizure (medical). The ninth, MIT-BIH, is an electrocardiogram dataset with expert-annotated ground-truth saliency regions introduced as an explanation benchmark by TimeX [36], enabling direct evaluation of explanation quality against human-identified diagnostic features.

Evaluation Protocol. All datasets follow a 60/20/20 train/validation/test split, with results averaged over 5 random seeds. The validation set is used exclusively for hyperparameter selection and threshold calibration (κ^*); all reported results are on the held-out test set. We evaluate under three label ratios $\alpha \in \{0.1, 0.2, 0.4\}$, where α denotes the fraction of training samples with observed labels; the remaining $(1 - \alpha)$ fraction constitutes $\mathcal{D}_u$. The backbone encoder is a Temporal Convolutional

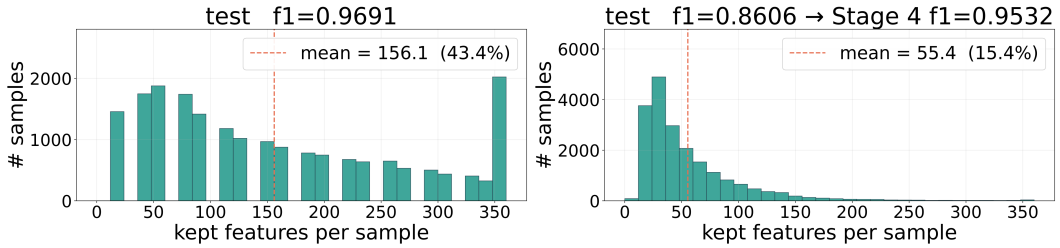

Figure 4: **Distribution of kept cells per sample on MIT-BIH ($T \cdot C = 360$, $\alpha = 0.4$).** Left: greedy selection (Stage 2) produces a broad distribution (mean kept fraction 43.4%). Right: the hard gate (Stage 3) compresses the distribution toward lower kept counts (mean kept fraction 15.4%). Dashed lines indicate the per-sample mean. After KD refinement and verification (Stage 4), macro-F1 recovers to a level comparable with Stage 1.

Network (TCN) across all experiments. The supervised baseline (SupCE) trains F_θ on $\mathcal{D}_l$ with cross-entropy only, without auxiliary objectives or gating.

Metrics. Classification performance is reported as accuracy (macro-F1 on MIT-BIH due to class imbalance) at three pipeline checkpoints: Stage 1 (base classifier on full input), Stage 3 (frozen classifier on gated input), and Stage 4 (student after KD refinement and verification on gated input). Sparsity is measured by the kept ratio $\|\mathbf{m}^{\text{hard}}\|_0/(T \cdot C)$. On MIT-BIH, we additionally evaluate explanation quality against ground-truth saliency using AUPRC, AUP, AUR, and ROC-AUC, with TIMING [18] applied to the same frozen Stage 1 classifier as the post-hoc attribution baseline.

4.2 Main Results

Experiments on datasets without explanation ground truth. Figure 2(a) plots accuracy against sparsity averaged over the eight datasets at each label ratio $\alpha \in \{0.1, 0.2, 0.4, 1.0\}$, and Figure 2(b) shows XJTU at $\alpha = 0.4$ as an example; per-dataset results appear in Appendix D.1. Each point reports the mean and standard error over 5 seeds. F-JEPA (Stage 1) operates on full input and appears as a dashed horizontal line, while Stage 3 and Stage 4 operate on gated input at reduced sparsity. Two patterns are consistent across datasets. First, Stage 3 (frozen classifier on gated input) drops accuracy relative to Stage 1, confirming that the gate removes input the classifier originally relied on. Second, Stage 4 (KD refinement and verification) recovers most or all of this drop: on Heartbeat, MFPT, XJTU, and UWaveGestureLibraryAll, Stage 4 matches or exceeds Stage 1 accuracy while retaining only 20–45% of the input (Appendix D.1). The recovery is largest on datasets where Stage 3 suffers the steepest drop (EpilepticSeizure, UWaveGestureLibraryAll, XJTU), indicating that KD is most critical precisely where gating is most aggressive. Stage 1 (F-JEPA) matches or exceeds f-VAT, the SOTA SS-TSC method, on 7 of 8 datasets by point estimate, with clear margins on 5 (Appendix D.1), confirming that the F-JEPA auxiliary produces representations of sufficient quality for the downstream stages.

Explanation quality on MIT-BIH. Figure 3 reports explanation faithfulness (AUPRC, AUP, AUR, ROC-AUC) against macro-F1 on MIT-BIH at $\alpha = 0.4$; We compare four variants: the raw TIMING attributions, the soft gate output (Stage 3 soft mask), and the binarized gate output before and after KD (Stage 3 and Stage 4). Stage 4 achieves the best trade-off: macro-F1 recovers to a level comparable with Stage 1 while matching or exceeding the raw TIMING attributions on AUR and ROC-AUC. On AUPRC, the soft gate output surpasses TIMING, indicating that BCE training acts as a denoiser over the raw attribution signal. On AUR, the hard gate scores highest, confirming a low false-negative rate: the gate rarely misses truly salient features. The gap between AUR and AUP reflects a sufficiency-plausibility asymmetry: the model relies on features beyond the annotated ground truth, producing over-selection rather than under-selection. Two factors contribute to this gap: imperfections in the TIMING attribution map propagate into the gate's supervision, and the classifier itself may depend on features outside the human-annotated saliency regions.

4.3 Further Analysis

Per-instance sparsity distribution. Figure 4 shows histograms of kept time steps per sample on MIT-BIH ($T \cdot C = 360$) for greedy selection (Stage 2, left) and the hard gate (Stage 3, right). Greedy selection produces a broad distribution with mean kept fraction 43.4%, reflecting variation in how many features each instance requires to preserve the classifier's prediction. The hard gate compresses this distribution: the mean kept fraction drops to 15.4%. This compression indicates that BCE training and threshold calibration identify a sparser sufficient set than greedy selection alone. The macro-F1 at the greedy stage is 0.97, while the hard gate alone achieves 0.86; after KD refinement and verification (Stage 4), macro-F1 recovers to a level comparable with the ungated baseline (Figure 3), confirming that the gate selects a sufficient subset even at this higher sparsity.

Student-teacher alignment in Stage 4. Figures 15 and 16 show that outcomes of the student model (Stage 4) align very well with those of the black box teacher (F-JEPA), indicating high consistency between them (see Appendix F.3). However, the student model is an ante-hoc explainable model.

Effects of label ratios on explanation quality. Figure 14 shows that explanation quality tied to the base SS-classifier's representation quality, which weakens under label scarcity, thus highlighting the need for stronger SS-TSC model like F-JEPA (see Appendix F.1).

Other studies. Due to space constraints, please refer to Appendix for many other analyses such as hyper-parameter sensitivity in H, per-instance sparsity distribution across label ratios in F.2, computation costs in J or qualitative visualizations in I.

4.4 Ablation Studies

Stage progression on MIT-BIH. Figure 5 isolates the contribution of each pipeline stage by reporting macro-F1 on MIT-BIH across all four label ratios ($\alpha \in \{0.1, 0.2, 0.4, 1.0\}$). The pattern is consistent across ratios: Stage 1 achieves high macro-F1, Stage 3 (frozen classifier on gated input) drops substantially, and Stage 4 (KD refinement and verification) recovers to a level close to Stage 1. Stage 3 alone loses 12–18 percentage points depending on the label ratio, with the largest drop and highest variance at $\alpha = 0.2$ and $\alpha = 0.4$. This confirms that the gate removes information the classifier relies on, and that operating on gated input without adaptation is insufficient. Stage 4 recovers nearly all of the lost performance, restoring macro-F1 to within 1–2 percentage points of Stage 1 across all ratios. The recovery is equally effective under label scarcity ($\alpha = 0.1$) and full supervision ($\alpha = 1.0$), indicating that KD on $\mathcal{D}_l \cup \mathcal{D}_u$ compensates for the limited labeled signal. The V-shaped trajectory from Stage 1 through Stage 3 to Stage 4 provides direct evidence that KD refinement in Stage 4 is not an optional refinement but an essential component.

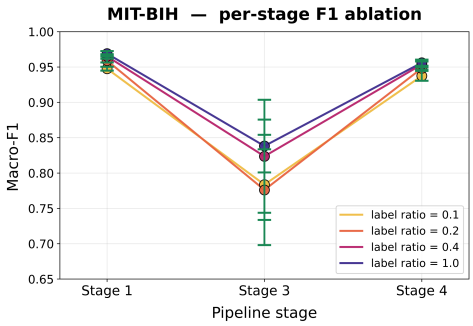

Figure 5: **Per-stage macro-F1 ablation on MIT-BIH across four label ratios.**

Other ablations. Please refer to Appendix G for other ablation studies such as target sparsity sweep on the XJTU dataset.

5 Related Work

Time Series Explanation. Explainability methods for TSC fall into two paradigms. Ante-hoc approaches embed explanations into the architecture via attention mechanisms or interpretable features [22, 51, 15, 40, 53, 30, 29], but impose constraints that limit model flexibility. Post-hoc methods generate attribution maps without modifying the model, operating at the time-point level via gradient-based [46, 3, 57, 56, 44, 60, 42, 18], perturbation-based [8], or surrogate approaches [36, 25], or at the segment level [43, 49, 48]. However, post-hoc explanations do not guarantee that the model follows the identified features; faithfulness can only be assessed through proxy evaluation. Our Sensory Gating framework is the first approach that bridges both paradigms: we use post-hoc attributions as supervision to train an ante-hoc gate, then verify via knowledge distillation that the model operates on the gate's selected features at inference.

Semi-supervised Time Series Classification. Current SS-TSC methods are built on pseudo-labels, regularization and pretraining, and they often combine these ideas. For example, the work in [10] combines pretraining with pseudo-labels through Temporal and Contextual Contrasting. Its training has four phases, including self-supervised pretraining, fine-tuning, generating pseudo-labels, and semi-supervised training. Other methods learn from unlabeled data through a self-supervised task. The work in [55] combines time series classification with a self-supervised temporal relation module, and SemiTime [11] similarly predicts temporal relations in time series. The work in [19] instead pairs classification with a forecasting task that learns from unlabeled but structured data. A third self-supervised task is the triplet loss in time, which the work in [17] uses to improve SS-TSC models. More recently, f-VAT [35] adapts an adversarial regularization loss to time series. However, all of these methods focus only on accuracy and do not address explainability. Our F-JEPA addresses the same representation learning problem, demonstrating that a forecasting-based objective is competitive with dedicated SSL approaches while providing the encoder quality needed for next step of the sensory gating framework.

6 Conclusion

We presented Sensory Gating, a framework that bridges post-hoc attribution and ante-hoc explanation for semi-supervised time series classification. The framework converts TIMING attributions into per-instance binary masks through greedy selection and BCE-supervised gate training, then verifies the gate's faithfulness via knowledge distillation. Binary masks arise by construction, avoiding the sparsity losses and differentiable relaxations required by end-to-end learnable-mask approaches. Experiments on eight datasets across four label ratios confirm that KD refinement and verification consistently recovers classification accuracy to within 1–2 percentage points of the ungated baseline while retaining 20–45% of the input. On MIT-BIH, where ground-truth saliency annotations are available, the gate's selected features align with domain-relevant regions and the soft gate output surpasses the raw TIMING attributions on AUPRC, indicating that BCE training denoises the attribution signal. The greedy selection further reveals that explanation difficulty varies per instance, providing an additional diagnostic signal for prediction reliability.

Acknowledgements

We thank the anonymous reviewers and the area chair, whose feedback on terminology, ablation coverage, structural clarity, and presentation materially improved this paper.

We thank Anthony Bagnall, Eamonn Keogh, Jason Lines, Aaron Bostrom, James Large, Matthew Middlehurst, Amaia Abanda, Gustavo Batista, Mustafa Baydogan, Angus Dempster, Houtao Deng, Michael Flynn, Germain Forestier, Ben Fulcher, Tomasz Gorecki, Josef Grabocka, David Guijo-Rubio, Georgiana Ifrim, Hassan Ismail Fawaz, Myong Jeong, Rohit Kate, Jessica Lin, Benjamin Lucas, Pierre-Francois Marteau, Usue Mori, Abdullah Mueen, George Oastler, Alejandro Pasos Ruiz, Francois Petitjean, Thanawin Rakthanmanon, Patrick Schafer, Pavel Senin, Ahmed Shifaz, Diego Silva, Alexandra Stefan, Geoff Webb, Chang Wei Tan, Hoang Anh Dau, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and others of `timeseriesclassification.com` for producing and maintaining the UCR and UEA time series archives used in this work.

We thank George Moody and Roger Mark for the MIT-BIH Arrhythmia Database, and Owen Queen, Thomas Hartvigsen, Teddy Koker, Huan He, Theodoros Tsiligkaridis, and Marinka Zitnik for the cardiologist-annotated MIT-ECG subset released with their NeurIPS 2023 work [36].

This research is part-funded by the European Union (Horizon Europe 2021–2027 Framework Programme, Grant Agreement number 101072456; views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union, which cannot be held responsible for them) and by the Engineering and Physical Sciences Research Council under grant number EP/X029174/1.

References

[1] Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian Erich Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 64 6 Pt 1:061907, 2001.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *CVPR*, pages 15619–15629, 2023.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[4] A. Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron George Bostrom, Paul Southam, and Eamonn J. Keogh. The uea multivariate time series classification archive, 2018. *ArXiv*, abs/1811.00075, 2018.

[5] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv*, abs/1803.01271, 2018.

[6] Graeme S. Chambers, Svetha Venkatesh, Geoff A. W. West, and Hung Hai Bui. Segmentation of intentional human gestures for sports video annotation. *MMM*, pages 124–129, 2004.

[7] Yanping Chen, Adena M. Why, Gustavo E. A. P. A. Batista, Agenor Mafra-Neto, and Eamonn J. Keogh. Flying insect classification with inexpensive sensors. *J. Insect Behav.*, 27:657 – 677, 2014.

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

[9] Hoang Anh Dau, A. Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn J. Keogh. The ucr time series archive. *IEEE/CAA J. Autom. Sin.*, 6:1293–1305, 2018.

[10] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. Self-supervised contrastive representation learning for semi-supervised time-series classification. *Trans. Pattern Anal. Mach. Intell.*, 45(12):15604–15618, 2023.

[11] Haoyi Fan, Fengbin Zhang, Ruidong Wang, Xunhua Huang, and Zuoyong Li. Semi-supervised time series classification by temporal relation prediction. In *ICASSP*, pages 3545–3549. IEEE, 2021.

[12] Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[13] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733, 2020.

[14] Sara Hooker, D. Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Neural Information Processing Systems*, 2018.

[15] 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.

[16] 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.

[17] Ali Ismail-Fawaz, Maxime Devanne, Jonathan Weber, and Germain Forestier. Enhancing time series classification with self-supervised learning. In *International Conference on Agents and Artificial Intelligence*, 2023.

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

[19] Shayan Jawed, Josif Grabocka, and Lars Schmidt-Thieme. Self-supervised learning for semi-supervised time series classification. *Advances in Knowledge Discovery and Data Mining*, 12084:499 – 511, 2020.

[20] Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Adversarial attacks on time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3309–3320, 2019.

[21] Ming Hsiao Ko, Geoff A. W. West, Svetha Venkatesh, and Mohan Kumar. Online context recognition in multisensor systems using dynamic time warping. *ISSNIP*, pages 283–288, 2005.

[22] Sangdi Lin and George C. Runger. Gcrnn: Group-constrained convolutional recurrent neural network. *TNNLS*, 29:4709–4718, 2018.

[23] Chengyu Liu, David B. Springer, Qiao Li, Benjamin Moody, Ricardo Abad Juan, Francisco Javier Chorro, Francisco Castells, José Millet Roig, Ikaro Silva, Alistair E. W. Johnson, Zeeshan Syed, Samuel Emil Schmidt, Chrysa D. Papadaniil, Leontios J. Hadjileontiadis, Hosein Naseri, Ali Moukadem, Alain Dieterlen, Christian Brandt, Hong Tang, Maryam Samieinasab, Mohammad Reza Samieinasab, Reza Sameni, Roger G. Mark, and Gari D. Clifford. An open access database for the evaluation of heart sound algorithms. *Physiol. Meas.*, 37:2181 – 2213, 2016.

[24] Jia-Yang Liu, Zhen Yu Wang, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–9, 2009.

[25] 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*, 2024.

[26] Kerry McAlonan, James R. Cavanaugh, and Robert Wurtz. Attentional modulation of thalamic reticular neurons. *J. Neurosci.*, 26:4444 – 4450, 2006.

[27] David A. McCormick and Thierry Bal. Sensory gating mechanisms of the thalamus. *Curr. Opin. Neurobiol.*, 4 4:550–6, 1994.

[28] Nijat Mehdiyev, Johannes Lahann, Andreas Emrich, David Enke, Peter Fettke, and Peter Loos. Time series classification using deep learning for process planning: A case from the process industry. *Procedia Computer Science*, 114:242–249, 2017.

[29] Dominique Mercier, Andreas R. Dengel, and Sheraz Ahmed. P2exnet: Patch-based prototype explanation network. In *ICONIP*, 2020.

[30] Dominique Mercier, Andreas R. Dengel, and Sheraz Ahmed. Patchx: Explaining deep models by intelligible pattern patches for time-series classification. *IJCN)*, pages 1–8, 2021.

[31] George B. Moody and Roger G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Eng. Med. Biol. Mag.*, 20:45–50, 2001.

[32] Menaa Nawaz and Jameel Ahmed. Cloud-based healthcare framework for real-time anomaly detection and classification of 1-d ecg signals. *PLOS ONE*, 17, 2022.

[33] Thu Trang Nguyen, Thach Le Nguyen, and Georgiana Ifrim. Robust explainer recommendation for time series classification. *Data Mining and Knowledge Discovery*, 38:3372 – 3413, 2023.

[34] Richard E. Nisbett and Timothy D. Wilson. Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, 84:231–259, 1977.

[35] Qingyi Pan and Yicheng Li. Functional virtual adversarial training for semi-supervised time series classification. In *NeurIPS*, 2025.

[36] 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.

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

[38] Davi Silva Rodrigues, Ana Catharina S Nastri, Marcello M Magri, Maura Salaroli de Oliveira, Ester C Sabino, Pedro HMF Figueiredo, Anna S Levin, Maristela P Freire, Leila S Harima, Fátima LS Nunes, et al. Predicting the outcome for covid-19 patients by applying time series classification to electronic health records. *Medical Informatics Decis. Mak.*, 22(1):187, 2022.

[39] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206 – 215, 2018.

[40] Leonid Schwenke and Martin Atzmueller. Show me what you're looking for visualizing abstracted transformer attention for enhancing their local interpretability on time series data. *FLAIRS*, 34, 2021.

[41] Noa Segall, Gene W. Hobbs, Christopher B. Granger, Amanda E. Anderson, Alberto S. Bonifacio, Jeffrey M. Taekman, and Melanie C. Wright. Patient load effects on response time to critical arrhythmias in cardiac telemetry: A randomized trial. *Critical Care Medicine*, 43:1036–1042, 2015.

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

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

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

[45] 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.

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

[47] Amirsajjad Taleban, Rodney Sparapani, Patrick Noffke, Sharone Zlochiver, Qiang Lu, Michael E Widlansky, and Jake Luo. Explainable artificial intelligence in electrocardiography: A systematic review. *Biomed. Signal Process. Control.*, 114:109325, 2026.

[48] 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.

[49] 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.

[50] Hugues Turbé, Mina Bjelogrlic, Christian Lovis, and Gianmarco Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, 5:250–260, 2022.

[51] 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*, pages 2624–2629, 2018.

[52] Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *Trans. Rel.*, 69:401–412, 2020.

[53] Yichang Wang, Rémi Emonet, Élisa Fromont, Simon Malinowski, and Romain Tavenard. Adversarial regularization for explainable-by-design time series classification. *ICTAI*, pages 1079–1087, 2020.

[54] Timothy D. Wilson and Jonathan W. Schooler. Thinking too much: introspection can reduce the quality of preferences and decisions. *J. Pers. Soc. Psychol.*, 60 2:181–92, 1991.

[55] Liang Xi, Zichao Yun, Han Liu, Rui-Dong Wang, Xun-Hua Huang, and Haoyi Fan. Semi-supervised time series classification model with self-supervised learning. *Eng. Appl. Artif. Intell.*, 116:105331, 2022.

[56] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[57] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *Int. J. Comput. Vis.*, 126(10):1084–1102, 2018.

[58] Zhibin Zhao, Tianfu Li, Jingyao Wu, Chuang Sun, Shibin Wang, Ruqiang Yan, and Xuefeng Chen. Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study. *ISA Trans.*, 2020.

[59] 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. *ArXiv*, abs/2410.02970, 2024.

[60] Linjiang Zhou, Chao Ma, Xiaochuan Shi, Dian Zhang, Wei Li, and Libing Wu. Salience-cam: Visual explanations from convolutional neural networks via salience score. In *IJCNN*, pages 1–8, 2021.

Technical appendices and supplementary material

The appendix is organized to first deepen the reader's understanding of each component, then broaden the empirical evidence beyond what the main paper can accommodate. We start with a self-contained description of F-JEPA (Appendix A), the auxiliary objective that drives semi-supervised representation learning in Stage 1. With this foundation in place, we turn to the full experimental details (Appendix C) and report comprehensive results across all four label ratios $\alpha \in \{0.1, 0.2, 0.4, 1.0\}$ in Appendix D. Appendix F offers three additional analyses that probe the framework's internals: how explanation sparsity varies across instances, how closely the Stage 4 student aligns with the Stage 1 teacher, and how explainability degrades as labeled data becomes scarce. We then revisit the ablation study from §4.4, extending it to XJTU with additional $s_{\max}$ configurations (Appendix G), and examine the sensitivity of our method to this hyperparameter on both XJTU and MIT-BIH (Appendix H). To give the reader a concrete sense of what the gate learns, Appendix I presents qualitative visualizations that compare soft masks, hard masks, TIMING attribution maps, and ground-truth saliency annotations side by side. Training, inference, and explanation time are reported in Appendix J. We close with a discussion of limitations (Appendix K) and broader impact (Appendix L).

A Forecasting-JEPA (F-JEPA) Details

This section provides a detailed description of the Forecasting Joint-Embedding Predictive Architecture (F-JEPA), the auxiliary self-supervised objective used in Stage 1.

Motivation. Semi-supervised TSC requires the encoder to learn representations from both labeled and unlabeled data. F-JEPA provides an auxiliary self-supervised signal on all data $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ by training the encoder to predict future temporal structure from past context in a learned embedding space. Unlike contrastive objectives that require data augmentation and negative pairs, F-JEPA operates on the raw temporal structure of the input, which is a natural inductive bias for sequential data.

Temporal splitting. Given an input time series $\mathbf{x} \in \mathbb{R}^{T \times C}$, we split it along the temporal axis into a context window $\mathbf{x}_{\text{ctx}} = \mathbf{x}_{1:T_c}$ and a target window $\mathbf{x}_{\text{tgt}} = \mathbf{x}_{T_c+1:T}$, where T_c is the context length. We set $T_c = \lfloor T/2 \rfloor$ by default, splitting each series at its midpoint.

Context encoding and prediction. The context window is encoded by the online encoder f_θ and reduced to a single vector via global average pooling:

$$\mathbf{r}_{\text{ctx}} = \text{pool}(f_\theta(\mathbf{x}_{\text{ctx}})) \in \mathbb{R}^d, \tag{10}$$

where d is the encoder's representation dimension. A linear predictor $h : \mathbb{R}^d \to \mathbb{R}^d$ maps this pooled context representation to a predicted target embedding:

$$\hat{\mathbf{z}} = h(\mathbf{r}_{\text{ctx}}) \in \mathbb{R}^d. \tag{11}$$

Target encoding. The target window is encoded by an exponential moving average (EMA) encoder f_ξ and pooled:

$$\mathbf{z} = \text{pool}(f_\xi(\mathbf{x}_{\text{tgt}})) \in \mathbb{R}^d. \tag{12}$$

The EMA encoder shares the same architecture as f_θ but its parameters ξ are not updated by gradient descent. Instead, ξ is updated after each training step as:

$$\xi \leftarrow \tau \cdot \xi + (1 - \tau) \cdot \theta, \tag{13}$$

where $\tau \in [0, 1)$ is the decay rate, following the BYOL [13] update rule. The EMA encoder produces a slowly evolving target that stabilizes training without requiring negative pairs.

Loss function. Both $\hat{\mathbf{z}}$ and $\mathbf{z}$ are ℓ_2-normalized before computing the loss. The F-JEPA loss is defined as the cosine distance between the predicted and target embeddings:

$$\mathcal{L}_{\text{F-JEPA}} = 1 - \frac{\hat{\mathbf{z}}^\top \mathbf{z}}{\|\hat{\mathbf{z}}\| \, \|\mathbf{z}\|}. \tag{14}$$

Table 1: **CPU inference time for batches of 64 instances.** Times are reported as mean ± standard deviation. Speed-ups are calculated from the reported mean batch times.

Dataset	T	Ours (ms)	Top-k TIMING + KD (ms)	Speed-up
CricketX	300	462.04 ± 4.96	$134{,}390.2 \pm 538.9$	$\mathbf{291}\times$
XJTU	1024	$1{,}484.19 \pm 8.52$	$431{,}331.0 \pm 967.8$	$\mathbf{291}\times$

Difference from the original JEPA. The original JEPA [2] operates in vision and optimizes a per-patch MSE loss across token-level representations produced by a Vision Transformer. F-JEPA differs in two respects. First, F-JEPA formulates the objective as temporal forecasting: the context and target correspond to past and future segments of the time series, whereas the original JEPA masks spatial patches within an image. Second, F-JEPA pools each window into a single vector and uses cosine distance (Eq. 14) rather than per-token MSE. This design choice is motivated by the pooling step: reducing each window to a single vector removes the token-level structure that makes MSE informative in the original JEPA, and MSE between pooled vectors is susceptible to representation collapse (all representations converging to a constant). Cosine distance with ℓ_2-normalization constrains representations to the unit hypersphere, preventing collapse without requiring additional architectural components such as batch normalization or prediction heads with hidden layers.

Integration with Stage 1. F-JEPA is applied as an auxiliary loss on all data $\mathcal{D}$ alongside the cross-entropy loss $\mathcal{L}_{\text{CE}}$ on labeled data $\mathcal{D}_l$:

$$\mathcal{L}(\mathcal{D}; \theta) = \mathcal{L}_{\text{CE}}(\mathcal{D}_l;\, \theta) + \lambda_{\text{F-JEPA}}\, \mathcal{L}_{\text{F-JEPA}}(\mathcal{D};\, \theta, \xi), \tag{15}$$

where $\lambda_{\text{F-JEPA}}$ is a regularization weight. Gradients from $\mathcal{L}_{\text{F-JEPA}}$ flow through f_θ and h only; the EMA encoder f_ξ receives no gradients and is updated solely via Eq. 13. The predictor h is discarded after Stage 1; only f_θ and the classification head are retained for subsequent stages.

B Practical Implications for Deployment and Research

The practical value of Sensory Gating has two aspects. For practitioners, it reduces the cost of producing explanations and incorporates the selected features directly into the deployed prediction pipeline. For researchers, it provides a lightweight subset-selection module that supports repeated analysis without recomputing post-hoc attributions for every input.

Deployment cost. Arrhythmia classification from ECG, as evaluated on MIT-BIH, provides a concrete motivation for considering explanation cost. Deep learning models achieve strong predictive performance across ECG tasks, but their clinical usefulness also depends on whether practitioners can interpret their outputs. A systematic review of explainable AI in electrocardiography identifies interpretability as an important requirement for clinical adoption, particularly when predictions inform consequential decisions [47]. When an explanation is part of the information needed to assess a prediction, its computation time becomes part of the workflow.

Research on cardiac telemetry further highlights the importance of timely responses to critical arrhythmias [41]. This motivates reducing avoidable processing delays, although our experiments do not measure clinical response times or patient outcomes. The computational burden also matters when systems process many signals. For example, an existing cloud-based ECG framework combines real-time anomaly detection with a separate batch-processing stage that classifies stored signals and presents the results to clinicians [32]. In such a workflow, generating an explanation for every prediction adds to the batch-processing cost. Sensory Gating moves attribution computation and mask construction to the offline training pipeline, after which deployment requires one gate forward pass and one student forward pass. The offline cost is incurred when preparing or updating the deployed model; GPU training times are reported in Appendix H.

Table 1 compares CPU inference time for our deployed pipeline with a top-k TIMING + KD pipeline at a batch size of 64. Our method achieves an approximately $291\times$ speed-up on both CricketX and XJTU.

The relative speed-up is similar across these two datasets despite their approximately $3.4\times$ difference in sequence length. In absolute terms, processing one batch takes approximately 0.46 seconds with

our pipeline versus 2.24 minutes with top-k TIMING + KD on CricketX, and 1.48 seconds versus 7.19 minutes on XJTU. Dividing the batch times by 64 gives average processing times of 7.2 and 23.2 milliseconds per instance for our pipeline, compared with 2.1 and 6.7 seconds for the baseline. These values describe average processing time under batching, rather than latency measured for individual requests. The results demonstrate the benefit of avoiding repeated attribution computation in the measured deployment setting.

Connecting explanations to deployed predictions. Deployment also requires a clear relationship between an explanation and the prediction it accompanies. For post-hoc explanations, this relationship is commonly assessed through perturbation-based evaluations that query the model after modifying features at multiple masking ratios [50, 33, 59]. These evaluations incur additional computation, and their interpretation depends on the perturbation procedure and evaluation criterion. They provide evidence about model behavior, but do not themselves constrain how the deployed model processes its input.

Sensory Gating introduces such a constraint directly into the prediction pipeline. The student receives the gated input: selected feature values pass through, while all remaining positions are replaced by the training-set mean. The displayed mask therefore identifies the original input values made available directly to the student. Only the gate and student are deployed; the teacher is used during training and evaluation.

This construction makes the selected subset part of the deployed computation, while teacher–student agreement measures how closely the resulting predictions reproduce those of the original model. The scope of this property is important: the gate itself observes the full input, and agreement between the student and teacher does not establish that they use identical reasoning. The architectural constraint concerns the student's direct input, whereas fidelity to the teacher and alignment with domain knowledge remain empirical questions. The practical benefit is therefore a more explicit connection between the displayed mask and the deployed prediction, together with lower explanation cost.

A fast subset learner for research. The lightweight gate also supports repeated investigation of the selected features across instances and experimental conditions. Once trained, it generates each mask in a single forward pass, allowing researchers to inspect feature selection and compare it with expected regions without running a fresh attribution procedure for every instance. This amortization can make broader analyses feasible within a fixed computational budget, although a new model or substantially different setting may require the gate to be retrained.

A further potential application is explanation-based detection of adversarial inputs. Prior work has demonstrated adversarial vulnerabilities in time series classifiers [20], while research in image classification has used SHAP-based signatures to distinguish adversarial examples from normal inputs [12]. The low inference cost of the sensory gate makes it a candidate for investigating related approaches in time series, where a diagnostic representation may need to be computed for every incoming instance. Such an application would require dedicated evaluation of detection performance and robustness; it is a research direction rather than an established capability of the present framework.

C Detailed Experimental Setup

C.1 UCR/UEA Datasets

CricketX. CricketX contains accelerometer recordings from four actors performing 12 cricket umpire gestures (e.g., Four, No Ball, Out, Six, Wide) for actor-independent gesture recognition [21, 6]. The task is to classify each time series into one of the 12 gesture classes.

Heartbeat. Heartbeat is derived from the PhysioNet/CinC Challenge 2016 [23], containing heart sound recordings classified as normal (healthy subjects) or abnormal (patients with confirmed cardiac diagnoses such as heart valve defects or coronary artery disease). Each recording is truncated to 5 seconds and converted into a spectrogram with a window size of 0.061 seconds and 70% overlap, where each channel corresponds to a frequency band.

InsectWingbeat. InsectWingbeat contains power spectra of insect flight sounds captured by optical sensors, originally collected by the UCR computational entomology group [7]. The task is

to classify each instance into one of 10 classes corresponding to male and female specimens of four mosquito species (*Ae. aegypti*, *Cx. tarsalis*, *Cx. quinquefasciatus*, *Cx. stigmatosoma*), two fly species (*Musca domestica*, *Drosophila simulans*), and other insects. This dataset is referred to as "InsectWingbeatSound" in the SemiTime benchmark [11].

NATOPS. NATOPS was originally part of a competition hosted at AALTD. The data is generated by sensors on the hands, elbows, wrists, and thumbs, recording x, y, z coordinates for each of the eight body locations (24 channels). The six classes correspond to distinct actions: I Have Command, All Clear, Not Clear, Spread Wings, Fold Wings, and Lock Wings.

UWaveGestureLibraryAll. UWaveGestureLibraryAll contains three-axis accelerometer recordings of eight simple hand gestures performed by multiple users [24].

C.2 Other datasets

XJTU. XJTU contains complete run-to-failure vibration data from 15 rolling element bearings acquired through accelerated degradation experiments [58, 52]. The dataset is publicly available and was adopted for semi-supervised TSC by SemiTime [11]. The task is to classify the bearing fault type from the vibration signals.

MFPT. The Machinery Failure Prevention Technology (MFPT) dataset was provided by the Society for Machinery Failure Prevention Technology [58]. The dataset was adopted for semi-supervised TSC by SemiTime [11].

EpilepticSeizure. EpilepticSeizure contains electroencephalographic (EEG) recordings spanning five categories: surface EEG from healthy volunteers with eyes open and eyes closed, intracranial EEG from epilepsy patients during seizure-free intervals (within and outside the seizure-generating area), and intracranial EEG recorded during epileptic seizures [1]. The task is to classify each single-channel time series into one of the five brain activity states. The dataset was adopted for semi-supervised TSC by SemiTime [11].

C.3 MIT-BIH Dataset with Explanation Ground Truth.

The MIT-BIH dataset has ECG recordings from 47 subjects recorded at the sampling rate of 360Hz. The raw dataset was then window-sliced into 92511 samples of 360 timestamps each [31]. Two cardiologists have labeled each beat independently and provide explanation ground truth originally used in TimeX paper [36].

C.4 Data Selection.

The dataset selection serves three purposes. Three datasets are drawn from the SemiTime benchmark to support comparison with prior semi-supervised time series classification work. Five datasets are drawn from the UCR/UEA archives to cover diverse conditions, including univariate and multivariate inputs, sequence lengths from 178 to 1,024, 2 to 15 classes, dataset sizes ranging from a few hundred to approximately 100,000 series, and four application domains. MIT-BIH serves a complementary purpose: it provides the ground-truth saliency regions used in the TimeX explanation benchmark. This distinction motivates the different evaluations. MIT-BIH is the only real-world dataset in our study with reference saliency annotations, allowing us to measure agreement between the selected features and these annotations. On the other eight datasets, we assess whether predictive performance can be maintained using the retained features and how much of the input is required. These evaluations address complementary questions: the eight benchmarks assess prediction preservation and sparsity across diverse conditions, while MIT-BIH evaluates alignment with annotated, domain-relevant regions. To reduce reliance on a single dataset with saliency annotations, we additionally evaluate the framework on four synthetic datasets with planted discriminative motifs [16]. These datasets provide ground truth by construction, enabling a complementary assessment of how well the selected features recover the known motifs.

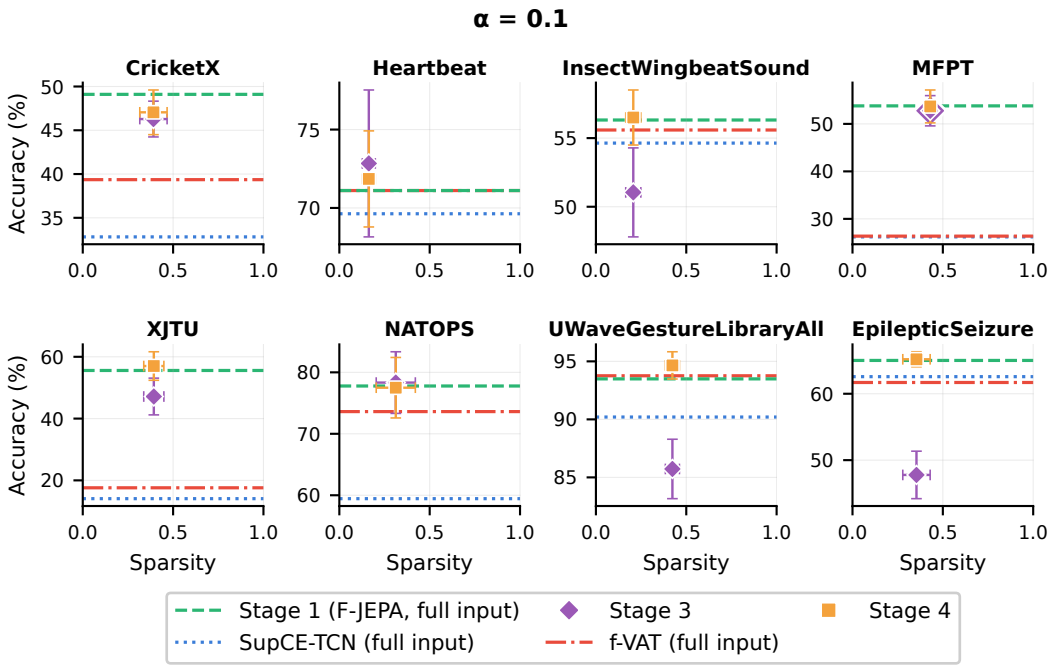

Figure 6: **Accuracy vs. sparsity at $\alpha = 0.1$ across eight datasets.** Each point shows mean $\pm$ standard error over 5 seeds. Full-input methods (SupCE-TCN, F-JEPA, f-VAT) use the whole input (sparsity $= 1.0$) and appear as horizontal lines, while Stage 3 and Stage 4 operate on gated input at reduced sparsity. Stage 4 (KD refinement and verification) consistently recovers accuracy toward the full-input baseline while retaining 20–45% of the input.

C.5 Deep Learning Architectures

We use Temporal Convolutional Network (TCN) [5], which is similar to f-VAT paper [35].

C.6 Hyperparameters

The framework introduces three primary hyperparameters: $\lambda_{\text{F-JEPA}}$, which balances cross-entropy and the F-JEPA auxiliary objective in Stage 1; λ_{hard}, which controls the optional hard cross-entropy term in Stage 4 (with $\lambda_{\text{kd}} = 1$ fixed); and $s_{\max}$, the target sparsity budget used to calibrate the gate threshold κ^*. The EMA decay coefficient τ is specific to F-JEPA and set to 0.99 throughout.

In all experiments, we use $\lambda_{\text{F-JEPA}} = 1$ and $\lambda_{\text{hard}} = 0$. Setting $\lambda_{\text{F-JEPA}} = 1$ reflects that the F-JEPA objective does not conflict with cross-entropy: both losses operate on complementary data subsets ($\mathcal{L}_{\text{CE}}$ on $\mathcal{D}_l$, $\mathcal{L}_{\text{F-JEPA}}$ on $\mathcal{D}$) and can be weighted equally without degrading classification accuracy. Setting $\lambda_{\text{hard}} = 0$ ensures that Stage 4 relies solely on knowledge distillation from the ungated teacher, which aligns with our objective: the student should reproduce the teacher's predictions on gated input, not fit hard labels directly.

We set $s_{\max} = 0.5$ for CricketX, Heartbeat, InsectWingbeat, NATOPS, UWaveGestureLibraryAll, XJTU, MFPT, and EpilepticSeizure, and $s_{\max} = 0.2$ for MIT-BIH. The appropriate sparsity budget is dataset-dependent, since the number of features sufficient for the classifier's decision varies across tasks. For MIT-BIH, the ground-truth saliency annotations indicate that diagnostically relevant features occupy approximately 12% of the input, which is consistent with our finding that the classifier recovers full performance after Stage 4 with a mean kept fraction of only 15.4% (Figure 4).

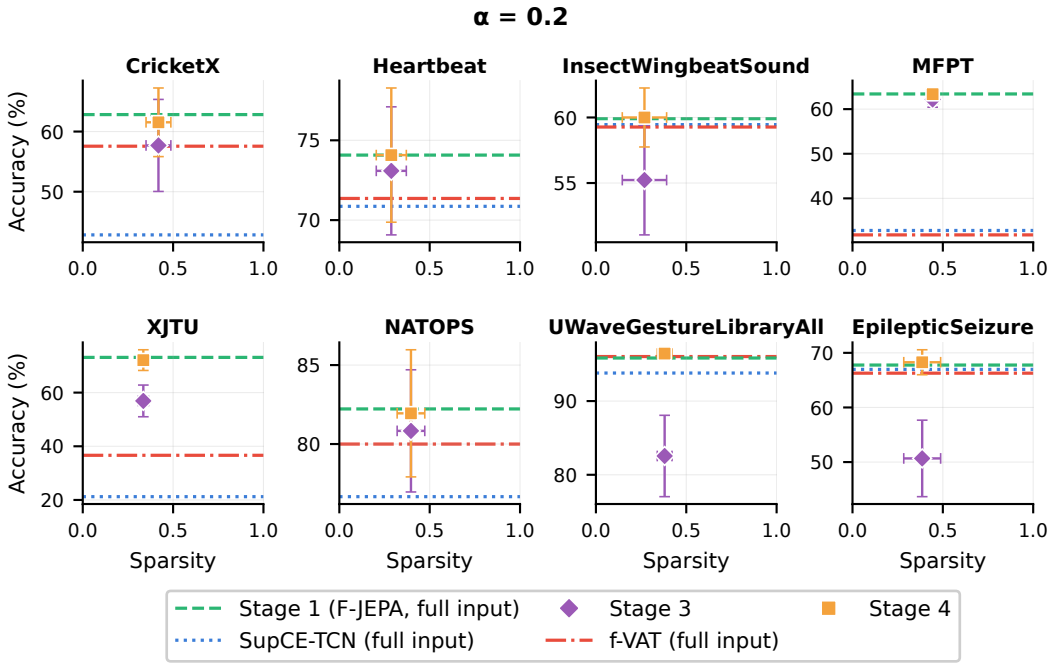

Figure 7: **Accuracy vs. sparsity at $\alpha = 0.2$ across eight datasets.** Each point shows mean $\pm$ standard error over 5 seeds. Full-input methods (SupCE-TCN, F-JEPA, f-VAT) use the whole input (sparsity $= 1.0$) and appear as horizontal lines, while Stage 3 and Stage 4 operate on gated input at reduced sparsity. Stage 4 (KD refinement and verification) consistently recovers accuracy toward the full-input baseline while retaining 20–45% of the input.

D Additional Experiments

D.1 UCR Datasets

Figures 6–9 report accuracy vs. sparsity for all eight datasets at label ratios $\alpha \in \{0.1, 0.2, 0.4, 1.0\}$. The two patterns identified in the main text hold across all ratios. First, Stage 3 (frozen classifier on gated input) consistently drops accuracy relative to Stage 1, confirming that the gate removes features the classifier relies on regardless of the supervision level. Second, Stage 4 (KD refinement and verification) recovers most or all of this drop across all ratios. At $\alpha = 1.0$ (full supervision), both Stage 1 and Stage 4 accuracy increase, and the Stage 3$\rightarrow$Stage 4 recovery gap narrows on several datasets (e.g., Heartbeat, NATOPS), indicating that stronger base representations make the gated input more informative even before KD. At $\alpha = 0.1$ (extreme label scarcity), Stage 4 still recovers accuracy on most datasets, confirming that KD on $\mathcal{D}_l \cup \mathcal{D}_u$ compensates for the limited labeled signal. F-JEPA (Stage 1) remains competitive with f-VAT across ratios, with consistent advantages on MFPT, XJTU, and NATOPS.

D.2 Applicability Beyond Deep Learning

Our framework is not limited to deep learning classifiers. To examine its applicability beyond this setting, we ran the pipeline with HYDRA as the classifier and Feature Ablation (FA) as the attribution method on MIT-BIH across all four label ratios and seeds. HYDRA+FA also provides a comparison with a strong classifier paired with a post-hoc explanation method. Table 2 reports the F1 scores and kept ratios alongside our deployed model at two feature budgets.

Our deployed model achieves higher F1 scores at every label ratio under both budgets. At the tighter budget, it retains approximately 12% of the features on average, compared with approximately 30% for HYDRA+FA, while achieving an average F1 score that is 3.1 percentage points higher. These averages are computed across the four label ratios.

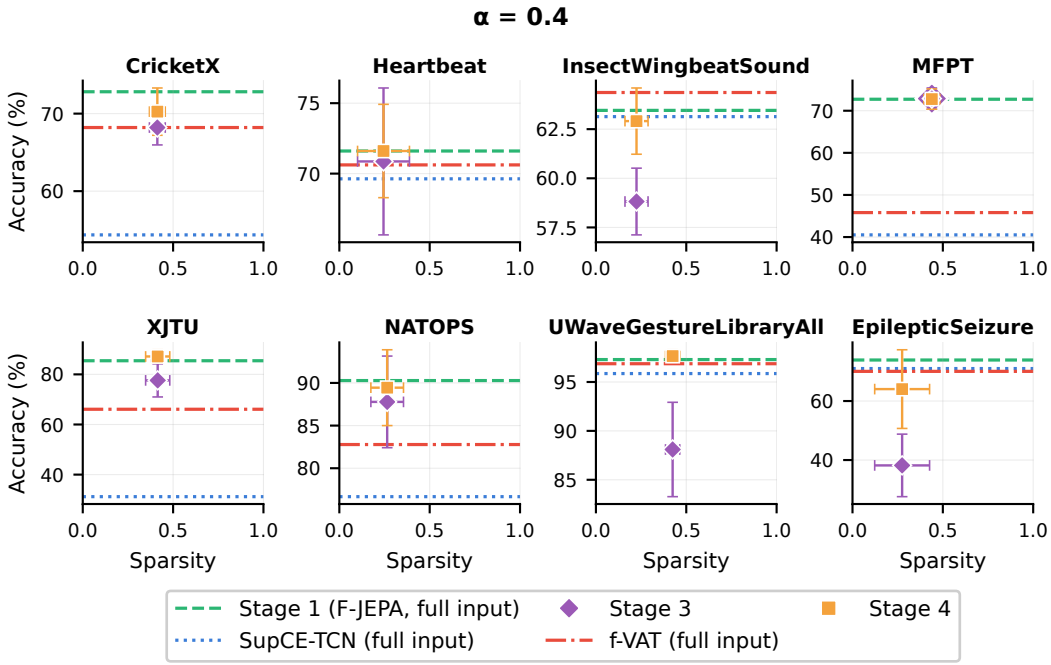

Figure 8: **Accuracy vs. sparsity at $\alpha = 0.4$ across eight datasets.** Each point shows mean $\pm$ standard error over 5 seeds. Full-input methods (SupCE-TCN, F-JEPA, f-VAT) use the whole input (sparsity $= 1.0$) and appear as horizontal lines, while Stage 3 and Stage 4 operate on gated input at reduced sparsity. Stage 4 (KD refinement and verification) consistently recovers accuracy toward the full-input baseline while retaining 20–45% of the input.

Table 2: **F1 scores and kept ratios on MIT-BIH.** Values in parentheses indicate the fraction of retained features.

α	Ours ($s_{\max} = 0.5$)	Ours ($s_{\max} = 0.15$)	HYDRA+FA
0.1	0.952 (0.38)	0.932 (0.10)	0.901 (0.34)
0.2	0.960 (0.41)	0.940 (0.10)	0.917 (0.32)
0.4	0.964 (0.44)	0.954 (0.15)	0.895 (0.24)
1.0	0.965 (0.42)	0.946 (0.11)	0.934 (0.28)

Table 3 evaluates the resulting masks against the MIT-BIH ground-truth saliency annotations using AUPRC. HYDRA+FA achieves higher mask AUPRC at every label ratio. However, this comparison also reflects substantial differences between the underlying attribution methods and classifiers, as shown by the raw attribution results in Table 4.

Averaged across label ratios, the raw attribution maps differ by approximately 0.212 AUPRC. After applying the framework, the corresponding masks differ by approximately 0.044 when our model uses $s_{\max} = 0.5$. For TIMING on TCN, mean AUPRC increases from approximately 0.463 for the raw attributions to 0.632 for the deployed masks. For HYDRA+FA, mean AUPRC remains similar, changing from approximately 0.674 to 0.676. Together, these changes narrow the initial gap by approximately 79%. This pattern suggests that differences in the starting attributions contribute to the remaining gap, although the comparison does not isolate attribution quality from classifier differences.

These results illustrate why we treat post-hoc explanation methods as inputs to the framework rather than solely as competing approaches. In this setting, the framework improves agreement with the annotated regions despite starting from weaker attributions, while the stronger HYDRA+FA attributions retain their higher AUPRC after conversion. Moreover, mean mask AUPRC is nearly unchanged between $s_{\max} = 0.5$ and $s_{\max} = 0.15$ (0.632 versus 0.633), suggesting that the tighter

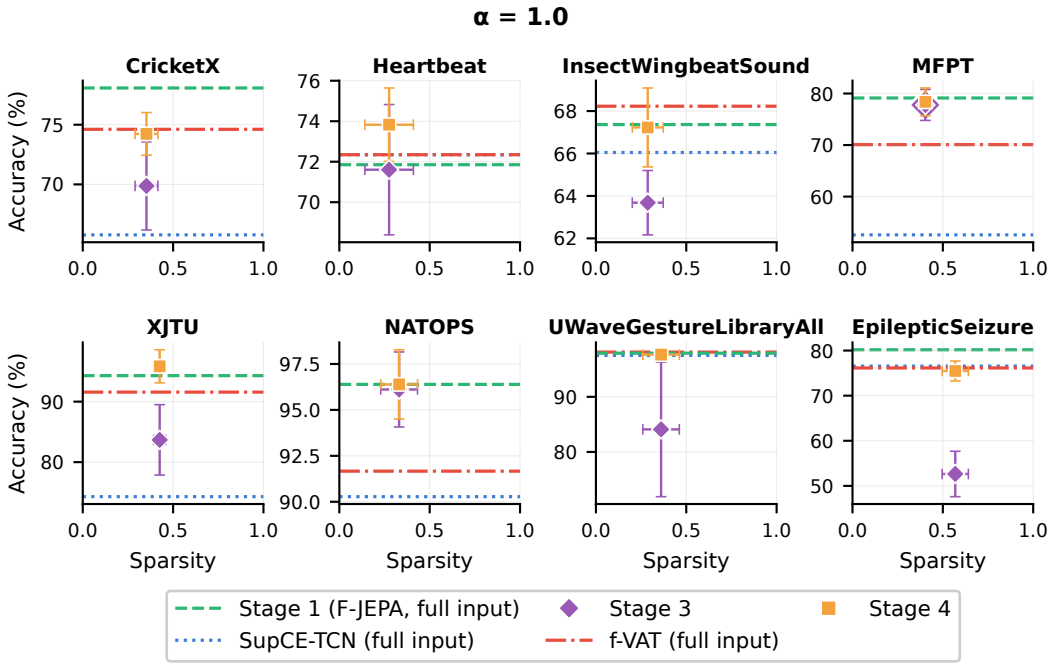

Figure 9: **Accuracy vs. sparsity at $\alpha = 1.0$ across eight datasets.** Each point shows mean $\pm$ standard error over 5 seeds. Full-input methods (SupCE-TCN, F-JEPA, f-VAT) use the whole input (sparsity $= 1.0$) and appear as horizontal lines, while Stage 3 and Stage 4 operate on gated input at reduced sparsity. Stage 4 (KD refinement and verification) consistently recovers accuracy toward the full-input baseline while retaining 20–45% of the input.

Table 3: **Mask AUPRC against MIT-BIH ground-truth saliency annotations.**

α	Ours ($s_{\max} = 0.5$)	Ours ($s_{\max} = 0.15$)	HYDRA+FA
0.1	0.608 ± 0.024	0.611 ± 0.018	0.672 ± 0.036
0.2	0.625 ± 0.018	0.628 ± 0.020	0.684 ± 0.043
0.4	0.639 ± 0.010	0.639 ± 0.010	0.659 ± 0.035
1.0	0.654 ± 0.014	0.655 ± 0.009	0.688 ± 0.041

feature budget does not explain the remaining gap within the tested range. These results support using a tighter budget when the distilled student can maintain adequate predictive performance.

D.3 Evaluation on Synthetic Data with Known Motifs

We additionally evaluate our framework on four synthetic datasets from ShapeX [16], where discriminative subsequences are planted by construction. These datasets provide known ground-truth motifs for evaluating the selected features. Table 5 reports the deployed student's test accuracy, kept ratio, and accuracy difference relative to Stage 1 over five seeds.

The student achieves mean test accuracy comparable to or slightly higher than the Stage 1 baseline on all four datasets, while retaining only 3.8–27.9% of the input features on average. These results show that the gated student maintains predictive performance while using a substantially reduced subset of the original input.

Table 6 evaluates explanation quality against the planted motifs on `mcce`, using 2,000 test instances. The soft gate achieves the highest AUPRC and ROC-AUC, whereas TIMING achieves the highest AUP and the hard gate achieves the highest AUR.

The comparison reveals a precision–recall trade-off rather than a uniform improvement across metrics. TIMING achieves the highest AUP (0.788), but its low AUR (0.072) indicates limited coverage of

Table 4: **Raw attribution AUPRC against MIT-BIH ground-truth saliency annotations.**

α	TIMING on TCN	HYDRA+FA
0.1	0.431 ± 0.023	0.672 ± 0.005
0.2	0.461 ± 0.021	0.676 ± 0.006
0.4	0.476 ± 0.012	0.674 ± 0.005
1.0	0.482 ± 0.010	0.675 ± 0.004

Table 5: **Classification performance and feature retention on four synthetic datasets.** Results are reported over five seeds. Δ denotes the accuracy difference relative to Stage 1 in percentage points.

Dataset	Test accuracy (%)	Kept features (%)	Δ vs. Stage 1
mcce	99.47 ± 0.39	27.9 ± 6.8	$+0.56 \pm 0.47$
mcch	99.93 ± 0.02	12.7 ± 1.7	$+0.08 \pm 0.07$
mtce	99.75 ± 0.12	26.2 ± 18.1	$+0.11 \pm 0.30$
mtch	99.94 ± 0.07	3.8 ± 1.4	$+0.09 \pm 0.09$

the planted motif across the evaluated thresholds. The soft gate improves the overall ranking of ground-truth features, as reflected by its higher AUPRC (0.649) and ROC-AUC (0.817), while also achieving substantially higher AUR (0.506). The deployed binary mask achieves the highest AUR (0.817), at the cost of lower AUP (0.440). For an explanation that determines which input features are available to the student, coverage of the discriminative region is particularly relevant. These results therefore highlight the importance of assessing motif coverage alongside precision when evaluating the deployed masks.

D.4 Comparison with Semi-Supervised Classification Methods

We compare F-JEPA with f-VAT and SemiTime across eight datasets, four label ratios, and five training seeds. Wilcoxon signed-rank tests indicate an overall advantage for F-JEPA over f-VAT ($p = 8.2 \times 10^{-15}$) and SemiTime ($p = 1.5 \times 10^{-2}$).

We compare F-JEPA with f-VAT and SemiTime on eight datasets across four label ratios, $\alpha \in \{0.1, 0.2, 0.4, 1.0\}$, using five seeds and the same data splits as the other experiments. The setting $\alpha = 1.0$ provides a fully supervised reference. Table 7 reports the complete test accuracy results.

F-JEPA achieves higher mean accuracy than f-VAT in 26 of the 32 settings and higher mean accuracy than SemiTime in 18 settings. It achieves the highest mean accuracy among the three methods in 13 settings. These comparisons show that F-JEPA is competitive across the evaluated datasets, although its relative performance varies with the dataset and label ratio.

To examine its applicability to another encoder, we also evaluate F-JEPA using the encoder architecture of SemiTime, denoted Semi+JEPA. Table 8 compares this variant with SemiTime using the same encoder architecture and reports the difference in mean accuracy.

With the encoder architecture held fixed, Semi+JEPA achieves higher mean accuracy than SemiTime in 31 of the 32 settings. The only exception is InsectWingbeatSound at $\alpha = 0.2$. These results support the applicability of the F-JEPA objective beyond the TCN encoder used in our main experiments.

E Additional Evaluation of Architectural Generality and Robustness

We extend the evaluation of Sensory Gating beyond the TCN backbone and examine its robustness to perturbations applied at test time. All experiments in this section use a labelled-data ratio of $\alpha = 0.4$ and five training seeds.

E.1 Transfer Across Encoder Architectures

Recurrent encoders on MIT-BIH. We replace the TCN encoder in F-JEPA with a BiGRU while keeping the gate architecture, greedy selection, threshold calibration, and knowledge distillation

Table 6: **Explanation quality against the planted motifs on `mcce`.** Results are evaluated on 2,000 test instances. Bold values indicate the highest mean for each metric.

Method	AUPRC	AUP	AUR	ROC-AUC
Gate (soft)	$\mathbf{0.6494 \pm 0.0351}$	0.5588 ± 0.0245	0.5061 ± 0.0896	$\mathbf{0.8174 \pm 0.0192}$
Gate (hard)	0.4799 ± 0.0405	0.4402 ± 0.0472	$\mathbf{0.8169 \pm 0.0333}$	0.7377 ± 0.0199
TIMING	0.5302 ± 0.0562	$\mathbf{0.7880 \pm 0.0342}$	0.0724 ± 0.0222	0.7574 ± 0.0387

Table 7: **Test accuracy (%) across eight datasets and four label ratios.** Results are reported over five seeds. Bold values indicate the highest mean accuracy in each row, including ties at the reported precision.

Dataset	α	F-JEPA	f-VAT	SemiTime
CricketX	0.1	$\mathbf{48.97 \pm 3.47}$	39.36 ± 4.70	32.19 ± 4.88
	0.2	$\mathbf{62.82 \pm 8.58}$	57.56 ± 2.84	39.98 ± 5.59
	0.4	$\mathbf{73.08 \pm 3.39}$	68.21 ± 4.96	52.14 ± 3.87
	1.0	$\mathbf{77.95 \pm 3.59}$	74.61 ± 3.03	57.73 ± 3.44
EpilepticSeizure	0.1	65.01 ± 0.70	61.70 ± 0.94	$\mathbf{74.44 \pm 1.03}$
	0.2	67.80 ± 1.44	66.28 ± 0.35	$\mathbf{75.03 \pm 1.63}$
	0.4	73.94 ± 1.36	70.05 ± 1.18	$\mathbf{77.15 \pm 1.00}$
	1.0	$\mathbf{80.22 \pm 1.84}$	76.14 ± 1.41	79.65 ± 0.89
Heartbeat	0.1	71.36 ± 3.20	71.11 ± 4.41	$\mathbf{73.09 \pm 0.55}$
	0.2	74.07 ± 5.52	71.36 ± 2.68	$\mathbf{74.32 \pm 2.21}$
	0.4	71.36 ± 6.01	70.62 ± 2.21	$\mathbf{74.07 \pm 2.31}$
	1.0	71.85 ± 3.20	$\mathbf{72.35 \pm 1.10}$	$\mathbf{72.35 \pm 4.42}$
InsectWingbeatSound	0.1	$\mathbf{56.27 \pm 1.99}$	55.59 ± 2.93	49.95 ± 5.60
	0.2	$\mathbf{60.09 \pm 2.04}$	59.27 ± 2.19	58.56 ± 1.34
	0.4	63.41 ± 2.66	$\mathbf{64.36 \pm 1.73}$	62.47 ± 3.15
	1.0	67.36 ± 1.89	$\mathbf{68.23 \pm 0.98}$	65.39 ± 1.73
MFPT	0.1	53.99 ± 3.20	26.36 ± 2.19	$\mathbf{62.54 \pm 2.88}$
	0.2	63.48 ± 1.80	31.82 ± 1.92	$\mathbf{70.81 \pm 3.10}$
	0.4	72.69 ± 2.04	45.81 ± 2.51	$\mathbf{76.52 \pm 2.78}$
	1.0	79.13 ± 3.73	70.08 ± 5.61	$\mathbf{82.82 \pm 2.33}$
NATOPS	0.1	$\mathbf{77.50 \pm 3.98}$	73.61 ± 5.56	64.44 ± 4.24
	0.2	$\mathbf{82.22 \pm 3.98}$	80.00 ± 4.23	77.50 ± 5.41
	0.4	$\mathbf{90.28 \pm 4.91}$	82.78 ± 5.43	88.61 ± 4.10
	1.0	$\mathbf{96.67 \pm 2.11}$	91.67 ± 8.33	95.56 ± 1.52
UWaveGestureLibraryAll	0.1	93.47 ± 1.33	$\mathbf{93.76 \pm 1.22}$	77.49 ± 3.91
	0.2	95.86 ± 0.42	$\mathbf{96.04 \pm 0.60}$	80.35 ± 3.05
	0.4	$\mathbf{97.32 \pm 0.29}$	96.87 ± 0.37	86.17 ± 1.83
	1.0	97.92 ± 0.70	$\mathbf{98.12 \pm 0.24}$	89.78 ± 0.49
XJTU	0.1	$\mathbf{55.63 \pm 3.10}$	17.60 ± 0.80	54.36 ± 34.87
	0.2	73.01 ± 4.30	36.64 ± 4.90	$\mathbf{87.60 \pm 2.30}$
	0.4	85.55 ± 2.17	66.08 ± 4.12	$\mathbf{94.71 \pm 1.23}$
	1.0	94.29 ± 2.43	91.57 ± 0.77	$\mathbf{94.90 \pm 2.50}$

unchanged. The purpose is to examine whether the pipeline transfers across encoder families. We therefore compare each Stage-4 student with its own Stage-1 full-input classifier, which provides the backbone-specific performance reference.

Table 9 shows that the student's mean F1 remains close to its Stage-1 reference for both backbones. The BiGRU achieves this with fewer parameters and a lower average retained fraction. Threshold calibration succeeds for all five seeds with both encoders. These results support transfer to a recurrent encoder in this setting, while the absolute predictive performance remains dependent on the backbone.

Explanation quality across backbones. We compare the gate with TIMING computed on each corresponding backbone using the MIT-BIH reference annotations. The soft gate provides a ranking

Table 8: **F-JEPA with the SemiTime encoder architecture.** Results are test accuracies (%) over five seeds. Δ denotes the mean accuracy difference between Semi+JEPA and SemiTime in percentage points.

Dataset	α	Semi+JEPA	SemiTime	Δ
CricketX	0.1	55.26 ± 3.61	32.19 ± 4.88	$+23.07$
	0.2	67.56 ± 2.92	39.98 ± 5.59	$+27.59$
	0.4	76.03 ± 2.01	52.14 ± 3.87	$+23.88$
	1.0	81.79 ± 2.67	57.73 ± 3.44	$+24.06$
EpilepticSeizure	0.1	78.15 ± 1.16	74.44 ± 1.03	$+3.71$
	0.2	80.90 ± 0.61	75.03 ± 1.63	$+5.88$
	0.4	82.92 ± 0.80	77.15 ± 1.00	$+5.77$
	1.0	83.98 ± 0.68	79.65 ± 0.89	$+4.33$
Heartbeat	0.1	73.83 ± 3.43	73.09 ± 0.55	$+0.74$
	0.2	75.80 ± 2.24	74.32 ± 2.21	$+1.48$
	0.4	74.32 ± 2.68	74.07 ± 2.31	$+0.25$
	1.0	75.56 ± 4.40	72.35 ± 4.42	$+3.21$
InsectWingbeatSound	0.1	53.36 ± 2.59	49.95 ± 5.60	$+3.41$
	0.2	56.95 ± 1.85	58.56 ± 1.34	-1.61
	0.4	62.68 ± 1.13	62.47 ± 3.15	$+0.22$
	1.0	67.68 ± 2.19	65.39 ± 1.73	$+2.29$
MFPT	0.1	74.39 ± 1.37	62.54 ± 2.88	$+11.85$
	0.2	79.57 ± 1.55	70.81 ± 3.10	$+8.75$
	0.4	86.36 ± 2.81	76.52 ± 2.78	$+9.84$
	1.0	91.58 ± 1.58	82.82 ± 2.33	$+8.76$
NATOPS	0.1	75.00 ± 8.95	64.44 ± 4.24	$+10.56$
	0.2	84.17 ± 4.35	77.50 ± 5.41	$+6.67$
	0.4	93.33 ± 4.21	88.61 ± 4.10	$+4.72$
	1.0	96.94 ± 2.67	95.56 ± 1.52	$+1.39$
UWaveGestureLibraryAll	0.1	92.33 ± 1.16	77.49 ± 3.91	$+14.84$
	0.2	94.81 ± 0.65	80.35 ± 3.05	$+14.46$
	0.4	97.02 ± 0.38	86.17 ± 1.83	$+10.85$
	1.0	97.61 ± 0.40	89.78 ± 0.49	$+7.83$
XJTU	0.1	91.41 ± 1.58	54.36 ± 34.87	$+37.05$
	0.2	97.97 ± 1.47	87.60 ± 2.30	$+10.37$
	0.4	99.36 ± 0.48	94.71 ± 1.23	$+4.65$
	1.0	99.41 ± 0.22	94.90 ± 2.50	$+4.51$

Table 9: Architectural comparison on MIT-BIH at $\alpha = 0.4$ across five seeds. Stage 1 evaluates the full-input classifier, and Stage 4 evaluates the KD student. The retained fraction ρ is reported as a percentage.

Backbone	Parameters	Stage-1 F1	Stage-4 F1	ρ (%)
TCN	742K	0.9650 ± 0.0030	0.9636 ± 0.0030	44.0 ± 6.6
BiGRU	100K	0.9568 ± 0.0071	0.9578 ± 0.0070	35.3 ± 5.0

of input features, which we evaluate using AUPRC and ROC-AUC. Table 10 also reports AUP and AUR.

The gate achieves higher AUPRC and ROC-AUC than TIMING on both backbones, indicating better agreement with the annotation ranking. The remaining metrics show a consistent trade-off: the gate has higher AUR, whereas TIMING has higher AUP. The improvement therefore concerns particular aspects of annotation agreement rather than every evaluation metric.

Additional encoder families and directionality. We further evaluate TCN, BiLSTM, Transformer, and unidirectional GRU (UniGRU) encoders on CricketX. The recurrent variants examine whether

Table 10: Agreement with MIT-BIH reference annotations across five seeds. TIMING is computed separately for each backbone. Bold values highlight the gate's higher AUPRC within each comparison.

Method	AUPRC	AUP	AUR	ROC-AUC
TCN				
Gate	$\mathbf{0.6394 \pm 0.0105}$	0.4656 ± 0.0280	0.6676 ± 0.0395	0.8042 ± 0.0078
TIMING	0.4765 ± 0.0124	0.7174 ± 0.0262	0.1362 ± 0.0109	0.7465 ± 0.0073
BiGRU				
Gate	$\mathbf{0.6487 \pm 0.0196}$	0.4442 ± 0.0428	0.7089 ± 0.0316	0.8107 ± 0.0073
TIMING	0.5346 ± 0.0282	0.7337 ± 0.0403	0.1635 ± 0.0054	0.7597 ± 0.0125

Table 11: Architectural comparison on CricketX at $\alpha = 0.4$ across five seeds. F1 and the retained fraction are expressed as percentages. Recovery denotes the reported Stage-4/Stage-1 F1 ratio.

Backbone	Stage-1 F1 (%)	Stage-4 F1 (%)	ρ (%)	Recovery
TCN	72.82 ± 3.23	70.26 ± 3.05	41.3 ± 4.4	0.9659 ± 0.0446
BiLSTM	63.85 ± 6.59	62.44 ± 5.34	40.6 ± 3.6	0.9806 ± 0.0403
Transformer	59.36 ± 5.01	57.69 ± 4.19	37.6 ± 8.5	0.9732 ± 0.0259
UniGRU	75.51 ± 2.91	75.38 ± 4.69	42.7 ± 4.0	0.9976 ± 0.0330

transfer depends on the recurrent unit or on bidirectional encoding. Table 11 reports each student's F1 relative to its own Stage-1 reference.

The reported mean recovery ratios range from 0.9659 to 0.9976, with mean retained fractions between 37.6% and 42.7%. Although the BiLSTM and Transformer have lower absolute F1 than the TCN, their students retain a comparable proportion of their own Stage-1 performance. The UniGRU results also show that bidirectional encoding is not required for successful transfer in this setting. Together, these experiments support the applicability of the pipeline across the tested convolutional, recurrent, and attention-based encoders.

A different semi-supervised objective. We evaluate the pipeline with an f-VAT base classifier while retaining the same gate architecture, greedy selection, threshold calibration, and distillation procedure. Table 12 compares f-VAT with the JEPA-aux reference on CricketX across four label ratios and five seeds.

We assess transfer by comparing each Stage-4 student with its own Stage-1 classifier. The reported recovery ratios range from 0.951 to 0.980 for JEPA-aux and from 0.947 to 1.026 for f-VAT, with mean retained fractions between 0.353 and 0.446. Averaged across the four label ratios, the recovery ratio is slightly higher for f-VAT (0.975) than for JEPA-aux (0.964), although f-VAT has lower absolute F1. These results support transfer beyond the JEPA-aux training objective. Together with the encoder comparisons, they show that the pipeline can accommodate separate changes to the encoder architecture and the base-classifier training objective in the tested settings.

E.2 Robustness to Test-Time Perturbations

We evaluate robustness on MIT-BIH using the TCN backbone. Perturbations are applied only at test time. The Stage-1 teacher, Stage-3 gate, and Stage-4 student remain frozen, with no retraining.

Additive Gaussian noise. Table 13 reports results for Gaussian noise with standard deviation $\sigma \in \{0, 0.05, 0.1\}$. Predictive performance changes little over these noise levels: teacher F1 decreases from 0.9670 to 0.9668, and student F1 decreases from 0.9653 to 0.9642. Annotation agreement is more sensitive, with AUPRC decreasing from 0.6400 to 0.6122.

Temporal shifts. We shift the test inputs by 9 or 18 timesteps, corresponding to 2.5% or 5% of the series length. The reference annotation masks receive the same shift. Table 14 shows a substantial decline in predictive performance. At a 5% shift, teacher F1 decreases from 0.9677 to 0.7397, while student F1 decreases from 0.9665 to 0.5952. The gated student is therefore more sensitive to these shifts than the full-input teacher. Reduced redundancy in the retained input is a possible explanation, although these experiments do not isolate its contribution.

Table 12: Transfer across base-classifier training objectives on CricketX across five seeds. Stage scores are F1 values expressed as percentages, and ρ denotes the retained input fraction. The final column reports the Stage-4/Stage-1 F1 ratio.

Base classifier	Stage 1	Stage 3	Stage 4	Retained ρ	S4/S1
$\alpha = 0.1$					
JEPA-aux	49.10 ± 2.94	46.28 ± 2.04	47.05 ± 2.55	0.391 ± 0.076	0.958
f-VAT	39.36 ± 4.21	36.92 ± 5.43	40.38 ± 3.74	0.419 ± 0.046	1.026
$\alpha = 0.2$					
JEPA-aux	62.82 ± 7.67	57.69 ± 7.65	61.54 ± 5.70	0.419 ± 0.069	0.980
f-VAT	57.56 ± 2.54	54.36 ± 2.27	55.51 ± 4.05	0.446 ± 0.035	0.964
$\alpha = 0.4$					
JEPA-aux	72.82 ± 3.23	68.21 ± 2.24	70.26 ± 3.05	0.413 ± 0.044	0.965
f-VAT	68.21 ± 4.43	61.41 ± 2.54	64.62 ± 2.85	0.405 ± 0.065	0.947
$\alpha = 1.0$					
JEPA-aux	78.08 ± 3.18	69.87 ± 3.69	74.23 ± 1.79	0.353 ± 0.063	0.951
f-VAT	74.62 ± 2.71	69.49 ± 3.90	71.79 ± 3.53	0.417 ± 0.049	0.962

Table 13: Robustness to additive Gaussian noise on MIT-BIH with the TCN backbone, $\alpha = 0.4$, and five seeds. All components remain frozen. The retained fraction ρ is expressed on a $[0, 1]$ scale.

σ	Retained ρ	Teacher F1	Student F1	AUPRC
0.00	0.438 ± 0.067	0.9670 ± 0.0039	0.9653 ± 0.0029	0.6400
0.05	0.423 ± 0.075	0.9667 ± 0.0035	0.9647 ± 0.0025	0.6351
0.10	0.453 ± 0.096	0.9668 ± 0.0037	0.9642 ± 0.0034	0.6122

Despite this loss in predictive performance, the retained fraction remains close to 0.44, and AUPRC remains between 0.6417 and 0.6429. The masks therefore maintain agreement with the shifted annotations even when classification deteriorates. The convolutional design of the gate may contribute to this behavior, but the results do not establish exact translation equivariance or faithfulness to the classifier's internal computation.

Label noise. We inject symmetric label noise into the training set at rates of 10%, 20%, and 40%, with the realized corruption rates reported in Table 15. We retrain the full pipeline for each noise condition and keep all components fixed at test time.

Absolute F1 declines as label noise increases, but the student continues to achieve performance close to or above its corresponding teacher. The Stage-4/Stage-1 ratio increases slightly on both datasets, from 0.951 to 0.974 on CricketX and from 1.016 to 1.054 on XJTU. On XJTU, the student's mean F1 exceeds the teacher's at every noise level, with a relative gain of approximately 1.6% under clean labels and 5.4% at 40% noise. Distillation with masked inputs may provide a regularizing effect under label noise, but this explanation remains untested. Across both datasets and all noise levels, the mean retained fraction ranges from 0.333 to 0.441.

E.3 Choice of Attribution Method

Stage 2 accepts an attribution map independently of how it is generated. Attributions are computed offline, so changing their source does not require changing the training procedures used in later stages. We use TIMING as our default because it is designed specifically for time-series attribution.

To examine whether the results depend on this choice, we replace TIMING with Integrated Gradients (IG), Feature Ablation, TimeX++, and Shapley Value Sampling. The Stage-1 setup, greedy selection procedure, gate architecture, threshold calibration, and knowledge distillation remain unchanged. We report the deployed KD student's predictive performance alongside the retained input fraction for each attribution source. The comparison therefore evaluates the full pipeline at each source's operating point; sparsity is not matched across methods.

Table 14: Robustness to temporal shifts on MIT-BIH with the TCN backbone, $\alpha = 0.4$, and five seeds. AUPRC is evaluated against correspondingly shifted reference annotations. All components remain frozen.

Shift k	k/T (%)	Retained ρ	Teacher F1	Student F1	AUPRC
0	0.0	0.438 ± 0.067	0.9677 ± 0.0024	0.9665 ± 0.0018	0.6417
9	2.5	0.441 ± 0.068	0.8083 ± 0.0210	0.7508 ± 0.0418	0.6429
18	5.0	0.442 ± 0.068	0.7397 ± 0.0245	0.5952 ± 0.0386	0.6425

Table 15: Sensitivity to symmetric training-label noise on CricketX and XJTU. Stage scores are F1 values expressed as percentages. The realized rate is the observed proportion of corrupted labels, and the final column reports the Stage-4/Stage-1 F1 ratio.

Noise	Realized	Stage 1	Stage 3	Stage 4	Retained ρ	S4/S1
CricketX						
0%	0.0%	78.08 ± 3.18	69.87 ± 3.69	74.23 ± 1.79	0.353 ± 0.063	0.951
10%	9.7%	73.85 ± 2.91	68.59 ± 3.01	70.64 ± 2.08	0.440 ± 0.030	0.957
20%	20.3%	65.51 ± 3.15	60.77 ± 4.16	63.46 ± 4.75	0.422 ± 0.020	0.969
40%	40.9%	53.46 ± 4.32	50.64 ± 5.18	52.05 ± 5.31	0.441 ± 0.033	0.974
XJTU						
0%	0.0%	94.29 ± 2.18	83.68 ± 5.82	95.84 ± 2.73	0.426 ± 0.033	1.016
10%	9.9%	85.44 ± 1.50	75.15 ± 1.80	86.88 ± 1.19	0.413 ± 0.038	1.017
20%	20.5%	72.16 ± 2.81	49.81 ± 8.73	75.52 ± 3.02	0.333 ± 0.059	1.047
40%	40.4%	44.32 ± 2.22	35.04 ± 2.69	46.72 ± 0.85	0.352 ± 0.048	1.054

We evaluate Feature Ablation on MIT-BIH only at $\alpha = 0.4$ because of its computational cost. Attribution generation covers the full 100,000-sample set at each label ratio, making evaluation across all four ratios expensive.

We additionally evaluate Shapley Value Sampling on CricketX and XJTU at $\alpha \in \{0.1, 0.2\}$. Its computational cost limits the comparison to these four settings.

Integrated Gradients and Shapley Value Sampling. IG yields similar predictive performance to TIMING across all three datasets and four label ratios. The differences in mean accuracy are at most 1.4 percentage points on CricketX and XJTU, while macro-F1 differs by at most 0.001 on MIT-BIH. IG also retains fewer features in ten of the twelve settings. For example, on XJTU at $\alpha = 0.4$, IG reaches 87.8% accuracy while retaining 33.3% of the input, compared with 87.1% accuracy and 41.5% retention for TIMING. Shapley Value Sampling likewise yields similar accuracies in the four evaluated settings. These results indicate that the pipeline's predictive utility is not specific to TIMING.

Feature Ablation. Feature Ablation gives similar performance on CricketX and at the evaluated MIT-BIH setting, but performs worse on XJTU. Its largest accuracy deficits on XJTU occur at $\alpha = 0.4$ and $\alpha = 1.0$, where it also retains fewer features than TIMING. These comparisons therefore do not isolate attribution quality from the effect of sparsity. Feature Ablation additionally requires evaluating replacements for individual features or feature groups, so its computational cost depends on both the number of groups and the cost of each model evaluation.

TimeX++. TimeX++ achieves similar macro-F1 to TIMING on MIT-BIH at broadly comparable retained fractions. On CricketX and XJTU, its mean accuracy is lower under partial supervision. At full supervision, it approaches or exceeds TIMING's accuracy, but retains substantially more input: 69.4% versus 35.3% on CricketX and 94.1% versus 42.6% on XJTU. The CricketX results also show greater variation across runs under partial supervision. TimeX++ uses a learned network to generate input-dependent explanations [25], extending the comparison to a different way of obtaining attribution maps. These results describe the performance of its maps within our pipeline; they do not isolate the effects of explainer training from those of subsequent feature selection.

Together, these experiments show that the framework supports gradient-based, perturbation-based, and learned attribution sources. IG and Shapley Value Sampling preserve predictive utility close

Table 16: Attribution-source comparison on CricketX ($T = 300$), using five seeds. Each method entry reports student accuracy (%) above the mean retained input percentage in parentheses. Ceiling denotes the full-input teacher's accuracy.

α	Ceiling	TIMING	IG	Feat. Abl.	TimeX++
0.1	49.1	47.1 ± 2.6 (39.1)	47.7 ± 3.1 (37.0)	47.6 ± 3.4 (36.7)	36.9 ± 12.6 (16.4)
0.2	62.8	61.5 ± 5.7 (41.9)	60.8 ± 6.5 (38.0)	59.1 ± 5.8 (32.6)	51.0 ± 17.1 (37.5)
0.4	72.8	70.3 ± 3.1 (41.3)	71.2 ± 1.7 (38.4)	70.3 ± 2.3 (40.9)	63.1 ± 14.5 (41.7)
1.0	78.1	74.2 ± 1.8 (35.3)	75.3 ± 2.3 (35.0)	73.6 ± 2.4 (33.2)	75.4 ± 5.5 (69.4)

Table 17: Attribution-source comparison on MIT-BIH ($T = 360$), using five seeds. Each method entry reports student macro-F1 above the mean retained input percentage in parentheses. Ceiling denotes the full-input teacher's macro-F1. Dashes indicate settings not evaluated.

α	Ceiling	TIMING	IG	Feat. Abl.	TimeX++
0.1	0.948	0.952 ± 0.002 (37.5)	0.951 ± 0.003 (43.6)	—	0.951 ± 0.003 (38.5)
0.2	0.959	0.960 ± 0.003 (41.0)	0.959 ± 0.004 (40.8)	—	0.958 ± 0.003 (36.7)
0.4	0.965	0.964 ± 0.003 (44.0)	0.963 ± 0.003 (39.3)	0.963 ± 0.002 (40.9)	0.963 ± 0.002 (41.2)
1.0	0.969	0.965 ± 0.002 (41.6)	0.965 ± 0.003 (38.5)	—	0.964 ± 0.005 (34.3)

to that obtained with TIMING in the tested settings, while Feature Ablation and TimeX++ reveal greater dependence on the dataset and operating point. The results therefore support flexibility in the choice of attribution source, while showing that predictive utility and sparsity must be assessed for each combination. They do not establish a general ranking of explanation quality. Additional results using the original TimeX are reported in Table 20; evaluation with occlusion remains future work.

TimeX. We also evaluate the original TimeX as the Stage-2 attribution source and rerun the full pipeline. TimeX uses a learned explanation network to generate attribution maps from individual inputs, jointly trained with a surrogate model to capture the behavior of the original classifier [36]. The remaining pipeline components and training procedures are unchanged.

Table 20 shows substantial utility losses in several settings. Although distillation improves the mean accuracy over the masked teacher in every setting, the student remains below the full-input teacher. For example, on XJTU at $\alpha = 0.4$, masking reduces the teacher's accuracy from 85.44% to 12.53%, while distillation recovers it to 54.35%. The gap is smaller on CricketX at $\alpha = 1.0$, where the student reaches 75.38% compared with the teacher's 78.08%, while retaining 64.6% of the input on average. Several settings also show substantial variation in accuracy and retained input fraction across runs.

One possible explanation is that maps learned through TimeX's surrogate-based objective are less suited to our greedy selection procedure, which evaluates selected features using the frozen teacher. However, we have not isolated this effect experimentally. These results identify a limitation of combining TimeX with our pipeline: supporting an attribution source does not guarantee that the resulting student will preserve the teacher's predictive performance. They do not establish that TimeX produces poorer explanations in general.

F Additional Analysis

F.1 Explainability under label scarcity

Figures 14 plot AUPRC and ROC-AUC against label ratio on MIT-BIH for three attribution variants: TIMING, soft mask, and hard mask. All three methods degrade as α decreases: AUPRC drops by 3–5 percentage points and ROC-AUC by 4–7 percentage points from $\alpha = 1.0$ to $\alpha = 0.1$. This

Table 18: Attribution-source comparison on XJTU ($T = 1024$), using five seeds. Each method entry reports student accuracy (%) above the mean retained input percentage in parentheses. Ceiling denotes the full-input teacher's accuracy.

α	Ceiling	TIMING	IG	Feat. Abl.	TimeX++
0.1	55.6	57.0 ± 4.6 (39.3)	56.1 ± 3.4 (36.1)	55.5 ± 3.0 (33.9)	44.9 ± 6.2 (29.7)
0.2	73.1	72.1 ± 3.9 (33.5)	73.5 ± 1.4 (40.2)	70.8 ± 7.8 (38.7)	63.1 ± 3.8 (44.7)
0.4	85.4	87.1 ± 2.7 (41.5)	87.8 ± 1.5 (33.3)	78.5 ± 8.3 (19.0)	83.5 ± 3.8 (65.0)
1.0	94.3	95.8 ± 2.7 (42.6)	95.3 ± 2.1 (40.6)	84.0 ± 22.7 (34.9)	96.6 ± 1.7 (94.1)

Table 19: Comparison with Shapley Value Sampling. Each method entry reports student accuracy (%) above the mean retained input percentage in parentheses. Ceiling denotes the full-input teacher's accuracy.

Dataset	α	Ceiling	TIMING	Shapley
CricketX	0.1	49.1	47.1 ± 2.6 (39.1)	47.7 ± 3.1 (41.3)
CricketX	0.2	62.8	61.5 ± 5.7 (41.9)	61.7 ± 4.8 (39.2)
XJTU	0.1	55.6	57.0 ± 4.6 (39.3)	55.4 ± 2.6 (34.5)
XJTU	0.2	73.1	72.1 ± 3.9 (33.5)	73.1 ± 3.3 (32.7)

confirms that explanation quality is tied to the base classifier's representation quality, which weakens under label scarcity. The soft mask consistently outperforms both TIMING and the hard mask at every label ratio, maintaining a stable margin of approximately 15–18 percentage points on AUPRC over TIMING. This gap indicates that BCE training provides a denoising effect that is robust to the supervision level. The hard mask trails TIMING on AUPRC due to thresholding, but matches or exceeds it on ROC-AUC at $\alpha \geq 0.2$, reflecting that binarization preserves ranking quality while sacrificing precision at the tails.

F.2 Per-instance sparsity distribution across label ratios

Figures 10–13 report the distribution of kept features per sample on MIT-BIH ($T \cdot C = 360$) across all four label ratios, complementing the $\alpha = 0.4$ analysis in the main paper (Figure 4). Each figure shows greedy selection (Stage 2, top row) and the hard gate (Stage 3, bottom row) on train, validation, and test splits. Two patterns are consistent across ratios. First, the hard gate compresses the greedy distribution toward lower kept counts at every ratio: the mean kept fraction drops from 35–47% (greedy) to 11–16% (hard gate), confirming that BCE training and threshold calibration identify a substantially sparser sufficient set. Second, Stage 4 (KD refinement and verification) recovers macro-F1 to a level comparable with Stage 1 at all ratios, despite the hard gate retaining only 11–16% of the input. The greedy kept fraction increases slightly with α (35.6% at $\alpha = 0.1$ vs. 47.5% at $\alpha = 1.0$), suggesting that stronger base classifiers trained with more labels rely on a broader set of features. The hard-gate distributions are stable across train, validation, and test splits within each ratio, indicating that the gate generalizes consistently and does not overfit to the training set.

F.3 Student–Teacher Alignment in Stage 4

Stage 4 trains the student classifier F_ϕ on gated input $\tilde{x}$ to reproduce the teacher F_θ's predictions on full input x via knowledge distillation. The distillation loss minimizes the KL divergence between student logits $y_S = F_\phi(\tilde{x})$ and teacher logits $y_T = F_\theta(x)$, without access to ground-truth labels. A correctly functioning KD objective should therefore align the student with the teacher's decision boundary rather than with the true class boundary.

Table 20: Results with TimeX as the Stage-2 attribution source. Teacher and student scores are accuracy (%), and ρ denotes the retained input fraction.

α	Teacher (full)	Teacher (masked)	Student	Kept ρ
CricketX				
0.1	49.10 ± 2.94	25.77 ± 13.29	33.59 ± 12.60	0.271 ± 0.341
0.2	62.82 ± 7.67	31.41 ± 16.41	43.21 ± 20.03	0.264 ± 0.277
0.4	72.82 ± 3.23	53.97 ± 22.80	60.38 ± 19.25	0.571 ± 0.400
1.0	78.08 ± 3.18	72.56 ± 7.58	75.38 ± 3.55	0.646 ± 0.242
XJTU				
0.1	55.57 ± 2.76	19.89 ± 9.21	43.68 ± 13.75	0.292 ± 0.172
0.2	73.12 ± 3.66	8.75 ± 4.36	46.61 ± 10.00	0.151 ± 0.088
0.4	85.44 ± 1.95	12.53 ± 2.76	54.35 ± 18.33	0.104 ± 0.051
1.0	94.29 ± 2.18	35.31 ± 16.40	85.39 ± 16.76	0.464 ± 0.211

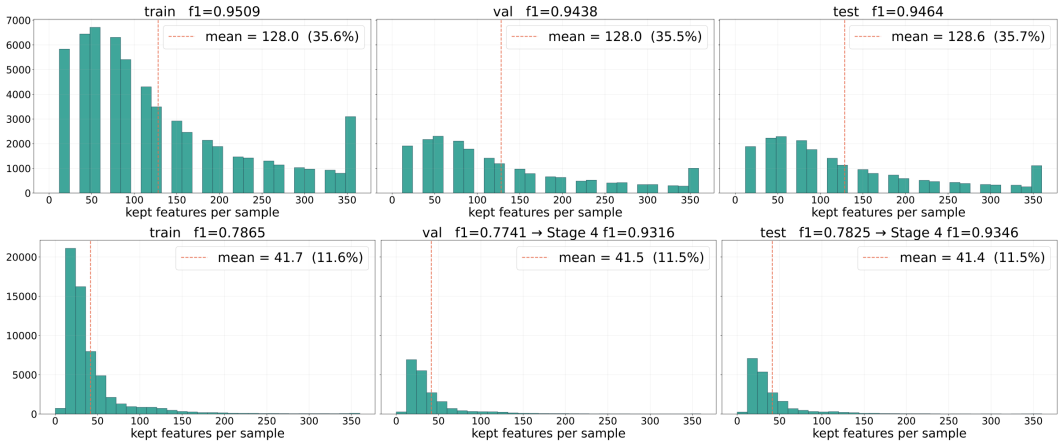

Figure 10: **Distribution of kept cells per sample on MIT-BIH** ($T \cdot C = 360$, $\alpha = 0.1$)**.** Left: greedy selection (Stage 2) produces a broad distribution. Right: the hard gate (Stage 3) compresses the distribution toward lower kept counts. Dashed lines indicate the per-sample mean. After KD refinement and verification (Stage 4), macro-F1 recovers to a level comparable with Stage 1.

We verify this by evaluating Stage 4 student predictions under two reference targets: (i) pseudo-labels generated by the Stage 1 teacher, defined as $\hat{y} = \arg\max F_\theta(x)$, and (ii) true labels. If the student aligns with the teacher, accuracy measured against pseudo-labels should meet or exceed accuracy measured against true labels, because the optimization target is the teacher's output distribution.

Figure 15 reports this comparison on XJTU and NATOPS across four label ratios ($\alpha \in \{0.1, 0.2, 0.4, 1.0\}$), averaged over five seeds. On both datasets, Stage 4 accuracy measured against teacher pseudo-labels consistently matches or exceeds accuracy measured against true labels at every label ratio. The gap between the two targets is largest at $\alpha = 0.1$, where the teacher's own accuracy is lowest and its pseudo-labels diverge most from true labels. As α increases and the teacher improves, the two reference targets converge and the gap narrows.

Figure 16 shows the same analysis on MIT-BIH using Macro-F1. The pattern is consistent: pseudo-label Macro-F1 exceeds true-label Macro-F1 at all label ratios. The absolute gap is smaller than on XJTU and NATOPS because the Stage 1 teacher already achieves Macro-F1 above 0.94 at $\alpha = 0.1$, leaving limited room for the two targets to diverge.

These results confirm that Stage 4 knowledge distillation achieves its intended objective: the student reproduces the teacher's decision boundary through gated input. This alignment is a prerequisite for the verification claim of the Sensory Gating framework, since faithfulness is defined as consistency between the gated student and the teacher. The Stage 1 true-label accuracy included in both figures further shows that the student matches or exceeds the teacher's performance, indicating that the gating pipeline does not degrade classification accuracy.

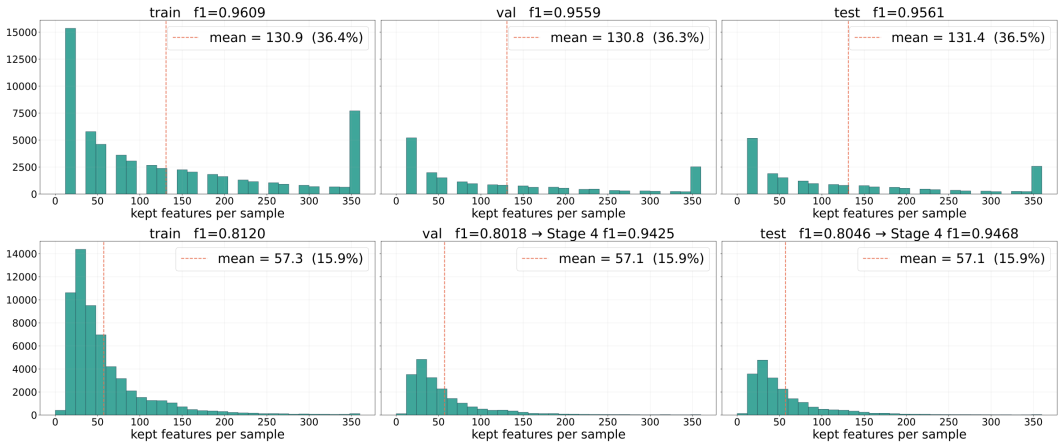

Figure 11: **Distribution of kept cells per sample on MIT-BIH ($T \cdot C = 360$, $\alpha = 0.2$).** Left: greedy selection (Stage 2) produces a broad distribution. Right: the hard gate (Stage 3) compresses the distribution toward lower kept counts. Dashed lines indicate the per-sample mean. After KD refinement and verification (Stage 4), macro-F1 recovers to a level comparable with Stage 1.

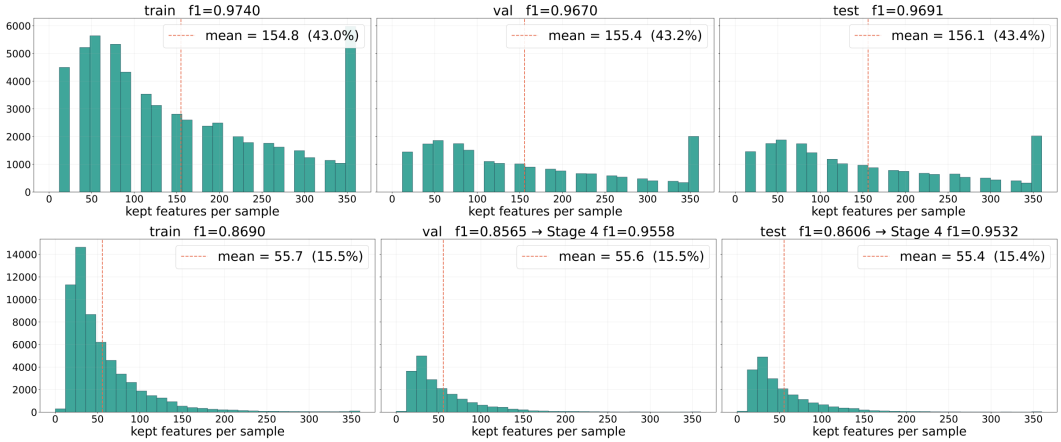

Figure 12: **Distribution of kept cells per sample on MIT-BIH ($T \cdot C = 360$, $\alpha = 0.4$).** Left: greedy selection (Stage 2) produces a broad distribution. Right: the hard gate (Stage 3) compresses the distribution toward lower kept counts. Dashed lines indicate the per-sample mean. After KD refinement and verification (Stage 4), macro-F1 recovers to a level comparable with Stage 1.

F.4 Matched-Sparsity Comparisons

We compare two simpler mask sources at the same feature budget: random selection and top-k TIMING attribution computed on each test instance. Both retain $k = \rho D$ features, where ρ is the retention ratio and D is the number of input features.

Each baseline has two variants. The *global* variant applies a uniform retention ratio, equal to the average across instances, to every instance. The *matched* variant uses the retention ratio assigned by greedy selection to each instance, giving the baseline access to an instance-specific budget. Both baselines are evaluated on the frozen teacher. At the end of this section, we also pass the matched masks through the same KD pipeline as our method (Table 24).

Table 21 compares the four baseline configurations with our deployed gate and KD student across nine datasets, four label ratios, and five seeds. Utility/ceiling denotes predictive utility under masking divided by the corresponding full-input teacher's utility. This comparison evaluates the deployed system against simpler masks, including the contribution of student training.

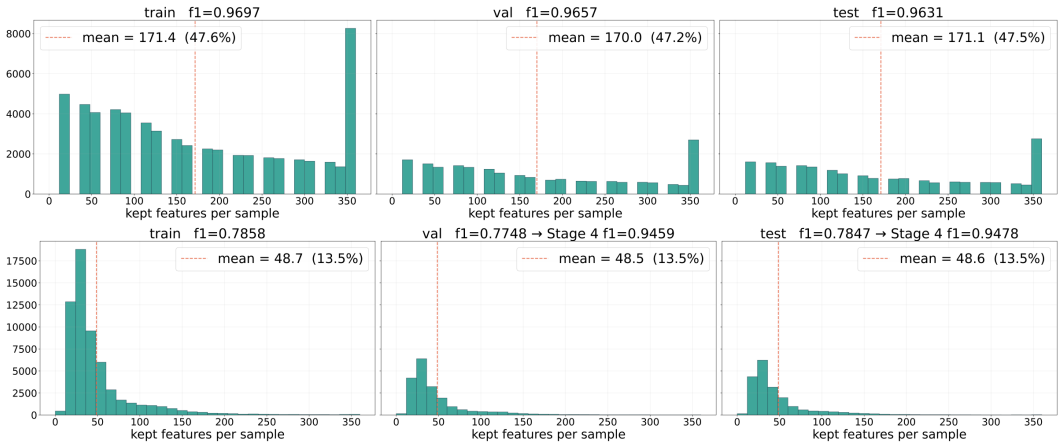

Figure 13: **Distribution of kept cells per sample on MIT-BIH ($T \cdot C = 360$, $\alpha = 1.0$).** Left: greedy selection (Stage 2) produces a broad distribution. Right: the hard gate (Stage 3) compresses the distribution toward lower kept counts. Dashed lines indicate the per-sample mean. After KD refinement and verification (Stage 4), macro-F1 recovers to a level comparable with Stage 1.

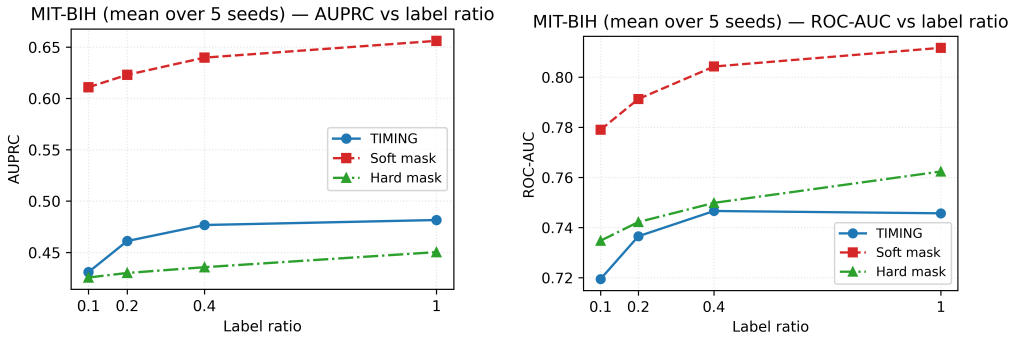

Figure 14: **Explanation quality vs. label ratio on MIT-BIH (mean over 5 seeds).** Left: AUPRC; right: ROC-AUC. All three attribution variants degrade as α decreases, confirming that explanation quality depends on the base classifier's representation quality. The soft mask consistently outperforms TIMING and the hard mask at every ratio, indicating that the BCE denoising effect is robust to label scarcity.

Random masks preserve substantially less utility than either our deployed system or top-k TIMING. Their utility-to-ceiling ratios are 0.574 and 0.564, compared with 0.943 and 0.945 for top-k TIMING and 0.992 for our system. Top-k TIMING preserves much of the teacher's utility, which is consistent with our gate being trained to imitate TIMING-derived greedy masks. We do not claim that the gate selects better features than its own supervision. The framework instead learns a direct mapping from inputs to masks for use with the deployed student.

Table 22 presents the matched comparison on three representative datasets.

The ordering of the reported means is consistent across these three datasets: our deployed system performs above top-k TIMING, and both outperform random selection. The gap to top-k TIMING is small on CricketX and MIT-BIH and larger on XJTU.

Table 23 examines budget allocation by comparing the two TIMING variants at the same average retention ratio.

Instance-specific budgets improve TIMING's mean utility on these datasets, consistent with the number of required features varying across instances. However, the aggregate improvement is small (0.945 versus 0.943), and random masks do not benefit from matched budgets. We therefore report this as a descriptive trend rather than a general advantage of instance-specific allocation.

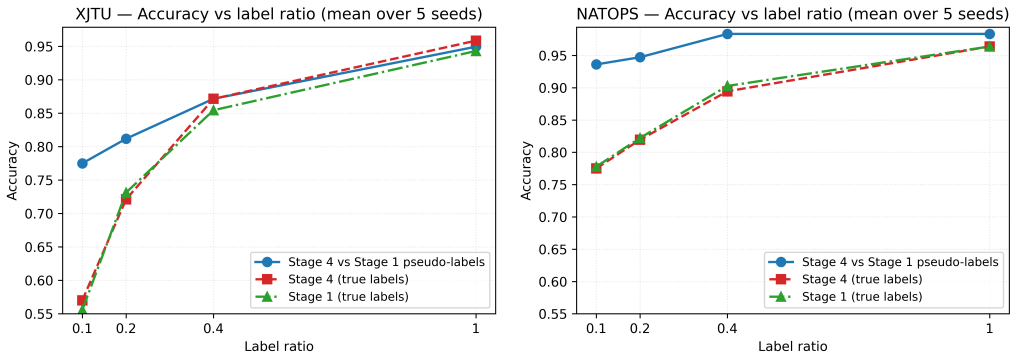

Figure 15: KD alignment verification on XJTU and NATOPS (mean over 5 seeds). Stage 4 student accuracy measured against Stage 1 teacher pseudo-labels (blue) matches or exceeds accuracy measured against true labels (red) at all label ratios, confirming that knowledge distillation successfully aligns the student with the teacher. Stage 1 true-label accuracy (green) is shown for reference.

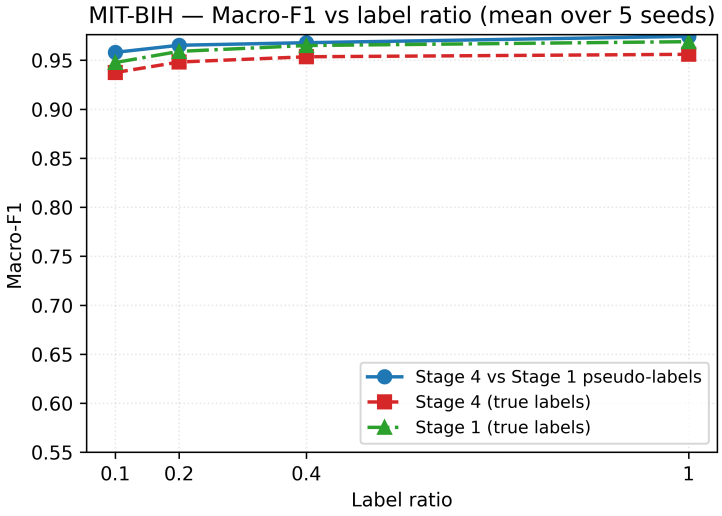

Figure 16: KD alignment verification on MIT-BIH (mean over 5 seeds). Stage 4 student Macro-F1 measured against Stage 1 teacher pseudo-labels (blue) consistently exceeds Macro-F1 measured against true labels (red) at all label ratios, confirming that knowledge distillation aligns the student with the teacher. Stage 1 true-label Macro-F1 (green) is shown for reference.

The approaches also differ in explanation cost at deployment. Top-k TIMING requires attribution computation for every test instance, whereas our gate produces a mask in a single forward pass. The per-instance explanation costs reported in Appendix Table 3 quantify this difference and support the use of the trained gate for efficient mask generation at deployment.

Matched masks under the same KD pipeline. We pass the matched top-k TIMING and matched random masks through the same KD pipeline as our method, with the same Stage-1 teacher, distillation objective, schedule, and seeds, so that only the mask source differs. Table 24 reports the results on CricketX and XJTU. Random masks fall behind our method on XJTU at every label ratio, with a mean gap of 9.4 % (68.62% versus 78.03%), so distillation alone does not recover the accuracy: which features are kept matters. On CricketX, the gap is 1.9 %, within the seed standard deviations of 2 to 8 %. Top-k TIMING masks under the same pipeline perform comparably to our gate: our method is ahead in four of the eight dataset-ratio settings and behind in the other four.

Table 21: Mean utility relative to the full-input teacher across nine datasets, four label ratios, and five seeds. Mask baselines are evaluated on the frozen teacher, while our deployed system uses the gate and KD student.

Mask / method	Mean utility/ceiling
Gate + KD student (ours, deployed)	$\mathbf{0.992}$
Top-k TIMING—global	0.943
Top-k TIMING—matched	0.945
Random selection—global	0.574
Random selection—matched	0.564

Table 22: Mean results on three representative datasets using the matched baseline variants. $\bar{\rho}$ denotes the mean retained input percentage. TIMING and random masks are evaluated on the frozen teacher.

Dataset	Metric	$\bar{\rho}$ (%)	Ceiling	Ours	TIMING	Random	Ours/ceiling
XJTU	Acc. (%)	39.2	77.1	$\mathbf{78.0}$	67.2	22.0	1.012
CricketX	Acc. (%)	39.4	65.7	$\mathbf{63.3}$	63.0	32.4	0.963
MIT-BIH	Macro-F1	41.0	0.960	$\mathbf{0.960}$	0.956	0.672	1.000

F.5 What Distillation Shows

Student–teacher agreement shows that a new model can reproduce the teacher's predictions from gated inputs. It does not establish that the teacher relied on the same selected features. We therefore distinguish direct tests on the frozen teacher from the input restriction imposed on the deployed student.

Direct evaluation on the frozen teacher. We freeze the Stage-1 teacher and generate binary masks with the sensory gate at threshold κ^*. No student or retraining is involved in this evaluation. For each original input $\mathbf{x}$ and its gate mask $\mathbf{M}$, we construct

$$\tilde{\mathbf{x}}_{\text{keep}} = \mathbf{M} \odot \mathbf{x} + (\mathbf{1} - \mathbf{M}) \odot \bar{\mathbf{x}}, \qquad \tilde{\mathbf{x}}_{\text{delete}} = (\mathbf{1} - \mathbf{M}) \odot \mathbf{x} + \mathbf{M} \odot \bar{\mathbf{x}},$$

where $\bar{\mathbf{x}}$ is the training-mean replacement. The keep input retains only the selected features, while the delete input replaces only those features.

Let U_{full}, U_{keep}, and U_{delete} denote the teacher's predictive utility on the original, keep, and delete inputs, respectively. We define the floor, U_{floor}, as its utility when every feature is replaced by the training mean. For $U_{\text{full}} > U_{\text{floor}}$, we compute

$$\text{Retention} = \frac{U_{\text{keep}} - U_{\text{floor}}}{U_{\text{full}} - U_{\text{floor}}}, \qquad \text{Necessity} = \frac{U_{\text{full}} - U_{\text{delete}}}{U_{\text{full}} - U_{\text{floor}}}.$$

Retention measures the fraction of above-floor utility preserved by the selected features. Necessity measures the fraction lost when those features are removed. Higher values provide evidence of predictive sufficiency and necessity under this replacement procedure.

Table 25 aggregates these quantities over the full experimental grid and compares gate masks with random masks containing the same number of selected features. Both quantities are referenced to the floor and therefore differ from the raw utility-to-ceiling ratios in Appendix F.4.

Gate-selected features preserve most of the teacher's above-floor utility, and their removal eliminates most of that utility. Random selections of the same size produce substantially smaller effects. These results support the relevance of the selected locations beyond the number of retained features, within the tested perturbation setting.

Table 26 reports the same evaluation by label ratio on XJTU. Retention ranges from 0.76 to 0.90, while necessity remains between 0.85 and 0.86.

The deployed gate and student. At inference, the system returns a student prediction, a soft mask, and a binary mask, as illustrated in Figure 22. The student receives $\tilde{\mathbf{x}}_{\text{keep}}$, so, for a fixed binary mask, changing the values of excluded features cannot change its input or prediction. This restriction on the student's direct input holds by construction. However, the gate computes the mask from the full series,

Table 23: Top-k TIMING with global and instance-specific matched budgets at the same average retention ratio. Both variants are evaluated on the frozen teacher.

Dataset	Metric	Global	Matched
XJTU	Accuracy (%)	64.3	**67.2**
CricketX	Accuracy (%)	62.6	**63.0**
MIT-BIH	Macro-F1	0.948	**0.956**

Table 24: Matched masks under the same KD pipeline as our method. Accuracy (%), mean $\pm$ standard deviation over five seeds. All students share the Stage-1 teacher, distillation objective, schedule, and seeds; only the mask source differs.

Dataset	α	Ours	Top-k TIMING + KD	Random + KD
CricketX	0.1	47.05 ± 2.85	48.46 ± 3.81	46.79 ± 1.63
	0.2	61.54 ± 6.38	61.28 ± 8.73	58.72 ± 7.83
	0.4	70.26 ± 3.41	72.69 ± 2.11	68.97 ± 2.06
	1.0	74.23 ± 2.00	77.56 ± 3.11	71.15 ± 3.11
	Mean	63.27	65.00	61.41
XJTU	0.1	57.01 ± 5.16	55.79 ± 3.51	48.85 ± 3.59
	0.2	72.11 ± 4.31	72.27 ± 3.53	61.55 ± 2.07
	0.4	87.15 ± 2.99	85.39 ± 3.05	77.39 ± 7.90
	1.0	95.84 ± 3.06	93.76 ± 3.39	86.67 ± 5.78
	Mean	78.03	76.80	68.62

so excluded features may still influence mask generation and thereby affect the complete system. The structural guarantee therefore concerns the student's direct input conditional on the mask; it does not by itself establish the faithfulness of the soft attribution ranking. The teacher supplies training targets but is not required at deployment, and Figures 6 to 9 compare the student's predictive utility with that of the teacher.

A tighter-budget evaluation. Table 27 repeats the frozen-teacher test on MIT-BIH at a retained fraction of approximately 15%. The teacher's keep-gate utility remains below its full-input ceiling, with retention between 0.79 and 0.83 and necessity between 0.70 and 0.72.

The KD student recovers utility closer to the ceiling, as reported in Figure 19. This is consistent with evidence that models trained on heavily masked inputs can retain substantial predictive accuracy [14]. That work also shows that masking can introduce distribution shifts, making it difficult to separate the effect of removed information from the frozen model's response to unfamiliar inputs. The frozen-teacher scores should therefore be interpreted as perturbation diagnostics, rather than guaranteed lower bounds on the information available in the selected features.

Training the student on gated inputs allows it to adapt to the input distribution used at deployment. Distillation provides the teacher's predictions as learning targets, while the binary mask restricts the student's direct access to input values. These roles are distinct: distillation supports predictive utility, and masking enforces the input restriction. Neither student–teacher agreement nor this restriction alone establishes how the original teacher reached its predictions.

G Additional Ablation Studies

Target sparsity sweep on XJTU. Figure 18 reports per-stage accuracy on XJTU at $\alpha = 0.4$ across seven target sparsity budgets $s_{\max} \in \{0.15, 0.25, 0.3, 0.35, 0.5, 0.6, 0.7\}$. Stage 3 accuracy decreases monotonically as $s_{\max}$ tightens, with the steepest drops below $s_{\max} = 0.3$, where the gate retains too few features for the frozen classifier to maintain its predictions. Stage 4 (KD refinement and verification) recovers accuracy at every sparsity level. At $s_{\max} \geq 0.35$, Stage 4 recovers to within 2–3 percentage points of the Stage 1 baseline (85.55%). At the most aggressive setting ($s_{\max} = 0.15$), Stage 3 drops to approximately 22%, yet KD still recovers roughly 40 percentage points, reaching approximately 60%. A residual gap to Stage 1 remains at this extreme sparsity, indicating that the gate retains too few features for full recovery. These results suggest that $s_{\max} \geq 0.35$ provides a practical operating range on XJTU, balancing sparsity and accuracy preservation.

Table 25: Retention and necessity on the frozen teacher, averaged over the full experimental grid. Random masks contain the same number of selected features as the corresponding gate masks. Both quantities are referenced to the training-mean floor.

Quantity	Gate	Random
Retention ↑	$\mathbf{0.893}$	0.447
Necessity ↑	$\mathbf{0.804}$	0.228

Table 26: Frozen-teacher evaluation on XJTU. Predictive utility is accuracy (%), and keep/delete results are mean ± standard deviation over five seeds. ρ is the mean retained input percentage. Ret. and Nec. denote retention and necessity.

α	ρ (%)	Ceiling	Keep-gate	Delete-gate	Floor	Ret. ↑	Nec. ↑
0.1	39.3	55.6	$\mathbf{47.1 \pm 6.0}$	$\mathbf{13.8 \pm 3.1}$	6.7	0.83	0.85
0.2	33.5	73.1	$\mathbf{56.9 \pm 5.9}$	$\mathbf{16.8 \pm 2.9}$	6.7	0.76	0.85
0.4	41.5	85.4	$\mathbf{77.6 \pm 6.6}$	$\mathbf{17.8 \pm 4.6}$	6.7	0.90	0.86
1.0	42.6	94.3	$\mathbf{83.7 \pm 5.8}$	$\mathbf{19.9 \pm 2.5}$	6.7	0.88	0.85

H Hyperparameter Sensitivity

The only hyperparameter that directly controls the sparsity-accuracy trade-off at inference is the target sparsity budget $s_{\max}$. We sweep $s_{\max}$ on two datasets: XJTU (Figure 18), which represents the eight datasets without explanation ground truth, and MIT-BIH (Figure 19), where ground-truth saliency is available.

On XJTU, Stage 4 consistently outperforms Stage 3 at every sparsity level. At $s_{\max} \geq 0.35$, Stage 4 recovers to within 2 percentage points of the Stage 1 baseline (85.55%). Below $s_{\max} = 0.3$, Stage 3 accuracy drops steeply, yet KD still provides substantial recovery: at $s_{\max} = 0.15$, Stage 4 reaches approximately 60% compared to 22% for Stage 3. The gap between Stage 4 and Stage 1 at extreme sparsity indicates that the gate retains too few features for full recovery, but KD remains effective as a partial corrector even in this regime.

On MIT-BIH, the sensitivity to $s_{\max}$ is minimal. Stage 4 recovers macro-F1 to within 2 percentage points of the Stage 1 baseline (0.965) at both $s_{\max} = 0.15$ and $s_{\max} = 0.2$, while Stage 3 remains below 0.83 at both settings. This stability reflects the concentrated nature of diagnostically relevant features in ECG data: a small fraction of the input suffices for classification, and further relaxing $s_{\max}$ provides diminishing returns.

Together, the two datasets illustrate complementary regimes. XJTU exhibits a smooth accuracy-sparsity trade-off where $s_{\max}$ must be tuned to balance explanation compactness against utility. MIT-BIH is robust to $s_{\max}$ within the tested range, suggesting that when ground-truth saliency is localized, the framework is insensitive to this hyperparameter.

I Qualitative Visualization

Figures 20–23 show the same MIT-BIH test sample across all four label ratios ($\alpha \in \{0.1, 0.2, 0.4, 1.0\}$) at $s_{\max} = 0.2$. Each figure displays four rows: the raw ECG with ground-truth saliency (teal shading), the TIMING attribution map, the soft gate output, and the hard gate mask (pink shading marks selected regions).

Across all ratios, the soft gate consistently sharpens the TIMING attributions, concentrating high-attribution values around the ground-truth region while suppressing noise elsewhere. The hard gate retains only the most salient segments, and the selected regions overlap with the ground-truth annotation at every label ratio. The calibrated threshold κ^* varies with α: lower label ratios yield a lower threshold ($\kappa^* = 0.60$ at $\alpha = 0.1$) while higher ratios produce a tighter threshold ($\kappa^* = 0.70$ at $\alpha = 0.4$ and $\alpha = 1.0$), reflecting that stronger base classifiers support more confident gate outputs. Despite this variation, the spatial location of the selected segments remains stable, indicating that the gate identifies the same diagnostically relevant region regardless of the supervision level.

Table 27: Frozen-teacher evaluation on MIT-BIH at an approximately 15% retention budget. Predictive utility is macro-F1. ρ is the mean retained input percentage; Ret. and Nec. denote retention and necessity.

α	ρ (%)	Ceiling	Keep-gate	Delete-gate	Floor	Ret. ↑	Nec. ↑
0.1	15.0	0.948	$\mathbf{0.806 \pm 0.030}$	$\mathbf{0.407 \pm 0.026}$	0.171	0.82	0.70
0.2	15.0	0.959	$\mathbf{0.795 \pm 0.026}$	$\mathbf{0.393 \pm 0.059}$	0.171	0.79	0.72
0.4	14.9	0.965	$\mathbf{0.826 \pm 0.034}$	$\mathbf{0.412 \pm 0.043}$	0.171	0.82	0.70
1.0	15.0	0.969	$\mathbf{0.832 \pm 0.019}$	$\mathbf{0.403 \pm 0.030}$	0.171	0.83	0.71

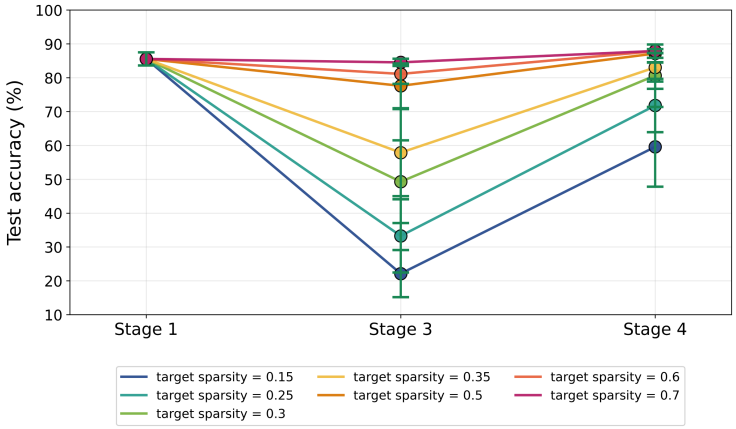

Figure 17: **Per-stage accuracy ablation on XJTU at $\alpha = 0.4$ across target sparsity budgets $s_{\max}$.** Stage 3 accuracy decreases as $s_{\max}$ tightens, with the steepest drops below $s_{\max} = 0.3$. Stage 4 (KD refinement and verification) recovers accuracy across all sparsity levels, with configurations above $s_{\max} = 0.35$ recovering to within 2–3 percentage points of Stage 1. Even at aggressive sparsity ($s_{\max} = 0.15$), KD recovers approximately 40 percentage points from Stage 3, though a gap to Stage 1 remains.

J Computational Cost

J.1 Compute Resources

All experiments were conducted on a single workstation with the following specifications.

Hardware. The workstation is a Dell Precision 5820 Tower equipped with an Intel Core i9-10900X CPU (10 cores, 20 threads, 3.70 GHz base, 4.70 GHz boost) and 64 GB RAM. Two NVIDIA RTX A5000 GPUs (24 GB VRAM each, Ampere architecture, compute capability 8.6) provide 48 GB total GPU memory. Storage is a single 1 TB NVMe SSD.

Software. The system runs Ubuntu 24.04.3 LTS with Linux kernel 6.8.0. All experiments use PyTorch 2.11.0 with CUDA 12.6. Each experiment uses a single GPU; the two GPUs are used to run independent seeds in parallel.

J.2 Inference Time

Table 28 reports single-instance forward latency (batch size 1) averaged over 5 timed runs. SupCE, f-VAT, and F-JEPA share the same inference architecture (TCNEncoder + ClassifierHead), since f-VAT's adversarial perturbation and F-JEPA's EMA encoder are used only during training. The Sensory Gating framework (Ours) adds the gate forward pass and mean-fill operation on top of the same student classifier. The overhead ranges from 0.30–0.56 ms per instance across datasets, corresponding to a 6–22% increase in latency. This overhead is negligible in practice: even on the slowest dataset (Heartbeat, 5.52 ms), inference remains well below 6 ms per instance.

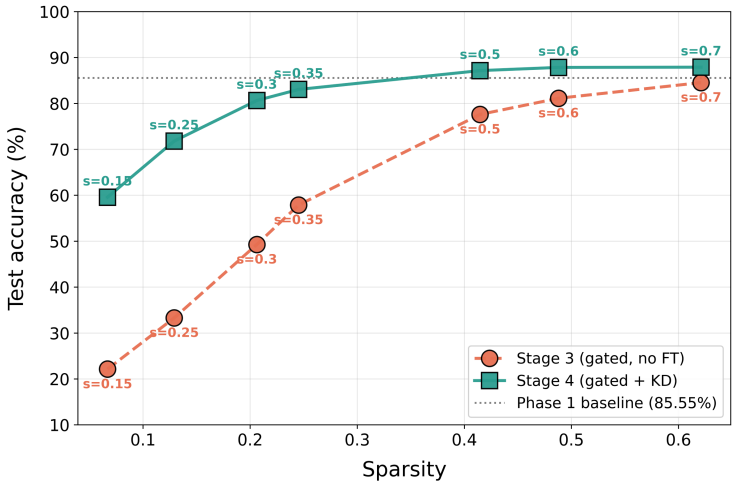

Figure 18: **Accuracy vs. sparsity on XJTU at $\alpha = 0.4$ across target sparsity budgets $s_{\max}$.** Stage 4 (KD refinement and verification) consistently outperforms Stage 3 (no fine-tuning) at every sparsity level. At $s_{\max} \geq 0.35$, Stage 4 recovers to within 2 percentage points of the Stage 1 baseline (85.55%, dotted line).

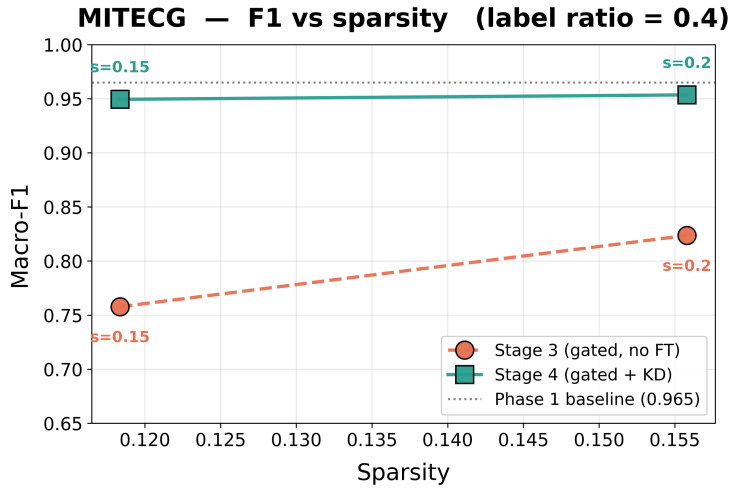

Figure 19: **Macro-F1 vs. sparsity on MIT-BIH at $\alpha = 0.4$ across target sparsity budgets $s_{\max}$.** Stage 4 (KD refinement and verification) recovers macro-F1 to within 2 percentage points of the Stage 1 baseline (0.965, dotted line) at both $s_{\max} = 0.15$ and $s_{\max} = 0.2$, while Stage 3 remains below 0.83.

J.3 Training Time

Table 29 reports total wall-clock training time for 1000 epochs at $\alpha = 0.4$, averaged over 5 runs. f-VAT, F-JEPA, and Ours (Sensory Gating) iterate over the full dataset, so training time is constant across label ratios; SupCE iterates the labeled subset directly and scales with α. Ours includes all four stages (Stage 1: F-JEPA training, Stage 2: TIMING attribution + greedy selection, Stage 3: gate training, Stage 4: KD refinement and verification).

F-JEPA is 1.6–6.1$\times$ faster than f-VAT across datasets, because f-VAT requires power iteration for adversarial perturbation computation (additional forward and backward passes per step), while F-JEPA's EMA encoder requires only a gradient-free forward pass. The full Sensory Gating pipeline adds 15–97% overhead on top of F-JEPA (Stages 2–4), but remains 1.1–2.9$\times$ faster than f-VAT on all

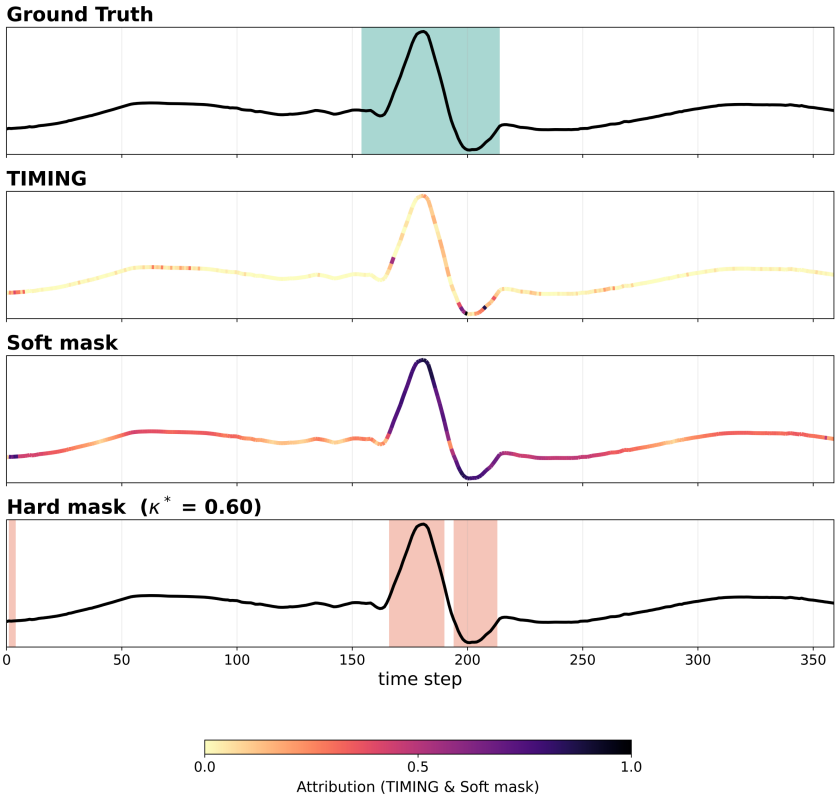

Figure 20: **Qualitative example on MIT-BIH at $\alpha = 0.1$, $\kappa^* = 0.60$, $s_{\max} = 0.2$.** The soft gate sharpens TIMING attributions around the ground-truth region (teal). The hard gate retains only the most salient segments (pink shading).

Table 28: **Inference latency per instance (ms, batch size 1).** SupCE, f-VAT, and F-JEPA share the same inference module. Ours adds the gate forward pass and mean-fill perturbation. Overhead is 0.30–0.56 ms (6–22%) across datasets.

Dataset	SupCE / f-VAT / F-JEPA (ms)	Ours (ms)	Overhead (ms)	Overhead (%)
CricketX	2.52	3.08	+0.56	22.2
EpilepticSeizure	2.69	2.99	+0.30	11.2
Heartbeat	5.07	5.52	+0.45	8.9
InsectWingbeatSound	2.59	3.03	+0.44	17.0
MFPT	2.52	2.86	+0.34	13.5
MIT-BIH	2.52	3.02	+0.50	19.8
NATOPS	2.56	2.97	+0.41	16.0
UWaveGestureLibraryAll	2.63	2.96	+0.33	12.5
XJTU	2.60	2.91	+0.31	11.9

datasets except EpilepticSeizure, where the two are comparable. On MIT-BIH (the largest dataset), the full pipeline completes in approximately 10.8 hours, compared to 19.9 hours for f-VAT.

J.4 Explanation Time

Table 30 compares per-sample explanation time on MIT-BIH. Stage 4 adds only 0.309 ms over Stage 1, corresponding to the forward pass through the sensory gate g_ω. This overhead is negligible compared to the 17.439 ms required by TIMING at batch size $B{=}1$, which computes gradients through the full encoder to produce a single attribution map. The Sensory Gating framework amortizes the cost of explanation: the gate produces a binary mask in a single forward pass at inference, avoiding the per-instance gradient computation that makes post-hoc attribution expensive at deployment.

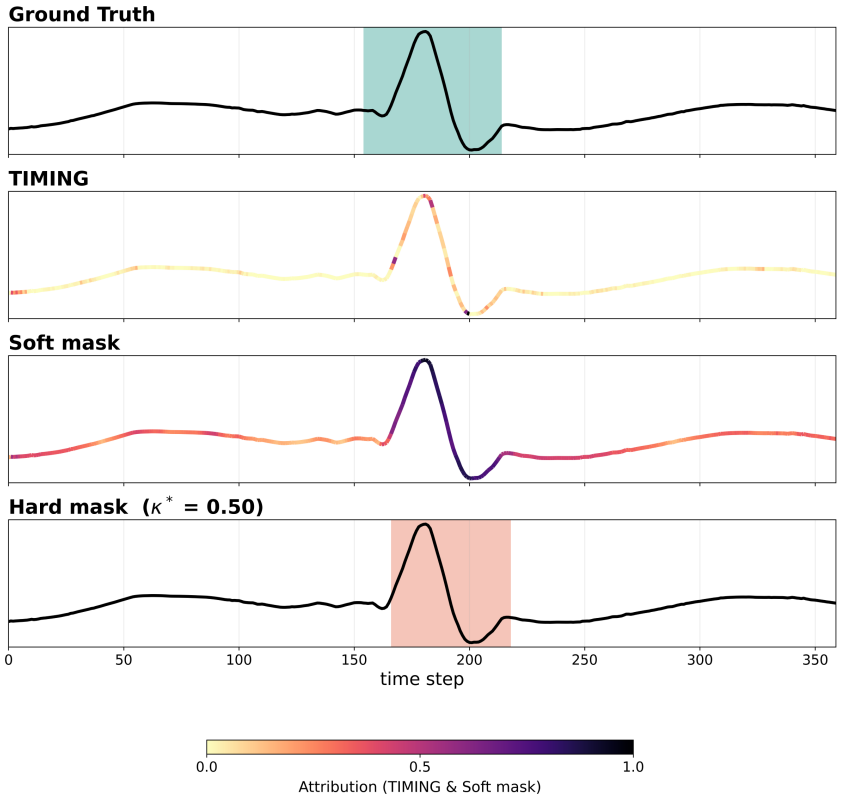

Figure 21: **Qualitative example on MIT-BIH at $\alpha = 0.2$, $\kappa^* = 0.50$, $s_{\max} = 0.2$.** The soft gate sharpens TIMING attributions around the ground-truth region (teal). The hard gate retains only the most salient segments (pink shading).

Table 29: **Training time (seconds) for 1000 epochs at $\alpha = 0.4$.** Ours includes all four pipeline stages. F-JEPA is consistently faster than f-VAT due to the absence of adversarial perturbation. The full Sensory Gating pipeline remains faster than f-VAT on 8 of 9 datasets.

Dataset	SupCE (s)	f-VAT (s)	F-JEPA (s)	Ours (s)
CricketX	49.5	502.4	321.0	414.6
EpilepticSeizure	628.4	6,445.8	4,803.0	6,077.4
Heartbeat	1,588.5	22,338.8	3,655.8	7,208.1
InsectWingbeatSound	124.9	1,255.9	894.3	1,174.1
MFPT	408.0	4,365.1	1,344.5	2,130.6
MIT-BIH	6,739.0	71,529.9	38,105.5	38,771.2
NATOPS	73.7	730.3	148.3	185.1
UWaveGestureLibraryAll	646.4	6,941.2	2,202.4	3,461.0
XJTU	347.1	3,306.9	1,023.5	1,618.3

K Limitations

The Sensory Gating framework has several limitations that we acknowledge openly.

First, the pipeline is sequential: four stages must be executed in order, and each stage depends on the output of the previous one. This design prevents end-to-end training and introduces cumulative latency during the offline preparation phases (Stages 1–3), though Stage 4 inference requires only a single forward pass through the gate and student.

Second, the quality of the gate's explanation is upper-bounded by the quality of the post-hoc attribution method used in Stage 2. We use TIMING [18] throughout, but if TIMING produces noisy or unreliable

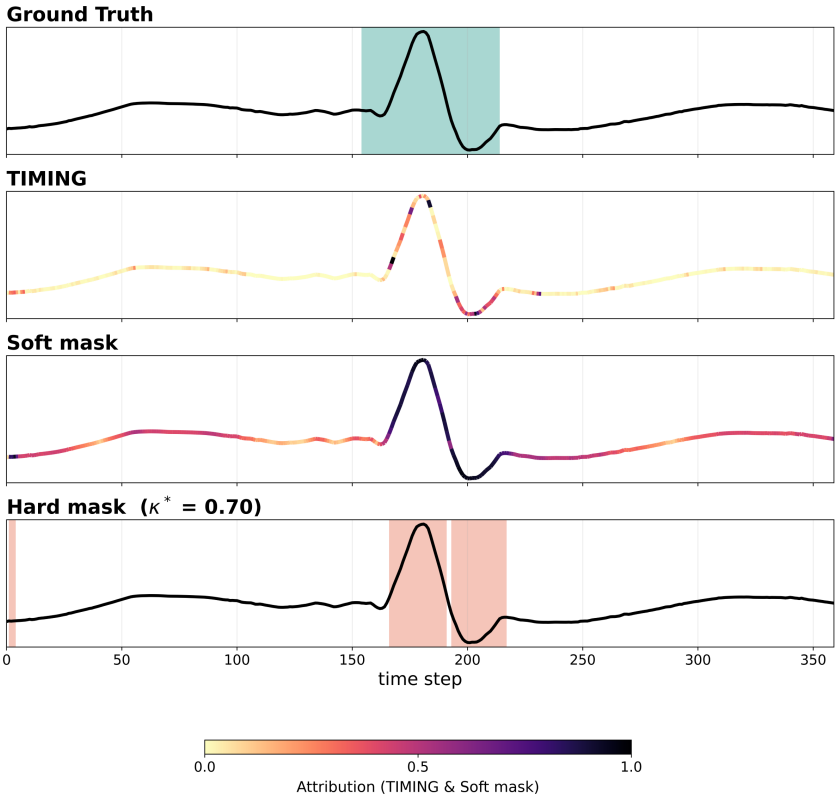

Figure 22: **Qualitative example on MIT-BIH at** $\alpha = 0.4$**,** $\kappa^* = 0.70$**,** $s_{\max} = 0.2$**.** The soft gate sharpens TIMING attributions around the ground-truth region (teal). The hard gate retains only the most salient segments (pink shading).

Table 30: Explanation time comparison (ms).

Method	Time (ms)
Stage 1	2.766
Stage 4	3.075
Explanation overhead (Stage 4 − Stage 1)	0.309
TIMING (B=1)	17.439

attributions for a given dataset, the greedy masks and downstream gate inherit those imperfections. Our AUPRC results on MIT-BIH show that BCE training partially denoises the raw attributions, but this recovery is not guaranteed in general.

Third, greedy selection (Stage 2) requires per-instance iterative evaluation, making it computationally heavier than a single forward pass. This cost is amortized at inference since the gate replaces greedy selection, but the offline preparation scales linearly with dataset size.

Fourth, the target sparsity budget $s_{\max}$ is dataset-dependent and must be selected by the practitioner. Our sensitivity analysis provides guidance on two datasets, but a principled method for choosing $s_{\max}$ without ground-truth saliency remains an open question.

L Broader Impact

This work addresses the intersection of semi-supervised learning and explainability for time series classification, with potential impact in safety-critical domains such as healthcare and industrial monitoring.

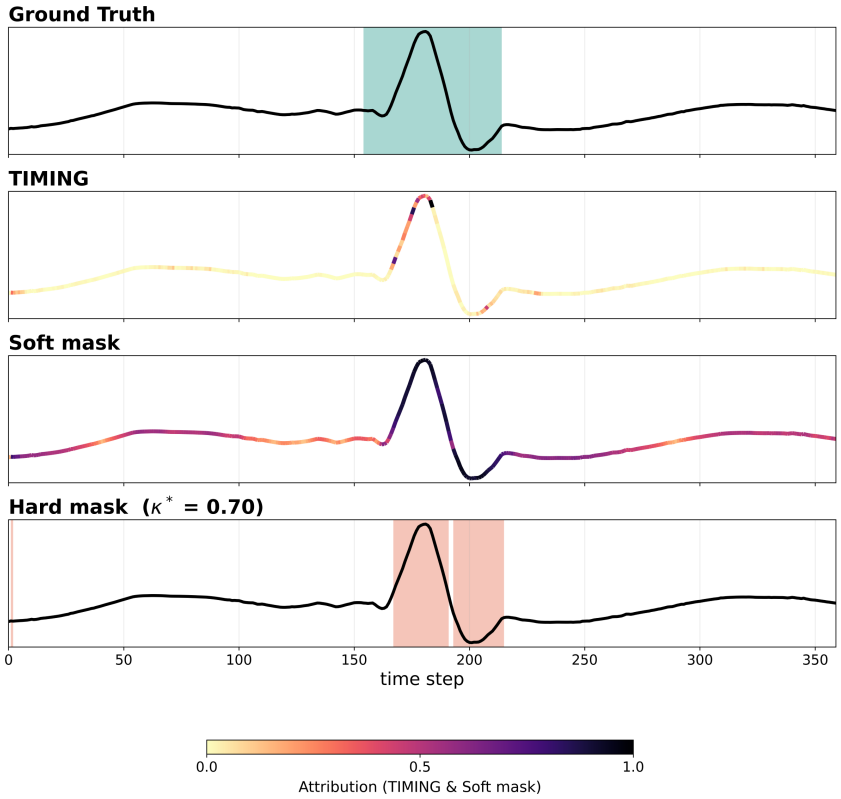

Figure 23: **Qualitative example on MIT-BIH at $\alpha = 1.0$, $\kappa^* = 0.70$, $s_{\max} = 0.2$.** The soft gate sharpens TIMING attributions around the ground-truth region (teal). The hard gate retains only the most salient segments (pink shading).

Positive impacts. By producing verified binary explanations at inference time, the framework enables practitioners to inspect which temporal features the model relies on for each prediction. In clinical settings, this transparency supports the requirements of the EU AI Act [45] by providing deployers with feature-level justification for model outputs. The per-instance variable sparsity further serves as a diagnostic signal: predictions that require unusually dense masks can be flagged for human review, adding a layer of safety beyond raw accuracy. The framework is architecture-agnostic and can be applied to existing trained classifiers without retraining the base model, lowering the barrier to adoption.

Potential risks. Gate-based explanations may create a false sense of interpretability if practitioners treat the selected features as causal rather than correlational. The gate identifies features that are sufficient for the classifier's prediction only. In high-stakes applications, domain experts should validate that the selected features align with established diagnostic criteria before relying on the explanations for decision-making. Additionally, the framework inherits biases present in the training data and the base classifier: if the classifier learns spurious correlations, the gate will faithfully select the spurious features and the KD refinement and verification stage will confirm them, since verification measures consistency with the teacher rather than correctness.

Intended use. This work is intended as a research contribution to the explainable AI community. We do not recommend deploying the framework in clinical or safety-critical systems without thorough validation by domain experts and compliance with applicable regulatory requirements.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state four contributions (sensory gate, KD refinement and verification, per-instance variable sparsity, F-JEPA), each supported by experiments in Section 4. Claims are scoped to the datasets and backbone tested (TCN on 9 datasets)

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix K discusses four limitations: sequential pipeline design, upper bound by TIMING attribution quality, computational cost of greedy selection, and dataset-dependent $s_{\max}$ selection.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results. The contribution is an empirical framework with design choices justified by experimental evidence rather than formal theorems. Background on TIMING (Section 2.2) references theoretical properties from the original paper [18].

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Appendix and Section 4 specifies all datasets (publicly available from UCR/UEA and SemiTime benchmarks), data splits (60/20/20), label ratios ($\alpha \in \{0.1, 0.2, 0.4\}$), backbone architecture (TCN), baselines, and evaluation metrics. Appendix reports all hyperparameters. The four-stage pipeline is fully described in main paper with loss functions (Eqs. 7–9). All results are averaged over 5 seeds with standard errors reported. Code will be released upon acceptance.

Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: All datasets are publicly available (UCR/UEA archive, SemiTime benchmark, MIT-BIH via TimeX). Anonymized code with reproduction scripts and environment specifications is included in the supplementary material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4.1 reports data splits (60/20/20), label ratios, backbone, baselines, and metrics. Appendix details all hyperparameters ($\lambda_{\text{F-JEPA}}$, λ_{hard}, τ, $s_{\max}$) and their selection rationale.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: All main results report mean ± standard error over 5 random seeds. The source of variability is random initialization and data split randomization.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix J.1 reports hardware specifications (Intel i9-10900X CPU, 2× NVIDIA RTX A5000 GPUs, 64 GB RAM). Appendix J.3 reports per-dataset training time and Appendix J.2 reports per-instance inference latency. The full research project required approximately 2–3× the reported compute due to preliminary experiments and hyperparameter exploration.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses only publicly available datasets, does not involve human subjects, and presents no foreseeable dual-use risks. Broader impact is discussed in Appendix L.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix L discusses positive impacts (verified explanations for safety-critical domains, per-instance trust signals), potential risks (false sense of interpretability, inheritance of classifier biases), and intended use scope.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper proposes an explainability framework for time series classification using publicly available benchmark datasets. It does not release pretrained models, scraped data, or artifacts with dual-use risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets are cited with their original publications. UCR/UEA datasets are cited via [9, 4], SemiTime benchmark datasets via [11], and MIT-BIH via [36]. TIMING is cited via [18]. All datasets are publicly available for research use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models. Code is provided in the supplementary material for reproduction purposes only.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects research. All experiments use publicly available benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only for writing assistance and editing. No LLM is a component of the proposed method or experimental pipeline.

    Guidelines:

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