Counterfactual Explanations for Time-Series Classification via Constrained Flow Matching

Akihiro Yamaguchi*
Toshiba Corporation

Shizuo Kaji†
Kyushu University

Kaname Matsue
Kyushu University

Ryusei Shingaki
Toshiba Corporation

Abstract

Counterfactual explanations for time-series classification generate synthetic instances that flip a prediction to a desired class while remaining plausible and making small, local changes. Existing methods often rely on classifier gradients, and many do not naturally extend to one-class settings. To bridge this gap, we propose CTCF, a model-agnostic method for hard-decision classifiers in supervised and one-class settings. CTCF decouples generation from classifier-specific control by reusing an unconditional Flow Matching generator and learning a flow-time-dependent *desired-class region* from interpolation states labeled by endpoint hard decisions, without classifier gradients or class probabilities. At inference time, CTCF uses a dual-control mechanism: fused-lasso endpoint-objective steering encourages small, segment-local changes, while linearized projection correction reduces violations of a conservative desired-class region. Theoretically, under ideal marginal matching, we show that the surrogate risk on interpolation states equals the corresponding ODE-trajectory risk. Experiments on UCR datasets demonstrate CTCF's effectiveness across quantitative metrics and qualitative case studies.

1 Introduction

Recent time-series classification (TSC) methods, including but not limited to neural network (NN) models, achieve high accuracy and efficiency [Middlehurst et al., 2024b, Dempster et al., 2024]. Yet their predictions are not always easy for humans to interpret. In safety- and accountability-critical applications, counterfactual (CF) explanations are therefore important: they show how an input should be changed to obtain a desired prediction [Wachter et al., 2018, Guidotti, 2022, Verma et al., 2024]. Standard requirements for CF quality include validity, proximity, sparsity, and plausibility. For time-series data, segment-wise locality is also important: changes should not spread over the whole sequence but should concentrate in a few contiguous intervals. Time-series CF evaluation therefore uses segment-count metrics such as NumSeg [Kan et al., 2024, Yamaguchi et al., 2026].

However, many existing CF time-series methods rely on classifier input gradients and thus assume differentiable classifiers [Wang et al., 2021, Filali Boubrahimi and Hamdi, 2022, Wang et al., 2024, Yamaguchi et al., 2024]. This assumption is restrictive on standard TSC benchmarks such as the UCR archive [Dau et al., 2019], where strong recent classifiers are not limited to end-to-end differentiable NN models [Middlehurst et al., 2024b, Dempster et al., 2024]. Some methods also require training instances at deployment time [Ates et al., 2021, Delaney et al., 2021, Bahri et al., 2024], which can conflict with data-governance or privacy constraints. Moreover, in one-class settings such as anomaly detection, CF generation is harder because supervised signals outside the desired class are limited.

*Corresponding author: akihiro.yamaguchi.h18@mail.toshiba, yamaguchi@imi.kyushu-u.ac.jp
†Shizuo Kaji is now at Kyoto University.

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

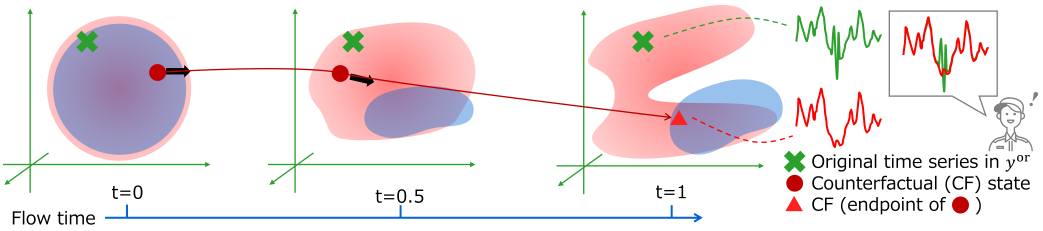

Figure 1: Inference-time control of CTCF. The desired-class region (blue) and probability distribution (pink) evolve over flow time. The red curve denotes the guided FM trajectory. A linearized projection correction is applied to an intermediate state (●), guiding the trajectory within the learned region. The corrected trajectory reaches a plausible counterfactual sample (▲) for the original time series (×), changing the classifier prediction from the original non-desired class y^{or} to the desired class y^{cf}.

To relax these restrictions, model-agnostic generative methods such as CVAE [Todo et al., 2023], LASTS [Spinnato et al., 2023], MCCE [Redelmeier et al., 2024], and CELT [Yamaguchi et al., 2026] have been proposed. Their main generation mechanisms are latent-space search [Spinnato et al., 2023], latent-space editing [Todo et al., 2023, Yamaguchi et al., 2026], or sample filtering [Redelmeier et al., 2024]. However, inference-time control of a pretrained generative model with explicit objectives and constraints remains underexplored for CF generation in TSC.

Such control requires sampling that can be modified during generation. Continuous-time generative models are well suited to this goal because they generate samples by evolving a state along a learned trajectory. FM and rectified flow provide simulation-free training and efficient ODE sampling [Lipman et al., 2023, Liu et al., 2023], while FlowChef and PCFM demonstrate inference-time steering and linearized projection correction for pretrained flow models [Patel et al., 2025, Utkarsh et al., 2025]. Continuous-time generative models have started to appear in time-series CF explanations, but existing studies are diffusion-based and target anomaly repair with a differentiable, linearly decomposable score [Ji et al., 2024] or block-maxima forecasting [Deng et al., 2024], not CFs for TSC. Thus, for classification-oriented time-series CF explanations, linking inference-time control of continuous-time generative models to decision-only black-box and one-class classifiers remains underexplored.

To address this gap, we propose **C**ounterfactual explanations for **T**ime-series **C**lassification via **C**onstrained **F**low Matching (CTCF), a model-agnostic counterfactual generation method for hard-decision and one-class TSC classifiers. CTCF decouples the reusable generator from classifier-specific control by using a pretrained unconditional FM model and, for each classifier and desired class, learning a flow-time-dependent *desired-class region*. This nonlinear region is trained to include desired-class samples and their interpolation states, while pushing non-desired-class samples and their interpolation states outside the region. Here, *flow time* is not a timestamp of the time series; it is the interpolation time along a probability path. At inference time, CTCF uses a dual-control mechanism. As shown in Figure 1, linearized projection correction reduces intermediate-state constraint violations; as shown in Figure 2, FlowChef-style steering with a fused lasso objective [Patel et al., 2025] encourages changes from the original time series to concentrate in a few contiguous segments. Together, these controls steer FM trajectories toward endpoints classified as the desired class while promoting proximity and segment-wise locality.

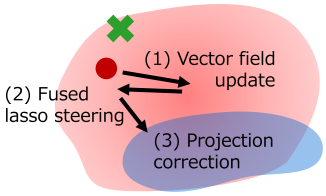

Figure 2: CTCF combines steering by a fused lasso objective with projection correction, handling locality of changes and region feasibility together.

Experiments on UCR time-series datasets [Dau et al., 2019] show that CTCF generates high-quality CFs in both supervised and one-class settings.

Our main contributions are as follows.

- We introduce a constrained-FM framework for CF generation in TSC that relies only on classifier hard decisions—without input gradients or class probabilities—and applies uniformly to supervised and one-class TSC classifiers.
- We learn a flow-time-dependent *desired-class region* and use it as an inference-time constraint.

- We prove that, under ideal marginal matching, the surrogate risk on efficiently sampled interpolation states equals the corresponding risk on ODE trajectories, justifying region training based on interpolation states.

2 Problem setup and preliminaries

2.1 Counterfactual generation for time-series classification

For a time series $\boldsymbol{x}^{\mathrm{or}} \in \mathbb{R}^L$ of length L, we aim to generate a CF $\boldsymbol{x}^{\mathrm{cf}} \in \mathbb{R}^L$ that flips the prediction of a classifier f from the original class y^{or} to the desired class y^{cf}. The classifier f may be either supervised or one-class. To support a broad range of classifiers, we query only hard decisions, not probabilities or other decision scores. We treat $f(\boldsymbol{x}) = y^{\mathrm{cf}}$ as desired and all other hard decisions, i.e., $f(\boldsymbol{x}) \neq y^{\mathrm{cf}}$, as non-desired; in one-class classification, this corresponds to "not the desired class." The supervised setting provides training instances from all classes, whereas the one-class setting provides mostly desired-class instances, making non-desired samples limited or unavailable.

We evaluate CF quality as follows. Validity indicates whether the generated CF is classified as the desired class, and proximity measures the amount of change from the original time series. Sparsity is the fraction of timestamps that receive only sufficiently small changes. Implausibility measures deviation from the data distribution and is computed as the nearest-neighbor distance to test data [Guidotti, 2022, Verma et al., 2024]. Because time-series changes should also concentrate in a few contiguous intervals, we use NumSeg, the number of substantially changed segments [Kan et al., 2024, Yamaguchi et al., 2026]. The metric definitions are given in Appendix A.

2.2 Unconditional Flow Matching

Our generative model is based on rectified-flow-style FM [Lipman et al., 2023, Liu et al., 2023]. FM is a simulation-free training framework for a continuous-time generative model that maps an initial distribution p_{init} to the data distribution p_{data} without numerical ODE integration during training. Throughout the paper, $t \in [0, 1]$ denotes *flow time*, not the time-series timestamp $l \in \{1, 2, \ldots, L\}$; it is the interpolation time on the probability path from p_{init} to p_{data}. Uppercase X_t denotes a random state, and lowercase $\boldsymbol{x}_t$ denotes its realization. We use $p_{\mathrm{init}} = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Rectified Flow defines the straight-line interpolation between independently sampled $X_0 \sim p_{\mathrm{init}}$ and $X_1 \sim p_{\mathrm{data}}$ as

$$X_t = (1 - t)X_0 + tX_1 \tag{1}$$

We call a realization $\boldsymbol{x}_t$ of this variable an *interpolation state*. Let p_t be the marginal distribution of X_t. Then $\{p_t\}_{t\in[0,1]}$ is a probability path that continuously evolves from p_{init} to p_{data}.

FM learns a neural vector field $\boldsymbol{u}_{\boldsymbol{\theta}} : \mathbb{R}^L \times [0, 1] \to \mathbb{R}^L$ that takes a state and flow time and outputs a velocity. Here, $\boldsymbol{\theta}$ denotes the trainable parameters of the vector-field network, and $\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is the velocity at state $\boldsymbol{x}_t$ and time t. Using the interpolation states in Eq. (1), the FM training objective is

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}), \quad \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) := \mathbb{E}_{t\sim\mathrm{Unif}[0,1],\, \boldsymbol{x}_0\sim p_{\mathrm{init}},\, \boldsymbol{x}_1\sim p_{\mathrm{data}}} \left[\|\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|_2^2 \right]. \tag{2}$$

Thus, training does not require ODE trajectories. Here, $\mathrm{Unif}[0, 1]$ denotes the uniform distribution over flow time. The key point is that $\boldsymbol{x}_t$ in Eq. (1) is not a generated state obtained by solving an ODE; it is an interpolation state directly constructed during training without ODE simulation. In contrast, the inference-time generated state Z_t is obtained as the solution of the ODE induced by the learned vector field $\boldsymbol{u}_{\boldsymbol{\theta}}$, so it does not correspond sample-wise to the interpolation state X_t. Under ideal training, however, solving this ODE from $Z_0 \sim p_{\mathrm{init}}$ makes the distribution of Z_t match the interpolation marginal distribution p_t at each time t. This marginal matching is the key fact used in Section 3.2 to connect the region-learning risk defined on interpolation states to the risk on ODE trajectories. In generation, we sample $\boldsymbol{z}_0 \sim p_{\mathrm{init}}$ and numerically solve the ODE defined by the learned $\boldsymbol{u}_{\boldsymbol{\theta}}$ from $t = 0$ to $t = 1$; the endpoint $\boldsymbol{z}_1$ is the generated sample.

We keep the generative model unconditional. That is, the vector field $\boldsymbol{u}_{\boldsymbol{\theta}}$ itself is not conditioned on the desired class; this information is supplied externally through the desired-class region learned in Section 3. This decoupling allows the same FM model to be reused independently of the classifier f to be explained.

3 Learning a desired-class region over flow time

We learn a desired-class region that varies over flow time. Intuitively, this extends DeepSVDD-style one-class representation learning [Ruff et al., 2018, Hojjati and Armanfard, 2024] to flow time. The region-learning risk is defined on interpolation states that can be directly constructed during training. As discussed in Section 2.2, interpolation states and inference-time ODE states do not correspond sample-wise, so Section 3.2 establishes their relation under marginal matching.

3.1 Optimization formulation

We define the desired-class region in feature space rather than directly in $\mathbb{R}^L \times [0,1]$. With the $\boldsymbol{w}$-parameterized feature-space mapping $\boldsymbol{\psi}_{\boldsymbol{w}} : \mathbb{R}^L \times [0,1] \to \mathbb{R}^d$, the region at flow time t is $\|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_t, t) - \boldsymbol{c}_t\|_2^2 \le r_{\boldsymbol{\nu}}^2(t)$, where $\boldsymbol{c}_t \in \mathbb{R}^d$ is the center and $r_{\boldsymbol{\nu}}^2(t) > 0$ is the $\boldsymbol{\nu}$-parameterized squared radius. In the time-independent case, this follows the DeepSVDD-style initialization: desired-class training instances are passed through the initial representation network, and their feature center is precomputed and fixed. In our flow-time setting, desired-class interpolation states may have different initial feature centers at different t. We therefore fit $\boldsymbol{c}_t$ before the main region training with a small regressor from t to these centers, and then keep it fixed. Because intermediate states $\boldsymbol{x}_t$ do not have independent class labels, we define desired and non-desired sides using the predicted label of the endpoint $\boldsymbol{x}_1$. Specifically, an interpolation state is treated as desired if $f(\boldsymbol{x}_1) = y^{\mathrm{cf}}$, and non-desired otherwise. This design handles supervised and one-class settings uniformly using only hard decisions, without classifier gradients or class probabilities. The time-dependent representation loss is

$$\ell_{\mathrm{rep}}(\boldsymbol{x}_t, t, \boldsymbol{x}_1) := \begin{cases} \|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_t, t) - \boldsymbol{c}_t\|_2^2, & \text{if } f(\boldsymbol{x}_1) = y^{\mathrm{cf}}, \\ \mathbb{I}(t \ge \tau) \max\big(0,\, r_{\boldsymbol{\nu}}^2(t) - \|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_t, t) - \boldsymbol{c}_t\|_2^2\big), & \text{if } f(\boldsymbol{x}_1) \ne y^{\mathrm{cf}}. \end{cases} \tag{3}$$

Here, $\mathbb{I}(\cdot)$ denotes the indicator function, which returns 1 if its argument is true and 0 otherwise. Thus, desired-side states are attracted to the center, whereas non-desired-side states are repelled only after the activation time τ. Near $t = 0$, $\boldsymbol{x}_t$ is dominated by class-agnostic $\boldsymbol{x}_0$; delaying repulsion avoids separating similar early states. If the training data contain no samples with $f(\boldsymbol{x}_1) \ne y^{\mathrm{cf}}$, only the first branch is used. The squared radius $r_{\boldsymbol{\nu}}^2(t)$ approximates the q-quantile of desired-side squared distances using the pinball loss $\rho_q(u) := \max(qu, (q-1)u)$ [Koenker and Bassett, 1978]:

$$\ell_{\mathrm{pin}}(\boldsymbol{x}_t, t, \boldsymbol{x}_1) := \mathbb{I}\big(f(\boldsymbol{x}_1) = y^{\mathrm{cf}}\big) \rho_q\big(\|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_t, t) - \boldsymbol{c}_t\|_2^2 - r_{\boldsymbol{\nu}}^2(t)\big). \tag{4}$$

The time-dependent desired-class region-learning loss is

$$\ell_{\mathrm{all}}(\boldsymbol{x}_t, t, \boldsymbol{x}_1) := \ell_{\mathrm{rep}}(\boldsymbol{x}_t, t, \boldsymbol{x}_1) + \lambda_{\mathrm{pin}}\, \ell_{\mathrm{pin}}(\boldsymbol{x}_t, t, \boldsymbol{x}_1), \tag{5}$$

and the final training problem is

$$\min_{\boldsymbol{w}, \boldsymbol{\nu}} \; \mathcal{L}_{\mathrm{DR}}(\boldsymbol{w}, \boldsymbol{\nu}),\; \mathcal{L}_{\mathrm{DR}}(\boldsymbol{w}, \boldsymbol{\nu}) := \mathbb{E}_{t \sim \mathrm{Unif}[0,1],\, \boldsymbol{x}_0 \sim p_{\mathrm{init}},\, \boldsymbol{x}_1 \sim p_{\mathrm{data}}} \big[\ell_{\mathrm{all}}(\boldsymbol{x}_t, t, \boldsymbol{x}_1)\big] + \frac{\lambda_{\mathrm{rep}}}{2} \|\boldsymbol{w}\|_2^2. \tag{6}$$

Implementation details are given in Appendix F.

3.2 Theoretical analysis

Equation (5) is a surrogate loss evaluated on interpolation states (X_t, X_1). Now consider the ODE induced by the learned vector field,

$$\frac{d}{dt} Z_t = \boldsymbol{u}_{\boldsymbol{\theta}}(Z_t, t), \qquad Z_0 \sim p_{\mathrm{init}}. \tag{7}$$

Because an ODE-trajectory state Z_t has no paired endpoint X_1, we marginalize the endpoint variable under $X_t = \boldsymbol{z}$ and define

$$\tilde{\ell}_{\mathrm{all}}(\boldsymbol{z}, t) := \mathbb{E}_{X_1}\big[\ell_{\mathrm{all}}(\boldsymbol{z}, t, X_1) \mid X_t = \boldsymbol{z}\big] \tag{8}$$

as a surrogate loss depending only on $(\boldsymbol{z}, t)$ and thus defined on ODE trajectories. The following theorem shows that, under marginal matching $Z_t \overset{d}{=} X_t$, the corresponding ODE-side risk equals the risk on interpolation states. The proof is in Appendix B.

Theorem 1. *Assume that ℓ_{all} is integrable and marginal matching $Z_t \overset{d}{=} X_t$ holds for $t \in [0,1]$. Then*

$$\mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\Big[\mathbb{E}_{Z_t}\big[\tilde{\ell}_{\mathrm{all}}(Z_t,t)\big]\Big] = \mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\Big[\mathbb{E}_{X_0,X_1}\big[\ell_{\mathrm{all}}((1-t)X_0+tX_1,t,X_1)\big]\Big]. \quad (9)$$

That is, the risk on interpolation states equals the risk on ODE trajectories.

For rectified flow, this identity $Z_t \overset{d}{=} X_t$ has been shown under idealized conditions that ignore finite-data, model-approximation, and numerical-integration errors [Liu et al., 2023]. Thus, Theorem 1 justifies simulation-free region learning based on interpolation states under ideal marginal matching. Under additional assumptions, the equality in Theorem 1 can be relaxed to an approximation-error bound without assuming perfect marginal matching; see Appendix C.

4 Constraint-preserving counterfactual generation

The desired-class region learned in the previous section gives a time-dependent constraint for checking whether states over flow time are consistent with intermediate states of the desired class, using only hard decisions from classifiers. At inference time, we use this region as a feasible set and steer the pretrained unconditional FM trajectory toward a CF endpoint. If only the endpoint is post hoc pushed toward the desired class, intermediate states can deviate from typical desired-class intermediate states without explicit penalty, and the generated trajectory may be inconsistent with the learned probability path of FM. CTCF therefore jointly handles small, local changes from the original time series and intermediate-state feasibility.

Optimization formulation. The endpoint $\boldsymbol{x}_1$ corresponding to a CF should stay close to the original time series $\boldsymbol{x}^{\mathrm{or}}$, and its changes should form contiguous segments rather than scattered timestamps. At the same time, each state $\boldsymbol{x}_t$ along the trajectory should remain inside the desired-class region at flow time t. We express these goals as the following path-level problem over a continuous-time path $\{\boldsymbol{x}_t\}_{t\in[0,1]}$ with CF endpoint $\boldsymbol{x}_1 \in \mathbb{R}^L$:

$$\min_{\{\boldsymbol{x}_t\}_{t\in[0,1]}} \quad \mathcal{L}_{\mathrm{GD}}(\boldsymbol{x}_1), \quad \mathcal{L}_{\mathrm{GD}}(\boldsymbol{x}) := \|\boldsymbol{D}(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{or}})\|_1 + \|\boldsymbol{x}-\boldsymbol{x}^{\mathrm{or}}\|_1 \quad (10)$$

$$\text{s.t.} \quad \|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_t,t)-\boldsymbol{c}_t\|_2^2 \le \gamma r_{\boldsymbol{\nu}}^2(t), \qquad \forall t\in[0,1]. \quad (11)$$

Here, $\boldsymbol{D} \in \mathbb{R}^{(L-1)\times L}$ is the first-difference matrix, and $\gamma \in (0,1]$ controls inference-time conservativeness. The objective is a fused-lasso endpoint objective on the perturbation $\boldsymbol{x}-\boldsymbol{x}^{\mathrm{or}}$ [Tibshirani et al., 2005]. The first term, $\|\boldsymbol{D}(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{or}})\|_1$, discourages high-frequency oscillations of the perturbation across adjacent timestamps, while the second term, $\|\boldsymbol{x}-\boldsymbol{x}^{\mathrm{or}}\|_1$, reduces the overall change size. Thus, the objective promotes segment-wise locality while maintaining proximity. The constraint in Eq. (11) requires each flow-time state to lie inside the feature-space hypersphere learned in Section 3.

Solution method. Rather than solving the constrained ODE optimization problem exactly, we combine FlowChef-style endpoint-objective steering [Patel et al., 2025] with PCFM-style linearized projection correction [Utkarsh et al., 2025]. Given the state $\boldsymbol{x}_t$ at time t, define the endpoint estimate as

$$\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t) = \boldsymbol{x}_t + (1-t)\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t). \quad (12)$$

The FlowChef-style prediction step is

$$\boldsymbol{x}_{t+\delta}^{\mathrm{pred}} = \boldsymbol{x}_t - \eta\nabla_{\boldsymbol{x}_t}\mathcal{L}_{\mathrm{GD}}(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)) + \delta\,\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t). \quad (13)$$

The last term is the Euler step of the pretrained FM model, and the middle term is local steering that decreases the endpoint objective in Eq. (10) for the predicted endpoint from the current state. For the ℓ_1 terms in $\mathcal{L}_{\mathrm{GD}}$, ∇ denotes a subgradient. This approximation gives an update direction for the endpoint objective without multi-step backpropagation through time. Next, if the prediction step leaves the desired-class region, we apply projection correction. Define the constraint violation as

$$e_t(\boldsymbol{x}) := \|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x},t)-\boldsymbol{c}_t\|_2^2 - \gamma r_{\boldsymbol{\nu}}^2(t). \quad (14)$$

If $e_{t+\delta}(\boldsymbol{x}_{t+\delta}^{\mathrm{pred}}) > 0$, define the violation gradient as

$$\boldsymbol{g}_{t+\delta} := \nabla_{\boldsymbol{x}} e_{t+\delta}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_{t+\delta}^{\mathrm{pred}}} = 2\left[\boldsymbol{J}_{\boldsymbol{x}}\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x},t+\delta)|_{\boldsymbol{x}=\boldsymbol{x}_{t+\delta}^{\mathrm{pred}}}\right]^{\top}\Big(\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_{t+\delta}^{\mathrm{pred}},t+\delta)-\boldsymbol{c}_{t+\delta}\Big). \quad (15)$$

Algorithm 1 Constraint-preserving CF generation (fixing η, γ)

Require: trained vector field $\boldsymbol{u}_{\boldsymbol{\theta}}$, desired-class region $\psi_{\boldsymbol{w}}, \boldsymbol{c}_t, r^2_{\boldsymbol{\nu}}(t)$, original instance $\boldsymbol{x}^{\mathrm{or}}$, desired class y^{cf}, initial state $\boldsymbol{x}_0$, number of ODE steps K, steering step size η, region scaling factor γ.
Ensure: CF candidate $\boldsymbol{x}^{\mathrm{cf}}$.

$\delta \leftarrow 1/K$
for $k = 0, \ldots, K-1$ **do**
  $t_k \leftarrow k\delta, \;\; t_{k+1} \leftarrow (k+1)\delta$
  $\hat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{x}_k + (1-t_k)\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, t_k)$
  $\boldsymbol{x}^{\mathrm{pred}}_{k+1} \leftarrow \boldsymbol{x}_k - \eta \nabla_{\boldsymbol{x}_k} \mathcal{L}_{\mathrm{GD}}(\hat{\boldsymbol{x}}_1) + \delta \boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, t_k)$
  $e \leftarrow \|\psi_{\boldsymbol{w}}(\boldsymbol{x}^{\mathrm{pred}}_{k+1}, t_{k+1}) - \boldsymbol{c}_{t_{k+1}}\|_2^2 - \gamma r^2_{\boldsymbol{\nu}}(t_{k+1})$
  if $e > 0$ **then**
    $\boldsymbol{g}^{\mathrm{proj}} \leftarrow \nabla_{\boldsymbol{x}^{\mathrm{pred}}_{k+1}} e, \;\; \boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}^{\mathrm{pred}}_{k+1} - \dfrac{e}{\|\boldsymbol{g}^{\mathrm{proj}}\|_2^2 + \varepsilon} \boldsymbol{g}^{\mathrm{proj}}$
  else
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}^{\mathrm{pred}}_{k+1}$
  end if
end for
return $\boldsymbol{x}^{\mathrm{cf}} \leftarrow \boldsymbol{x}_K$

We then correct the state by

$$\boldsymbol{x}_{t+\delta} = \boldsymbol{x}^{\mathrm{pred}}_{t+\delta} - \frac{e_{t+\delta}(\boldsymbol{x}^{\mathrm{pred}}_{t+\delta})}{\|\boldsymbol{g}_{t+\delta}\|_2^2 + \varepsilon} \boldsymbol{g}_{t+\delta}. \tag{16}$$

Here, $\varepsilon > 0$ is a small constant for numerical stability.

Theoretical analysis. Projection correction brings a state $\boldsymbol{x}^{\mathrm{pred}}_{t+\delta}$ that was moved outside the learned desired-class region by the prediction step back toward the region with a small modification. Because the desired-class region is generally nonlinear, solving an exact projection problem at every Euler step would be expensive. We therefore linearize $e_{t+\delta}$ around the prediction point $\boldsymbol{x}^{\mathrm{pred}}_{t+\delta}$ and use the minimum-norm correction to the linearized constraint boundary. This correction can be interpreted as a local projection in the direction $-\boldsymbol{g}_{t+\delta}$, which decreases the violation. The following proposition shows that Eq. (16) is exactly the minimum-norm correction under the linearized constraint.

Proposition 1. *Assume* $e_{t+\delta}(\boldsymbol{x}^{\mathrm{pred}}_{t+\delta}) > 0$, $\boldsymbol{g}_{t+\delta} \neq \boldsymbol{0}$, *and* $\varepsilon = 0$. *Then Eq.* (16) *gives the unique minimum-norm solution*

$$\min_{\Delta \boldsymbol{x} \in \mathbb{R}^L} \frac{1}{2}\|\Delta \boldsymbol{x}\|_2^2 \qquad \textit{s.t.} \qquad e_{t+\delta}\left(\boldsymbol{x}^{\mathrm{pred}}_{t+\delta} + \Delta \boldsymbol{x}\right) \approx e_{t+\delta}(\boldsymbol{x}^{\mathrm{pred}}_{t+\delta}) + \boldsymbol{g}^{\top}_{t+\delta} \Delta \boldsymbol{x} = 0 \tag{17}$$

given by

$$\Delta \boldsymbol{x}^{\star} = -\frac{e_{t+\delta}(\boldsymbol{x}^{\mathrm{pred}}_{t+\delta})}{\|\boldsymbol{g}_{t+\delta}\|_2^2} \boldsymbol{g}_{t+\delta}. \tag{18}$$

Thus, the projection step can be viewed as an orthogonal projection onto the linearized feasible set. The proof is in Appendix D.

Algorithm. Algorithm 1 shows CTCF inference for fixed η and γ, where $\boldsymbol{x}_k$ denotes a discretized flow state. At each Euler step, CTCF first estimates the predicted endpoint $\hat{\boldsymbol{x}}_1$ from the current vector field and steers the state using the local gradient of the endpoint objective in Eq. (10). It then applies the linearized projection correction in Eq. (16) only if the prediction point $\boldsymbol{x}^{\mathrm{pred}}_{k+1}$ violates the learned desired-class region at time t_{k+1}. In practice, we choose η and γ from small candidate sets $\{10^{-5}, 10^{-4}, 10^{-3}\}$ and $\{0.1, 0.5, 1.0\}$, respectively, by evaluating all nine combinations. The selection criterion prioritizes validity; ties are broken by choosing the candidate with the smallest proximity. The full algorithm is given in Appendix E.

5 Experiments

We evaluate CTCF following the evaluation protocol of CELT [Yamaguchi et al., 2026]. Additional experimental details are given in Appendix G.

5.1 Experimental setup

Classifiers. We use time-series classifiers whose gradients are unavailable in both supervised and one-class settings. In the supervised setting, we use QUANT [Dempster et al., 2024], which is recommended in a recent TSC benchmark [Middlehurst et al., 2024b] for its accuracy and speed. QUANT first extracts features from time-series instances without labels and then trains a supervised classifier on these features. Following this design, in the one-class setting we use a classifier that applies LOF [Breunig et al., 2000, Han et al., 2022] to QUANT features.

Datasets. We use 26 datasets from the UCR Archive [Dau et al., 2019], setting class 1 as the desired class and class 2 as the original class. This two-class evaluation follows Yamaguchi et al. [2026] for comparison with existing methods; it does not restrict CTCF to binary classifiers. For some datasets, the official UCR training split contains too few instances for stable one-class classifier training. We therefore adjust the split by reserving the last 30 series from each of class 1 and class 2 for testing and using all remaining series for training. The supervised setting uses training instances from both classes, while the one-class setting uses only desired-class training instances. The split is shared across the classifier, CF generator, region learner, FM model, and supervised/one-class settings. Before CF generation, we select at most 10 correctly classified original-class test instances $\boldsymbol{x}^{\mathrm{or}}$ satisfying $f(\boldsymbol{x}^{\mathrm{or}}) = y^{\mathrm{or}}$ in their fixed test-set order.

Baselines. We compare CTCF with seven methods: four existing methods and three ablations. The existing methods are CELT [Yamaguchi et al., 2026], CVAE [Todo et al., 2023], LASTS [Spinnato et al., 2023], and MCCE [Redelmeier et al., 2024]. All are model-agnostic, support hard-decision classifiers, and do not require access to training data at deployment time. Because CVAE relies on supervised contrastive learning, we compare with it only in the supervised setting. The ablations are "w/o proj," which removes the projection step in §4; "w/o steer," which removes steering by the fused-lasso endpoint objective; and "w/o time," which removes the time-dependent components of the desired-class region learner in §3. In "w/o time," $\boldsymbol{\psi}_{\boldsymbol{w}}$, $\boldsymbol{c}_t$, and $r^2_{\boldsymbol{\nu}}$ do not depend on flow time t.

5.2 Evaluation of validity

Figures 3(a) and 3(b) show critical-difference (CD) diagrams for validity on 26 datasets in the supervised and one-class settings. The numbers in parentheses are mean ranks; lower is better. Methods connected by a bold line are not significantly different at the 95% confidence level [Demšar, 2006]. The CD values in Figures 3(a) and 3(b) are 2.06 and 1.77, respectively. In the supervised setting, CTCF obtains the best mean rank. In the one-class setting, it obtains a mean rank that is not significantly different from the best existing method. These results show that CTCF achieves competitive validity against existing methods.

Compared with the ablations, CTCF has a better mean rank than all three ablations in both supervised and one-class settings. In particular, "w/o proj" degrades substantially in the supervised setting, indicating that projection correction is important for guiding an unconditional FM model trained on both classes toward the desired class. In the one-class setting, "w/o proj" is also a strong baseline because the FM model itself is trained only on desired-class training instances. The comparison with "w/o steer" shows that adding fused-lasso steering to the projection-corrected trajectory does not sacrifice validity. Overall, these results support the effectiveness of projection, steering, and the flow-time-dependent region.

5.3 Evaluation of proximity, sparsity, NumSeg, and implausibility

For proximity, sparsity, NumSeg, and implausibility, we compare CTCF with each baseline on the common set of valid CFs. These metrics can attain their best values for unchanged samples such as $\boldsymbol{x}^{\mathrm{cf}} = \boldsymbol{x}^{\mathrm{or}} \in \mathbb{R}^L$, but such samples do not flip the prediction and are not meaningful CF explanations. The common-valid-set protocol also ensures that CTCF and each comparison method are evaluated on the same original instances $\boldsymbol{x}^{\mathrm{or}}$, enabling like-for-like comparisons. Figures 3(c) and 3(d) show proximity and sparsity; figures for all four metrics are in Appendix G.3. The results show that CTCF with fused-lasso steering achieves better proximity and sparsity than existing methods.

Table 1 summarizes pairwise Wilcoxon signed-rank tests under the common-valid-set protocol for the four metrics. The tests are two-sided with significance level $\alpha = 0.05$. Each cell reports a p-value,

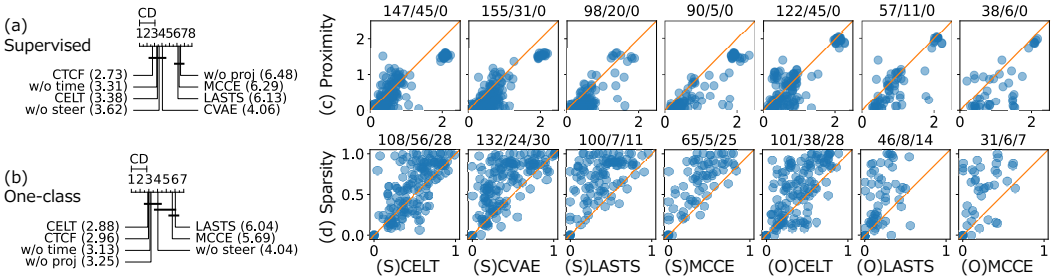

Figure 3: Summary of experimental results. The left side shows CD diagrams for validity, where a lower mean rank is better. The right side shows examples of pairwise comparisons under the common-valid-set protocol: proximity in the top row and sparsity in the bottom row. In each scatter-plot title, $v_1/v_2/v_3$ denotes the numbers of wins/losses/draws for CTCF. (S) and (O) denote supervised and one-class settings. Full pairwise comparisons are in Appendix G.3.

Table 1: Pairwise Wilcoxon signed-rank test results under the common-valid-set protocol. Each cell reports a p-value, and win/loss/draw indicates whether CTCF is better/worse/tied against the comparison method. (n.s.) indicates no significant difference at $\alpha = 0.05$. $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively. All p-values are from two-sided tests.

Setting	Comparison	Proximity $\downarrow$	Sparsity $\uparrow$	NumSeg $\downarrow$	Implausibility $\downarrow$
Supervised	vs. CELT	**win** 2.11×10^{-11}	**win** 4.08×10^{-7}	**win** 7.70×10^{-3}	**win** 3.69×10^{-2}
	vs. CVAE	**win** 7.61×10^{-22}	**win** 4.74×10^{-21}	**win** 2.61×10^{-7}	**loss** (n.s.) 5.24×10^{-1}
	vs. LASTS	**win** 8.46×10^{-14}	**win** 8.77×10^{-18}	**win** 2.48×10^{-9}	**win** 1.77×10^{-7}
	vs. MCCE	**win** 1.26×10^{-16}	**win** 3.82×10^{-12}	**win** 1.90×10^{-6}	**loss** 3.40×10^{-3}
	vs. w/o proj	**win** 2.39×10^{-10}	**win** 4.33×10^{-9}	**win** (n.s.) 1.88×10^{-1}	**win** 6.93×10^{-4}
	vs. w/o steer	**win** 6.19×10^{-35}	**win** 8.06×10^{-29}	**win** 1.56×10^{-6}	**win** 5.95×10^{-6}
	vs. w/o time	**loss** (n.s.) 7.79×10^{-2}	**loss** (n.s.) 6.58×10^{-1}	**win** (n.s.) 2.26×10^{-1}	**win** (n.s.) 6.82×10^{-2}
One-class	vs. CELT	**win** 4.28×10^{-7}	**win** 1.15×10^{-10}	**win** 5.03×10^{-9}	**win** (n.s.) 7.17×10^{-1}
	vs. LASTS	**win** 8.38×10^{-8}	**win** 1.30×10^{-7}	**win** 4.85×10^{-2}	**loss** (n.s.) 7.65×10^{-1}
	vs. MCCE	**win** 4.79×10^{-6}	**win** 5.80×10^{-6}	**win** 1.09×10^{-2}	**loss** (n.s.) 8.85×10^{-1}
	vs. w/o proj	**win** 3.62×10^{-5}	**win** 7.17×10^{-3}	**loss** (n.s.) 2.99×10^{-1}	**win** (n.s.) 9.70×10^{-1}
	vs. w/o steer	**win** 5.06×10^{-27}	**win** 3.78×10^{-21}	**win** 8.73×10^{-3}	**win** 2.58×10^{-2}
	vs. w/o time	**win** 8.46×10^{-3}	**win** (n.s.) 1.96×10^{-1}	**draw** (n.s.) 4.38×10^{-1}	**win** (n.s.) 7.77×10^{-1}

and win/loss/draw indicates whether CTCF is better/worse/tied against the corresponding method; (n.s.) indicates no significant difference.

Compared with the ablations, CTCF is not significantly worse on any of these four metrics. Against existing methods, CTCF is significantly better in all comparisons for proximity, sparsity, and NumSeg. Implausibility is mixed: CTCF is significantly worse than MCCE in the supervised setting, while the other comparisons against existing methods are not significant. MCCE, however, performs worse on other metrics. These results indicate that CTCF has some trade-off in implausibility but significantly outperforms existing methods in proximity, sparsity, and NumSeg.

5.4 Additional evaluations

We further evaluate the effectiveness of the learned desired-class region, applicability to multiclass classification, dependence on the classifier, and hyperparameter sensitivity. The maximum dataset-wise mean execution time of Algorithm 1 was 0.41 min on one L40S GPU (Appendix G.2).

To compare CTCF's intermediate-state feasibility assessment with an endpoint-based alternative, we replace the learned desired-class region with an autoencoder (AE)-based constraint. This constraint uses the reconstruction error of an AE trained on endpoints classified as the desired class and is evaluated at the predicted endpoint $\hat{\boldsymbol{x}}_1$ at each Euler step. We retain the FM model and steering-and-correction framework. Across 26 datasets, CTCF achieved higher mean validity, more wins than losses on each CF-quality metric, and better mean ranks on all five metrics in both settings. Five of eight common-valid Wilcoxon comparisons significantly favored CTCF; the rest were nonsignificant.

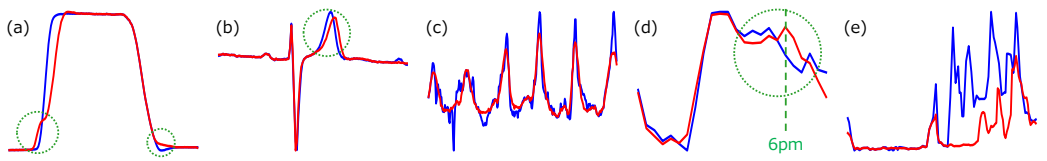

Figure 4: Original time series (blue) and generated CFs (red) on (a) GunPoint, (b) ECGFiveDays, (c) ToeSegmentation1, (d) ItalyPowerDemand, and (e) PowerCons. Green circles indicate local patterns that distinguish the two classes.

These results support the usefulness of CTCF's intermediate-state feasibility assessment relative to the endpoint-AE proxy (Appendix G.5).

To test CTCF beyond the two-class evaluation protocol, we compared it with its three ablations in supervised experiments retaining all classes of the 12 native multiclass datasets (3–50 classes). CTCF matched or outperformed all three ablations in mean rank on all five metrics. Eleven of the 12 common-valid tests over the four CF-quality metrics significantly favored CTCF, with no significant degradation. In these pairwise tests, removing either projection or steering significantly worsened all four CF-quality metrics, while removing flow-time dependence significantly worsened sparsity, NumSeg, and implausibility but not proximity. These results support the component-level trends observed in the original two-class evaluation (Appendix G.6).

To test whether these component-level trends extend beyond QUANT, we compared CTCF with the same three ablations using MultiROCKET-HYDRA [Tan et al., 2022, Dempster et al., 2023] on all 26 datasets in both settings, applying Isolation Forest [Liu et al., 2008] to its features in the one-class setting. CTCF achieved the best mean rank on all five metrics in the supervised setting; in the one-class setting, it ranked best for validity and proximity and tied for the best sparsity rank. Eight and six of the 12 common-valid tests over the four CF-quality metrics significantly favored CTCF in the supervised and one-class settings, respectively, with no significant degradation. These results support the component-level trends beyond the QUANT-based classifier (Appendix G.7).

To assess sensitivity to training hyperparameters, we evaluated all 36 configurations on five supervised datasets (50 originals). We had fixed $(\tau, q, \lambda_{\text{pin}}) = (0.5, 0.95, 1)$ across all datasets and both settings before the main experiments, without dataset-specific tuning. In one-factor comparisons, we varied one parameter while holding the other two at their adopted values. The adopted configuration was not optimal for every metric, but ranked second of 36 on a joint score for validity and sparsity. For inference, we used the same nine (η, γ) combinations across all 26 datasets and both settings, prioritizing validity and then lower proximity for each instance. Every candidate value of η and γ was selected in both settings, but no single value was selected for all instances. Together, these findings support using a common training-time configuration and selecting η and γ instance-wise from a small grid shared across datasets and settings (Appendix G.8).

5.5 Case studies

We evaluate the CFs generated by CTCF in the one-class setting using five UCR datasets whose explanatory evidence is relatively well understood. In this setting, following Section 5.1, instances from the original class y^{or} are not included in the training data. Figure 4 shows original time series (blue) and their generated CFs (red). All five examples are valid CFs. To assess whether the illustrated examples have typical metric values, we compare their proximity, sparsity, NumSeg, and implausibility with the distributions over all valid CTCF counterfactuals for each dataset. Across five examples and four metrics, 17 of the 20 values lie within the corresponding interquartile ranges, and three lie beyond them in the favorable direction (Appendix G.9).

GunPoint classifies motion patterns for drawing a gun (gun class) versus pointing (pointing class). In the gun class, a bump appears when the subject draws the gun from the holster, whereas in the pointing class a dip caused by overshoot appears when the arm is lowered [Ye and Keogh, 2009, Lines et al., 2012]. Figure 4(a) shows that CTCF adds the bump and reduces the overshoot dip, clearly capturing both patterns of the gun class. **ECGFiveDays** classifies patient conditions from electrocardiogram signals. A cardiologist identified the T wave as the discriminative region [Keogh and Rakthanmanon, 2013]. Figure 4(b) shows that CTCF generates local changes corresponding to

the T wave. **ToeSegmentation1** classifies normal and abnormal walking. Figure 4(c) shows that, from an abnormal-walk original instance, the generated normal-walk CF repeats regular and stable walking patterns. **ItalyPowerDemand** classifies daily electricity demand in Italy into summer and winter. The generated winter time series has a peak around 6 p.m., consistent with prior studies [Cabello et al., 2020, Yamaguchi et al., 2025]. **PowerCons** classifies household electric power consumption into warm and cold seasons [Dau et al., 2019]. The generated warm-season CFs show lower evening-to-night consumption than the cold-season originals, consistent with lower warm-month evening peaks and reduced heating use [Palmer et al., 2013, Godoy-Shimizu et al., 2014].

6 Related work

6.1 Counterfactual generation for time-series classification

Research on time-series CF generation can be broadly divided into model-specific methods [Karlsson et al., 2020, Yan and Wang, 2023] and model-agnostic methods. Among model-agnostic methods, some rely on classifier input gradients [Wang et al., 2021, Filali Boubrahimi and Hamdi, 2022, Wang et al., 2024, Yamaguchi et al., 2024], while others require access to training instances at deployment time [Ates et al., 2021, Delaney et al., 2021, Bahri et al., 2024]. These assumptions can be obstacles in applications with decision-only black-box classifiers or strict data-access constraints.

Generative approaches avoid these restrictions. CVAE and CELT both learn structured latent spaces; the former uses supervised contrastive loss, whereas the latter further performs minimal time-local segment edits at deployment time [Todo et al., 2023, Yamaguchi et al., 2026]. LASTS searches local neighborhoods in a VAE latent space, and MCCE, a tabular method applicable to time series, samples and filters valid and plausible candidates [Spinnato et al., 2023, Redelmeier et al., 2024]. In contrast, CTCF uses an unconditional FM generator that is independent of the classifier and desired class, and controls its continuous-time trajectory with a learned desired-class region over flow time.

6.2 Continuous-time generative models and inference-time control

In computer vision, diffusion models and FM have achieved major success in high-quality generation. FM [Lipman et al., 2023] and rectified flow [Liu et al., 2023] provide efficient frameworks for training and sampling in continuous-time generative modeling. From the perspective of inference-time control, Patel et al. [2025] propose a steering rule that avoids expensive backpropagation through ODE trajectories, and Utkarsh et al. [2025], Cheng et al. [2025] provide constraint correction for pretrained flow models. We bring these ideas to time-series CF generation, using a desired-class region learned from classifier hard decisions as the constraint rather than physical constraints.

Diffusion-based CF explanations have also been studied. Most existing methods for image classification use gradients or logits/probabilities of the target classifier [Augustin et al., 2022, Jeanneret et al., 2024, Augustin et al., 2024, Weng et al., 2024]. Derivative-free directions also exist [Kim et al., 2025], but they differ from our setting, which supports hard decisions. For time series, Ji et al. [2024] generate anomaly repair as CFs but assume a linearly decomposable anomaly score. Deng et al. [2024] focus on block-maxima forecasting, unlike our setting of class flips in TSC. CTCF addresses this gap by combining FM inference-time steering with learned time-dependent desired-class constraints, generating CFs while keeping intermediate states on the flow trajectory consistent with the constraints.

7 Conclusion

We proposed CTCF, a CF explanation method applicable to hard-decision and one-class time-series classifiers. CTCF uses a pretrained unconditional FM model as a generator and learns a desired-class region along flow time. At inference time, CTCF combines FlowChef-style endpoint-objective steering with linearized projection correction onto this region. Theoretically, we showed that the learning objective defined on interpolation states coincides with the corresponding objective on ODE trajectories under ideal marginal matching. This justifies simulation-free learning of the desired-class region from interpolation states. On UCR datasets, CTCF was effective across five metrics in supervised and one-class settings; case studies further demonstrated interpretable generated CFs.

Acknowledgments and Disclosure of Funding

The research was the collaborative work of Toshiba Corporation and Kyushu University, based on funding from Toshiba Corporation.

References

Emre Ates, Burak Aksar, Vitus J. Leung, and Ayse K. Coskun. Counterfactual explanations for multivariate time series. In *ICAPAI*, pages 1–8. IEEE Computer Society, 2021.

Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion visual counterfactual explanations. In *NeurIPS*, volume 35, pages 364–377. Curran Associates Inc., 2022.

Maximilian Augustin, Yannic Neuhaus, and Matthias Hein. DiG-IN: Diffusion guidance for investigating networks – uncovering classifier differences, neuron visualisations, and visual counterfactual explanations. In *CVPR*, pages 11093–11103. IEEE Computer Society, 2024.

Omar Bahri, Peiyu Li, Soukaina Filali Boubrahimi, and Shah Muhammad Hamdi. Discord-based counterfactual explanations for time series classification. *Data Min. Knowl. Discov.*, 38(6):3347–3371, 2024.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *SIGMOD*, pages 93–104. ACM, 2000.

Nestor Cabello, Elham Naghizade, Jianzhong Qi, and Lars Kulik. Fast and accurate time series classification through supervised interval search. In *ICDM*, pages 948–953. IEEE Computer Society, 2020.

Chaoran Cheng, Boran Han, Danielle C. Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W. Mahoney, and Bernie Wang. Gradient-free generation for hard-constrained systems. In *ICLR*. OpenReview.net, 2025.

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

Eoin Delaney, Derek Greene, and Mark T. Keane. Instance-based counterfactual explanations for time series classification. In *ICCBR*, pages 32–47. Springer-Verlag, 2021.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. Hydra: competing convolutional kernels for fast and accurate time series classification. *Data Min. Knowl. Discov.*, 37(5):1779–1805, 2023.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. Quant: a minimalist interval method for time series classification. *Data Min. Knowl. Discov.*, 38(4):2377–2402, 2024.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

Yue Deng, Asadullah Hill Galib, Pang-Ning Tan, and Lifeng Luo. Unraveling block maxima forecasting models with counterfactual explanation. In *KDD*, pages 562–573. ACM, 2024.

Soukaïna Filali Boubrahimi and Shah Muhammad Hamdi. On the mining of time series data counterfactual explanations using barycenters. In *CIKM*, pages 3943–3947. ACM, 2022.

Daniel Godoy-Shimizu, Jason Palmer, and Nicola Terry. What can we learn from the household electricity survey? *Buildings*, 4(4):737–761, 2014.

Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Min. Knowl. Discov.*, 36(6):1–55, 2022.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: anomaly detection benchmark. In *NeurIPS*, pages 32142–32159. Curran Associates Inc., 2022.

Hadi Hojjati and Narges Armanfard. Dasvdd: Deep autoencoding support vector data descriptor for anomaly detection. *IEEE Trans. on Knowl. and Data Eng.*, 36(8):3739–3750, 2024.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.*, 33(4):917–963, 2019.

Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. Diffusion models for counterfactual explanations. *Comput. Vis. Image Underst.*, 249:104207, 2024.

Xiayan Ji, Anton Xue, Eric Wong, Oleg Sokolsky, and Insup Lee. Ar-pro: Counterfactual explanations for anomaly repair with formal properties. In *NeurIPS*, pages 16133–16159. Curran Associates Inc., 2024.

Ziwen Kan et al. Benchmarking counterfactual interpretability in deep learning models for time series classification. *arXiv preprint arXiv:2408.12666*, 2024. URL `https://arxiv.org/abs/2408.12666`.

Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis. Locally and globally explainable time series tweaking. *Knowl. Inf. Syst.*, 62(5):1671–1700, 2020.

Eamonn Keogh and Thanawin Rakthanmanon. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SDM*, pages 668–676. SIAM, 2013.

Won Jun Kim, Hyungjin Chung, Jaemin Kim, Sangmin Lee, Byeongsu Sim, and Jong Chul Ye. Derivative-free diffusion manifold-constrained gradient for unified XAI. In *CVPR*, pages 23795–23805. IEEE Computer Society, 2025.

Roger Koenker and Gilbert Bassett, Jr. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

Jason Lines, Luke M. Davis, Jon Hills, and Anthony Bagnall. A shapelet transform for time series classification. In *KDD*, pages 289–297. ACM, 2012.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for generative modeling. In *ICLR*. OpenReview.net, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow Matching guide and code, 2024. `https://github.com/facebookresearch/flow_matching`.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, pages 413–422. IEEE Computer Society, 2008.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with Rectified Flow. In *ICLR*. OpenReview.net, 2023.

Matthew Middlehurst, Ali Ismail-Fawaz, Antoine Guillaume, Christopher Holder, David Guijo-Rubio, Guzal Bulatova, Leonidas Tsaprounis, Lukasz Mentel, Martin Walter, Patrick Schäfer, and Anthony Bagnall. aeon: a python toolkit for learning from time series. *J. Mach. Learn. Res.*, 25(1), 2024a.

Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Min. Knowl. Discov.*, 38(4):1958–2031, 2024b. ISSN 1384-5810.

Jason Palmer, Nicola Terry, and Tom Kane. Further analysis of the household electricity survey: Early findings: Demand side management. Department of Energy and Climate Change, 2013. `https://assets.publishing.service.gov.uk/media/5a7c2aa340f0b645ba3c71de/early_findings_revised.pdf`.

Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. FlowChef: Steering of Rectified Flow models for controlled generations. In *ICCV*, pages 15308–15318. IEEE Computer Society, 2025.

Annabelle Redelmeier, Martin Jullum, Kjersti Aas, and Anders Løland. Mcce: Monte carlo sampling of valid and realistic counterfactual explanations for tabular data. *Data Min. Knowl. Discov.*, 38(4):1830–1861, 2024.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, volume 80, pages 4393–4402. PMLR, 2018.

Francesco Spinnato, Riccardo Guidotti, Anna Monreale, Mirco Nanni, Dino Pedreschi, and Fosca Giannotti. Understanding any time series classifier with a subsequence-based explainer. *ACM Trans. Knowl. Discov. Data*, 18(2), 2023.

Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey I. Webb. Multirocket: multiple pooling operators and transformations for fast and effective time series classification. *Data Min. Knowl. Discov.*, 36 (5):1623–1646, 2022.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B*, 67(1):91–108, 2005.

William Todo, Merwann Selmani, Béatrice Laurent, and Jean-Michel Loubes. Counterfactual explanation for multivariate times series using a contrastive variational autoencoder. In *ICASSP*, pages 1–5. IEEE Computer Society, 2023.

Alexander Tong and contributors. TorchCFM: A conditional Flow Matching library, 2024. `https://github.com/atong01/conditional-flow-matching`.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Trans. Mach. Learn. Res.*, 2024.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained Flow Matching: Sampling generative models with hard constraints. In *NeurIPS*. Curran Associates Inc., 2025.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Comput. Surv.*, 56(12): 1–42, 2024.

S Wachter, B Mittelstadt, and C Russell. Counterfactual explanations without opening the black box: automated decisions and the gdpr. *Harvard Journal of Law and Technology*, 31(2):841–887, 2018.

Zhendong Wang, Isak Samsten, Rami Mochaourab, and Panagiotis Papapetrou. Learning time series counterfactuals via latent space representations. In *Discovery Science*, pages 369–384. Springer-Verlag, 2021.

Zhendong Wang, Isak Samsten, Ioanna Miliou, Rami Mochaourab, and Panagiotis Papapetrou. Glacier: guided locally constrained counterfactual explanations for time series classification. *Machine Learning*, 113(3):1–31, 2024.

Nina Weng, Paraskevas Pegios, Eike Petersen, Aasa Feragen, and Siavash Bigdeli. FastDiME: Fast diffusion-based counterfactuals for shortcut removal and generation. In *ECCV*, pages 338–357. Springer, 2024.

Akihiro Yamaguchi, Ken Ueno, Ryusei Shingaki, and Hisashi Kashima. Learning counterfactual explanations with intervals for time-series classification. In *CIKM*, pages 4158–4162. ACM, 2024.

Akihiro Yamaguchi, Ken Ueno, and Hisashi Kashima. Learning location-guided time-series shapelets. *IEEE Trans. on Knowl. and Data Eng.*, 37(5):2712–2726, 2025.

Akihiro Yamaguchi, Shizuo Kaji, Kaname Matsue, and Ryusei Shingaki. Counterfactual explanations via latent structure for time series classification. In *AISTATS*. PMLR, 2026.

Jingquan Yan and Hao Wang. Self-interpretable time series prediction with counterfactual explanations. In *ICML*, volume 202, pages 39110–39125. PMLR, 2023.

Lexiang Ye and Eamonn Keogh. Time series shapelets: A new primitive for data mining. In *KDD*, pages 947–956. ACM, 2009.

Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019.

A Evaluation metrics

This section defines the counterfactual quality metrics used in Sections 2 and 5.3. Let $\mathcal{S}_{\mathrm{eval}}$ be the set of original instances to evaluate. Validity is the fraction of counterfactuals that flip to the desired class:

$$\mathrm{validity} := \frac{1}{|\mathcal{S}_{\mathrm{eval}}|} \sum_{\boldsymbol{x}^{\mathrm{or}} \in \mathcal{S}_{\mathrm{eval}}} \mathbb{I}\big(f(\boldsymbol{x}^{\mathrm{cf}}) = y^{\mathrm{cf}}\big). \tag{19}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if its argument is true and 0 otherwise. Proximity is the average ℓ_1 distance between the original instance and the counterfactual:

$$\mathrm{prox}(\boldsymbol{x}^{\mathrm{or}}, \boldsymbol{x}^{\mathrm{cf}}) := \frac{1}{L} \sum_{l=1}^{L} |x_l^{\mathrm{cf}} - x_l^{\mathrm{or}}|. \tag{20}$$

Smaller values indicate greater closeness to the original instance. Sparsity is the fraction of timestamps with sufficiently small changes:

$$\mathrm{sparsity}(\boldsymbol{x}^{\mathrm{or}}, \boldsymbol{x}^{\mathrm{cf}}) := \frac{1}{L} \sum_{l=1}^{L} \mathbb{I}\big(|x_l^{\mathrm{cf}} - x_l^{\mathrm{or}}| \le \epsilon_{\mathrm{sp}}\big) \tag{21}$$

following Todo et al. [2023], Yamaguchi et al. [2026]. Under this definition, larger values mean that fewer timestamps are changed. Plausibility measures whether a counterfactual is realistic under the data distribution. We use implausibility, defined as the nearest-neighbor distance to the held-out test set $\mathcal{D}_{\mathrm{test}}$:

$$\mathrm{impl}(\boldsymbol{x}^{\mathrm{cf}}) := \min_{\boldsymbol{x} \in \mathcal{D}_{\mathrm{test}}} \|\boldsymbol{x}^{\mathrm{cf}} - \boldsymbol{x}\|_1. \tag{22}$$

Smaller values mean that the generated counterfactual is closer to test data and thus more plausible [Guidotti, 2022, Verma et al., 2024]. NumSeg is the number of contiguous segments formed by changed timestamps. First define the change indicator

$$b_l = \mathbb{I}\Big(|x_l^{\mathrm{cf}} - x_l^{\mathrm{or}}| > \epsilon_{\mathrm{sg}}\big(\max_{1 \le j \le L} x_j^{\mathrm{or}} - \min_{1 \le j \le L} x_j^{\mathrm{or}}\big)\Big), \qquad l = 1, \dots, L. \tag{23}$$

We then fill any gap of at most $g_L := \mathrm{round}(0.01L)$ unchanged timestamps between changed timestamps, so nearby changes are counted as one segment. Let $\tilde{b}_l$ denote the resulting indicator. Setting $\tilde{b}_0 := 0$, we define

$$\mathrm{NumSeg}(\boldsymbol{x}^{\mathrm{or}}, \boldsymbol{x}^{\mathrm{cf}}) := \sum_{l=1}^{L} \tilde{b}_l (1 - \tilde{b}_{l-1}) \tag{24}$$

following Kan et al. [2024], Yamaguchi et al. [2026]. Smaller values indicate that changes concentrate in fewer contiguous segments. Following Yamaguchi et al. [2026], we set $\epsilon_{\mathrm{sp}} = 0.2$ on the standardized time series and $\epsilon_{\mathrm{sg}} = 0.1$, and use the 1% gap-merging rule above.

B Proof of theorem 1

We show the equality between the risk on interpolation states and the ODE-side risk. First, for any fixed $t \in [0, 1]$, we show that

$$\mathbb{E}_{X_0, X_1}\big[\ell_{\mathrm{all}}((1-t)X_0 + tX_1, t, X_1)\big] = \mathbb{E}_{X_t}\big[\tilde{\ell}_{\mathrm{all}}(X_t, t)\big]. \tag{25}$$

Writing the expectation as an integral with respect to P_{X_t} gives

$$\mathbb{E}_{X_t}\big[\tilde{\ell}_{\mathrm{all}}(X_t, t)\big] = \int_{\mathbb{R}^L} \tilde{\ell}_{\mathrm{all}}(\boldsymbol{x}, t)\, dP_{X_t}(\boldsymbol{x}). \tag{26}$$

Let $P_{X_1 | X_t = \boldsymbol{x}}$ be the conditional distribution of X_1 given $X_t = \boldsymbol{x}$. Substituting Eq. (8) yields

$$\int_{\mathbb{R}^L} \left(\int_{\mathbb{R}^L} \ell_{\mathrm{all}}(\boldsymbol{x}, t, \boldsymbol{x}_1)\, dP_{X_1 | X_t = \boldsymbol{x}}(\boldsymbol{x}_1) \right) dP_{X_t}(\boldsymbol{x}). \tag{27}$$

By the definition of conditional distribution, this equals

$$\int_{\mathbb{R}^L \times \mathbb{R}^L} \ell_{\mathrm{all}}(\boldsymbol{x}, t, \boldsymbol{x}_1)\, dP_{(X_t, X_1)}(\boldsymbol{x}, \boldsymbol{x}_1) = \mathbb{E}_{X_t, X_1}\big[\ell_{\mathrm{all}}(X_t, t, X_1)\big]. \tag{28}$$

Since $X_t = (1-t)X_0 + tX_1$, Eq. (25) follows.

Next, the assumption $Z_t \overset{d}{=} X_t$ implies that, for any integrable function $\tilde{\ell}_{\mathrm{all}}(\cdot, t)$,

$$\mathbb{E}_{Z_t}\big[\tilde{\ell}_{\mathrm{all}}(Z_t, t)\big] = \mathbb{E}_{X_t}\big[\tilde{\ell}_{\mathrm{all}}(X_t, t)\big]. \tag{29}$$

Combining this with Eq. (25), for each fixed t we obtain

$$\mathbb{E}_{Z_t}\big[\tilde{\ell}_{\mathrm{all}}(Z_t, t)\big] = \mathbb{E}_{X_0, X_1}\big[\ell_{\mathrm{all}}((1-t)X_0 + tX_1, t, X_1)\big]. \tag{30}$$

Averaging both sides over $t \sim \mathrm{Unif}[0,1]$ proves Eq. (9).

C Approximation-error bound under marginal mismatch

Theorem 1 establishes equality between the interpolation-side and ODE-side region-learning risks under perfect marginal matching. The following proposition relaxes this assumption and bounds the risk difference in terms of the mismatch between the learned ODE marginals and the interpolation marginals.

For fixed region parameters, let $R_{\mathrm{ODE}}(\boldsymbol{\theta})$ and R_{int} denote the ODE-side and interpolation-side risks in Eq. (9), respectively:

$$R_{\mathrm{ODE}}(\boldsymbol{\theta}) := \mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\Big[\mathbb{E}_{Z_t}\big[\tilde{\ell}_{\mathrm{all}}(Z_t, t)\big]\Big],$$
$$R_{\mathrm{int}} := \mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\Big[\mathbb{E}_{X_0, X_1}\big[\ell_{\mathrm{all}}((1-t)X_0 + tX_1, t, X_1)\big]\Big].$$

Here, Z_t is the state induced by the learned ODE, and X_t is the interpolation state in Eq. (1). Write

$$\widehat{p}_t^{\boldsymbol{\theta}} := \mathrm{Law}(Z_t), \qquad p_t := \mathrm{Law}(X_t).$$

Proposition 2. *Assume that the losses defining these risks are integrable in their state and flow-time arguments under the corresponding joint probability laws. Suppose that the conditional loss $\tilde{\ell}_{\mathrm{all}}(\cdot, t)$ defined in Eq.* (8) *is uniformly C_ℓ-Lipschitz in its state argument:*

$$\Big|\tilde{\ell}_{\mathrm{all}}(\boldsymbol{z}, t) - \tilde{\ell}_{\mathrm{all}}(\boldsymbol{z}', t)\Big| \le C_\ell \|\boldsymbol{z} - \boldsymbol{z}'\|_2$$

for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^L$ and $t \in [0,1]$. Assume also that $\widehat{p}_t^{\boldsymbol{\theta}}$ and p_t have finite first moments for each t. Then

$$|R_{\mathrm{ODE}}(\boldsymbol{\theta}) - R_{\mathrm{int}}| \le C_\ell\, \mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\big[W_1\big(\widehat{p}_t^{\boldsymbol{\theta}}, p_t\big)\big] =: \varepsilon_{\mathrm{FM}}, \tag{31}$$

where W_1 is the 1-Wasserstein distance based on the Euclidean norm. Consequently,

$$R_{\mathrm{int}} - \varepsilon_{\mathrm{FM}} \le R_{\mathrm{ODE}}(\boldsymbol{\theta}) \le R_{\mathrm{int}} + \varepsilon_{\mathrm{FM}}.$$

Proof. By the definition of the conditional loss in Eq. (8) and the law of total expectation,

$$R_{\mathrm{int}} = \mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\Big[\mathbb{E}_{X_t}\big[\tilde{\ell}_{\mathrm{all}}(X_t, t)\big]\Big].$$

For each fixed t, the Lipschitz assumption and the Kantorovich–Rubinstein inequality give

$$\Big|\mathbb{E}_{Z_t}[\tilde{\ell}_{\mathrm{all}}(Z_t, t)] - \mathbb{E}_{X_t}[\tilde{\ell}_{\mathrm{all}}(X_t, t)]\Big| \le C_\ell W_1\big(\widehat{p}_t^{\boldsymbol{\theta}}, p_t\big)\,.$$

Taking the expectation over $t \sim \mathrm{Unif}[0,1]$ and applying the triangle inequality yields the bound in Eq. (31):

$$|R_{\mathrm{ODE}}(\boldsymbol{\theta}) - R_{\mathrm{int}}| = \Big|\mathbb{E}_t\Big[\mathbb{E}_{Z_t}[\tilde{\ell}_{\mathrm{all}}(Z_t, t)] - \mathbb{E}_{X_t}[\tilde{\ell}_{\mathrm{all}}(X_t, t)]\Big]\Big|$$
$$\le \mathbb{E}_t\Big[\Big|\mathbb{E}_{Z_t}[\tilde{\ell}_{\mathrm{all}}(Z_t, t)] - \mathbb{E}_{X_t}[\tilde{\ell}_{\mathrm{all}}(X_t, t)]\Big|\Big] \le C_\ell\, \mathbb{E}_t\big[W_1\big(\widehat{p}_t^{\boldsymbol{\theta}}, p_t\big)\big] = \varepsilon_{\mathrm{FM}}.$$

□

The quantity $\varepsilon_{\mathrm{FM}}$ bounds the region-learning risk error caused by marginal mismatch. Ideal marginal matching gives $\varepsilon_{\mathrm{FM}} = 0$ and recovers the equality in Theorem 1. This approximation-error bound expresses the combined effects of finite data, model approximation, and optimization through marginal mismatch; it is not a finite-sample bound that evaluates these effects separately.

D Proof of proposition 1

Fix $t+\delta$ and consider the linearly constrained optimization problem

$$\min_{\Delta \boldsymbol{x} \in \mathbb{R}^L} \ \frac{1}{2}\|\Delta \boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad e_{t+\delta}(\boldsymbol{x}_{t+\delta}^{\text{pred}}) + \boldsymbol{g}_{t+\delta}^\top \Delta \boldsymbol{x} = 0. \tag{32}$$

Its Lagrangian is

$$\mathcal{J}(\Delta \boldsymbol{x}, \lambda) = \frac{1}{2}\|\Delta \boldsymbol{x}\|_2^2 + \lambda\big(e_{t+\delta}(\boldsymbol{x}_{t+\delta}^{\text{pred}}) + \boldsymbol{g}_{t+\delta}^\top \Delta \boldsymbol{x}\big). \tag{33}$$

The stationarity condition gives

$$\nabla_{\Delta \boldsymbol{x}} \mathcal{J}(\Delta \boldsymbol{x}, \lambda) = \Delta \boldsymbol{x} + \lambda \boldsymbol{g}_{t+\delta} = \boldsymbol{0}, \tag{34}$$

so

$$\Delta \boldsymbol{x} = -\lambda \boldsymbol{g}_{t+\delta}. \tag{35}$$

Substituting this into the constraint gives

$$e_{t+\delta}(\boldsymbol{x}_{t+\delta}^{\text{pred}}) - \lambda \|\boldsymbol{g}_{t+\delta}\|_2^2 = 0. \tag{36}$$

Since $\boldsymbol{g}_{t+\delta} \neq \boldsymbol{0}$,

$$\lambda = \frac{e_{t+\delta}(\boldsymbol{x}_{t+\delta}^{\text{pred}})}{\|\boldsymbol{g}_{t+\delta}\|_2^2} \tag{37}$$

is uniquely determined. Hence

$$\Delta \boldsymbol{x}^\star = -\frac{e_{t+\delta}(\boldsymbol{x}_{t+\delta}^{\text{pred}})}{\|\boldsymbol{g}_{t+\delta}\|_2^2} \boldsymbol{g}_{t+\delta}. \tag{38}$$

The objective is strictly convex and the constraint is linear, so this is the unique minimum-norm solution. Therefore, Eq. (16) gives the minimum-norm correction under the constraint linearized around the prediction point.

E Detailed counterfactual-generation algorithm

Algorithm 2 extends Algorithm 1 with hyperparameter selection. In our implementation of FlowChef-style endpoint-objective steering [Patel et al., 2025], the endpoint guidance gradient at the current Euler step is computed by autograd as

$$\boldsymbol{g}_t^{\text{GD}} := \nabla_{\boldsymbol{x}_t} \mathcal{L}_{\text{GD}}\big(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)\big), \quad \hat{\boldsymbol{x}}_1(\boldsymbol{x}_t) = \boldsymbol{x}_t + (1-t)\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t). \tag{39}$$

This gradient is taken with respect to the current state $\boldsymbol{x}_t$ through the local endpoint estimate $\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)$; we do not backpropagate through the full ODE trajectory. That is, we do not unroll a multi-step chain such as

$$\nabla_{\boldsymbol{x}_t} \mathcal{L}_{\text{GD}}(\boldsymbol{x}_1) = \left(\frac{\partial \boldsymbol{x}_{t+\delta}}{\partial \boldsymbol{x}_t}\right)^\top \cdots \left(\frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_{1-\delta}}\right)^\top \nabla_{\boldsymbol{x}_1} \mathcal{L}_{\text{GD}}(\boldsymbol{x}_1). \tag{40}$$

Instead, we approximate the steering direction toward the endpoint objective by a single-step chain rule through the endpoint estimate:

$$\nabla_{\boldsymbol{x}_t} \mathcal{L}_{\text{GD}}\big(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)\big) = \left(\frac{\partial \hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\right)^\top \nabla_{\hat{\boldsymbol{x}}_1} \mathcal{L}_{\text{GD}}(\hat{\boldsymbol{x}}_1). \tag{41}$$

Algorithm 2 Detailed counterfactual-generation algorithm (including hyperparameter selection)

Require: trained $\boldsymbol{u}_{\boldsymbol{\theta}}$, $\boldsymbol{\psi}_{\boldsymbol{w}}$, $\boldsymbol{c}_t$, $r^2_{\boldsymbol{\nu}}(t)$, hard-decision classifier f, original instance $\boldsymbol{x}^{\mathrm{or}}$, desired class y^{cf}, steps K, candidate sets $\mathcal{H}_\eta$ and $\mathcal{H}_\gamma$, and a small positive constant ε.
Ensure: selected counterfactual $\boldsymbol{x}^{\mathrm{best}}$.

$V^{\mathrm{best}} \leftarrow -\infty$, $P^{\mathrm{best}} \leftarrow +\infty$, $\boldsymbol{x}^{\mathrm{best}} \leftarrow \emptyset$
$\xi \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $\eta \in \mathcal{H}_\eta$ **do**
  for $\gamma \in \mathcal{H}_\gamma$ **do**
    $\delta \leftarrow 1/K$
    $\boldsymbol{x}_0 \leftarrow \xi$
    for $k = 0, \ldots, K-1$ **do**
      $t_k \leftarrow k\delta$, $t_{k+1} \leftarrow \min(1, (k+1)\delta)$
      $\boldsymbol{v}_k \leftarrow \boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, t_k)$
      $\hat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{x}_k + (1 - t_k)\boldsymbol{v}_k$
      $\mathcal{L}_{\mathrm{GD}} \leftarrow \|\boldsymbol{D}(\hat{\boldsymbol{x}}_1 - \boldsymbol{x}^{\mathrm{or}})\|_1 + \|\hat{\boldsymbol{x}}_1 - \boldsymbol{x}^{\mathrm{or}}\|_1$
      $\boldsymbol{g}_k^{\mathrm{GD}} \leftarrow \nabla_{\boldsymbol{x}_k} \mathcal{L}_{\mathrm{GD}}$
      $\boldsymbol{x} \leftarrow \boldsymbol{x}_k - \eta \boldsymbol{g}_k^{\mathrm{GD}} + \delta \boldsymbol{v}_k$ ▷ prediction step
      $e \leftarrow \|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}, t_{k+1}) - \boldsymbol{c}_{t_{k+1}}\|_2^2 - \gamma r^2_{\boldsymbol{\nu}}(t_{k+1})$
      if $e > 0$ **then**
        $\boldsymbol{g}^{\mathrm{proj}} \leftarrow \nabla_{\boldsymbol{x}} e$
        $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x} - \dfrac{e}{\|\boldsymbol{g}^{\mathrm{proj}}\|_2^2 + \varepsilon} \boldsymbol{g}^{\mathrm{proj}}$ ▷ linearized projection
      else
        $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}$
      end if
    end for
    $\boldsymbol{x}^{\mathrm{cand}} \leftarrow \boldsymbol{x}_K$
    $V \leftarrow \mathbb{I}\big(f(\boldsymbol{x}^{\mathrm{cand}}) = y^{\mathrm{cf}}\big)$
    $P \leftarrow L^{-1}\|\boldsymbol{x}^{\mathrm{cand}} - \boldsymbol{x}^{\mathrm{or}}\|_1$
    if $V > V^{\mathrm{best}}$ **or** $(V = V^{\mathrm{best}}$ **and** $P < P^{\mathrm{best}})$ **then**
      $\boldsymbol{x}^{\mathrm{best}} \leftarrow \boldsymbol{x}^{\mathrm{cand}}$, $V^{\mathrm{best}} \leftarrow V$, $P^{\mathrm{best}} \leftarrow P$
    end if
  end for
end for
return $\boldsymbol{x}^{\mathrm{best}}$

During counterfactual generation, the trained FM model and desired-class region learner are fixed; their parameters are not updated. We use the Euler solver with $K = 1000$ steps. The candidate sets are $\mathcal{H}_\eta = \{10^{-5}, 10^{-4}, 10^{-3}\}$ and $\mathcal{H}_\gamma = \{0.1, 0.5, 1.0\}$. We set $\varepsilon = 10^{-8}$.

For each original instance, all nine (η, γ) pairs use the same sampled initial state $\boldsymbol{x}_0$, as shown in Algorithm 2, so their differences are not confounded by different initial states.

F Implementation details of learning models

This section gives implementation details for the FM model and desired-class region learner used in the experiments. Both models are trained separately on the training data of each dataset. We handle univariate time series, so the number of channels is $M_{\mathrm{ch}} = 1$, and the mini-batch input tensor has shape (B, M_{ch}, L). Unless otherwise stated, we use PyTorch defaults and do not use learning-rate scheduling, early stopping, or dropout.

F.1 Flow Matching model

The basic design follows Lipman et al. [2024], Tong and contributors [2024]: the model takes an interpolation state $\boldsymbol{x}_t \in \mathbb{R}^{M_{\mathrm{ch}} \times L}$ and flow time $t \in (0, 1)$ and outputs the vector field $\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \in \mathbb{R}^{M_{\mathrm{ch}} \times L}$. All convolutions are one-dimensional for time-series counterfactual generation, and all kernel sizes are 3. Table 2 lists the hyperparameters.

Input conditioning. Flow time t is transformed into a time channel using a sinusoidal positional embedding and an MLP. Specifically, t is mapped to a 60-dimensional sinusoidal embedding, which is passed through Linear$(60, 240)$–SiLU–Linear$(240, L)$ to obtain a time channel of shape $(B, 1, L)$. This time channel is concatenated with the interpolation state along the channel dimension, so the

subsequent network input has shape $(B, M_{\mathrm{ch}}+1, L)$. No class channel is concatenated to the network input; the vector field $\boldsymbol{u}_{\boldsymbol{\theta}}$ is trained as an unconditional FM model not conditioned on the desired class or classifier. This design allows the same FM model to be reused as a generator independent of the classifier and desired class.

Network architecture. Table 3 shows the FM architecture. A residual block adds the output of Conv$(\mathrm{padding}=1)$–SiLU–Conv$(\mathrm{padding}=1)$ to the block input and then applies SiLU. A DownBlock applies stride-2 convolution for downsampling followed by one residual block. An UpBlock applies transposed convolution for upsampling followed by one residual block. At the end of the encoder–decoder, Conv$(64, M_{\mathrm{ch}}, \mathrm{padding}=1)$ outputs the vector field. To avoid length mismatches caused by stride-2 downsampling and upsampling, the output is interpolated to the original length L.

Table 2: Training hyperparameters of the FM model.

Item	Value
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	0
Batch size	128
Epochs	10,000
Gradient clipping	global gradient norm 1.0

Table 3: Architecture of the FM model. Since we use univariate time series, $M_{\mathrm{ch}}=1$.

Component	Operation	Output
Time channel	Sinusoidal embedding $(60) \to$ Linear$(60, 240) \to$ SiLU $\to$ Linear$(240, L) \to$ unsqueeze	$(B, 1, L)$
Input concatenation	Interpolation state (B, M_{ch}, L) and time channel $(B, 1, L) \to$ channel-wise concatenation	$(B, M_{\mathrm{ch}}+1, L)$
Input conv	Conv$(M_{\mathrm{ch}}+1, 64, 3, \mathrm{padding}=1) \to$ SiLU	$(B, 64, L)$
DownBlock 1	Conv$(64, 128, 3, \mathrm{stride}=2, \mathrm{padding}=1) \to$ Residual block	$(B, 128, \lceil L/2 \rceil)$
DownBlock 2	Conv$(128, 256, 3, \mathrm{stride}=2, \mathrm{padding}=1) \to$ Residual block	$(B, 256, \lceil L/4 \rceil)$
Middle	Residual block	$(B, 256, \lceil L/4 \rceil)$
UpBlock 1	ConvTranspose$(256, 128, 4, \mathrm{stride}=2, \mathrm{padding}=1) \to$ Residual block	$(B, 128, 2\lceil L/4 \rceil)$
UpBlock 2	ConvTranspose$(128, 64, 4, \mathrm{stride}=2, \mathrm{padding}=1) \to$ Residual block	$(B, 64, 4\lceil L/4 \rceil)$
Output	Conv$(64, M_{\mathrm{ch}}, 3, \mathrm{padding}=1) \to$ F.interpolate$(\mathrm{size}=L)$	(B, M_{ch}, L)

F.2 Desired-class region learner

The desired-class region learner learns, for each flow time t, a region $\|\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}, t) - \boldsymbol{c}_t\|_2^2 \le r_{\boldsymbol{\nu}}^2(t)$. Its inputs are an interpolation state $\boldsymbol{x}_t$ and flow time t, and its output is a 128-dimensional feature representation. As in FM, interpolation states are constructed as $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$, where $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{x}_1$ is a training instance. Table 4 lists the hyperparameters.

Feature representation network with time channel. Flow time is mapped to a 20-dimensional sinusoidal embedding and then transformed by an MLP, Linear$(20, 80)$–SiLU–Linear$(80, L)$, into a time channel of shape $(B, 1, L)$. This time channel is concatenated with the interpolation state, and the resulting tensor of shape $(B, M_{\mathrm{ch}}+1, L)$ is fed to an FCN encoder. The FCN encoder is a three-layer bias-free 1D FCN: Conv$(M_{\mathrm{ch}}+1, 128, 8, \mathrm{padding}=4)$–ReLU, Conv$(128, 256, 5, \mathrm{padding}=2)$–ReLU, and Conv$(256, 128, 3, \mathrm{padding}=1)$–ReLU, followed by AdaptiveAvgPool1d(1) over the timestamp dimension to obtain $\boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_t, t) \in \mathbb{R}^{128}$. This architecture adopts FCN-style temporal convolutions commonly used in time-series classification [Ismail Fawaz et al., 2019], while disabling convolution biases to match the bias-free design of Deep SVDD [Ruff et al., 2018].

Center and squared-radius functions. The center function $\boldsymbol{c}_t$ is a two-layer MLP, Linear$(1, 64)$–SiLU–Linear$(64, 128)$, taking scalar flow time t as input. This center model is pretrained and then fixed during main region training. The squared-radius function $r_{\boldsymbol{\nu}}^2(t)$ is a two-layer MLP, Linear$(1, 32)$–SiLU–Linear$(32, 1)$, followed by softplus to keep the output positive. For initialization, we compute the 0.95-quantile of squared distances for desired-class interpolation states and set the final-layer bias using the inverse softplus. This function is then updated during main region training.

Training stages. The region learner is trained in three stages. In Stage 0, to initialize the FCN encoder of the feature representation network, we train a convolutional autoencoder for 2,000 epochs. Its decoder reconstructs the original time series using Linear$(128, 256)$–ReLU–Linear$(256, M_{\mathrm{ch}}L)$. The loss is mean-squared reconstruction error. In Stage 1, using the pretrained encoder, we train $\boldsymbol{c}_t$ on desired-class interpolation states for 200 epochs and then fix it. Stage 2 performs main region training. Specifically, the feature representation network, including the MLP for time embedding, and the squared-radius function $r^2_{\boldsymbol{\nu}}(t)$ are jointly optimized for 1,500 epochs. During the first 100 epochs, radius warm-up updates only $r^2_{\boldsymbol{\nu}}(t)$ using desired-class samples. We set $\tau = 0.5$ in Eq. (3).

Table 4: Training hyperparameters of the desired-class region learner.

Item	Value
Batch size	128
Representation dimension	128
Autoencoder pretraining	2,000 epochs; Adam; learning rate 10^{-3}; weight decay 10^{-6}
Center fitting	200 epochs; Adam; learning rate 10^{-3}; weight decay 10^{-6}
Main region training	1,500 epochs; Adam; learning rate 10^{-4}; $\lambda_{\mathrm{rep}} = 10^{-6}$
Radius quantile	$q = 0.95$
Radius pinball coefficient	$\lambda_{\mathrm{pin}} = 1.0$
Activation time for non-desired class	$\tau = 0.5$
Radius warm-up	100 epochs

Table 5: Architecture of the desired-class region learner. Since we use univariate time series, $M_{\mathrm{ch}} = 1$.

Component	Operation	Output
Time channel	Sinusoidal embedding $(20) \to$ Linear$(20, 80) \to$ SiLU $\to$ Linear$(80, L) \to$ unsqueeze	$(B, 1, L)$
Input concatenation	Interpolation state (B, M_{ch}, L) and time channel $(B, 1, L) \to$ channel-wise concatenation	$(B, M_{\mathrm{ch}} + 1, L)$
Encoder block 1	Conv$(M_{\mathrm{ch}} + 1, 128, 8, \mathrm{padding} = 4, \mathrm{bias} = \mathrm{False}) \to$ ReLU	$(B, 128, L + 1)$
Encoder block 2	Conv$(128, 256, 5, \mathrm{padding} = 2, \mathrm{bias} = \mathrm{False}) \to$ ReLU	$(B, 256, L + 1)$
Encoder block 3	Conv$(256, 128, 3, \mathrm{padding} = 1, \mathrm{bias} = \mathrm{False}) \to$ ReLU	$(B, 128, L + 1)$
Pooling	AdaptiveAvgPool1d$(1) \to$ squeeze	$(B, 128)$
Center model	Linear$(1, 64) \to$ SiLU $\to$ Linear$(64, 128)$	$(B, 128)$
Radius head	Linear$(1, 32) \to$ SiLU $\to$ Linear$(32, 1) \to$ softplus $\to$ squeeze	(B)
AE decoder for pretraining	Linear$(128, 256) \to$ ReLU $\to$ Linear$(256, M_{\mathrm{ch}}L) \to$ reshape	(B, M_{ch}, L)

G Experimental details

This section provides additional details for Section 5. Unless otherwise stated, the results of existing methods for validity, proximity, sparsity, NumSeg, and implausibility are the same as those reported in the main text and appendix of Yamaguchi et al. [2026]. However, we rename MAD as proximity. Also, because their implausibility calculation included training data and thus suffered from data leakage, we evaluate implausibility only on held-out test data following Eq. (22). All experiments were run on an Amazon AWS g6e.12xlarge instance with Ubuntu OS, four NVIDIA L40S GPUs, and 384 GiB RAM. The additional evaluations in Sections 5.4 and 5.5 are described separately in Sections G.5–G.9.

G.1 Details about experimental setup

This section adds details to Section 5.1 on the time-series classifiers, existing methods, and UCR datasets. All of these settings are shared with Yamaguchi et al. [2026].

Pre-trained classifiers. For supervised classification, we use `QUANTClassifier` from the aeon library [Middlehurst et al., 2024a]. For one-class classification, we use `QUANTTransformer` to generate the same features as `QUANTClassifier`, and then apply `LOF` from the PyOD library [Zhao et al., 2019]. We use default parameters in aeon v0.9.0 and PyOD v1.1.2. For evaluations with alternative classifiers, see Section G.7.

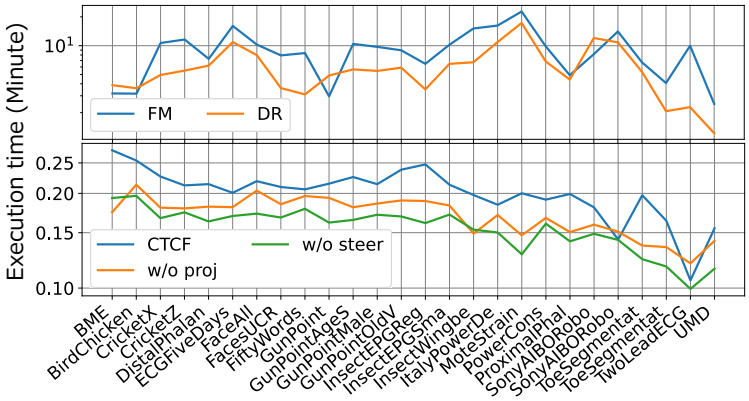

Figure 5: Training time (top) and inference-time (bottom) for each dataset. The top panel reports training time for Flow Matching (FM) and the desired-class region (DR). The bottom panel reports per-sample time for CTCF and ablations without steering or projection steps (w/o steer and w/o proj).

Existing methods. As in Yamaguchi et al. [2026], we make the following changes from the official repository implementations. We use UCR Archive datasets whose time-series instances are already split, whereas Todo et al. [2023] targeted long time series from PTB-XL. Therefore, we do not apply the random-cropping preprocessing used in CVAE. In addition, CVAE may not work correctly for some series lengths, such as odd L. Following the CELT architecture, we therefore apply upsampling after the last transposed-convolution step when needed so that the output length matches the input length L. Because the growing-sphere search in LASTS can run indefinitely, we set a 10-minute timeout for each generation.

Datasets. We select 26 datasets from the UCR Archive and use only two classes: class 1 (y^{cf}) and class 2 (y^{or}). The selection criteria are: (i) each of y^{cf} and y^{or} has at least 30 training instances; (ii) there are no missing values; (iii) all time series have the same length; (iv) training the CVAE model on the dataset does not cause a GPU out-of-memory error; and (v) test accuracy is at least 50% in both one-class and supervised settings. The selected datasets are listed in Tables 6–15. Details of each dataset are available on the UCR Archive web page [Dau et al., 2019].

Instance selection. In the original two-class experiments, the last 30 series from each selected class form the held-out test pool. From this pool, we select at most 10 correctly classified original-class instances in their fixed test-set order, before CF generation, as described in Section 5.1. No instance is selected based on the resulting CFs. The native multiclass experiment uses the prespecified round-robin rule described in Appendix G.6.

G.2 Training and inference time

Figure 5 shows training time (top) and inference time (bottom) for the proposed method. We used an Amazon AWS g6e.12xlarge instance, with one GPU per fit. In the main CTCF experiments, we fit the FM model and desired-class region learner $26 \times 2 = 52$ times each. The mean training time per fit was 12.47 min for the FM model and 8.20 min for the region learner in the supervised setting, and 4.95 min and 3.33 min, respectively, in the one-class setting.

Inference is much faster than training; the maximum dataset-wise mean execution time of Algorithm 1 was 0.41 min (approximately 25 seconds). The worst cold-start case took 43.16 min, including a 1.999-min nine-candidate sweep. Therefore, generating counterfactual candidates from a small set of combinations of the steering strength η and region conservativeness γ is computationally acceptable. CTCF is slightly slower than w/o steer and w/o proj, but remains in the same order of magnitude, so the added computation of projection and steering is not dominant.

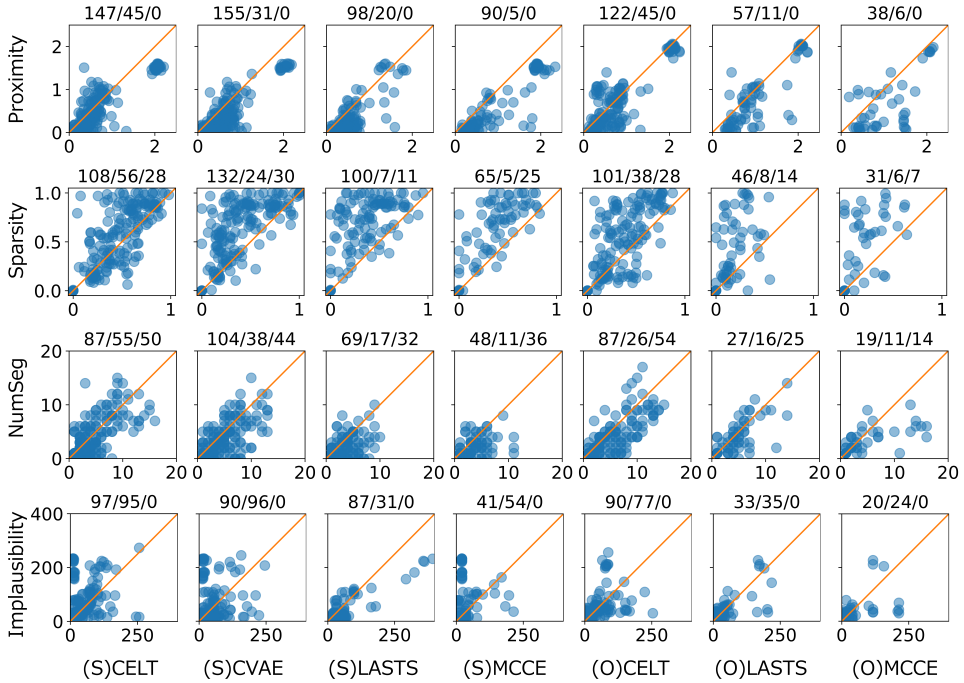

Figure 6: Pairwise comparisons for proximity, sparsity, NumSeg, and implausibility. In each scatter-plot title, $v_1/v_2/v_3$ denotes the numbers of wins/losses/draws for CTCF. (S) and (O) denote supervised and one-class settings.

G.3 Details about experimental results

This section provides details for the metric evaluation of validity, proximity, sparsity, NumSeg, and implausibility in Sections 5.2 and 5.3.

Validity. Before drawing the CD diagrams in Section 5.2, we performed the Friedman test. The p-values were 6.4×10^{-13} in the supervised setting and 6.1×10^{-11} in the one-class setting, indicating statistically significant differences among methods. As a post-hoc test, we used the Nemenyi test. The critical differences were 2.06 in the supervised setting and 1.77 in the one-class setting, matching the CD values used in Figure 3 (left).

Proximity, sparsity, NumSeg, and implausibility. As described in Section 5.3, proximity, sparsity, NumSeg, and implausibility are compared pairwise on the common set of instances for which both CTCF and the corresponding existing method generated valid CFs. Figure 3 (right) shows only part of the metric results; Figure 6 shows the results for all four metrics. Wilcoxon signed-rank test results are reported in Table 1 of the main text.

G.4 Dataset-wise performance tables

This section reports dataset-wise performance values under the experimental setup in Section 5. Tables 6–7 report validity in the supervised and one-class settings, respectively. Tables 8–15 report mean values for proximity, sparsity, NumSeg, and implausibility in the supervised and one-class settings; "±" denotes the standard deviation. As stated in Section 5.3, proximity, sparsity, NumSeg, and implausibility are computed only for valid CFs. If a method generates no valid CF on a dataset, we write "—" and assign the worst rank.

All Tables 6–15 include the same existing methods and three ablations, w/o proj, w/o steer, and w/o time, as in Section 5. In one-class settings, CVAE [Todo et al., 2023] is not included. Numbers in parentheses denote method ranks, where 1 is best. We also report the mean rank and Number of Wins at the bottom of each table.

Tables 6 and 7 directly support Figure 3 (left) in Section 5.2. In contrast, Tables 10–15 use a different protocol from Table 1 in Section 5.3 and Figure 6. Specifically, Tables 10–15 evaluate each method on its own set of valid instances, whereas Table 1 and Figure 6 evaluate CTCF and each existing method on their common valid set to ensure comparison under the same conditions. For fairer comparisons, see Table 1 and Figure 6.

Table 6: Dataset-wise validity (↑) in the supervised setting, including ablations.

	CELT	CTCF	CVAE	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	0.90 (5.5)	1.00 (2.5)	1.00 (2.5)	0.50 (7.0)	0.90 (5.5)	0.40 (8.0)	1.00 (2.5)	1.00 (2.5)
BirdChicken	0.80 (4.0)	0.90 (3.0)	0.70 (5.0)	0.00 (7.5)	0.00 (7.5)	1.00 (1.5)	0.10 (6.0)	1.00 (1.5)
CricketX	1.00 (1.0)	0.50 (2.5)	0.00 (7.0)	0.00 (7.0)	0.00 (7.0)	0.20 (5.0)	0.40 (4.0)	0.50 (2.5)
CricketZ	0.90 (1.0)	0.40 (4.5)	0.70 (2.0)	0.00 (7.0)	0.00 (7.0)	0.00 (7.0)	0.40 (4.5)	0.50 (3.0)
DistalPhalanxOutlineAgeGroup	0.10 (6.5)	1.00 (2.5)	0.10 (6.5)	1.00 (2.5)	0.00 (8.0)	0.30 (5.0)	1.00 (2.5)	1.00 (2.5)
ECGFiveDays	1.00 (2.0)	1.00 (2.0)	1.00 (2.0)	0.75 (6.0)	0.20 (8.0)	0.30 (7.0)	0.90 (4.5)	0.90 (4.5)
FaceAll	1.00 (1.0)	0.90 (2.0)	0.50 (4.0)	0.00 (7.0)	0.00 (7.0)	0.00 (7.0)	0.70 (3.0)	0.30 (5.0)
FacesUCR	1.00 (1.5)	1.00 (1.5)	0.70 (4.0)	0.00 (7.5)	0.00 (7.5)	0.40 (6.0)	0.60 (5.0)	0.80 (3.0)
FiftyWords	1.00 (1.5)	1.00 (1.5)	0.50 (7.0)	0.20 (8.0)	0.60 (6.0)	0.70 (4.5)	0.70 (4.5)	0.90 (3.0)
GunPoint	1.00 (2.5)	1.00 (2.5)	0.80 (7.0)	0.80 (7.0)	0.90 (5.0)	0.80 (7.0)	1.00 (2.5)	1.00 (2.5)
GunPointAgeSpan	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)	0.80 (6.0)	0.70 (7.5)	0.70 (7.5)	1.00 (3.0)	1.00 (3.0)
GunPointMaleVersusFemale	0.40 (6.5)	0.90 (2.0)	0.40 (6.5)	0.70 (4.0)	1.00 (1.0)	0.20 (8.0)	0.80 (3.0)	0.50 (5.0)
GunPointOldVersusYoung	1.00 (2.5)	1.00 (2.5)	0.50 (7.0)	0.80 (5.5)	1.00 (2.5)	0.30 (8.0)	1.00 (2.5)	0.80 (5.5)
InsectEPGRegularTrain	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	0.30 (7.0)	1.00 (3.5)	0.20 (8.0)	1.00 (3.5)	1.00 (3.5)
InsectEPGSmallTrain	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	0.20 (7.0)	1.00 (3.5)	0.10 (8.0)	1.00 (3.5)	1.00 (3.5)
InsectWingbeatSound	1.00 (1.5)	1.00 (1.5)	0.70 (5.5)	0.70 (5.5)	0.40 (7.0)	0.30 (8.0)	0.80 (3.5)	0.80 (3.5)
ItalyPowerDemand	0.40 (8.0)	1.00 (2.0)	0.75 (5.0)	0.85 (4.0)	0.70 (6.0)	0.50 (7.0)	1.00 (2.0)	1.00 (2.0)
MoteStrain	0.80 (6.0)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)	0.00 (8.0)	0.50 (7.0)	1.00 (3.0)	1.00 (3.0)
PowerCons	1.00 (1.5)	0.90 (4.5)	1.00 (1.5)	0.90 (4.5)	0.40 (7.0)	0.30 (8.0)	0.90 (4.5)	0.90 (4.5)
ProximalPhalanxOutlineAgeGroup	0.10 (6.5)	0.90 (2.5)	1.00 (1.0)	0.80 (4.5)	0.00 (8.0)	0.10 (6.5)	0.80 (4.5)	0.90 (2.5)
SonyAIBORobotSurface1	1.00 (2.5)	1.00 (2.5)	0.90 (5.5)	0.00 (7.5)	0.00 (7.5)	0.90 (5.5)	1.00 (2.5)	1.00 (2.5)
SonyAIBORobotSurface2	0.90 (1.5)	0.70 (4.0)	0.90 (1.5)	0.00 (7.5)	0.00 (7.5)	0.20 (6.0)	0.60 (5.0)	0.80 (3.0)
ToeSegmentation1	1.00 (2.0)	1.00 (2.0)	1.00 (2.0)	0.00 (7.5)	0.00 (7.5)	0.60 (6.0)	0.90 (4.5)	0.90 (4.5)
ToeSegmentation2	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)	0.33 (8.0)	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)
TwoLeadECG	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	0.50 (7.5)	0.50 (7.5)	1.00 (3.5)	1.00 (3.5)
UMD	0.60 (5.5)	1.00 (2.5)	1.00 (2.5)	0.10 (8.0)	0.20 (7.0)	0.60 (5.5)	1.00 (2.5)	1.00 (2.5)
Rank Mean	3.38	2.73	4.06	6.13	6.29	6.48	3.62	3.31
Number of Wins	18	18	13	3	5	2	13	13

Table 7: Dataset-wise validity (↑) in the one-class setting, including ablations.

	CELT	CTCF	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	1.00 (3.0)	1.00 (3.0)	0.50 (6.0)	0.00 (7.0)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
BirdChicken	1.00 (3.0)	1.00 (3.0)	0.00 (6.5)	1.00 (3.0)	1.00 (3.0)	0.00 (6.5)	1.00 (3.0)
CricketX	1.00 (1.5)	1.00 (1.5)	0.62 (6.0)	0.00 (7.0)	0.88 (4.0)	0.88 (4.0)	0.88 (4.0)
CricketZ	1.00 (3.5)	1.00 (3.5)	0.67 (7.0)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)
DistalPhalanxOutlineAgeGroup	1.00 (3.0)	1.00 (3.0)	0.00 (6.5)	0.00 (6.5)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
ECGFiveDays	0.90 (2.5)	0.90 (2.5)	0.05 (6.0)	0.00 (7.0)	0.90 (2.5)	0.60 (5.0)	0.90 (2.5)
FaceAll	1.00 (3.0)	1.00 (3.0)	0.30 (6.0)	0.00 (7.0)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
FacesUCR	1.00 (3.0)	1.00 (3.0)	0.60 (6.0)	0.00 (7.0)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
FiftyWords	1.00 (2.5)	1.00 (2.5)	0.10 (6.0)	0.00 (7.0)	1.00 (2.5)	0.90 (5.0)	1.00 (2.5)
GunPoint	1.00 (1.0)	0.89 (3.0)	0.50 (6.0)	0.00 (7.0)	0.89 (3.0)	0.78 (5.0)	0.89 (3.0)
GunPointAgeSpan	1.00 (2.5)	1.00 (2.5)	0.78 (7.0)	1.00 (2.5)	0.89 (5.5)	0.89 (5.5)	1.00 (2.5)
GunPointMaleVersusFemale	1.00 (1.5)	0.70 (3.0)	0.50 (7.0)	1.00 (1.5)	0.60 (5.0)	0.60 (5.0)	0.60 (5.0)
GunPointOldVersusYoung	1.00 (3.5)	1.00 (3.5)	0.20 (7.0)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)
InsectEPGRegularTrain	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	0.00 (7.0)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)
InsectEPGSmallTrain	1.00 (3.0)	1.00 (3.0)	0.20 (6.0)	0.00 (7.0)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
InsectWingbeatSound	1.00 (1.0)	0.75 (3.0)	0.50 (6.0)	0.00 (7.0)	0.75 (3.0)	0.62 (5.0)	0.75 (3.0)
ItalyPowerDemand	0.70 (6.0)	1.00 (2.5)	0.95 (5.0)	0.00 (7.0)	1.00 (2.5)	1.00 (2.5)	1.00 (2.5)
MoteStrain	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)	1.00 (4.0)
PowerCons	1.00 (3.5)	1.00 (3.5)	0.00 (7.0)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)
ProximalPhalanxOutlineAgeGroup	1.00 (3.0)	1.00 (3.0)	0.00 (6.5)	0.00 (6.5)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
SonyAIBORobotSurface1	1.00 (3.0)	1.00 (3.0)	0.00 (6.5)	0.00 (6.5)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
SonyAIBORobotSurface2	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)	0.00 (7.0)	1.00 (3.0)	0.50 (6.0)	1.00 (3.0)
ToeSegmentation1	1.00 (3.0)	1.00 (3.0)	0.00 (7.0)	0.75 (6.0)	1.00 (3.0)	1.00 (3.0)	1.00 (3.0)
ToeSegmentation2	1.00 (2.5)	1.00 (2.5)	0.00 (6.5)	0.00 (6.5)	1.00 (2.5)	0.60 (5.0)	1.00 (2.5)
TwoLeadECG	1.00 (2.5)	1.00 (2.5)	0.10 (6.0)	0.00 (7.0)	1.00 (2.5)	0.60 (5.0)	1.00 (2.5)
UMD	1.00 (3.5)	1.00 (3.5)	0.00 (7.0)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)	1.00 (3.5)
Rank Mean	2.88	2.96	6.04	5.69	3.25	4.04	3.13
Number of Wins	25	23	3	8	21	15	22

Table 8: Dataset-wise proximity (↓) in the supervised setting, including ablations.

	CELT	CTCF	CVAE	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	0.20 ± 0.08 (3.0)	0.12 ± 0.05 (2.0)	0.32 ± 0.14 (5.0)	0.42 ± 0.15 (7.0)	0.40 ± 0.43 (6.0)	0.29 ± 0.11 (4.0)	0.54 ± 0.37 (8.0)	0.10 ± 0.04 (1.0)
BirdChicken	0.58 ± 0.34 (5.0)	0.10 ± 0.20 (3.0)	0.48 ± 0.24 (4.0)	— (7.5)	— (7.5)	0.03 ± 0.01 (1.5)	1.05 ± 0.00 (6.0)	0.03 ± 0.01 (1.5)
CricketX	0.64 ± 0.21 (1.0)	1.14 ± 0.20 (5.0)	— (7.0)	— (7.0)	— (7.0)	0.98 ± 0.04 (3.0)	1.12 ± 0.13 (4.0)	0.93 ± 0.11 (2.0)
CricketZ	0.59 ± 0.27 (1.0)	0.97 ± 0.18 (4.0)	0.81 ± 0.23 (2.0)	— (7.0)	— (7.0)	— (7.0)	1.11 ± 0.16 (5.0)	0.96 ± 0.21 (3.0)
DistalPhalanxOutlineAgeGroup	0.12 ± 0.00 (3.5)	0.09 ± 0.05 (2.0)	0.12 ± 0.00 (3.5)	0.24 ± 0.05 (7.0)	— (8.0)	0.21 ± 0.04 (6.0)	0.21 ± 0.04 (5.0)	0.09 ± 0.04 (1.0)
ECGFiveDays	0.22 ± 0.10 (3.0)	0.10 ± 0.06 (1.0)	0.26 ± 0.08 (4.0)	0.34 ± 0.12 (6.0)	0.28 ± 0.04 (5.0)	0.48 ± 0.03 (7.0)	0.48 ± 0.11 (8.0)	0.12 ± 0.06 (2.0)
FaceAll	0.68 ± 0.14 (2.0)	0.65 ± 0.23 (1.0)	0.81 ± 0.15 (4.0)	— (7.0)	— (7.0)	— (7.0)	0.90 ± 0.11 (5.0)	0.76 ± 0.14 (3.0)
FacesUCR	0.65 ± 0.22 (3.0)	0.67 ± 0.30 (4.0)	0.81 ± 0.09 (5.0)	— (7.5)	— (7.5)	0.59 ± 0.30 (1.0)	0.96 ± 0.14 (6.0)	0.61 ± 0.34 (2.0)
FiftyWords	0.56 ± 0.20 (3.0)	0.49 ± 0.36 (2.0)	0.72 ± 0.30 (4.0)	0.74 ± 0.05 (6.0)	0.74 ± 0.15 (5.0)	0.81 ± 0.19 (7.0)	1.02 ± 0.17 (8.0)	0.43 ± 0.31 (1.0)
GunPoint	0.18 ± 0.11 (3.0)	0.14 ± 0.11 (1.0)	0.20 ± 0.09 (4.0)	0.52 ± 0.16 (7.0)	0.43 ± 0.14 (5.0)	0.50 ± 0.25 (6.0)	0.60 ± 0.18 (8.0)	0.14 ± 0.14 (2.0)
GunPointAgeSpan	0.24 ± 0.10 (3.0)	0.06 ± 0.05 (2.0)	0.33 ± 0.27 (5.0)	0.49 ± 0.22 (6.0)	0.29 ± 0.08 (4.0)	0.55 ± 0.70 (7.0)	1.13 ± 0.60 (8.0)	0.06 ± 0.04 (1.0)
GunPointMaleVersusFemale	0.56 ± 0.36 (2.0)	0.67 ± 0.30 (3.0)	0.33 ± 0.17 (1.0)	1.32 ± 0.28 (7.0)	1.41 ± 0.41 (8.0)	0.91 ± 0.14 (6.0)	0.90 ± 0.25 (5.0)	0.72 ± 0.31 (4.0)
GunPointOldVersusYoung	1.45 ± 0.73 (7.0)	1.00 ± 0.55 (2.0)	0.78 ± 0.25 (1.0)	1.24 ± 0.62 (5.0)	1.54 ± 0.75 (8.0)	1.09 ± 0.56 (3.0)	1.33 ± 0.56 (6.0)	1.11 ± 0.63 (4.0)
InsectEPGRegularTrain	2.08 ± 0.04 (8.0)	1.49 ± 0.03 (4.0)	2.00 ± 0.07 (7.0)	1.26 ± 0.03 (2.0)	1.91 ± 0.04 (6.0)	1.15 ± 0.04 (1.0)	1.62 ± 0.06 (5.0)	1.30 ± 0.11 (3.0)
InsectEPGSmallTrain	2.04 ± 0.04 (8.0)	1.56 ± 0.03 (4.0)	2.04 ± 0.09 (7.0)	1.36 ± 0.01 (3.0)	1.89 ± 0.04 (6.0)	1.22 ± 0.00 (1.0)	1.68 ± 0.05 (5.0)	1.27 ± 0.14 (2.0)
InsectWingbeatSound	0.61 ± 0.08 (5.0)	0.59 ± 0.13 (4.0)	0.79 ± 0.09 (8.0)	0.48 ± 0.10 (1.0)	0.59 ± 0.14 (3.0)	0.63 ± 0.10 (6.0)	0.74 ± 0.11 (7.0)	0.52 ± 0.14 (2.0)
ItalyPowerDemand	0.63 ± 0.30 (8.0)	0.35 ± 0.19 (1.0)	0.36 ± 0.12 (2.0)	0.48 ± 0.17 (6.0)	0.44 ± 0.12 (5.0)	0.43 ± 0.19 (4.0)	0.54 ± 0.23 (7.0)	0.41 ± 0.24 (3.0)
MoteStrain	0.48 ± 0.12 (3.0)	0.36 ± 0.13 (1.0)	0.67 ± 0.10 (6.0)	0.53 ± 0.10 (4.0)	— (8.0)	0.58 ± 0.18 (5.0)	0.70 ± 0.13 (7.0)	0.47 ± 0.17 (2.0)
PowerCons	0.51 ± 0.16 (3.0)	0.29 ± 0.20 (1.0)	0.74 ± 0.22 (5.0)	0.73 ± 0.25 (4.0)	0.94 ± 0.30 (7.0)	0.95 ± 0.45 (8.0)	0.92 ± 0.33 (6.0)	0.30 ± 0.23 (2.0)
ProximalPhalanxOutlineAgeGroup	0.12 ± 0.00 (6.0)	0.10 ± 0.05 (3.0)	0.08 ± 0.02 (2.0)	0.13 ± 0.04 (7.0)	— (8.0)	0.11 ± 0.00 (5.0)	0.11 ± 0.04 (4.0)	0.08 ± 0.04 (1.0)
SonyAIBORobotSurface1	0.39 ± 0.11 (1.0)	0.46 ± 0.11 (4.0)	0.45 ± 0.08 (3.0)	— (7.5)	— (7.5)	0.49 ± 0.10 (5.0)	0.56 ± 0.10 (6.0)	0.44 ± 0.13 (2.0)
SonyAIBORobotSurface2	0.56 ± 0.10 (2.0)	0.64 ± 0.26 (4.0)	0.63 ± 0.09 (3.0)	— (7.5)	— (7.5)	0.85 ± 0.09 (6.0)	0.77 ± 0.18 (5.0)	0.56 ± 0.26 (1.0)
ToeSegmentation1	0.51 ± 0.21 (3.0)	0.48 ± 0.30 (2.0)	0.81 ± 0.23 (4.0)	— (7.5)	— (7.5)	0.87 ± 0.19 (5.0)	1.00 ± 0.13 (6.0)	0.46 ± 0.30 (1.0)
ToeSegmentation2	0.55 ± 0.09 (4.0)	0.18 ± 0.07 (2.0)	0.85 ± 0.02 (7.0)	0.73 ± 0.00 (5.0)	0.81 ± 0.08 (6.0)	0.26 ± 0.21 (3.0)	0.89 ± 0.08 (8.0)	0.18 ± 0.07 (1.0)
TwoLeadECG	0.22 ± 0.09 (5.0)	0.06 ± 0.01 (2.0)	0.14 ± 0.03 (3.0)	0.30 ± 0.08 (6.0)	0.39 ± 0.13 (8.0)	0.21 ± 0.04 (4.0)	0.35 ± 0.10 (7.0)	0.06 ± 0.01 (1.0)
UMD	0.22 ± 0.10 (3.0)	0.11 ± 0.03 (1.0)	0.52 ± 0.15 (5.0)	0.70 ± 0.00 (6.0)	0.40 ± 0.05 (4.0)	0.78 ± 0.36 (7.0)	0.84 ± 0.40 (8.0)	0.18 ± 0.06 (2.0)
Rank Mean	3.79	2.50	4.29	5.90	6.48	4.83	6.27	1.94
Number of Wins	3	7	2	1	0	4	0	10

Table 9: Dataset-wise proximity (↓) in the one-class setting, including ablations.

	CELT	CTCF	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	0.65 ± 0.35 (4.0)	0.58 ± 0.30 (2.0)	0.66 ± 0.00 (5.0)	— (7.0)	0.58 ± 0.30 (2.0)	1.17 ± 0.01 (6.0)	0.58 ± 0.30 (2.0)
BirdChicken	1.32 ± 0.00 (5.0)	0.06 ± 0.00 (2.0)	— (6.5)	0.73 ± 0.00 (4.0)	0.06 ± 0.00 (2.0)	— (6.5)	0.06 ± 0.00 (2.0)
CricketX	0.35 ± 0.03 (1.0)	0.91 ± 0.21 (4.0)	1.02 ± 0.05 (5.0)	— (7.0)	0.89 ± 0.22 (2.5)	1.21 ± 0.36 (6.0)	0.89 ± 0.22 (2.5)
CricketZ	0.30 ± 0.04 (1.0)	0.53 ± 0.41 (3.0)	1.02 ± 0.07 (6.0)	0.45 ± 0.09 (2.0)	0.53 ± 0.41 (4.5)	1.22 ± 0.25 (7.0)	0.53 ± 0.41 (4.5)
DistalPhalanxOutlineAgeGroup	0.25 ± 0.06 (4.0)	0.12 ± 0.05 (2.0)	— (6.5)	— (6.5)	0.12 ± 0.05 (2.0)	0.30 ± 0.06 (5.0)	0.12 ± 0.05 (2.0)
ECGFiveDays	0.26 ± 0.08 (3.0)	0.24 ± 0.12 (1.5)	0.57 ± 0.00 (6.0)	— (7.0)	0.27 ± 0.12 (4.0)	0.44 ± 0.06 (5.0)	0.24 ± 0.13 (1.5)
FaceAll	0.77 ± 0.09 (1.0)	0.82 ± 0.15 (3.0)	0.86 ± 0.09 (5.0)	— (7.0)	0.82 ± 0.15 (3.0)	0.93 ± 0.12 (6.0)	0.82 ± 0.15 (3.0)
FacesUCR	0.83 ± 0.09 (1.0)	0.84 ± 0.21 (3.0)	0.91 ± 0.10 (5.0)	— (7.0)	0.84 ± 0.21 (3.0)	1.00 ± 0.12 (6.0)	0.84 ± 0.21 (3.0)
FiftyWords	0.74 ± 0.19 (4.0)	0.36 ± 0.38 (2.0)	1.06 ± 0.00 (6.0)	— (7.0)	0.36 ± 0.38 (2.0)	1.04 ± 0.16 (5.0)	0.36 ± 0.38 (2.0)
GunPoint	0.41 ± 0.15 (4.0)	0.28 ± 0.22 (2.0)	0.60 ± 0.17 (6.0)	— (7.0)	0.28 ± 0.22 (2.0)	0.56 ± 0.25 (5.0)	0.28 ± 0.22 (2.0)
GunPointAgeSpan	0.35 ± 0.09 (3.0)	0.31 ± 0.31 (1.0)	1.02 ± 0.67 (6.0)	1.01 ± 0.51 (5.0)	0.48 ± 0.31 (4.0)	1.24 ± 0.54 (7.0)	0.31 ± 0.23 (2.0)
GunPointMaleVersusFemale	0.53 ± 0.26 (2.0)	0.55 ± 0.47 (3.0)	0.52 ± 0.13 (1.0)	1.13 ± 0.29 (7.0)	0.90 ± 0.60 (5.0)	0.99 ± 0.53 (6.0)	0.88 ± 0.58 (4.0)
GunPointOldVersusYoung	1.70 ± 0.65 (6.0)	1.44 ± 0.68 (1.0)	2.08 ± 0.14 (7.0)	1.61 ± 0.68 (4.0)	1.54 ± 0.72 (3.0)	1.64 ± 0.67 (5.0)	1.51 ± 0.75 (2.0)
InsectEPGRegularTrain	2.05 ± 0.05 (4.0)	2.01 ± 0.04 (2.0)	2.07 ± 0.04 (5.0)	— (7.0)	2.01 ± 0.04 (2.0)	2.09 ± 0.04 (6.0)	2.01 ± 0.04 (2.0)
InsectEPGSmallTrain	2.04 ± 0.05 (5.0)	1.95 ± 0.04 (2.0)	2.10 ± 0.12 (6.0)	— (7.0)	1.95 ± 0.04 (2.0)	2.04 ± 0.04 (4.0)	1.95 ± 0.04 (2.0)
InsectWingbeatSound	0.66 ± 0.04 (1.0)	0.74 ± 0.04 (4.0)	0.71 ± 0.01 (2.0)	— (7.0)	0.74 ± 0.04 (4.0)	0.77 ± 0.05 (6.0)	0.74 ± 0.04 (4.0)
ItalyPowerDemand	0.47 ± 0.14 (4.0)	0.36 ± 0.22 (1.0)	0.57 ± 0.24 (6.0)	— (7.0)	0.37 ± 0.23 (2.5)	0.56 ± 0.29 (5.0)	0.37 ± 0.23 (2.5)
MoteStrain	0.34 ± 0.00 (4.0)	0.16 ± 0.00 (2.0)	0.50 ± 0.00 (5.0)	0.56 ± 0.00 (7.0)	0.16 ± 0.00 (2.0)	0.54 ± 0.00 (6.0)	0.16 ± 0.00 (2.0)
PowerCons	1.15 ± 0.27 (4.0)	0.85 ± 0.14 (1.0)	— (7.0)	1.34 ± 0.24 (5.0)	0.85 ± 0.14 (2.5)	1.48 ± 0.21 (6.0)	0.85 ± 0.14 (2.5)
ProximalPhalanxOutlineAgeGroup	0.15 ± 0.02 (4.0)	0.05 ± 0.02 (2.5)	— (6.5)	— (6.5)	0.05 ± 0.02 (2.5)	0.16 ± 0.04 (5.0)	0.05 ± 0.03 (1.0)
SonyAIBORobotSurface1	0.43 ± 0.06 (4.0)	0.22 ± 0.03 (2.0)	— (6.5)	— (6.5)	0.22 ± 0.03 (2.0)	0.63 ± 0.02 (5.0)	0.22 ± 0.03 (2.0)
SonyAIBORobotSurface2	0.46 ± 0.06 (1.0)	0.58 ± 0.28 (3.0)	0.85 ± 0.08 (5.0)	— (7.0)	0.58 ± 0.28 (3.0)	0.93 ± 0.00 (6.0)	0.58 ± 0.28 (3.0)
ToeSegmentation1	0.63 ± 0.13 (4.0)	0.22 ± 0.09 (2.0)	— (7.0)	0.86 ± 0.09 (5.0)	0.22 ± 0.09 (2.0)	1.02 ± 0.08 (6.0)	0.22 ± 0.09 (2.0)
ToeSegmentation2	0.84 ± 0.18 (4.0)	0.24 ± 0.12 (2.0)	— (6.5)	— (6.5)	0.24 ± 0.12 (2.0)	1.20 ± 0.13 (5.0)	0.24 ± 0.12 (2.0)
TwoLeadECG	0.22 ± 0.13 (4.0)	0.07 ± 0.03 (2.0)	0.35 ± 0.00 (6.0)	— (7.0)	0.07 ± 0.03 (2.0)	0.28 ± 0.10 (5.0)	0.07 ± 0.03 (2.0)
UMD	0.78 ± 0.08 (3.0)	0.37 ± 0.33 (1.0)	— (7.0)	0.44 ± 0.28 (2.0)	0.86 ± 0.08 (4.5)	1.02 ± 0.02 (6.0)	0.86 ± 0.08 (4.5)
Rank Mean	3.27	2.15	5.63	6.08	2.77	5.63	2.46
Number of Wins	6	18	1	0	12	0	14

Table 10: Dataset-wise sparsity (↑) in the supervised setting, including ablations.

	CELT	CTCF	CVAE	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	0.78 ± 0.09 (3.0)	0.85 ± 0.06 (2.0)	0.57 ± 0.15 (5.0)	0.51 ± 0.11 (8.0)	0.63 ± 0.14 (4.0)	0.55 ± 0.20 (6.0)	0.53 ± 0.18 (7.0)	0.88 ± 0.04 (1.0)
BirdChicken	0.42 ± 0.20 (4.0)	0.90 ± 0.21 (3.0)	0.31 ± 0.12 (5.0)	— (7.5)	— (7.5)	0.98 ± 0.02 (2.0)	0.16 ± 0.00 (6.0)	0.98 ± 0.02 (1.0)
CricketX	0.29 ± 0.15 (1.0)	0.11 ± 0.04 (4.0)	— (7.0)	— (7.0)	— (7.0)	0.11 ± 0.00 (3.0)	0.09 ± 0.03 (5.0)	0.13 ± 0.03 (2.0)
CricketZ	0.35 ± 0.16 (1.0)	0.15 ± 0.06 (3.0)	0.23 ± 0.11 (2.0)	— (7.0)	— (7.0)	— (7.0)	0.12 ± 0.04 (5.0)	0.14 ± 0.05 (4.0)
DistalPhalanxOutlineAgeGroup	0.84 ± 0.00 (3.0)	0.85 ± 0.11 (2.0)	0.79 ± 0.00 (4.0)	0.50 ± 0.10 (7.0)	— (8.0)	0.59 ± 0.04 (5.0)	0.57 ± 0.11 (6.0)	0.86 ± 0.10 (1.0)
ECGFiveDays	0.73 ± 0.14 (3.0)	0.84 ± 0.09 (1.0)	0.64 ± 0.13 (5.0)	0.61 ± 0.20 (6.0)	0.66 ± 0.01 (4.0)	0.59 ± 0.14 (7.0)	0.49 ± 0.16 (8.0)	0.81 ± 0.11 (2.0)
FaceAll	0.24 ± 0.08 (2.0)	0.27 ± 0.14 (1.0)	0.17 ± 0.03 (5.0)	— (7.0)	— (7.0)	— (7.0)	0.18 ± 0.05 (4.0)	0.18 ± 0.04 (3.0)
FacesUCR	0.25 ± 0.07 (4.0)	0.29 ± 0.16 (3.0)	0.17 ± 0.02 (5.0)	— (7.5)	— (7.5)	0.40 ± 0.19 (1.0)	0.17 ± 0.03 (6.0)	0.36 ± 0.20 (2.0)
FiftyWords	0.55 ± 0.10 (2.0)	0.53 ± 0.25 (3.0)	0.29 ± 0.13 (6.0)	0.15 ± 0.02 (8.0)	0.32 ± 0.09 (5.0)	0.32 ± 0.11 (4.0)	0.18 ± 0.05 (7.0)	0.55 ± 0.27 (1.0)
GunPoint	0.80 ± 0.12 (3.0)	0.85 ± 0.13 (1.0)	0.72 ± 0.17 (4.0)	0.28 ± 0.15 (8.0)	0.47 ± 0.16 (5.0)	0.46 ± 0.25 (6.0)	0.32 ± 0.14 (7.0)	0.84 ± 0.16 (2.0)
GunPointAgeSpan	0.66 ± 0.10 (3.0)	0.90 ± 0.11 (2.0)	0.51 ± 0.34 (6.0)	0.28 ± 0.18 (7.0)	0.59 ± 0.12 (5.0)	0.62 ± 0.38 (4.0)	0.22 ± 0.16 (8.0)	0.93 ± 0.07 (1.0)
GunPointMaleVersusFemale	0.40 ± 0.27 (2.0)	0.28 ± 0.20 (3.0)	0.53 ± 0.28 (1.0)	0.01 ± 0.03 (8.0)	0.06 ± 0.08 (7.0)	0.20 ± 0.01 (5.0)	0.18 ± 0.11 (6.0)	0.28 ± 0.25 (4.0)
GunPointOldVersusYoung	0.13 ± 0.18 (5.0)	0.23 ± 0.29 (1.0)	0.11 ± 0.08 (6.0)	0.17 ± 0.23 (3.0)	0.10 ± 0.13 (7.0)	0.16 ± 0.13 (4.0)	0.07 ± 0.11 (8.0)	0.22 ± 0.26 (2.0)
InsectEPGRegularTrain	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.24 ± 0.01 (1.0)	0.00 ± 0.00 (5.5)	0.03 ± 0.07 (2.0)
InsectEPGSmallTrain	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.00 ± 0.00 (5.5)	0.25 ± 0.00 (1.0)	0.00 ± 0.00 (5.5)	0.01 ± 0.01 (2.0)
InsectWingbeatSound	0.59 ± 0.10 (1.0)	0.45 ± 0.14 (4.0)	0.38 ± 0.16 (7.0)	0.43 ± 0.15 (6.0)	0.55 ± 0.11 (2.0)	0.44 ± 0.16 (5.0)	0.35 ± 0.11 (8.0)	0.53 ± 0.16 (3.0)
ItalyPowerDemand	0.28 ± 0.19 (6.0)	0.48 ± 0.18 (1.0)	0.38 ± 0.13 (3.0)	0.27 ± 0.13 (8.0)	0.34 ± 0.05 (5.0)	0.36 ± 0.21 (4.0)	0.28 ± 0.16 (7.0)	0.43 ± 0.25 (2.0)
MoteStrain	0.36 ± 0.10 (3.0)	0.54 ± 0.09 (1.0)	0.18 ± 0.03 (7.0)	0.28 ± 0.07 (5.0)	— (8.0)	0.30 ± 0.08 (4.0)	0.23 ± 0.07 (6.0)	0.41 ± 0.20 (2.0)
PowerCons	0.54 ± 0.11 (3.0)	0.75 ± 0.12 (1.0)	0.35 ± 0.08 (4.0)	0.21 ± 0.11 (8.0)	0.33 ± 0.15 (5.0)	0.31 ± 0.17 (7.0)	0.32 ± 0.13 (6.0)	0.73 ± 0.15 (2.0)
ProximalPhalanxOutlineAgeGroup	0.75 ± 0.00 (7.0)	0.88 ± 0.12 (3.0)	0.95 ± 0.06 (1.0)	0.81 ± 0.13 (6.0)	— (8.0)	0.81 ± 0.00 (5.0)	0.86 ± 0.11 (4.0)	0.92 ± 0.11 (2.0)
SonyAIBORobotSurface1	0.40 ± 0.12 (1.0)	0.37 ± 0.09 (3.0)	0.32 ± 0.06 (5.0)	— (7.5)	— (7.5)	0.33 ± 0.05 (4.0)	0.26 ± 0.07 (6.0)	0.39 ± 0.12 (2.0)
SonyAIBORobotSurface2	0.34 ± 0.08 (2.0)	0.29 ± 0.19 (3.0)	0.24 ± 0.06 (5.0)	— (7.5)	— (7.5)	0.27 ± 0.02 (4.0)	0.20 ± 0.09 (6.0)	0.37 ± 0.18 (1.0)
ToeSegmentation1	0.42 ± 0.16 (3.0)	0.48 ± 0.24 (2.0)	0.20 ± 0.07 (5.0)	— (7.5)	— (7.5)	0.24 ± 0.10 (4.0)	0.15 ± 0.06 (6.0)	0.50 ± 0.24 (1.0)
ToeSegmentation2	0.37 ± 0.18 (4.0)	0.76 ± 0.13 (3.0)	0.19 ± 0.09 (6.0)	0.17 ± 0.00 (8.0)	0.21 ± 0.02 (5.0)	0.79 ± 0.11 (1.0)	0.17 ± 0.10 (7.0)	0.76 ± 0.12 (2.0)
TwoLeadECG	0.62 ± 0.15 (4.0)	0.91 ± 0.04 (2.0)	0.77 ± 0.08 (3.0)	0.49 ± 0.16 (6.0)	0.41 ± 0.14 (7.0)	0.62 ± 0.16 (5.0)	0.39 ± 0.12 (8.0)	0.92 ± 0.03 (1.0)
UMD	0.71 ± 0.16 (3.0)	0.86 ± 0.04 (1.0)	0.47 ± 0.04 (8.0)	0.53 ± 0.00 (7.0)	0.61 ± 0.12 (6.0)	0.64 ± 0.11 (4.0)	0.61 ± 0.09 (5.0)	0.73 ± 0.09 (2.0)
Rank Mean	3.23	2.46	4.85	6.87	6.17	4.23	6.27	1.92
Number of Wins	4	8	2	0	0	4	0	8

Table 11: Dataset-wise sparsity (↑) in the one-class setting, including ablations.

	CELT	CTCF	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	0.48 ± 0.04 (4.0)	0.52 ± 0.04 (2.0)	0.43 ± 0.00 (5.0)	— (7.0)	0.52 ± 0.04 (2.0)	0.37 ± 0.08 (6.0)	0.52 ± 0.04 (2.0)
BirdChicken	0.07 ± 0.00 (5.0)	0.92 ± 0.00 (2.0)	— (6.5)	0.17 ± 0.00 (4.0)	0.92 ± 0.00 (2.0)	— (6.5)	0.92 ± 0.00 (2.0)
CricketX	0.45 ± 0.04 (1.0)	0.20 ± 0.12 (4.0)	0.09 ± 0.03 (6.0)	— (7.0)	0.20 ± 0.12 (2.5)	0.14 ± 0.12 (5.0)	0.20 ± 0.12 (2.5)
CricketZ	0.49 ± 0.06 (1.0)	0.49 ± 0.33 (4.0)	0.09 ± 0.02 (7.0)	0.39 ± 0.04 (5.0)	0.49 ± 0.32 (2.5)	0.12 ± 0.06 (6.0)	0.49 ± 0.32 (2.5)
DistalPhalanxOutlineAgeGroup	0.50 ± 0.12 (4.0)	0.77 ± 0.13 (2.0)	— (6.5)	— (6.5)	0.77 ± 0.13 (2.0)	0.43 ± 0.12 (5.0)	0.77 ± 0.13 (2.0)
ECGFiveDays	0.70 ± 0.09 (2.0)	0.71 ± 0.14 (1.0)	0.33 ± 0.00 (6.0)	— (7.0)	0.63 ± 0.18 (4.0)	0.44 ± 0.19 (5.0)	0.70 ± 0.14 (3.0)
FaceAll	0.21 ± 0.04 (1.0)	0.19 ± 0.06 (3.0)	0.15 ± 0.04 (5.0)	— (7.0)	0.19 ± 0.06 (3.0)	0.15 ± 0.03 (6.0)	0.19 ± 0.06 (3.0)
FacesUCR	0.17 ± 0.04 (4.0)	0.20 ± 0.11 (2.0)	0.13 ± 0.02 (5.0)	— (7.0)	0.20 ± 0.11 (2.0)	0.12 ± 0.04 (6.0)	0.20 ± 0.11 (2.0)
FiftyWords	0.26 ± 0.08 (4.0)	0.70 ± 0.26 (1.0)	0.17 ± 0.00 (5.0)	— (7.0)	0.70 ± 0.26 (2.5)	0.16 ± 0.04 (6.0)	0.70 ± 0.26 (2.5)
GunPoint	0.29 ± 0.25 (5.0)	0.60 ± 0.38 (2.0)	0.14 ± 0.08 (6.0)	— (7.0)	0.60 ± 0.38 (2.0)	0.31 ± 0.28 (4.0)	0.60 ± 0.38 (2.0)
GunPointAgeSpan	0.42 ± 0.13 (4.0)	0.65 ± 0.29 (1.0)	0.18 ± 0.16 (6.0)	0.27 ± 0.17 (5.0)	0.48 ± 0.25 (3.0)	0.10 ± 0.15 (7.0)	0.63 ± 0.23 (2.0)
GunPointMaleVersusFemale	0.35 ± 0.24 (2.0)	0.48 ± 0.35 (1.0)	0.32 ± 0.12 (3.0)	0.09 ± 0.09 (7.0)	0.31 ± 0.29 (5.0)	0.19 ± 0.17 (6.0)	0.32 ± 0.29 (4.0)
GunPointOldVersusYoung	0.05 ± 0.08 (6.0)	0.15 ± 0.24 (2.0)	0.00 ± 0.00 (7.0)	0.07 ± 0.13 (4.0)	0.13 ± 0.24 (3.0)	0.06 ± 0.11 (5.0)	0.16 ± 0.25 (1.0)
InsectEPGRegularTrain	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)	— (7.0)	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)
InsectEPGSmallTrain	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)	— (7.0)	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)	0.00 ± 0.00 (3.5)
InsectWingbeatSound	0.50 ± 0.06 (1.0)	0.47 ± 0.07 (3.0)	0.44 ± 0.03 (5.0)	— (7.0)	0.47 ± 0.07 (3.0)	0.44 ± 0.07 (6.0)	0.47 ± 0.07 (3.0)
ItalyPowerDemand	0.30 ± 0.12 (4.0)	0.49 ± 0.22 (3.0)	0.24 ± 0.13 (6.0)	— (7.0)	0.49 ± 0.22 (1.5)	0.29 ± 0.15 (5.0)	0.49 ± 0.22 (1.5)
MoteStrain	0.40 ± 0.00 (4.0)	0.68 ± 0.00 (2.0)	0.20 ± 0.00 (6.0)	0.15 ± 0.00 (7.0)	0.68 ± 0.00 (2.0)	0.25 ± 0.00 (5.0)	0.68 ± 0.00 (2.0)
PowerCons	0.26 ± 0.17 (4.0)	0.39 ± 0.14 (2.0)	— (7.0)	0.16 ± 0.09 (5.0)	0.39 ± 0.14 (2.0)	0.13 ± 0.11 (6.0)	0.39 ± 0.14 (2.0)
ProximalPhalanxOutlineAgeGroup	0.73 ± 0.09 (4.0)	0.99 ± 0.02 (1.5)	— (6.5)	— (6.5)	0.99 ± 0.02 (1.5)	0.69 ± 0.11 (5.0)	0.97 ± 0.04 (3.0)
SonyAIBORobotSurface1	0.31 ± 0.07 (4.0)	0.63 ± 0.07 (2.0)	— (6.5)	— (6.5)	0.63 ± 0.07 (2.0)	0.20 ± 0.02 (5.0)	0.63 ± 0.07 (2.0)
SonyAIBORobotSurface2	0.33 ± 0.01 (4.0)	0.34 ± 0.14 (2.0)	0.16 ± 0.02 (6.0)	— (7.0)	0.34 ± 0.14 (2.0)	0.23 ± 0.00 (5.0)	0.34 ± 0.14 (2.0)
ToeSegmentation1	0.24 ± 0.09 (4.0)	0.67 ± 0.13 (2.0)	— (7.0)	0.15 ± 0.03 (6.0)	0.67 ± 0.13 (2.0)	0.17 ± 0.04 (5.0)	0.67 ± 0.13 (2.0)
ToeSegmentation2	0.22 ± 0.10 (4.0)	0.61 ± 0.21 (2.0)	— (6.5)	— (6.5)	0.61 ± 0.21 (2.0)	0.08 ± 0.04 (5.0)	0.61 ± 0.21 (2.0)
TwoLeadECG	0.63 ± 0.19 (4.0)	0.92 ± 0.05 (2.0)	0.33 ± 0.00 (6.0)	— (7.0)	0.92 ± 0.05 (2.0)	0.53 ± 0.16 (5.0)	0.92 ± 0.05 (2.0)
UMD	0.56 ± 0.02 (3.0)	0.69 ± 0.12 (1.0)	— (7.0)	0.60 ± 0.04 (2.0)	0.56 ± 0.01 (4.5)	0.53 ± 0.03 (6.0)	0.56 ± 0.01 (4.5)
Rank Mean	3.46	2.17	5.79	6.23	2.58	5.33	2.44
Number of Wins	6	20	2	0	16	2	16

Table 12: Dataset-wise NumSeg (↓) in the supervised setting, including ablations.

	CELT	CTCF	CVAE	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	3.89 ± 0.99 (5.0)	2.10 ± 0.83 (1.0)	4.40 ± 2.33 (6.0)	5.20 ± 2.23 (8.0)	5.11 ± 2.51 (7.0)	3.25 ± 1.09 (4.0)	3.10 ± 1.04 (3.0)	2.40 ± 0.92 (2.0)
BirdChicken	4.88 ± 2.03 (4.0)	0.78 ± 2.20 (3.0)	6.29 ± 1.67 (6.0)	— (7.5)	— (7.5)	0.00 ± 0.00 (1.5)	6.00 ± 0.00 (5.0)	0.00 ± 0.00 (1.5)
CricketX	7.10 ± 2.88 (1.0)	8.60 ± 3.01 (3.0)	— (7.0)	— (7.0)	— (7.0)	9.00 ± 1.00 (5.0)	8.50 ± 2.96 (2.0)	8.80 ± 1.72 (4.0)
CricketZ	9.33 ± 2.26 (5.0)	8.25 ± 2.17 (3.0)	8.57 ± 2.38 (4.0)	— (7.0)	— (7.0)	— (7.0)	8.00 ± 2.00 (2.0)	7.40 ± 3.26 (1.0)
DistalPhalanxOutlineAgeGroup	2.00 ± 0.00 (3.5)	1.50 ± 1.69 (1.0)	2.00 ± 0.00 (3.5)	6.60 ± 1.62 (7.0)	— (8.0)	5.67 ± 1.25 (6.0)	4.50 ± 1.50 (5.0)	1.70 ± 1.95 (2.0)
ECGFiveDays	1.90 ± 1.14 (3.0)	0.90 ± 0.94 (1.0)	2.30 ± 0.71 (4.0)	2.60 ± 0.88 (8.0)	2.50 ± 0.50 (7.0)	2.33 ± 1.89 (5.5)	2.33 ± 1.15 (5.5)	1.33 ± 0.82 (2.0)
FaceAll	11.30 ± 2.41 (4.0)	9.67 ± 2.58 (2.0)	11.40 ± 1.85 (5.0)	— (7.0)	— (7.0)	— (7.0)	9.00 ± 1.93 (1.0)	10.00 ± 1.63 (3.0)
FacesUCR	11.30 ± 2.61 (6.0)	9.00 ± 2.45 (2.5)	10.86 ± 1.81 (5.0)	— (7.5)	— (7.5)	9.00 ± 2.12 (2.5)	9.83 ± 2.79 (4.0)	8.75 ± 2.82 (1.0)
FiftyWords	2.80 ± 1.17 (1.0)	3.60 ± 1.43 (3.0)	4.60 ± 1.74 (7.0)	4.00 ± 0.00 (4.0)	5.67 ± 1.49 (8.0)	3.57 ± 0.73 (2.0)	4.29 ± 1.03 (5.0)	4.56 ± 2.59 (6.0)
GunPoint	1.90 ± 0.70 (2.0)	2.10 ± 1.70 (3.0)	2.62 ± 1.36 (5.0)	4.06 ± 1.48 (8.0)	3.67 ± 1.15 (7.0)	2.50 ± 1.12 (4.0)	3.20 ± 0.98 (6.0)	1.50 ± 1.12 (1.0)
GunPointAgeSpan	2.70 ± 1.35 (4.0)	2.80 ± 2.04 (6.0)	3.60 ± 1.91 (7.0)	2.75 ± 0.66 (5.0)	3.71 ± 1.39 (8.0)	1.43 ± 1.18 (1.0)	2.40 ± 1.20 (2.0)	2.50 ± 1.57 (3.0)
GunPointMaleVersusFemale	3.25 ± 1.30 (7.0)	2.44 ± 1.42 (4.0)	2.75 ± 1.48 (6.0)	1.14 ± 0.35 (1.0)	2.00 ± 1.34 (2.0)	2.50 ± 0.50 (5.0)	2.25 ± 1.30 (3.0)	3.60 ± 2.15 (8.0)
GunPointOldVersusYoung	1.90 ± 1.22 (7.0)	1.00 ± 0.00 (1.0)	2.00 ± 0.63 (8.0)	1.88 ± 1.36 (6.0)	1.60 ± 0.80 (3.0)	1.67 ± 0.47 (5.0)	1.40 ± 0.66 (2.0)	1.62 ± 0.70 (4.0)
InsectEPGRegularTrain	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)
InsectEPGSmallTrain	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)	1.00 ± 0.00 (4.5)
InsectWingbeatSound	3.70 ± 1.68 (1.0)	6.00 ± 2.05 (7.5)	5.29 ± 0.88 (4.0)	4.43 ± 2.66 (2.0)	5.50 ± 0.87 (5.0)	6.00 ± 2.45 (7.5)	5.75 ± 1.98 (6.0)	5.00 ± 2.45 (3.0)
ItalyPowerDemand	4.00 ± 1.22 (4.5)	3.50 ± 1.02 (1.0)	4.07 ± 0.77 (6.0)	4.24 ± 0.88 (7.0)	4.29 ± 0.70 (8.0)	3.80 ± 0.75 (2.5)	4.00 ± 1.00 (4.5)	3.80 ± 1.33 (2.5)
MoteStrain	4.62 ± 1.22 (2.0)	4.00 ± 1.41 (1.0)	5.40 ± 1.28 (5.5)	4.90 ± 0.70 (3.0)	— (8.0)	5.00 ± 0.63 (4.0)	5.60 ± 1.02 (7.0)	5.40 ± 1.69 (5.5)
PowerCons	6.00 ± 1.84 (4.0)	3.56 ± 1.77 (1.0)	7.20 ± 2.44 (8.0)	7.00 ± 1.89 (7.0)	5.25 ± 1.92 (3.0)	6.33 ± 1.25 (5.5)	6.33 ± 2.00 (5.5)	3.78 ± 2.48 (2.0)
ProximalPhalanxOutlineAgeGroup	2.00 ± 0.00 (7.0)	0.78 ± 1.13 (4.0)	0.20 ± 0.40 (2.0)	1.25 ± 1.30 (6.0)	— (8.0)	0.00 ± 0.00 (1.0)	1.00 ± 1.12 (5.0)	0.44 ± 0.96 (3.0)
SonyAIBORobotSurface1	6.20 ± 1.17 (1.0)	7.30 ± 1.62 (5.0)	7.00 ± 1.33 (2.5)	— (7.5)	— (7.5)	7.78 ± 1.31 (6.0)	7.10 ± 1.30 (4.0)	7.00 ± 2.05 (2.5)
SonyAIBORobotSurface2	5.89 ± 1.45 (5.0)	4.71 ± 1.98 (2.0)	6.11 ± 1.52 (6.0)	— (7.5)	— (7.5)	3.50 ± 1.50 (1.0)	5.50 ± 1.50 (4.0)	5.38 ± 1.65 (3.0)
ToeSegmentation1	6.90 ± 2.12 (2.0)	8.80 ± 3.60 (6.0)	8.70 ± 2.49 (5.0)	— (7.5)	— (7.5)	8.67 ± 1.37 (3.5)	8.67 ± 2.87 (3.5)	6.78 ± 3.99 (1.0)
ToeSegmentation2	5.33 ± 2.87 (5.0)	4.33 ± 2.87 (3.5)	7.33 ± 1.70 (7.0)	3.00 ± 0.00 (2.0)	10.33 ± 0.94 (8.0)	2.00 ± 0.82 (1.0)	6.67 ± 0.94 (6.0)	4.33 ± 3.40 (3.5)
TwoLeadECG	2.30 ± 0.90 (5.0)	0.50 ± 0.67 (2.0)	1.80 ± 0.98 (3.0)	3.10 ± 0.70 (6.0)	3.20 ± 0.98 (7.5)	2.00 ± 1.10 (4.0)	3.20 ± 1.60 (7.5)	0.40 ± 0.49 (1.0)
UMD	2.50 ± 0.50 (2.0)	3.80 ± 1.25 (6.0)	3.00 ± 0.63 (4.5)	4.00 ± 0.00 (7.0)	3.00 ± 1.00 (4.5)	1.67 ± 0.75 (1.0)	2.80 ± 1.33 (3.0)	5.60 ± 2.20 (8.0)
Rank Mean	3.85	3.13	5.23	5.94	6.52	3.90	4.25	3.17
Number of Wins	6	9	2	3	2	8	3	8

Table 13: Dataset-wise NumSeg (↓) in the one-class setting, including ablations.

	CELT	CTCF	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	3.00 ± 0.00 (5.0)	2.00 ± 0.00 (2.5)	12.00 ± 0.00 (6.0)	— (7.0)	2.00 ± 0.00 (2.5)	2.00 ± 0.00 (2.5)	2.00 ± 0.00 (2.5)
BirdChicken	7.00 ± 0.00 (4.0)	1.00 ± 0.00 (2.0)	— (6.5)	11.00 ± 0.00 (5.0)	1.00 ± 0.00 (2.0)	— (6.5)	1.00 ± 0.00 (2.0)
CricketX	10.25 ± 2.11 (6.0)	7.25 ± 1.56 (3.0)	5.20 ± 0.75 (1.0)	— (7.0)	7.29 ± 1.67 (4.5)	5.43 ± 1.84 (2.0)	7.29 ± 1.67 (4.5)
CricketZ	10.33 ± 0.75 (6.0)	5.67 ± 0.94 (2.0)	5.00 ± 0.71 (1.0)	14.67 ± 1.11 (7.0)	5.83 ± 1.07 (3.5)	6.00 ± 2.08 (5.0)	5.83 ± 1.07 (3.5)
DistalPhalanxOutlineAgeGroup	6.50 ± 1.89 (4.0)	2.67 ± 1.97 (2.0)	— (6.5)	— (6.5)	2.67 ± 1.97 (2.0)	7.00 ± 0.82 (5.0)	2.67 ± 1.97 (2.0)
ECGFiveDays	2.11 ± 1.10 (2.0)	2.11 ± 1.20 (2.0)	5.00 ± 0.00 (6.0)	— (7.0)	2.22 ± 1.03 (4.0)	2.33 ± 0.75 (5.0)	2.11 ± 1.20 (2.0)
FaceAll	11.40 ± 2.58 (6.0)	10.10 ± 2.07 (3.0)	9.33 ± 1.70 (1.0)	— (7.0)	10.10 ± 2.07 (3.0)	10.80 ± 2.40 (5.0)	10.10 ± 2.07 (3.0)
FacesUCR	10.90 ± 1.81 (6.0)	9.90 ± 2.02 (3.0)	9.83 ± 3.02 (1.0)	— (7.0)	9.90 ± 2.02 (3.0)	10.00 ± 2.68 (5.0)	9.90 ± 2.02 (3.0)
FiftyWords	5.10 ± 1.97 (6.0)	2.00 ± 1.10 (2.0)	5.00 ± 0.00 (5.0)	— (7.0)	2.00 ± 1.10 (2.0)	4.00 ± 0.94 (4.0)	2.00 ± 1.10 (2.0)
GunPoint	3.33 ± 1.56 (5.0)	2.50 ± 1.41 (2.0)	4.11 ± 1.20 (6.0)	— (7.0)	2.50 ± 1.41 (2.0)	3.00 ± 0.93 (4.0)	2.50 ± 1.41 (2.0)
GunPointAgeSpan	3.11 ± 0.99 (6.0)	3.00 ± 1.05 (4.0)	3.00 ± 2.20 (4.0)	2.56 ± 0.68 (2.0)	3.00 ± 1.12 (4.0)	1.88 ± 0.60 (1.0)	3.22 ± 1.69 (7.0)
GunPointMaleVersusFemale	2.90 ± 1.45 (6.0)	2.14 ± 0.64 (2.0)	4.00 ± 1.10 (7.0)	2.40 ± 1.20 (5.0)	2.17 ± 0.69 (3.5)	2.00 ± 0.58 (1.0)	2.17 ± 0.69 (3.5)
GunPointOldVersusYoung	1.20 ± 0.40 (2.0)	1.30 ± 0.46 (3.5)	1.00 ± 0.00 (1.0)	1.60 ± 1.02 (7.0)	1.40 ± 0.66 (5.5)	1.40 ± 0.66 (5.5)	1.30 ± 0.46 (3.5)
InsectEPGRegularTrain	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)	— (7.0)	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)
InsectEPGSmallTrain	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)	— (7.0)	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)	1.00 ± 0.00 (3.5)
InsectWingbeatSound	5.25 ± 0.97 (6.0)	5.00 ± 0.58 (3.0)	4.75 ± 0.83 (1.0)	— (7.0)	5.00 ± 0.58 (3.0)	5.20 ± 0.75 (5.0)	5.00 ± 0.58 (3.0)
ItalyPowerDemand	4.86 ± 1.46 (6.0)	3.00 ± 1.18 (2.0)	4.05 ± 1.32 (4.0)	— (7.0)	3.00 ± 1.18 (2.0)	4.10 ± 1.14 (5.0)	3.00 ± 1.18 (2.0)
MoteStrain	5.00 ± 0.00 (3.5)	5.00 ± 0.00 (3.5)	5.00 ± 0.00 (3.5)	6.00 ± 0.00 (7.0)	5.00 ± 0.00 (3.5)	5.00 ± 0.00 (3.5)	5.00 ± 0.00 (3.5)
PowerCons	6.25 ± 1.48 (5.5)	5.25 ± 0.43 (3.5)	— (7.0)	6.25 ± 2.49 (5.5)	5.00 ± 0.71 (2.0)	4.50 ± 1.50 (1.0)	5.25 ± 0.43 (3.5)
ProximalPhalanxOutlineAgeGroup	3.57 ± 2.06 (4.5)	0.29 ± 0.45 (2.0)	— (6.5)	— (6.5)	0.29 ± 0.45 (2.0)	3.57 ± 2.38 (4.5)	0.29 ± 0.45 (2.0)
SonyAIBORobotSurface1	6.33 ± 2.05 (4.0)	4.00 ± 0.00 (2.0)	— (6.5)	— (6.5)	4.00 ± 0.00 (2.0)	7.67 ± 1.70 (5.0)	4.00 ± 0.00 (2.0)
SonyAIBORobotSurface2	5.50 ± 0.50 (2.5)	5.50 ± 0.50 (2.5)	6.00 ± 0.00 (5.5)	— (7.0)	5.50 ± 0.50 (2.5)	6.00 ± 0.00 (5.5)	5.50 ± 0.50 (2.5)
ToeSegmentation1	9.25 ± 1.64 (6.0)	7.00 ± 3.00 (2.0)	— (7.0)	9.00 ± 2.83 (5.0)	7.00 ± 3.00 (2.0)	7.25 ± 2.38 (4.0)	7.00 ± 3.00 (2.0)
ToeSegmentation2	9.80 ± 4.96 (2.0)	10.20 ± 8.80 (4.0)	— (6.5)	— (6.5)	10.20 ± 8.80 (4.0)	6.33 ± 2.36 (1.0)	10.20 ± 8.80 (4.0)
TwoLeadECG	1.90 ± 0.83 (4.0)	0.90 ± 0.54 (2.0)	3.00 ± 0.00 (6.0)	— (7.0)	0.90 ± 0.54 (2.0)	2.50 ± 0.96 (5.0)	0.90 ± 0.54 (2.0)
UMD	2.67 ± 0.94 (4.0)	6.00 ± 2.94 (6.0)	— (7.0)	4.33 ± 1.70 (5.0)	2.33 ± 0.47 (2.0)	2.33 ± 0.47 (2.0)	2.33 ± 0.47 (2.0)
Rank Mean	4.58	2.79	4.60	6.37	2.90	3.85	2.92
Number of Wins	5	15	9	0	15	9	16

Table 14: Dataset-wise implausibility (↓) in the supervised setting, including ablations.

	CELT	CTCF	CVAE	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	20.40 ± 7.27 (5.0)	13.28 ± 3.69 (2.0)	26.61 ± 12.18 (7.0)	33.56 ± 8.23 (8.0)	19.83 ± 7.41 (4.0)	17.59 ± 4.93 (3.0)	22.12 ± 4.31 (6.0)	11.38 ± 3.32 (1.0)
BirdChicken	170.44 ± 66.88 (5.0)	46.28 ± 80.26 (3.0)	141.11 ± 42.92 (4.0)	— (7.5)	— (7.5)	17.94 ± 5.10 (1.0)	299.22 ± 0.00 (6.0)	18.04 ± 5.20 (2.0)
CricketX	121.83 ± 38.02 (1.0)	179.98 ± 39.45 (2.0)	— (7.0)	— (7.0)	— (7.0)	232.51 ± 2.98 (5.0)	197.19 ± 15.68 (3.0)	209.35 ± 10.69 (4.0)
CricketZ	92.74 ± 19.69 (1.0)	212.64 ± 19.52 (4.0)	126.52 ± 60.65 (2.0)	— (7.0)	— (7.0)	— (7.0)	218.82 ± 20.10 (5.0)	196.85 ± 12.91 (3.0)
DistalPhalanxOutlineAgeGroup	6.18 ± 0.00 (2.0)	6.04 ± 2.03 (1.0)	8.51 ± 0.00 (6.0)	7.53 ± 1.17 (4.0)	— (8.0)	9.21 ± 2.24 (7.0)	7.86 ± 0.89 (5.0)	6.23 ± 1.82 (3.0)
ECGFiveDays	15.62 ± 5.25 (4.0)	12.51 ± 4.33 (1.0)	13.93 ± 3.48 (2.0)	31.34 ± 2.47 (8.0)	19.83 ± 3.96 (6.0)	18.52 ± 4.19 (5.0)	23.01 ± 9.02 (7.0)	14.59 ± 4.77 (3.0)
FaceAll	64.52 ± 15.28 (4.0)	64.38 ± 13.33 (2.0)	42.32 ± 10.10 (1.0)	— (7.0)	— (7.0)	— (7.0)	64.81 ± 15.55 (5.0)	64.42 ± 4.99 (3.0)
FacesUCR	47.14 ± 9.12 (4.0)	48.15 ± 12.49 (6.0)	42.08 ± 10.34 (1.0)	— (7.5)	— (7.5)	45.91 ± 19.83 (3.0)	45.59 ± 13.36 (2.0)	47.92 ± 18.88 (5.0)
FiftyWords	87.12 ± 14.40 (4.0)	83.93 ± 49.05 (3.0)	76.50 ± 14.59 (1.0)	162.58 ± 0.54 (8.0)	114.25 ± 38.13 (5.0)	132.85 ± 23.48 (7.0)	123.77 ± 24.14 (6.0)	79.33 ± 48.01 (2.0)
GunPoint	22.75 ± 12.12 (4.0)	16.40 ± 10.75 (2.0)	19.61 ± 9.82 (3.0)	47.89 ± 24.85 (8.0)	29.00 ± 6.19 (6.0)	26.13 ± 19.14 (5.0)	32.13 ± 15.53 (7.0)	15.11 ± 10.86 (1.0)
GunPointAgeSpan	18.01 ± 6.71 (4.0)	9.47 ± 7.70 (2.0)	27.42 ± 22.87 (6.0)	44.34 ± 16.17 (8.0)	18.45 ± 7.49 (5.0)	11.94 ± 10.35 (3.0)	38.43 ± 27.93 (7.0)	8.68 ± 5.92 (1.0)
GunPointMaleVersusFemale	19.10 ± 12.60 (2.0)	53.79 ± 36.74 (5.0)	13.85 ± 4.52 (1.0)	57.80 ± 5.24 (6.0)	32.69 ± 16.16 (3.0)	82.83 ± 25.68 (8.0)	64.70 ± 37.22 (7.0)	52.71 ± 35.53 (4.0)
GunPointOldVersusYoung	17.17 ± 8.42 (3.0)	61.77 ± 18.26 (8.0)	26.48 ± 29.49 (4.0)	38.10 ± 17.62 (6.0)	9.31 ± 3.31 (1.0)	14.50 ± 5.72 (2.0)	40.40 ± 18.57 (7.0)	34.87 ± 13.76 (5.0)
InsectEPGRegularTrain	11.17 ± 1.02 (1.0)	228.79 ± 4.51 (5.0)	15.89 ± 5.09 (2.0)	370.37 ± 15.48 (6.0)	20.77 ± 0.00 (3.0)	535.55 ± 24.28 (8.0)	217.55 ± 9.56 (4.0)	408.38 ± 66.93 (7.0)
InsectEPGSmallTrain	13.32 ± 1.63 (1.0)	172.32 ± 10.21 (4.0)	15.39 ± 3.03 (2.0)	308.25 ± 15.84 (6.0)	21.17 ± 0.00 (3.0)	506.46 ± 0.00 (8.0)	181.38 ± 4.35 (5.0)	433.09 ± 69.66 (7.0)
InsectWingbeatSound	71.22 ± 15.96 (2.0)	103.86 ± 10.90 (7.0)	53.72 ± 20.64 (1.0)	99.62 ± 4.90 (6.0)	78.35 ± 12.57 (3.0)	111.66 ± 18.93 (8.0)	98.42 ± 25.03 (5.0)	94.99 ± 11.42 (4.0)
ItalyPowerDemand	5.15 ± 2.99 (7.0)	3.81 ± 1.46 (5.0)	3.92 ± 1.81 (6.0)	5.20 ± 1.58 (8.0)	3.22 ± 1.80 (3.0)	2.82 ± 1.41 (2.0)	2.27 ± 0.49 (1.0)	3.37 ± 1.21 (4.0)
MoteStrain	27.89 ± 4.62 (5.0)	24.76 ± 5.72 (3.0)	19.71 ± 7.86 (1.0)	20.43 ± 0.35 (2.0)	— (8.0)	27.42 ± 5.51 (4.0)	32.00 ± 8.13 (7.0)	29.79 ± 8.52 (6.0)
PowerCons	46.36 ± 10.28 (4.0)	35.84 ± 21.19 (2.0)	51.77 ± 10.87 (6.0)	60.06 ± 1.05 (8.0)	40.94 ± 8.48 (3.0)	47.89 ± 8.83 (5.0)	59.30 ± 17.75 (7.0)	34.26 ± 20.13 (1.0)
ProximalPhalanxOutlineAgeGroup	3.77 ± 0.00 (2.0)	4.10 ± 1.18 (4.0)	3.45 ± 0.51 (1.0)	5.62 ± 2.19 (7.0)	— (8.0)	5.14 ± 0.00 (5.0)	5.16 ± 1.00 (6.0)	4.03 ± 0.95 (3.0)
SonyAIBORobotSurface1	22.18 ± 3.53 (1.0)	24.14 ± 2.40 (4.0)	23.53 ± 2.27 (2.0)	— (7.5)	— (7.5)	24.73 ± 2.66 (5.0)	25.39 ± 3.11 (6.0)	23.55 ± 3.50 (3.0)
SonyAIBORobotSurface2	28.60 ± 8.21 (4.0)	26.58 ± 7.12 (3.0)	26.38 ± 7.45 (2.0)	— (7.5)	— (7.5)	35.53 ± 1.03 (6.0)	29.89 ± 4.04 (5.0)	25.79 ± 6.99 (1.0)
ToeSegmentation1	108.02 ± 29.42 (1.0)	120.84 ± 67.17 (4.0)	119.44 ± 27.01 (3.0)	— (7.5)	— (7.5)	195.40 ± 11.91 (5.0)	198.12 ± 15.88 (6.0)	113.34 ± 64.46 (2.0)
ToeSegmentation2	158.06 ± 7.16 (4.0)	62.36 ± 25.14 (2.0)	195.51 ± 13.10 (6.0)	179.30 ± 0.00 (5.0)	197.82 ± 12.61 (7.0)	88.98 ± 70.01 (3.0)	221.63 ± 9.21 (8.0)	60.75 ± 21.54 (1.0)
TwoLeadECG	7.13 ± 1.53 (4.0)	5.17 ± 0.69 (2.0)	7.33 ± 2.03 (5.0)	10.41 ± 1.32 (7.0)	12.88 ± 0.00 (8.0)	5.56 ± 1.04 (3.0)	7.63 ± 1.91 (6.0)	4.55 ± 1.00 (1.0)
UMD	18.15 ± 1.85 (2.0)	15.57 ± 4.57 (1.0)	48.00 ± 19.73 (7.0)	68.45 ± 0.00 (8.0)	26.79 ± 1.38 (4.0)	29.77 ± 5.29 (5.0)	30.49 ± 10.25 (6.0)	26.10 ± 10.29 (3.0)
Rank Mean	3.12	3.35	3.42	6.83	5.63	5.00	5.58	3.08
Number of Wins	6	3	7	0	1	1	1	7

Table 15: Dataset-wise implausibility (↓) in the one-class setting, including ablations.

	CELT	CTCF	LASTS	MCCE	w/o proj	w/o steer	w/o time
BME	23.29 ± 0.28 (4.0)	18.80 ± 3.30 (2.0)	66.13 ± 0.00 (6.0)	— (7.0)	18.80 ± 3.30 (2.0)	23.30 ± 10.06 (5.0)	18.80 ± 3.30 (2.0)
BirdChicken	254.06 ± 0.00 (5.0)	29.68 ± 0.00 (1.0)	— (6.5)	213.41 ± 0.00 (4.0)	31.24 ± 0.00 (2.5)	— (6.5)	31.24 ± 0.00 (2.5)
CricketX	82.78 ± 4.46 (1.0)	190.51 ± 46.63 (6.0)	179.33 ± 24.22 (2.0)	— (7.0)	181.11 ± 42.16 (4.5)	180.11 ± 46.59 (3.0)	181.11 ± 42.16 (4.5)
CricketZ	82.75 ± 9.49 (1.0)	125.85 ± 89.05 (3.0)	184.18 ± 18.79 (6.0)	123.36 ± 15.08 (2.0)	130.72 ± 93.70 (4.5)	196.42 ± 41.93 (7.0)	130.72 ± 93.70 (4.5)
DistalPhalanxOutlineAgeGroup	7.32 ± 0.74 (1.0)	8.40 ± 2.78 (3.0)	— (6.5)	— (6.5)	8.40 ± 2.78 (3.0)	9.20 ± 0.96 (5.0)	8.40 ± 2.78 (3.0)
ECGFiveDays	18.66 ± 7.36 (5.0)	16.68 ± 4.34 (1.0)	58.44 ± 0.00 (6.0)	— (7.0)	18.04 ± 4.40 (4.0)	17.65 ± 5.38 (3.0)	16.70 ± 4.33 (2.0)
FaceAll	47.91 ± 12.22 (6.0)	44.48 ± 9.97 (3.0)	35.00 ± 0.76 (1.0)	— (7.0)	44.48 ± 9.97 (3.0)	46.36 ± 6.79 (5.0)	44.48 ± 9.97 (3.0)
FacesUCR	42.46 ± 7.58 (3.0)	44.96 ± 7.75 (5.0)	39.60 ± 2.40 (1.0)	— (7.0)	44.96 ± 7.75 (5.0)	40.04 ± 10.35 (2.0)	44.96 ± 7.75 (5.0)
FiftyWords	85.07 ± 23.61 (6.0)	52.34 ± 30.59 (1.0)	80.13 ± 0.00 (4.0)	— (7.0)	52.45 ± 30.49 (3.0)	82.07 ± 15.87 (5.0)	52.41 ± 30.48 (2.0)
GunPoint	16.93 ± 3.83 (5.0)	13.77 ± 6.08 (2.0)	26.99 ± 2.27 (6.0)	— (7.0)	13.77 ± 6.08 (2.0)	15.94 ± 5.56 (4.0)	13.77 ± 6.08 (2.0)
GunPointAgeSpan	23.72 ± 5.16 (4.0)	23.53 ± 21.45 (3.0)	36.26 ± 4.66 (7.0)	14.20 ± 2.93 (1.0)	26.50 ± 21.18 (5.0)	34.02 ± 31.25 (6.0)	19.66 ± 10.33 (2.0)
GunPointMaleVersusFemale	14.40 ± 9.91 (1.0)	29.07 ± 19.01 (5.0)	43.08 ± 21.77 (7.0)	38.33 ± 2.43 (6.0)	24.99 ± 15.39 (3.0)	21.14 ± 8.49 (2.0)	26.97 ± 16.03 (4.0)
GunPointOldVersusYoung	10.78 ± 4.66 (2.0)	26.87 ± 6.23 (7.0)	23.71 ± 11.54 (6.0)	5.03 ± 1.00 (1.0)	13.94 ± 9.94 (4.0)	12.25 ± 6.36 (3.0)	16.57 ± 10.61 (5.0)
InsectEPGRegularTrain	21.38 ± 10.59 (2.0)	30.61 ± 7.93 (4.0)	11.63 ± 1.40 (1.0)	— (7.0)	30.61 ± 7.93 (4.0)	31.87 ± 8.36 (6.0)	30.61 ± 7.93 (4.0)
InsectEPGSmallTrain	13.11 ± 2.02 (1.0)	30.93 ± 8.98 (3.0)	42.95 ± 30.75 (6.0)	— (7.0)	30.93 ± 8.98 (3.0)	34.14 ± 7.76 (5.0)	30.93 ± 8.98 (3.0)
InsectWingbeatSound	51.96 ± 8.41 (1.0)	73.79 ± 19.92 (4.0)	53.20 ± 0.46 (2.0)	— (7.0)	73.79 ± 19.92 (4.0)	75.13 ± 26.95 (6.0)	73.79 ± 19.92 (4.0)
ItalyPowerDemand	4.04 ± 1.40 (6.0)	3.00 ± 0.85 (2.0)	3.15 ± 0.16 (5.0)	— (7.0)	3.06 ± 0.87 (3.5)	2.61 ± 0.67 (1.0)	3.06 ± 0.87 (3.5)
MoteStrain	14.83 ± 0.00 (4.0)	13.53 ± 0.00 (2.0)	19.48 ± 0.00 (6.0)	27.60 ± 0.00 (7.0)	13.53 ± 0.00 (2.0)	17.42 ± 0.00 (5.0)	13.53 ± 0.00 (2.0)
PowerCons	69.41 ± 16.16 (6.0)	48.46 ± 6.21 (4.0)	— (7.0)	51.43 ± 29.26 (5.0)	48.09 ± 6.53 (2.0)	39.75 ± 4.86 (1.0)	48.26 ± 6.38 (3.0)
ProximalPhalanxOutlineAgeGroup	3.89 ± 0.54 (4.0)	3.41 ± 1.17 (2.0)	— (6.5)	— (6.5)	3.43 ± 1.15 (3.0)	4.47 ± 1.41 (5.0)	3.38 ± 1.45 (1.0)
SonyAIBORobotSurface1	24.77 ± 0.43 (4.0)	14.98 ± 2.07 (2.0)	— (6.5)	— (6.5)	14.98 ± 2.07 (2.0)	29.55 ± 1.99 (5.0)	14.98 ± 2.07 (2.0)
SonyAIBORobotSurface2	23.28 ± 0.73 (1.0)	30.31 ± 9.97 (3.0)	39.59 ± 0.38 (6.0)	— (7.0)	30.31 ± 9.97 (3.0)	35.77 ± 0.00 (5.0)	30.31 ± 9.97 (3.0)
ToeSegmentation1	148.56 ± 14.92 (4.0)	61.27 ± 23.60 (2.0)	— (7.0)	211.70 ± 2.12 (6.0)	61.27 ± 23.60 (2.0)	195.32 ± 24.87 (5.0)	61.27 ± 23.60 (2.0)
ToeSegmentation2	170.90 ± 28.09 (4.0)	82.58 ± 41.13 (2.0)	— (6.5)	— (6.5)	82.58 ± 41.13 (2.0)	235.90 ± 10.83 (5.0)	82.58 ± 41.13 (2.0)
TwoLeadECG	5.28 ± 0.70 (4.0)	4.58 ± 0.87 (2.0)	11.69 ± 0.00 (6.0)	— (7.0)	4.58 ± 0.87 (2.0)	7.65 ± 1.44 (5.0)	4.58 ± 0.87 (2.0)
UMD	32.66 ± 8.15 (3.0)	24.54 ± 7.74 (1.0)	— (7.0)	29.77 ± 15.73 (2.0)	40.10 ± 5.73 (5.0)	39.23 ± 11.50 (4.0)	40.14 ± 5.71 (6.0)
Rank Mean	3.38	2.88	5.29	5.81	3.19	4.40	3.04
Number of Wins	7	11	3	2	7	2	8

Table 16: Comparison between CTCF's flow-time-dependent desired-class region and the endpoint-AE proxy. Validity is the average across 26 datasets, reported as CTCF / endpoint AE. For each setting, the first row of the other metrics reports win/loss/draw counts from CTCF's perspective and the second reports p-values from two-sided Wilcoxon signed-rank tests on common-valid instances. (n.s.) indicates no significant difference at $\alpha = 0.05$. ↑ and ↓ indicate that higher and lower values are better, respectively.

Setting	Validity ↑	Proximity ↓	Sparsity ↑	NumSeg ↓	Implausibility ↓
Supervised	**0.93** / 0.83	120/79/2 $p = 0.0161$	113/56/32 $p = 1.13 \times 10^{-6}$	76/54/71 $p = 0.0623$ (n.s.)	116/85/0 $p = 0.00821$
One-class	**0.97** / 0.95	73/55/1 $p = 0.223$ (n.s.)	68/45/16 $p = 0.0317$	54/26/49 $p = 2.66 \times 10^{-4}$	68/61/0 $p = 0.429$ (n.s.)

G.5 Comparison with an endpoint-AE feasibility proxy

The w/o time ablation isolates flow-time dependence within our DeepSVDD-style region, but does not compare this feasibility module with a classical endpoint proxy. We therefore performed a controlled proxy-replacement experiment using an endpoint autoencoder (AE).

The key design distinction is where feasibility is defined. The endpoint AE is trained on endpoints classified by f as the desired class. Although its reconstruction error is evaluated through the local endpoint estimate $\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)$ at every Euler step, its signal remains endpoint-based: it does not directly assess whether the current state $\boldsymbol{x}_t$ is consistent with desired-class interpolation states at flow time t. CTCF instead learns a time-indexed feature-space region from interpolation states labeled by the endpoint hard decision $f(\boldsymbol{x}_1)$, checks the predicted intermediate state at each Euler step, and applies a linearized correction when that state violates the region.

The comparison retains the unconditional FM backbone and the steering-and-correction framework, while replacing the feasibility module and its violation function. We chose an AE because it requires only learned parameters and a single threshold, rather than stored training instances, at deployment. It omits the time channel but otherwise uses the same three convolutional blocks and 128-dimensional pooled representation as CTCF's region encoder, followed by a two-layer MLP decoder. It was trained independently with mean-squared reconstruction error on training endpoints satisfying $f(\boldsymbol{x}_1) = y^{\mathrm{cf}}$ and is distinct from the AE used to initialize CTCF's region encoder.

Let $A_{\boldsymbol{\phi}}$ denote the endpoint AE and define

$$R_{\mathrm{AE}}(\boldsymbol{z}) := L^{-1}\|\boldsymbol{z} - A_{\boldsymbol{\phi}}(\boldsymbol{z})\|_2^2. \tag{42}$$

We set κ_{AE} to the 0.95-quantile of R_{AE} on desired-class training endpoints, using the same quantile level as CTCF, and replaced the violation in Eq. (14) with

$$e_t^{\mathrm{AE}}(\boldsymbol{x}) := R_{\mathrm{AE}}(\boldsymbol{x} + (1-t)\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)) - \gamma\kappa_{\mathrm{AE}}, \tag{43}$$

where $\boldsymbol{x} + (1-t)\boldsymbol{u}_{\boldsymbol{\theta}}(\boldsymbol{x}, t) = \hat{\boldsymbol{x}}_1(\boldsymbol{x})$ is the endpoint estimate in Eq. (39).

We evaluated both methods on the 26 UCR datasets used in Section 5. Validity was averaged across the datasets. The other metrics used valid CFs, and the paired comparisons followed the common-valid-set protocol in Section 5.3. Table 16 reports mean validity and the paired comparisons. Table 17 reports mean ranks across the 26 datasets, with ranks computed among all methods, namely CTCF, the endpoint-AE proxy, CELT, CVAE (supervised setting only), LASTS, MCCE, and the three ablations (w/o proj, w/o steer, and w/o time). Tables 31 and 32 report dataset-wise results for the endpoint-AE proxy in the supervised and one-class settings, respectively.

CTCF achieved higher average validity, more wins than losses on all four paired metrics, and better mean ranks on all five metrics in both settings. Five of the eight paired differences significantly favored CTCF: proximity, sparsity, and implausibility in the supervised setting, and sparsity and NumSeg in the one-class setting; the remaining three showed no significant difference. These results support the usefulness of CTCF's flow-time-indexed intermediate-state feasibility assessment relative to the tested endpoint-AE proxy within the same FM steering-and-correction framework.

Table 17: Mean ranks across the 26 datasets, including all methods, for CTCF and the endpoint-AE proxy. Each cell reports CTCF / endpoint AE; lower ranks are better.

Setting	Validity	Proximity	Sparsity	NumSeg	Implausibility
Supervised	**3.17** / 3.81	**2.77** / 3.69	**2.65** / 4.08	**3.44** / 4.06	**3.73** / 4.35
One-class	**3.46** / 3.60	**2.35** / 3.31	**2.54** / 3.44	**3.17** / 3.92	**3.35** / 3.50

Table 18: Results restricted to the 14 native binary datasets. Each cell reports mean rank / Number of Wins, with ties counted as wins. Lower mean ranks are better. Validity uses all evaluated instances; the other metrics use each method's own valid CFs. A dataset-method pair without a valid CF receives a tied-worst rank. CVAE does not support the one-class setting.

Setting	Metric	CELT	CTCF	CVAE	LASTS	MCCE	w/o proj	w/o steer	w/o time
Supervised	Validity	3.54 / 10	2.89 / 10	4.04 / 8	5.82 / 2	6.18 / 3	6.43 / 2	3.61 / 8	3.50 / 8
	Proximity	3.71 / 1	2.07 / 5	3.71 / 2	6.14 / 0	6.71 / 0	5.04 / 1	6.64 / 0	1.96 / 6
	Sparsity	3.29 / 1	1.93 / 6	4.64 / 1	6.93 / 0	6.29 / 0	4.36 / 1	6.79 / 0	1.79 / 5
	NumSeg	3.96 / 1	2.82 / 5	5.64 / 0	5.93 / 1	6.54 / 0	3.54 / 4	4.68 / 0	2.89 / 4
	Implausibility	3.71 / 2	3.29 / 1	3.64 / 2	6.86 / 0	5.71 / 1	4.07 / 1	6.21 / 1	2.50 / 6
One-class	Validity	2.96 / 13	2.96 / 12	—	6.04 / 2	5.14 / 6	3.32 / 11	4.46 / 6	3.11 / 12
	Proximity	3.71 / 1	1.82 / 12	—	5.82 / 1	6.07 / 0	2.71 / 7	5.61 / 0	2.25 / 8
	Sparsity	4.00 / 0	1.86 / 12	—	6.11 / 0	6.14 / 0	2.46 / 10	5.32 / 0	2.11 / 11
	NumSeg	4.18 / 3	2.64 / 9	—	5.46 / 2	6.04 / 0	2.93 / 8	3.79 / 5	2.96 / 9
	Implausibility	3.93 / 2	2.71 / 8	—	6.32 / 0	5.57 / 2	2.79 / 6	4.04 / 2	2.64 / 6

G.6 Native multiclass evaluation

CTCF is not structurally limited to binary classifiers. For any desired label y^{cf}, it uses only whether $f(\boldsymbol{x}) = y^{\mathrm{cf}}$: the desired label is treated as desired, and all other outputs are treated as non-desired. Thus, f may have any number of output classes, although a desired-class region is learned separately for each desired label.

In the original class-1/class-2 setting, we followed the protocol of Yamaguchi et al. [2026] to compare with CELT [Yamaguchi et al., 2026], CVAE [Todo et al., 2023], LASTS [Spinnato et al., 2023], and MCCE [Redelmeier et al., 2024] where supported; CVAE does not support the one-class setting. We used the validity, proximity, sparsity, and NumSeg values reported in Yamaguchi et al. [2026] and recomputed implausibility using held-out test data.

To test whether the conclusions depend on this protocol, we performed two complementary analyses: a reanalysis of the 14 native binary datasets and an all-class experiment on the remaining 12 native multiclass datasets.

Reanalysis of native binary datasets. Table 18 uses the original aggregation rules. Validity is computed over all evaluated instances, whereas the other four metrics use each method's own valid CFs. A dataset-method pair without a valid CF receives a tied-worst rank. Each cell reports mean rank / Number of Wins, with ties counted as wins; CVAE does not support the one-class setting.

On the 14 native binary datasets, CTCF's mean rank was at least as good as that of the best existing method in all 10 setting-metric combinations. Even when the three ablations were included, CTCF ranked first or second in every combination. Thus, the main result also holds on datasets that require no reduction from multiple classes to two classes.

All-class experiment on native multiclass datasets. For the additional supervised experiment on the 12 native multiclass datasets, which have 3–50 classes, we retained all classes. For each class, we retained up to the last 30 instances in the official UCR test split as the held-out test set and added any preceding test-split instances to the official training set. QUANT [Dempster et al., 2024] and the unconditional FM were trained on data from all classes, and a separate region learner was trained for

Table 19: Native multiclass supervised results with QUANT over 12 datasets. Mean validity is averaged over the datasets after macro-averaging over desired classes. The remaining columns report mean rank / Number of Wins, with ties counted as wins; lower mean ranks are better.

Method	Mean validity	Validity	Proximity	Sparsity	NumSeg	Implausibility
CTCF	0.789	1.33 / 11	1.67 / 6	1.75 / 6	1.92 / 5	2.00 / 4
w/o proj	0.254	4.00 / 0	2.75 / 1	2.67 / 3	2.08 / 6	2.92 / 2
w/o steer	0.712	2.25 / 2	3.92 / 0	3.75 / 0	2.92 / 2	3.00 / 2
w/o time	0.689	2.42 / 5	1.67 / 5	1.83 / 3	3.08 / 1	2.08 / 4

Table 20: Pairwise native multiclass results on common-valid instances. For each comparison, the first row reports win/loss/draw counts from CTCF's perspective and the second reports p-values from two-sided Wilcoxon signed-rank tests. n is the number of common-valid instances. (n.s.) indicates no significant difference at $\alpha = 0.05$. $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively.

Comparison	Proximity $\downarrow$	Sparsity $\uparrow$	NumSeg $\downarrow$	Implausibility $\downarrow$
vs. w/o proj ($n = 171$)	110/29/32 $p = 4.78 \times 10^{-13}$	87/43/41 $p = 1.40 \times 10^{-5}$	62/30/79 $p = 1.48 \times 10^{-3}$	97/44/30 $p = 1.23 \times 10^{-8}$
vs. w/o steer ($n = 617$)	598/16/3 $p = 2.97 \times 10^{-97}$	503/50/64 $p = 2.42 \times 10^{-76}$	276/140/201 $p = 2.99 \times 10^{-12}$	429/188/0 $p = 1.06 \times 10^{-27}$
vs. w/o time ($n = 473$)	227/231/15 $p = 7.53 \times 10^{-1}$ (n.s.)	235/195/43 $p = 8.47 \times 10^{-3}$	217/92/164 $p = 2.47 \times 10^{-15}$	271/197/5 $p = 7.51 \times 10^{-4}$

each desired label y^{cf}. For each desired class, we selected up to 10 correctly classified non-desired test instances in round-robin order over the non-desired classes in ascending class-label order, with the instances within each class shuffled using seed 1, before CF generation. The other settings were unchanged. Because the baseline results reused from Yamaguchi et al. [2026] are available only for the two-class protocol, this all-class extension compares CTCF with its three controlled ablations.

Validity was macro-averaged across desired classes and then averaged over the 12 datasets. The other four metrics were macro-averaged across desired classes for which valid CFs were obtained. All four methods generated at least one valid CF on every dataset, so ranks were available for all five metrics over all 12 datasets. Table 19 summarizes the mean validity, mean ranks, and Number of Wins. Table 33 provides dataset-wise results for validity, proximity, sparsity, and NumSeg.

For validity, the dataset-level win/loss/draw counts from the perspective of CTCF were 12/0/0 against w/o proj, 10/0/2 against w/o steer, and 7/1/4 against w/o time. Table 20 reports common-valid comparisons. W/L/D denotes win/loss/draw counts from CTCF's perspective, and p is from a two-sided Wilcoxon signed-rank test.

CTCF matched or outperformed all three ablations in mean rank on all five metrics and significantly improved 11 of the 12 common-valid comparisons, with no significant degradation. Projection had the largest effect on validity, which decreased from 0.789 to 0.254 without projection. Removing steering significantly worsened all four CF-quality metrics, while removing time dependence significantly worsened sparsity, NumSeg, and implausibility but not proximity. These results show that CTCF can operate with a classifier trained on all original classes and support the component-level trends found in the original two-class evaluation.

G.7 Evaluation with MultiROCKET-HYDRA

Architectural matching is not a design requirement of CTCF. The classifier f, desired-class region learner, and FM network do not share architectures, parameters, or internal representations. They communicate through hard decisions and flow-time states.

The classifier supplies only endpoint hard decisions. These decisions label the interpolation states used to train the desired-class region learner, which takes $(\boldsymbol{x}_t, t)$ as input and learns a region from interpolation states whose endpoints receive the desired-class hard decision. The FM network

Table 21: CTCF with the QUANT-based and MultiROCKET-HYDRA-based classifiers being explained. Values are dataset macro-averages. Validity uses all 26 datasets; the other metrics use datasets where CTCF generated valid CFs with both classifiers (25 supervised and 23 one-class). ↓ and ↑ indicate that lower and higher values are better, respectively.

Setting	Metric	QUANT-based	MultiROCKET-HYDRA-based
Supervised	Validity ↑	0.927	0.919
	Proximity ↓	0.45	0.44
	Sparsity ↑	0.55	0.55
	NumSeg ↓	4.04	4.29
	Implausibility ↓	58.58	64.32
One-class	Validity ↑	0.971	0.631
	Proximity ↓	0.45	0.58
	Sparsity ↑	0.56	0.44
	NumSeg ↓	4.42	4.60
	Implausibility ↓	41.58	51.96

Table 22: Supervised and one-class ablation results with the MultiROCKET-HYDRA-based classifier being explained. Each metric cell reports dataset macro-average / mean rank / Number of Wins. Validity averages, mean ranks, and win counts use all 26 datasets; the other metric averages use the 23 datasets in each setting where all four methods generated valid CFs. CF-quality ranks and wins use each method's own valid CFs, with tied-worst ranks assigned when no valid CF was generated and ties counted as wins. Lower mean ranks are better.

Setting	Method	Validity ↑	Proximity ↓	Sparsity ↑	NumSeg ↓	Implausibility ↓
Supervised	CTCF	0.919 / 1.73 / 23	0.41 / 1.42 / 16	0.56 / 1.52 / 14	4.48 / 1.85 / 14	58.53 / 1.77 / 13
	w/o proj	0.508 / 3.73 / 2	0.64 / 2.94 / 2	0.39 / 2.83 / 3	5.01 / 2.58 / 7	71.08 / 2.94 / 4
	w/o steer	0.850 / 2.10 / 16	0.76 / 3.77 / 0	0.29 / 3.71 / 1	5.14 / 2.56 / 5	79.69 / 3.23 / 3
	w/o time	0.819 / 2.44 / 14	0.46 / 1.87 / 8	0.53 / 1.94 / 9	5.29 / 3.02 / 2	61.76 / 2.06 / 6
One-class	CTCF	0.631 / 1.94 / 19	0.58 / 1.96 / 13	0.44 / 2.06 / 13	4.60 / 2.27 / 12	51.96 / 2.62 / 8
	w/o proj	0.547 / 2.85 / 7	0.64 / 2.35 / 9	0.42 / 2.44 / 9	4.48 / 2.19 / 11	55.08 / 2.46 / 5
	w/o steer	0.471 / 3.17 / 8	0.78 / 3.60 / 4	0.30 / 3.44 / 5	5.13 / 2.98 / 10	63.16 / 2.98 / 8
	w/o time	0.631 / 2.04 / 17	0.64 / 2.10 / 9	0.42 / 2.06 / 9	4.80 / 2.56 / 8	52.85 / 1.94 / 14

independently learns an unconditional data flow. During inference, the learned region controls the FM trajectory, and f evaluates the generated candidates. When f is changed, we reuse the trained FM and retrain only the region learner parameters using the new classifier's hard decisions.

In the original experiments, f was QUANT-based, the region learner used a three-layer 1D FCN encoder, and the FM used a 1D encoder-decoder. To test this architectural decoupling, we replaced QUANT with the structurally different MultiROCKET-HYDRA classifier [Tan et al., 2022, Dempster et al., 2023] on all 26 datasets in both settings. In the one-class setting, we applied Isolation Forest [Liu et al., 2008] to its higher-dimensional features instead of applying LOF to QUANT features. The CTCF architecture and other experimental settings were unchanged. Table 21 shows how CTCF's metric values change with the classifier, while Table 22 compares CTCF with its ablations under the new classifier.

In Table 21, validity uses all 26 datasets, while the other metrics use datasets where CTCF generated valid CFs with both classifiers (25 supervised and 23 one-class). In Table 22, validity and all mean ranks use all 26 datasets, with tied-worst ranks assigned when no valid CF was generated; the other metric averages use the 23 datasets in each setting where all four methods generated valid CFs. Thus, the two tables use different dataset subsets for CF-quality macro-averages. For CF-quality ranks and Number of Wins, each method is evaluated on its own valid CFs, as in Tables 34 and 35. Paired tests were conducted separately on common-valid instances, as reported in Tables 23 and 24.

Table 23: Pairwise supervised results with the MultiROCKET-HYDRA-based classifier being explained on common-valid instances. For each comparison, the first row reports win/loss/draw counts from CTCF's perspective and the second reports p-values from two-sided Wilcoxon signed-rank tests. n is the number of common-valid instances. (n.s.) indicates no significant difference at $\alpha = 0.05$. $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively.

Comparison	Proximity $\downarrow$	Sparsity $\uparrow$	NumSeg $\downarrow$	Implausibility $\downarrow$
vs. w/o proj ($n = 129$)	90/7/32 $p = 1.47 \times 10^{-15}$	89/9/31 $p = 5.33 \times 10^{-15}$	47/31/51 $p = 6.31 \times 10^{-2}$ (n.s.)	76/24/29 $p = 1.15 \times 10^{-7}$
vs. w/o steer ($n = 219$)	219/0/0 $p = 1.10 \times 10^{-37}$	204/6/9 $p = 1.08 \times 10^{-35}$	98/53/68 $p = 2.97 \times 10^{-4}$	156/63/0 $p = 1.38 \times 10^{-12}$
vs. w/o time ($n = 208$)	106/93/9 $p = 1.30 \times 10^{-1}$ (n.s.)	104/83/21 $p = 1.19 \times 10^{-1}$ (n.s.)	86/49/73 $p = 1.56 \times 10^{-4}$	104/97/7 $p = 8.78 \times 10^{-1}$ (n.s.)

Table 24: Pairwise one-class results with the MultiROCKET-HYDRA-based classifier being explained on common-valid instances. For each comparison, the first row reports win/loss/draw counts from CTCF's perspective and the second reports p-values from two-sided Wilcoxon signed-rank tests. n is the number of common-valid instances. (n.s.) indicates no significant difference at $\alpha = 0.05$. $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively.

Comparison	Proximity $\downarrow$	Sparsity $\uparrow$	NumSeg $\downarrow$	Implausibility $\downarrow$
vs. w/o proj ($n = 109$)	54/15/40 $p = 2.67 \times 10^{-5}$	44/19/46 $p = 1.99 \times 10^{-3}$	25/16/68 $p = 4.65 \times 10^{-1}$ (n.s.)	39/32/38 $p = 2.69 \times 10^{-1}$ (n.s.)
vs. w/o steer ($n = 100$)	93/7/0 $p = 2.21 \times 10^{-15}$	70/20/10 $p = 1.05 \times 10^{-9}$	38/28/34 $p = 4.46 \times 10^{-2}$	50/50/0 $p = 7.89 \times 10^{-1}$ (n.s.)
vs. w/o time ($n = 118$)	49/50/19 $p = 6.42 \times 10^{-1}$ (n.s.)	49/43/26 $p = 8.43 \times 10^{-1}$ (n.s.)	36/26/56 $p = 4.81 \times 10^{-2}$	48/58/12 $p = 1.61 \times 10^{-1}$ (n.s.)

In the supervised setting, QUANT and MultiROCKET-HYDRA gave similar CTCF validity (0.927 and 0.919), proximity (0.45 and 0.44), and sparsity (both 0.55). With MultiROCKET-HYDRA, CTCF had the best macro-average and mean rank for all five metrics. Eight of the 12 common-valid comparisons significantly favored CTCF, with at least one improvement over every ablation and no significant degradation (Table 23).

In the one-class setting, validity changed from 0.971 to 0.631, showing that CTCF's metric values can depend on the classifier's hard-decision region. Nevertheless, CTCF obtained the best mean ranks for validity and proximity and tied for the best sparsity rank. Six of the 12 common-valid comparisons significantly favored CTCF, again with at least one improvement over every ablation and no significant degradation (Table 24).

With MultiROCKET-HYDRA, each ablation significantly worsened at least one CF-quality metric in both settings. These results are consistent with the overall component-level trends observed with QUANT and support CTCF's applicability beyond that classifier. Tables 34 and 35 report the dataset-wise validity, proximity, sparsity, and NumSeg results in the supervised and one-class settings, respectively.

G.8 Hyperparameter analysis

Training hyperparameters. We used one common training configuration across all datasets and both classification settings, rather than dataset-specific tuning. We fixed $(\tau, q, \lambda_{\text{pin}}) = (0.5, 0.95, 1)$ before the main experiments. The 36-setting analysis below was conducted after receiving the reviews and was not used to select these values.

Here, τ activates repulsion of non-desired states in Eq. (3), q is the quantile level, and the squared radius $r_{\boldsymbol{\nu}}^2(t)$ in Eq. (4) approximates the q-quantile of desired-side squared distances. The coefficient λ_{pin} weights the pinball loss in Eq. (5). The coefficient λ_{rep} is the ℓ_2 regularization coefficient in Eq. (6); following Deep SVDD [Ruff et al., 2018], we fixed it at 10^{-6} and did not tune it.

Table 25: One-factor comparisons of training hyperparameters. The reference is the adopted configuration $(\tau, q, \lambda_{\text{pin}}) = (0.5, 0.95, 1)$; in each other row, only the indicated parameter differs from the reference. Validity and VS in Eq. (44) use all 50 instances, while the four CF-quality metrics use the 45 instances valid under all 36 configurations. ↓ and ↑ indicate that lower and higher values are better, respectively.

Setting	Adopted	Validity ↑	VS ↑	Proximity ↓	Sparsity ↑	NumSeg ↓	Implausibility ↓
Reference	yes	0.9400	0.6127	0.3042	0.6489	3.4644	32.8587
$\tau = 0$	–	0.9200	0.5601	0.3587	0.6020	3.5111	31.1366
$\tau = 0.25$	–	0.9600	0.5605	0.3790	0.5925	3.6089	40.8470
$\tau = 0.75$	–	0.9600	0.6069	0.3032	0.6433	3.4467	31.9443
$q = 0.90$	–	0.9600	0.5955	0.3133	0.6304	3.7644	33.6117
$q = 0.99$	–	0.9600	0.6043	0.3168	0.6345	3.4311	35.8997
$\lambda_{\text{pin}} = 0.1$	–	0.9400	0.6149	0.2961	0.6503	3.4422	32.9759
$\lambda_{\text{pin}} = 10$	–	0.9400	0.6120	0.3040	0.6481	3.4844	32.9311

Table 26: One-factor variation (max–min). The smallest range in each row is in bold.

Metric	Varying τ	Varying q	Varying λ_{pin}
Validity	0.0400	0.0200	**0.0000**
VS	0.0526	0.0172	**0.0029**
Proximity	0.0758	0.0126	**0.0081**
Sparsity	0.0564	0.0185	**0.0022**
NumSeg	0.1622	0.3333	**0.0422**
Implausibility	9.7104	3.0410	**0.1172**

We evaluated all 36 combinations of $\tau \in \{0, 0.25, 0.5, 0.75\}$, $q \in \{0.90, 0.95, 0.99\}$, and $\lambda_{\text{pin}} \in \{0.1, 1, 10\}$ on the five datasets in Section 5.5, using 10 supervised test instances per dataset. For every training setting and instance, all other conditions were fixed, and one CF was selected from the common nine-point inference grid by prioritizing Validity and using lower Proximity to break ties.

To summarize Validity and Sparsity jointly, we define

$$\text{VS} = \frac{1}{5} \sum_{d=1}^{5} \frac{1}{N_d} \sum_{i=1}^{N_d} v_{di} s_{di}. \tag{44}$$

Here, N_d is the number of evaluated instances in dataset d, v_{di} is the validity indicator for instance i, and s_{di} is its Sparsity. Thus, an invalid CF contributes zero. Validity and VS use all 50 instances. The other four metrics use the same 45 instances that were valid under all 36 settings, with within-dataset means macro-averaged over the five datasets.

Table 25 presents one-factor comparisons, in which we varied only one parameter while fixing the other two at their original values. ↑ and ↓ indicate the preferred direction.

In the one-factor comparisons, the prespecified setting had the highest VS when varying either τ or q. Changing λ_{pin} from 1 to 0.1 increased VS only from 0.6127 to 0.6149. The maximum Validity difference from the prespecified value was 0.02, corresponding to one of the 50 instances.

Table 26 reports the max–min range in each one-factor comparison. Values should be compared within the same metric.

Changing λ_{pin} produced the smallest range for every metric. Over $\lambda_{\text{pin}} \in \{0.1, 1, 10\}$, Validity remained 0.94 and the VS range was 0.0029. Thus, within the tested ranges, sensitivity to λ_{pin} was lower than sensitivity to τ or q.

To also examine parameter interactions, Table 27 shows the position of the original setting among all 36 settings. N is the number of evaluated instances. Gap to best is the difference from the best value, accounting for the preferred direction of each metric.

Across all 36 combinations, the prespecified setting ranked second in VS, with a gap of 0.0022 from the best, and its other four CF-quality metrics were at least as good as their medians. Its Validity-only

Table 27: Cross-check across all 36 training settings. Adopted refers to the prespecified configuration $(\tau, q, \lambda_{\text{pin}}) = (0.5, 0.95, 1)$. The distribution summary is the median and first and third quartiles over the 36 settings. $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively.

Metric	N	Adopted	Median [Q1, Q3] over 36	Adopted rank	Gap to best
Validity $\uparrow$	50	0.9400	0.9600 [0.9400, 0.9600]	27/36	0.0400
VS $\uparrow$	50	0.6127	0.5916 [0.5706, 0.6036]	2/36	0.0022
Proximity $\downarrow$	45	0.3042	0.3253 [0.3071, 0.3489]	8/36	0.0211
Sparsity $\uparrow$	45	0.6489	0.6300 [0.6147, 0.6414]	3/36	0.0035
NumSeg $\downarrow$	45	3.4644	3.5956 [3.4483, 3.6856]	13/36	0.1600
Implausibility $\downarrow$	45	32.8587	33.2055 [31.9693, 35.8997]	16/36	5.0905

Table 28: Selection frequencies and metric summaries for inference-time parameters across 26 datasets in the supervised setting (253 instances). Validity is averaged over the final selections with the indicated parameter value. The other metrics report the median and first and third quartiles (Q1 and Q3) over the valid final selections with that value.

Parameter	Value	Counts (%)	Validity	Statistic	Proximity	Sparsity	NumSeg	Implausibility
η	10^{-5}	21 (8.3%)	0.7143	Median	0.9308	0.1221	5.0000	74.6039
				Q1	0.7568	0.0975	2.0000	29.1670
				Q3	1.0272	0.2067	7.5000	158.7014
η	10^{-4}	82 (32.4%)	0.8537	Median	0.6914	0.2913	5.0000	65.4297
				Q1	0.4263	0.1670	2.0000	24.3164
				Q3	0.9577	0.4124	8.0000	159.9932
η	10^{-3}	150 (59.3%)	0.9933	Median	0.1514	0.7917	2.0000	16.8144
				Q1	0.0740	0.5556	1.0000	7.5568
				Q3	0.3307	0.8933	4.0000	42.4010
γ	0.1	118 (46.6%)	0.9576	Median	0.2391	0.6190	3.0000	19.6938
				Q1	0.1079	0.3200	2.0000	6.8426
				Q3	0.6612	0.8516	6.0000	52.3482
γ	0.5	60 (23.7%)	1.0000	Median	0.1105	0.8315	2.5000	19.8874
				Q1	0.0692	0.4119	0.7500	11.0100
				Q3	0.4712	0.9011	5.2500	30.8559
γ	1.0	75 (29.6%)	0.8133	Median	0.7674	0.2383	1.0000	99.4640
				Q1	0.2801	0.0000	1.0000	35.6751
				Q3	1.5044	0.6867	5.0000	182.3661

rank was 27/36. Thus, it was not optimal for every metric, but provided a strong balance between Validity and Sparsity.

Inference-time parameters. K is the number of Euler integration steps and affects accuracy and computation time. We fixed $K = 1000$ for every dataset in both settings. Prior FM work also evaluated fixed-step Euler integration with 1000 steps [Tong et al., 2024]. We did not evaluate sensitivity to K or tune it by dataset.

η is the fused-lasso steering step size in Eq. (13), and γ is the region scaling factor in Eqs. (11) and (14). We used the same candidate sets for all 26 datasets and both settings:

$$\mathcal{H}_\eta = \{10^{-5}, 10^{-4}, 10^{-3}\}, \qquad \mathcal{H}_\gamma = \{0.1, 0.5, 1.0\}.$$

For each instance, Algorithm 2 selects among the nine candidates by prioritizing Validity and then lower Proximity.

Tables 28 and 29 report the selection frequency and metric summaries for each value over the 253 supervised and 178 one-class instances, respectively. Validity is the mean over all final selections; the other metrics are Median [Q1, Q3] over valid final selections.

Table 29: Selection frequencies and metric summaries for inference-time parameters across 26 datasets in the one-class setting (178 instances). Validity is averaged over the final selections with the indicated parameter value. The other metrics report the median and first and third quartiles (Q1 and Q3) over the valid final selections with that value.

Parameter	Value	Counts (%)	Validity	Statistic	Proximity	Sparsity	NumSeg	Implausibility
				Median	0.7694	0.2572	3.5000	51.2274
η	10^{-5}	14 (7.9%)	0.5714	Q1	0.6921	0.1371	2.7500	22.1187
				Q3	0.8837	0.5322	5.0000	60.3415
				Median	1.0230	0.1282	4.0000	36.7798
η	10^{-4}	63 (35.4%)	0.9841	Q1	0.7882	0.0000	1.0000	25.3863
				Q3	1.9504	0.2562	7.0000	64.8378
				Median	0.1610	0.7500	2.0000	17.8626
η	10^{-3}	101 (56.7%)	1.0000	Q1	0.0821	0.4792	1.0000	5.3303
				Q3	0.4028	0.8867	4.0000	39.2387
				Median	0.4569	0.4727	4.0000	30.3197
γ	0.1	127 (71.3%)	0.9843	Q1	0.1553	0.1298	1.0000	13.8719
				Q3	0.9639	0.7500	6.0000	45.8886
				Median	0.3596	0.5429	2.0000	21.4389
γ	0.5	40 (22.5%)	0.8750	Q1	0.0769	0.1467	1.0000	6.6378
				Q3	1.0179	0.8800	4.0000	45.5385
				Median	0.3349	0.6133	2.0000	20.8593
γ	1.0	11 (6.2%)	1.0000	Q1	0.0792	0.4219	1.0000	5.0267
				Q3	0.6833	0.9079	2.5000	37.7831

Tables 28 and 29 report how often each value was selected and summarize the metrics for the instances on which it was selected. The Counts (%) column shows that every candidate value was selected in both settings, but no single value was selected for all instances. Together with the training-time analysis above, these findings support using a common training-time configuration and selecting η and γ instance-wise from a small grid shared across datasets and settings.

G.9 Case-study representativeness

Figure 4 illustrates that CTCF can capture known class-related patterns. One type of failure case is a valid CF with poor proximity, sparsity, NumSeg, or implausibility. We therefore compared the five examples in Figure 4 with all valid CFs generated by CTCF for each dataset and evaluated their positions within the distributions of proximity, sparsity, NumSeg, and implausibility. Table 30 shows the results.

Of the 20 comparisons, 17 were within the interquartile range (IQR), three were on the favorable side, and none were on the unfavorable side. Thus, the examples in Figure 4 are not biased toward exceptionally favorable cases, at least with respect to these four metrics.

Figure 7 complements this analysis with additional valid CFs in the one-class setting. The comparisons below use the same dataset-specific distributions summarized in Table 30.

- For GunPoint (Figure 7(a)), proximity is 0.67, above the upper quartile of 0.44, and sparsity is 0.07, below the lower quartile of 0.21. Thus, this example exhibits both large changes and changes across many timestamps.
- For ECGFiveDays (Figure 7(b)), only proximity lies beyond the corresponding IQR in the unfavorable direction: its value is 0.46, exceeding the upper quartile of 0.32.
- For ToeSegmentation1 (Figure 7(c)), sparsity is 0.50, below the lower quartile of 0.57, while proximity and implausibility are 0.35 and 96.67, above their respective upper quartiles of 0.27 and 75.47.

Table 30: Positions of the Figure 4 examples within the distributions of all valid CFs generated by CTCF in the one-class setting. Test indices are zero-based. "Favorable" indicates that the value lies beyond the interquartile range (IQR) in the favorable direction; "Within" indicates that it lies within the IQR. ↓ and ↑ indicate that lower and higher values are better, respectively.

Dataset	Test index	Metric	Figure 4 value	Median [Q1, Q3]	IQR position
GunPoint	31	Proximity ↓	0.08	0.24 [0.07, 0.44]	Within
GunPoint	31	Sparsity ↑	0.89	0.75 [0.21, 0.91]	Within
GunPoint	31	NumSeg ↓	2.00	2.50 [1.75, 4.00]	Within
GunPoint	31	Implausibility ↓	12.46	14.12 [10.68, 18.27]	Within
ECGFiveDays	42	Proximity ↓	0.13	0.30 [0.11, 0.32]	Within
ECGFiveDays	42	Sparsity ↑	0.82	0.65 [0.62, 0.85]	Within
ECGFiveDays	42	NumSeg ↓	2.00	2.00 [1.00, 3.00]	Within
ECGFiveDays	42	Implausibility ↓	17.02	15.00 [13.28, 20.44]	Within
ToeSegmentation1	44	Proximity ↓	0.25	0.20 [0.15, 0.27]	Within
ToeSegmentation1	44	Sparsity ↑	0.59	0.68 [0.57, 0.78]	Within
ToeSegmentation1	44	NumSeg ↓	10.00	7.00 [4.00, 10.00]	Within
ToeSegmentation1	44	Implausibility ↓	68.41	55.71 [41.51, 75.47]	Within
ItalyPowerDemand	40	Proximity ↓	0.20	0.27 [0.21, 0.50]	Favorable
ItalyPowerDemand	40	Sparsity ↑	0.62	0.50 [0.29, 0.61]	Favorable
ItalyPowerDemand	40	NumSeg ↓	3.00	3.00 [3.00, 3.75]	Within
ItalyPowerDemand	40	Implausibility ↓	2.97	2.86 [2.55, 3.10]	Within
PowerCons	57	Proximity ↓	0.77	0.86 [0.75, 0.97]	Within
PowerCons	57	Sparsity ↑	0.57	0.37 [0.26, 0.50]	Favorable
PowerCons	57	NumSeg ↓	5.00	5.00 [5.00, 5.25]	Within
PowerCons	57	Implausibility ↓	43.54	47.54 [43.07, 52.93]	Within

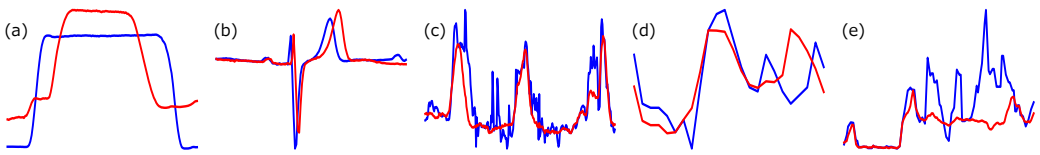

Figure 7: Original time series (blue) and valid CFs (red) illustrating less favorable values for particular metrics in the one-class setting on (a) GunPoint, (b) ECGFiveDays, (c) ToeSegmentation1, (d) ItalyPowerDemand, and (e) PowerCons.

- For ItalyPowerDemand (Figure 7(d)), changes span five segments, exceeding the upper quartile of 3.75 for NumSeg. Its proximity, sparsity, and implausibility also lie beyond their corresponding IQRs in the unfavorable direction.
- For PowerCons (Figure 7(e)), implausibility is relatively high (51.54 versus a median of 47.54), although the other metrics are comparatively good: proximity and sparsity are better than their respective medians, and NumSeg matches the median. Its implausibility nevertheless remains within the IQR [43.07, 52.93].

These examples illustrate that achieving the desired-class label does not by itself ensure favorable proximity, sparsity, NumSeg, or implausibility.

Table 31: Dataset-wise results for the endpoint-AE feasibility proxy in the supervised setting. Validity is computed over all selected instances. Proximity, sparsity, and NumSeg are computed over valid counterfactuals and are reported as mean ± standard deviation (omitted for a single valid counterfactual). A dash (—) indicates that no valid counterfactual was generated.

Dataset	Validity	Proximity	Sparsity	NumSeg
BME	1.00	0.15 ± 0.15	0.79 ± 0.18	2.30 ± 0.90
BirdChicken	1.00	0.36 ± 0.18	0.48 ± 0.22	5.40 ± 2.58
CricketX	0.00	—	—	—
CricketZ	0.00	—	—	—
DistalPhalanxOutlineAgeGroup	1.00	0.16 ± 0.06	0.72 ± 0.13	3.10 ± 1.76
ECGFiveDays	1.00	0.22 ± 0.05	0.78 ± 0.07	1.90 ± 0.54
FaceAll	1.00	0.88 ± 0.12	0.19 ± 0.04	10.40 ± 1.11
FacesUCR	0.70	0.76 ± 0.19	0.23 ± 0.10	10.14 ± 2.95
FiftyWords	1.00	0.42 ± 0.23	0.56 ± 0.19	3.70 ± 1.42
GunPoint	1.00	0.18 ± 0.06	0.74 ± 0.09	2.20 ± 0.75
GunPointAgeSpan	1.00	0.19 ± 0.16	0.74 ± 0.22	2.00 ± 1.18
GunPointMaleVersusFemale	1.00	0.53 ± 0.35	0.36 ± 0.34	2.40 ± 1.02
GunPointOldVersusYoung	1.00	1.10 ± 0.54	0.21 ± 0.27	1.00 ± 0.00
InsectEPGRegularTrain	1.00	1.26 ± 0.04	0.00 ± 0.00	1.00 ± 0.00
InsectEPGSmallTrain	1.00	1.33 ± 0.04	0.00 ± 0.00	1.00 ± 0.00
InsectWingbeatSound	1.00	0.38 ± 0.09	0.66 ± 0.15	4.20 ± 1.72
ItalyPowerDemand	1.00	0.39 ± 0.16	0.42 ± 0.17	4.00 ± 1.34
MoteStrain	1.00	0.49 ± 0.21	0.36 ± 0.15	4.70 ± 1.42
PowerCons	1.00	0.53 ± 0.21	0.50 ± 0.17	5.30 ± 1.55
ProximalPhalanxOutlineAgeGroup	0.50	0.09 ± 0.03	0.91 ± 0.09	0.80 ± 1.17
SonyAIBORobotSurface1	0.90	0.48 ± 0.04	0.26 ± 0.05	6.89 ± 1.29
SonyAIBORobotSurface2	0.30	0.61 ± 0.13	0.23 ± 0.04	6.67 ± 1.89
ToeSegmentation1	0.10	0.85	0.22	7.00
ToeSegmentation2	1.00	0.26 ± 0.07	0.66 ± 0.10	4.33 ± 1.25
TwoLeadECG	1.00	0.12 ± 0.06	0.85 ± 0.10	1.00 ± 1.10
UMD	1.00	0.13 ± 0.07	0.85 ± 0.12	2.50 ± 1.12

Table 32: Dataset-wise results for the endpoint-AE feasibility proxy in the one-class setting. Validity is computed over all selected instances. Proximity, sparsity, and NumSeg are computed over valid counterfactuals and are reported as mean ± standard deviation (omitted for a single valid counterfactual).

Dataset	Validity	Proximity	Sparsity	NumSeg
BME	1.00	1.12 ± 0.24	0.45 ± 0.03	2.67 ± 1.49
BirdChicken	1.00	0.90	0.11	6.00
CricketX	1.00	0.26 ± 0.05	0.58 ± 0.09	9.75 ± 2.33
CricketZ	1.00	0.28 ± 0.05	0.51 ± 0.10	11.67 ± 2.36
DistalPhalanxOutlineAgeGroup	1.00	0.10 ± 0.05	0.84 ± 0.14	1.33 ± 1.37
ECGFiveDays	1.00	0.36 ± 0.17	0.68 ± 0.15	2.80 ± 0.60
FaceAll	1.00	0.73 ± 0.14	0.25 ± 0.04	10.00 ± 1.18
FacesUCR	1.00	0.76 ± 0.10	0.22 ± 0.05	11.50 ± 1.57
FiftyWords	1.00	0.75 ± 0.35	0.35 ± 0.24	3.60 ± 0.80
GunPoint	1.00	0.45 ± 0.34	0.41 ± 0.38	2.89 ± 1.66
GunPointAgeSpan	1.00	0.29 ± 0.14	0.60 ± 0.19	2.60 ± 1.50
GunPointMaleVersusFemale	1.00	0.60 ± 0.32	0.24 ± 0.31	2.10 ± 1.45
GunPointOldVersusYoung	1.00	1.10 ± 0.84	0.34 ± 0.34	1.40 ± 0.80
InsectEPGRegularTrain	1.00	1.97 ± 0.04	0.00 ± 0.00	1.00 ± 0.00
InsectEPGSmallTrain	1.00	1.84 ± 0.05	0.00 ± 0.00	1.00 ± 0.00
InsectWingbeatSound	0.50	0.71 ± 0.04	0.50 ± 0.07	5.00 ± 1.22
ItalyPowerDemand	1.00	0.47 ± 0.42	0.48 ± 0.32	2.50 ± 1.02
MoteStrain	1.00	0.49	0.23	5.00
PowerCons	1.00	1.23 ± 0.13	0.22 ± 0.11	4.00 ± 0.71
ProximalPhalanxOutlineAgeGroup	1.00	0.05 ± 0.02	0.96 ± 0.05	0.71 ± 0.88
SonyAIBORobotSurface1	1.00	0.27 ± 0.06	0.51 ± 0.08	6.33 ± 0.94
SonyAIBORobotSurface2	0.50	0.90	0.22	3.00
ToeSegmentation1	0.75	0.26 ± 0.07	0.55 ± 0.12	9.33 ± 3.30
ToeSegmentation2	1.00	0.29 ± 0.10	0.54 ± 0.16	12.80 ± 8.82
TwoLeadECG	1.00	0.12 ± 0.11	0.84 ± 0.15	1.40 ± 1.02
UMD	1.00	0.68 ± 0.08	0.61 ± 0.03	2.00 ± 0.00

Table 33: Dataset-wise CTCF and ablation results for the native multiclass experiment with QUANT. Validity uses all selected instances; the other metrics use each method's valid counterfactuals and are reported as mean ± standard deviation. Parentheses give ranks among the four variants. Rank Mean and Number of Wins summarize all 12 datasets, with ties counted as wins.

Dataset	Method	Validity	Proximity	Sparsity	NumSeg
BME	CTCF	0.90 (2.0)	0.28 ± 0.39 (1.0)	0.80 ± 0.15 (1.0)	2.29 ± 0.85 (1.0)
	w/o proj	0.40 (4.0)	0.34 ± 0.34 (3.0)	0.76 ± 0.13 (3.0)	2.74 ± 1.09 (2.0)
	w/o steer	0.83 (3.0)	0.64 ± 0.44 (4.0)	0.58 ± 0.15 (4.0)	3.01 ± 0.82 (3.0)
	w/o time	0.93 (1.0)	0.29 ± 0.35 (2.0)	0.77 ± 0.14 (2.0)	3.89 ± 2.37 (4.0)
CricketX	CTCF	0.55 (1.0)	0.64 ± 0.32 (2.0)	0.35 ± 0.21 (2.0)	7.38 ± 2.45 (4.0)
	w/o proj	0.15 (4.0)	0.72 ± 0.38 (3.0)	0.35 ± 0.28 (3.0)	6.44 ± 2.11 (1.0)
	w/o steer	0.48 (2.0)	0.94 ± 0.21 (4.0)	0.18 ± 0.09 (4.0)	7.19 ± 2.21 (2.0)
	w/o time	0.38 (3.0)	0.62 ± 0.36 (1.0)	0.38 ± 0.26 (1.0)	7.31 ± 2.94 (3.0)
CricketZ	CTCF	0.66 (1.0)	0.64 ± 0.33 (2.0)	0.35 ± 0.23 (2.0)	8.39 ± 3.35 (2.0)
	w/o proj	0.19 (4.0)	0.54 ± 0.36 (1.0)	0.45 ± 0.30 (1.0)	7.43 ± 4.34 (1.0)
	w/o steer	0.53 (2.0)	0.98 ± 0.21 (4.0)	0.15 ± 0.08 (4.0)	8.76 ± 2.96 (3.0)
	w/o time	0.46 (3.0)	0.64 ± 0.34 (3.0)	0.35 ± 0.24 (3.0)	9.66 ± 3.76 (4.0)
DistalPhalanxOutlineAgeGroup	CTCF	1.00 (1.5)	0.13 ± 0.08 (1.0)	0.79 ± 0.17 (1.0)	2.70 ± 2.31 (1.0)
	w/o proj	0.37 (4.0)	0.27 ± 0.06 (4.0)	0.54 ± 0.10 (4.0)	4.53 ± 2.25 (3.0)
	w/o steer	0.97 (3.0)	0.25 ± 0.08 (3.0)	0.54 ± 0.15 (3.0)	5.40 ± 2.26 (4.0)
	w/o time	1.00 (1.5)	0.15 ± 0.09 (2.0)	0.74 ± 0.18 (2.0)	3.60 ± 2.78 (2.0)
FaceAll	CTCF	0.66 (1.0)	0.58 ± 0.28 (1.0)	0.39 ± 0.23 (1.0)	7.86 ± 3.56 (2.0)
	w/o proj	0.13 (4.0)	0.75 ± 0.28 (3.0)	0.29 ± 0.20 (3.0)	7.67 ± 2.83 (1.0)
	w/o steer	0.58 (2.0)	0.93 ± 0.12 (4.0)	0.16 ± 0.04 (4.0)	10.25 ± 1.72 (4.0)
	w/o time	0.54 (3.0)	0.62 ± 0.26 (2.0)	0.34 ± 0.19 (2.0)	10.04 ± 3.89 (3.0)
FacesUCR	CTCF	0.76 (1.0)	0.56 ± 0.24 (1.0)	0.35 ± 0.20 (1.0)	9.11 ± 3.44 (1.0)
	w/o proj	0.11 (4.0)	0.79 ± 0.24 (3.0)	0.25 ± 0.14 (3.0)	10.56 ± 2.54 (2.0)
	w/o steer	0.64 (2.0)	0.85 ± 0.16 (4.0)	0.18 ± 0.04 (4.0)	10.98 ± 2.23 (4.0)
	w/o time	0.46 (3.0)	0.61 ± 0.25 (2.0)	0.31 ± 0.17 (2.0)	10.88 ± 3.31 (3.0)
FiftyWords	CTCF	0.35 (1.0)	0.61 ± 0.29 (2.0)	0.39 ± 0.21 (2.0)	6.09 ± 2.39 (2.0)
	w/o proj	0.03 (4.0)	0.72 ± 0.16 (3.0)	0.36 ± 0.15 (3.0)	5.28 ± 1.60 (1.0)
	w/o steer	0.23 (2.0)	0.87 ± 0.12 (4.0)	0.23 ± 0.07 (4.0)	6.38 ± 1.92 (3.0)
	w/o time	0.14 (3.0)	0.59 ± 0.29 (1.0)	0.43 ± 0.24 (1.0)	6.68 ± 3.37 (4.0)
InsectEPGRegularTrain	CTCF	1.00 (2.0)	1.10 ± 0.61 (3.0)	0.03 ± 0.07 (3.0)	1.00 ± 0.00 (1.5)
	w/o proj	0.37 (4.0)	1.06 ± 0.45 (2.0)	0.20 ± 0.09 (1.0)	1.11 ± 0.31 (4.0)
	w/o steer	1.00 (2.0)	1.20 ± 0.57 (4.0)	0.02 ± 0.06 (4.0)	1.00 ± 0.00 (1.5)
	w/o time	1.00 (2.0)	0.94 ± 0.53 (1.0)	0.11 ± 0.14 (2.0)	1.03 ± 0.18 (3.0)
InsectEPGSmallTrain	CTCF	1.00 (2.0)	1.08 ± 0.59 (3.0)	0.02 ± 0.05 (4.0)	1.00 ± 0.00 (1.5)
	w/o proj	0.30 (4.0)	1.07 ± 0.64 (2.0)	0.20 ± 0.15 (1.0)	1.32 ± 0.65 (4.0)
	w/o steer	1.00 (2.0)	1.14 ± 0.58 (4.0)	0.02 ± 0.07 (3.0)	1.00 ± 0.00 (1.5)
	w/o time	1.00 (2.0)	0.92 ± 0.51 (1.0)	0.10 ± 0.14 (2.0)	1.07 ± 0.25 (3.0)
InsectWingbeatSound	CTCF	0.63 (1.0)	0.49 ± 0.18 (1.0)	0.58 ± 0.14 (1.0)	4.68 ± 1.62 (2.0)
	w/o proj	0.10 (4.0)	0.62 ± 0.22 (3.0)	0.57 ± 0.16 (3.0)	4.50 ± 1.95 (1.0)
	w/o steer	0.55 (2.0)	0.69 ± 0.13 (4.0)	0.40 ± 0.10 (4.0)	5.82 ± 1.39 (4.0)
	w/o time	0.45 (3.0)	0.49 ± 0.23 (2.0)	0.58 ± 0.16 (2.0)	4.84 ± 2.40 (3.0)
ProximalPhalanxOutlineAgeGroup	CTCF	1.00 (1.0)	0.10 ± 0.05 (1.0)	0.83 ± 0.14 (1.0)	2.17 ± 2.19 (2.0)
	w/o proj	0.37 (4.0)	0.15 ± 0.06 (3.0)	0.72 ± 0.15 (4.0)	3.44 ± 2.38 (4.0)
	w/o steer	0.97 (2.0)	0.15 ± 0.05 (4.0)	0.72 ± 0.14 (3.0)	3.36 ± 2.32 (3.0)
	w/o time	0.93 (3.0)	0.11 ± 0.05 (2.0)	0.82 ± 0.13 (2.0)	1.79 ± 1.91 (1.0)
UMD	CTCF	0.97 (1.5)	0.32 ± 0.32 (2.0)	0.68 ± 0.24 (2.0)	3.70 ± 2.67 (3.0)
	w/o proj	0.53 (4.0)	0.72 ± 0.46 (3.0)	0.67 ± 0.18 (3.0)	1.82 ± 0.90 (1.0)
	w/o steer	0.77 (3.0)	0.90 ± 0.37 (4.0)	0.54 ± 0.14 (4.0)	2.77 ± 1.37 (2.0)
	w/o time	0.97 (1.5)	0.24 ± 0.29 (1.0)	0.79 ± 0.15 (1.0)	4.28 ± 3.23 (4.0)
Rank Mean	CTCF	1.33	1.67	1.75	1.92
	w/o proj	4.00	2.75	2.67	2.08
	w/o steer	2.25	3.92	3.75	2.92
	w/o time	2.42	1.67	1.83	3.08
Number of Wins	CTCF	11	6	6	5
	w/o proj	0	1	3	6
	w/o steer	2	0	0	2
	w/o time	5	5	3	1

Table 34: Dataset-wise CTCF and ablation results with the MultiROCKET-HYDRA-based classifier in the supervised setting. Validity uses all selected instances; the other metrics use each method's valid counterfactuals and are reported as mean ± standard deviation (omitted for a single valid counterfactual). Parentheses give ranks among the four variants; — denotes no valid counterfactual. Rank Mean and Number of Wins summarize all 26 datasets, with ties counted as wins.

Dataset	Method	Validity	Proximity	Sparsity	NumSeg
BME	CTCF	1.00 (2.0)	0.10 ± 0.05 (2.0)	0.85 ± 0.10 (2.0)	2.50 ± 1.50 (1.0)
	w/o proj	0.50 (4.0)	0.25 ± 0.08 (3.0)	0.75 ± 0.17 (3.0)	3.40 ± 1.02 (3.0)
	w/o steer	1.00 (2.0)	0.43 ± 0.36 (4.0)	0.61 ± 0.15 (4.0)	3.10 ± 1.58 (2.0)
	w/o time	1.00 (2.0)	0.09 ± 0.04 (1.0)	0.88 ± 0.05 (1.0)	4.10 ± 1.92 (4.0)
BirdChicken	CTCF	1.00 (1.5)	0.25 ± 0.31 (1.0)	0.73 ± 0.33 (2.0)	2.90 ± 3.36 (1.5)
	w/o proj	1.00 (1.5)	0.25 ± 0.32 (2.0)	0.74 ± 0.33 (1.0)	2.90 ± 3.36 (1.5)
	w/o steer	0.70 (3.0)	0.98 ± 0.14 (4.0)	0.10 ± 0.03 (4.0)	7.71 ± 1.28 (4.0)
	w/o time	0.60 (4.0)	0.44 ± 0.41 (3.0)	0.57 ± 0.33 (3.0)	3.83 ± 2.61 (3.0)
CricketX	CTCF	1.00 (1.5)	0.68 ± 0.25 (1.0)	0.30 ± 0.15 (1.0)	10.00 ± 2.65 (4.0)
	w/o proj	0.80 (4.0)	0.80 ± 0.18 (3.0)	0.27 ± 0.12 (2.0)	7.62 ± 2.50 (1.0)
	w/o steer	1.00 (1.5)	0.98 ± 0.16 (4.0)	0.15 ± 0.10 (4.0)	7.70 ± 3.69 (2.0)
	w/o time	0.90 (3.0)	0.80 ± 0.17 (2.0)	0.22 ± 0.08 (3.0)	9.11 ± 2.08 (3.0)
CricketZ	CTCF	0.90 (1.5)	0.83 ± 0.25 (1.0)	0.22 ± 0.11 (1.0)	6.44 ± 2.22 (3.0)
	w/o proj	0.70 (3.5)	1.10 ± 0.13 (3.0)	0.17 ± 0.12 (2.0)	5.43 ± 2.13 (1.0)
	w/o steer	0.90 (1.5)	1.14 ± 0.12 (4.0)	0.12 ± 0.05 (4.0)	6.22 ± 1.99 (2.0)
	w/o time	0.70 (3.5)	1.09 ± 0.13 (2.0)	0.13 ± 0.05 (3.0)	7.14 ± 2.95 (4.0)
DistalPhalanxOutlineAgeGroup	CTCF	1.00 (1.5)	0.16 ± 0.05 (2.0)	0.70 ± 0.12 (2.0)	3.20 ± 2.09 (2.0)
	w/o proj	0.00 (4.0)	— (4.0)	— (4.0)	— (4.0)
	w/o steer	0.90 (3.0)	0.22 ± 0.05 (3.0)	0.59 ± 0.12 (3.0)	4.44 ± 2.27 (3.0)
	w/o time	1.00 (1.5)	0.10 ± 0.03 (1.0)	0.83 ± 0.06 (1.0)	2.10 ± 1.30 (1.0)
ECGFiveDays	CTCF	1.00 (2.0)	0.10 ± 0.02 (1.0)	0.85 ± 0.02 (1.0)	1.10 ± 0.83 (1.0)
	w/o proj	0.50 (4.0)	0.37 ± 0.14 (3.0)	0.57 ± 0.24 (3.0)	1.80 ± 0.75 (3.0)
	w/o steer	1.00 (2.0)	0.48 ± 0.12 (4.0)	0.46 ± 0.21 (4.0)	2.30 ± 1.19 (4.0)
	w/o time	1.00 (2.0)	0.16 ± 0.10 (2.0)	0.79 ± 0.09 (2.0)	1.30 ± 0.90 (2.0)
FaceAll	CTCF	1.00 (1.0)	0.53 ± 0.20 (1.0)	0.32 ± 0.12 (1.0)	11.00 ± 2.00 (2.5)
	w/o proj	0.40 (4.0)	0.79 ± 0.19 (3.0)	0.18 ± 0.04 (3.0)	11.00 ± 2.74 (2.5)
	w/o steer	0.80 (2.0)	0.88 ± 0.15 (4.0)	0.16 ± 0.04 (4.0)	10.75 ± 1.85 (1.0)
	w/o time	0.70 (3.0)	0.69 ± 0.21 (2.0)	0.24 ± 0.08 (2.0)	12.14 ± 1.81 (4.0)
FacesUCR	CTCF	1.00 (1.0)	0.79 ± 0.22 (1.0)	0.22 ± 0.11 (1.0)	9.80 ± 3.54 (1.0)
	w/o proj	0.50 (4.0)	0.88 ± 0.13 (3.0)	0.16 ± 0.02 (4.0)	11.20 ± 3.12 (4.0)
	w/o steer	0.90 (2.0)	0.92 ± 0.16 (4.0)	0.16 ± 0.05 (3.0)	10.22 ± 2.57 (2.0)
	w/o time	0.70 (3.0)	0.81 ± 0.15 (2.0)	0.19 ± 0.06 (2.0)	10.29 ± 2.60 (3.0)
FiftyWords	CTCF	1.00 (2.0)	0.14 ± 0.03 (2.0)	0.76 ± 0.08 (2.0)	3.40 ± 1.91 (1.0)
	w/o proj	0.80 (4.0)	0.64 ± 0.08 (3.0)	0.38 ± 0.06 (3.0)	4.12 ± 1.05 (3.0)
	w/o steer	1.00 (2.0)	0.73 ± 0.11 (4.0)	0.25 ± 0.08 (4.0)	4.90 ± 0.83 (4.0)
	w/o time	1.00 (2.0)	0.11 ± 0.02 (1.0)	0.82 ± 0.06 (1.0)	3.70 ± 2.05 (2.0)
GunPoint	CTCF	1.00 (1.5)	0.09 ± 0.04 (1.0)	0.87 ± 0.07 (1.0)	1.10 ± 0.83 (1.0)
	w/o proj	0.70 (4.0)	0.39 ± 0.24 (3.0)	0.51 ± 0.23 (3.0)	2.71 ± 0.88 (3.0)
	w/o steer	1.00 (1.5)	0.49 ± 0.22 (4.0)	0.32 ± 0.24 (4.0)	3.40 ± 1.02 (4.0)
	w/o time	0.90 (3.0)	0.25 ± 0.26 (2.0)	0.69 ± 0.34 (2.0)	1.56 ± 1.34 (2.0)
GunPointAgeSpan	CTCF	1.00 (2.0)	0.06 ± 0.02 (1.0)	0.93 ± 0.05 (2.0)	3.30 ± 1.90 (3.0)
	w/o proj	0.80 (4.0)	0.58 ± 0.48 (3.0)	0.46 ± 0.31 (3.0)	2.38 ± 0.86 (1.0)
	w/o steer	1.00 (2.0)	0.73 ± 0.35 (4.0)	0.36 ± 0.23 (4.0)	2.70 ± 1.55 (2.0)
	w/o time	1.00 (2.0)	0.06 ± 0.05 (2.0)	0.93 ± 0.07 (1.0)	4.20 ± 2.64 (4.0)
GunPointMaleVersusFemale	CTCF	0.60 (2.0)	0.46 ± 0.42 (2.0)	0.50 ± 0.29 (2.0)	2.83 ± 1.07 (2.0)
	w/o proj	0.10 (4.0)	0.93 (3.0)	0.17 (4.0)	4.00 (3.0)
	w/o steer	0.50 (3.0)	1.05 ± 0.48 (4.0)	0.23 ± 0.29 (3.0)	2.40 ± 1.74 (1.0)
	w/o time	1.00 (1.0)	0.31 ± 0.40 (1.0)	0.67 ± 0.27 (1.0)	7.10 ± 4.18 (4.0)
GunPointOldVersusYoung	CTCF	1.00 (1.5)	0.63 ± 0.57 (1.0)	0.46 ± 0.30 (1.0)	1.00 ± 0.00 (1.0)
	w/o proj	0.50 (4.0)	0.87 ± 0.26 (3.0)	0.34 ± 0.11 (2.0)	2.80 ± 0.75 (3.0)
	w/o steer	1.00 (1.5)	0.93 ± 0.61 (4.0)	0.18 ± 0.16 (4.0)	2.70 ± 1.27 (2.0)
	w/o time	0.80 (3.0)	0.78 ± 0.64 (2.0)	0.29 ± 0.22 (3.0)	2.88 ± 1.17 (4.0)
InsectEPGRegularTrain	CTCF	1.00 (1.0)	1.43 ± 0.04 (1.0)	0.00 ± 0.00 (1.5)	1.00 ± 0.00 (1.5)
	w/o proj	0.00 (3.5)	— (3.5)	— (3.5)	— (3.5)
	w/o steer	0.30 (2.0)	1.60 ± 0.06 (2.0)	0.00 ± 0.00 (1.5)	1.00 ± 0.00 (1.5)
	w/o time	0.00 (3.5)	— (3.5)	— (3.5)	— (3.5)
InsectEPGSmallTrain	CTCF	0.00 (3.0)	— (3.0)	— (3.0)	— (3.0)
	w/o proj	0.30 (1.0)	0.41 ± 0.11 (1.0)	0.76 ± 0.04 (1.0)	3.33 ± 0.47 (1.0)
	w/o steer	0.00 (3.0)	— (3.0)	— (3.0)	— (3.0)
	w/o time	0.00 (3.0)	— (3.0)	— (3.0)	— (3.0)

Continued on next page

Table 34: Dataset-wise CTCF and ablation results with the MultiROCKET-HYDRA-based classifier in the supervised setting (continued).

Dataset	Method	Validity	Proximity	Sparsity	NumSeg
InsectWingbeatSound	CTCF	0.60 (2.0)	0.64 ± 0.07 (2.0)	0.45 ± 0.12 (2.0)	4.50 ± 0.76 (2.0)
	w/o proj	0.20 (4.0)	0.61 ± 0.06 (1.0)	0.55 ± 0.02 (1.0)	4.00 ± 1.00 (1.0)
	w/o steer	0.60 (2.0)	0.70 ± 0.06 (4.0)	0.40 ± 0.11 (4.0)	4.83 ± 1.07 (3.0)
	w/o time	0.60 (2.0)	0.65 ± 0.07 (3.0)	0.43 ± 0.12 (3.0)	5.83 ± 2.19 (4.0)
ItalyPowerDemand	CTCF	1.00 (2.0)	0.58 ± 0.37 (1.0)	0.35 ± 0.28 (2.0)	3.00 ± 1.18 (1.0)
	w/o proj	0.60 (4.0)	0.63 ± 0.27 (3.0)	0.22 ± 0.20 (3.0)	4.67 ± 1.70 (4.0)
	w/o steer	1.00 (2.0)	0.83 ± 0.35 (4.0)	0.15 ± 0.16 (4.0)	4.10 ± 1.76 (3.0)
	w/o time	1.00 (2.0)	0.58 ± 0.34 (2.0)	0.36 ± 0.24 (1.0)	3.70 ± 2.05 (2.0)
MoteStrain	CTCF	1.00 (1.5)	0.56 ± 0.17 (2.0)	0.31 ± 0.15 (2.0)	4.80 ± 1.17 (2.5)
	w/o proj	0.50 (4.0)	0.68 ± 0.14 (3.0)	0.22 ± 0.08 (3.0)	4.80 ± 0.40 (2.5)
	w/o steer	1.00 (1.5)	0.75 ± 0.15 (4.0)	0.21 ± 0.13 (4.0)	4.70 ± 0.90 (1.0)
	w/o time	0.90 (3.0)	0.55 ± 0.14 (1.0)	0.33 ± 0.10 (1.0)	5.22 ± 1.62 (4.0)
PowerCons	CTCF	1.00 (1.5)	0.40 ± 0.28 (1.0)	0.64 ± 0.18 (1.0)	4.90 ± 2.07 (1.0)
	w/o proj	0.40 (4.0)	0.82 ± 0.30 (3.0)	0.40 ± 0.11 (3.0)	5.50 ± 2.29 (2.0)
	w/o steer	0.90 (3.0)	0.84 ± 0.27 (4.0)	0.34 ± 0.11 (4.0)	6.00 ± 0.94 (4.0)
	w/o time	1.00 (1.5)	0.40 ± 0.16 (2.0)	0.59 ± 0.14 (2.0)	5.70 ± 1.55 (3.0)
ProximalPhalanxOutlineAgeGroup	CTCF	0.90 (2.5)	0.05 ± 0.02 (1.0)	0.98 ± 0.03 (1.0)	0.11 ± 0.31 (1.0)
	w/o proj	0.20 (4.0)	0.19 ± 0.02 (4.0)	0.64 ± 0.08 (4.0)	4.00 ± 2.00 (4.0)
	w/o steer	0.90 (2.5)	0.11 ± 0.04 (3.0)	0.84 ± 0.12 (3.0)	1.44 ± 1.77 (3.0)
	w/o time	1.00 (1.0)	0.07 ± 0.03 (2.0)	0.92 ± 0.07 (2.0)	0.60 ± 0.92 (2.0)
SonyAIBORobotSurface1	CTCF	1.00 (2.0)	0.27 ± 0.11 (1.0)	0.59 ± 0.11 (1.0)	4.30 ± 1.10 (1.0)
	w/o proj	0.80 (4.0)	0.52 ± 0.10 (3.0)	0.27 ± 0.06 (3.0)	7.00 ± 1.50 (4.0)
	w/o steer	1.00 (2.0)	0.57 ± 0.08 (4.0)	0.24 ± 0.03 (4.0)	6.60 ± 1.20 (3.0)
	w/o time	1.00 (2.0)	0.27 ± 0.11 (2.0)	0.56 ± 0.13 (2.0)	5.00 ± 1.34 (2.0)
SonyAIBORobotSurface2	CTCF	1.00 (1.5)	0.45 ± 0.17 (2.0)	0.38 ± 0.12 (2.0)	5.20 ± 1.40 (1.0)
	w/o proj	0.40 (4.0)	0.65 ± 0.13 (3.0)	0.27 ± 0.11 (3.0)	6.00 ± 1.58 (4.0)
	w/o steer	1.00 (1.5)	0.73 ± 0.23 (4.0)	0.21 ± 0.08 (4.0)	5.40 ± 1.62 (2.0)
	w/o time	0.90 (3.0)	0.43 ± 0.10 (1.0)	0.38 ± 0.07 (1.0)	5.67 ± 0.94 (3.0)
ToeSegmentation1	CTCF	0.90 (1.0)	0.85 ± 0.23 (1.0)	0.22 ± 0.17 (1.0)	8.33 ± 2.67 (4.0)
	w/o proj	0.60 (3.5)	1.13 ± 0.19 (4.0)	0.12 ± 0.06 (3.0)	7.50 ± 1.50 (2.0)
	w/o steer	0.70 (2.0)	1.12 ± 0.16 (3.0)	0.10 ± 0.03 (4.0)	7.43 ± 2.26 (1.0)
	w/o time	0.60 (3.5)	1.03 ± 0.22 (2.0)	0.13 ± 0.07 (2.0)	8.17 ± 2.79 (3.0)
ToeSegmentation2	CTCF	1.00 (2.0)	0.76 ± 0.25 (2.0)	0.27 ± 0.21 (1.0)	8.10 ± 1.92 (1.0)
	w/o proj	0.90 (4.0)	0.86 ± 0.28 (3.0)	0.22 ± 0.21 (3.0)	8.11 ± 1.29 (2.0)
	w/o steer	1.00 (2.0)	1.04 ± 0.13 (4.0)	0.13 ± 0.05 (4.0)	8.20 ± 2.04 (3.0)
	w/o time	1.00 (2.0)	0.74 ± 0.23 (1.0)	0.25 ± 0.19 (2.0)	9.30 ± 2.00 (4.0)
TwoLeadECG	CTCF	1.00 (2.0)	0.06 ± 0.01 (2.0)	0.91 ± 0.02 (2.0)	0.30 ± 0.46 (2.0)
	w/o proj	0.70 (4.0)	0.21 ± 0.09 (3.0)	0.67 ± 0.22 (3.0)	2.29 ± 1.03 (3.0)
	w/o steer	1.00 (2.0)	0.36 ± 0.12 (4.0)	0.49 ± 0.18 (4.0)	2.70 ± 1.42 (4.0)
	w/o time	1.00 (2.0)	0.05 ± 0.01 (1.0)	0.94 ± 0.03 (1.0)	0.20 ± 0.40 (1.0)
UMD	CTCF	1.00 (2.0)	0.12 ± 0.06 (1.0)	0.83 ± 0.05 (1.0)	4.20 ± 1.47 (3.0)
	w/o proj	0.30 (4.0)	0.57 ± 0.35 (3.0)	0.58 ± 0.03 (3.0)	2.00 ± 0.00 (1.0)
	w/o steer	1.00 (2.0)	0.77 ± 0.42 (4.0)	0.56 ± 0.12 (4.0)	2.60 ± 1.28 (2.0)
	w/o time	1.00 (2.0)	0.13 ± 0.07 (2.0)	0.81 ± 0.08 (2.0)	4.90 ± 2.47 (4.0)
Rank Mean	CTCF	1.73	1.42	1.52	1.85
	w/o proj	3.73	2.94	2.83	2.58
	w/o steer	2.10	3.77	3.71	2.56
	w/o time	2.44	1.87	1.94	3.02
Number of Wins	CTCF	23	16	14	14
	w/o proj	2	2	3	7
	w/o steer	16	0	1	5
	w/o time	14	8	9	2

Table 35: Dataset-wise CTCF and ablation results with the MultiROCKET-HYDRA-based classifier in the one-class setting. Validity uses all selected instances; the other metrics use each method's valid counterfactuals and are reported as mean ± standard deviation (omitted for a single valid counterfactual). Parentheses give ranks among the four variants; — denotes no valid counterfactual. Rank Mean and Number of Wins summarize all 26 datasets, with ties counted as wins.

Dataset	Method	Validity	Proximity	Sparsity	NumSeg
BME	CTCF	0.60 (2.0)	0.20 ± 0.13 (2.5)	0.79 ± 0.17 (3.5)	2.67 ± 0.47 (2.0)
	w/o proj	0.60 (2.0)	0.20 ± 0.13 (2.5)	0.79 ± 0.17 (3.5)	2.67 ± 0.47 (2.0)
	w/o steer	0.20 (4.0)	0.25 ± 0.02 (4.0)	0.81 ± 0.01 (1.0)	4.00 ± 1.00 (4.0)
	w/o time	0.60 (2.0)	0.20 ± 0.13 (1.0)	0.79 ± 0.17 (2.0)	2.67 ± 0.47 (2.0)
BirdChicken	CTCF	0.50 (3.0)	0.09 ± 0.04 (1.0)	0.86 ± 0.09 (1.0)	1.75 ± 1.30 (1.0)
	w/o proj	0.60 (2.0)	0.51 ± 0.51 (3.0)	0.54 ± 0.39 (3.0)	3.50 ± 2.87 (3.0)
	w/o steer	0.12 (4.0)	1.20 (4.0)	0.11 (4.0)	7.00 (4.0)
	w/o time	0.67 (1.0)	0.28 ± 0.31 (2.0)	0.68 ± 0.35 (2.0)	2.67 ± 2.75 (2.0)
CricketX	CTCF	0.60 (2.5)	1.18 ± 0.24 (3.0)	0.13 ± 0.07 (3.0)	5.83 ± 2.11 (2.0)
	w/o proj	0.60 (2.5)	0.99 ± 0.35 (1.0)	0.19 ± 0.16 (1.0)	7.00 ± 2.89 (4.0)
	w/o steer	0.50 (4.0)	1.36 ± 0.07 (4.0)	0.09 ± 0.03 (4.0)	5.20 ± 1.33 (1.0)
	w/o time	0.90 (1.0)	1.06 ± 0.27 (2.0)	0.18 ± 0.10 (2.0)	6.22 ± 1.31 (3.0)
CricketZ	CTCF	0.30 (3.5)	1.14 ± 0.22 (2.0)	0.12 ± 0.04 (2.0)	5.33 ± 1.70 (1.0)
	w/o proj	0.40 (2.0)	1.20 ± 0.17 (3.0)	0.10 ± 0.03 (3.0)	6.00 ± 2.24 (3.5)
	w/o steer	0.30 (3.5)	1.34 ± 0.10 (4.0)	0.07 ± 0.00 (4.0)	6.00 ± 0.82 (3.5)
	w/o time	0.90 (1.0)	1.09 ± 0.26 (1.0)	0.16 ± 0.08 (1.0)	5.89 ± 2.88 (2.0)
DistalPhalanxOutlineAgeGroup	CTCF	1.00 (2.0)	0.16 ± 0.08 (2.0)	0.71 ± 0.22 (2.0)	2.78 ± 2.20 (1.0)
	w/o proj	0.89 (4.0)	0.19 ± 0.10 (3.0)	0.67 ± 0.22 (3.0)	4.00 ± 3.08 (2.5)
	w/o steer	1.00 (2.0)	0.23 ± 0.08 (4.0)	0.55 ± 0.14 (4.0)	5.44 ± 2.06 (4.0)
	w/o time	1.00 (2.0)	0.15 ± 0.09 (1.0)	0.72 ± 0.23 (1.0)	4.00 ± 3.46 (2.5)
ECGFiveDays	CTCF	0.70 (1.5)	0.38 ± 0.08 (2.0)	0.51 ± 0.17 (2.0)	2.57 ± 0.90 (3.0)
	w/o proj	0.60 (3.0)	0.40 ± 0.07 (3.0)	0.47 ± 0.17 (3.0)	2.33 ± 0.94 (1.0)
	w/o steer	0.50 (4.0)	0.46 ± 0.11 (4.0)	0.35 ± 0.16 (4.0)	2.80 ± 1.17 (4.0)
	w/o time	0.70 (1.5)	0.38 ± 0.06 (1.0)	0.53 ± 0.17 (1.0)	2.43 ± 0.90 (2.0)
FaceAll	CTCF	0.90 (3.0)	0.77 ± 0.19 (1.0)	0.22 ± 0.05 (1.0)	10.44 ± 2.06 (3.0)
	w/o proj	1.00 (1.5)	0.85 ± 0.18 (3.0)	0.19 ± 0.05 (3.0)	10.10 ± 2.77 (2.0)
	w/o steer	0.80 (4.0)	0.95 ± 0.15 (4.0)	0.16 ± 0.03 (4.0)	9.12 ± 1.54 (1.0)
	w/o time	1.00 (1.5)	0.82 ± 0.20 (2.0)	0.20 ± 0.05 (2.0)	11.20 ± 2.32 (4.0)
FacesUCR	CTCF	0.80 (2.0)	0.87 ± 0.18 (1.0)	0.17 ± 0.03 (1.0)	10.25 ± 1.48 (4.0)
	w/o proj	0.60 (4.0)	0.92 ± 0.14 (2.0)	0.17 ± 0.02 (3.0)	10.00 ± 1.83 (2.5)
	w/o steer	0.80 (2.0)	0.97 ± 0.14 (4.0)	0.14 ± 0.03 (4.0)	9.62 ± 3.60 (1.0)
	w/o time	0.80 (2.0)	0.93 ± 0.14 (3.0)	0.17 ± 0.03 (2.0)	10.00 ± 1.94 (2.5)
FiftyWords	CTCF	0.80 (1.0)	0.74 ± 0.16 (3.0)	0.31 ± 0.05 (1.0)	3.88 ± 0.78 (1.0)
	w/o proj	0.70 (2.5)	0.71 ± 0.15 (1.0)	0.30 ± 0.05 (2.0)	4.29 ± 1.28 (2.0)
	w/o steer	0.50 (4.0)	0.84 ± 0.19 (4.0)	0.23 ± 0.04 (4.0)	5.20 ± 1.47 (4.0)
	w/o time	0.70 (2.5)	0.71 ± 0.15 (2.0)	0.29 ± 0.06 (3.0)	4.57 ± 1.18 (3.0)
GunPoint	CTCF	0.90 (1.0)	0.30 ± 0.25 (3.0)	0.56 ± 0.35 (3.0)	2.56 ± 1.50 (3.0)
	w/o proj	0.70 (2.0)	0.16 ± 0.13 (1.0)	0.74 ± 0.24 (1.0)	2.14 ± 1.46 (2.0)
	w/o steer	0.20 (4.0)	0.63 ± 0.16 (4.0)	0.12 ± 0.06 (4.0)	4.00 ± 1.00 (4.0)
	w/o time	0.60 (3.0)	0.23 ± 0.24 (2.0)	0.70 ± 0.30 (2.0)	1.83 ± 0.90 (1.0)
GunPointAgeSpan	CTCF	0.30 (1.5)	0.42 ± 0.28 (1.0)	0.50 ± 0.35 (1.5)	2.00 ± 0.00 (2.0)
	w/o proj	0.30 (1.5)	0.42 ± 0.28 (2.0)	0.50 ± 0.35 (1.5)	2.00 ± 0.00 (2.0)
	w/o steer	0.10 (3.5)	0.72 (3.0)	0.19 (3.0)	2.00 (2.0)
	w/o time	0.10 (3.5)	1.55 (4.0)	0.15 (4.0)	3.00 (4.0)
GunPointMaleVersusFemale	CTCF	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o proj	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o steer	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o time	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
GunPointOldVersusYoung	CTCF	0.50 (1.0)	0.53 ± 0.22 (2.0)	0.35 ± 0.17 (2.0)	3.00 ± 1.67 (3.0)
	w/o proj	0.40 (2.0)	1.38 ± 0.69 (4.0)	0.13 ± 0.17 (4.0)	2.00 ± 1.22 (1.5)
	w/o steer	0.20 (4.0)	0.49 ± 0.01 (1.0)	0.42 ± 0.03 (1.0)	3.50 ± 0.50 (4.0)
	w/o time	0.30 (3.0)	0.84 ± 0.72 (3.0)	0.33 ± 0.28 (3.0)	2.00 ± 1.41 (1.5)
InsectEPGRegularTrain	CTCF	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o proj	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o steer	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o time	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
InsectEPGSmallTrain	CTCF	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o proj	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o steer	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)
	w/o time	0.00 (2.5)	— (2.5)	— (2.5)	— (2.5)

Continued on next page

Table 35: Dataset-wise CTCF and ablation results with the MultiROCKET-HYDRA-based classifier in the one-class setting (continued).

Dataset	Method	Validity	Proximity	Sparsity	NumSeg
InsectWingbeatSound	CTCF	0.60 (1.5)	0.73 ± 0.06 (1.0)	0.49 ± 0.05 (3.0)	5.33 ± 0.94 (3.0)
	w/o proj	0.50 (3.5)	0.77 ± 0.07 (3.0)	0.51 ± 0.05 (2.0)	5.20 ± 0.75 (2.0)
	w/o steer	0.50 (3.5)	0.78 ± 0.06 (4.0)	0.47 ± 0.05 (4.0)	5.40 ± 1.02 (4.0)
	w/o time	0.60 (1.5)	0.74 ± 0.05 (2.0)	0.53 ± 0.04 (1.0)	4.67 ± 0.75 (1.0)
ItalyPowerDemand	CTCF	1.00 (2.5)	0.66 ± 0.35 (4.0)	0.30 ± 0.23 (4.0)	3.60 ± 1.43 (2.5)
	w/o proj	1.00 (2.5)	0.37 ± 0.31 (1.0)	0.52 ± 0.30 (1.0)	2.60 ± 1.50 (1.0)
	w/o steer	1.00 (2.5)	0.66 ± 0.36 (3.0)	0.30 ± 0.17 (3.0)	3.80 ± 1.40 (4.0)
	w/o time	1.00 (2.5)	0.52 ± 0.34 (2.0)	0.38 ± 0.27 (2.0)	3.60 ± 1.20 (2.5)
MoteStrain	CTCF	0.70 (2.5)	0.75 ± 0.11 (3.0)	0.17 ± 0.05 (4.0)	5.00 ± 1.20 (2.0)
	w/o proj	0.50 (4.0)	0.63 ± 0.10 (1.0)	0.27 ± 0.04 (1.0)	5.40 ± 0.49 (4.0)
	w/o steer	0.80 (1.0)	0.78 ± 0.08 (4.0)	0.20 ± 0.03 (3.0)	5.12 ± 1.17 (3.0)
	w/o time	0.70 (2.5)	0.65 ± 0.17 (2.0)	0.22 ± 0.13 (2.0)	4.57 ± 1.05 (1.0)
PowerCons	CTCF	1.00 (1.5)	0.89 ± 0.39 (1.0)	0.38 ± 0.18 (1.0)	5.00 ± 1.00 (2.0)
	w/o proj	0.78 (4.0)	1.18 ± 0.33 (4.0)	0.27 ± 0.16 (4.0)	4.00 ± 1.31 (1.0)
	w/o steer	0.90 (3.0)	1.06 ± 0.37 (3.0)	0.28 ± 0.13 (3.0)	5.44 ± 1.42 (4.0)
	w/o time	1.00 (1.5)	0.96 ± 0.45 (2.0)	0.36 ± 0.21 (2.0)	5.11 ± 1.52 (3.0)
ProximalPhalanxOutlineAgeGroup	CTCF	0.80 (1.5)	0.08 ± 0.07 (2.0)	0.88 ± 0.16 (2.0)	2.25 ± 2.95 (2.0)
	w/o proj	0.70 (3.0)	0.05 ± 0.06 (1.0)	0.95 ± 0.13 (1.0)	0.86 ± 2.10 (1.0)
	w/o steer	0.40 (4.0)	0.15 ± 0.04 (4.0)	0.75 ± 0.11 (4.0)	2.75 ± 2.17 (4.0)
	w/o time	0.80 (1.5)	0.09 ± 0.07 (3.0)	0.85 ± 0.15 (3.0)	2.38 ± 2.69 (3.0)
SonyAIBORobotSurface1	CTCF	1.00 (1.5)	0.50 ± 0.10 (1.0)	0.33 ± 0.11 (1.0)	6.80 ± 1.66 (1.0)
	w/o proj	0.80 (4.0)	0.56 ± 0.09 (3.0)	0.24 ± 0.07 (3.0)	7.50 ± 2.12 (2.0)
	w/o steer	1.00 (1.5)	0.60 ± 0.10 (4.0)	0.22 ± 0.07 (4.0)	7.60 ± 1.56 (3.0)
	w/o time	0.90 (3.0)	0.52 ± 0.08 (2.0)	0.28 ± 0.05 (2.0)	7.67 ± 1.94 (4.0)
SonyAIBORobotSurface2	CTCF	0.80 (1.0)	0.63 ± 0.23 (1.0)	0.32 ± 0.17 (1.0)	5.88 ± 1.05 (4.0)
	w/o proj	0.60 (4.0)	0.78 ± 0.24 (3.0)	0.22 ± 0.09 (3.0)	5.17 ± 1.67 (2.0)
	w/o steer	0.70 (2.5)	0.90 ± 0.21 (4.0)	0.19 ± 0.08 (4.0)	5.14 ± 0.83 (1.0)
	w/o time	0.70 (2.5)	0.74 ± 0.30 (2.0)	0.24 ± 0.15 (2.0)	5.43 ± 1.05 (3.0)
ToeSegmentation1	CTCF	0.40 (2.0)	0.84 ± 0.32 (3.0)	0.27 ± 0.24 (3.0)	6.00 ± 1.58 (1.0)
	w/o proj	0.30 (3.0)	0.68 ± 0.30 (2.0)	0.33 ± 0.25 (2.0)	6.33 ± 1.70 (2.0)
	w/o steer	0.10 (4.0)	1.09 (4.0)	0.14 (4.0)	9.00 (4.0)
	w/o time	0.50 (1.0)	0.58 ± 0.32 (1.0)	0.37 ± 0.24 (1.0)	6.80 ± 1.60 (3.0)
ToeSegmentation2	CTCF	0.50 (1.0)	0.61 ± 0.37 (1.0)	0.38 ± 0.31 (1.0)	6.80 ± 1.17 (3.0)
	w/o proj	0.30 (2.5)	0.68 ± 0.39 (2.0)	0.37 ± 0.26 (2.0)	6.33 ± 0.47 (2.0)
	w/o steer	0.20 (4.0)	1.16 ± 0.09 (4.0)	0.11 ± 0.02 (4.0)	5.50 ± 0.50 (1.0)
	w/o time	0.30 (2.5)	1.02 ± 0.18 (3.0)	0.15 ± 0.06 (3.0)	7.67 ± 1.70 (4.0)
TwoLeadECG	CTCF	0.70 (3.0)	0.18 ± 0.10 (1.0)	0.71 ± 0.21 (1.0)	1.43 ± 0.73 (1.0)
	w/o proj	0.70 (3.0)	0.19 ± 0.11 (2.0)	0.66 ± 0.22 (3.0)	1.71 ± 1.03 (3.0)
	w/o steer	0.70 (3.0)	0.33 ± 0.14 (4.0)	0.52 ± 0.18 (4.0)	2.29 ± 1.03 (4.0)
	w/o time	0.90 (1.0)	0.20 ± 0.11 (3.0)	0.67 ± 0.20 (2.0)	1.56 ± 0.68 (2.0)
UMD	CTCF	1.00 (1.0)	0.64 ± 0.34 (2.0)	0.56 ± 0.03 (2.0)	4.60 ± 2.50 (4.0)
	w/o proj	0.67 (4.0)	0.91 ± 0.06 (3.0)	0.55 ± 0.02 (3.0)	2.00 ± 0.00 (1.5)
	w/o steer	0.71 (3.0)	1.04 ± 0.15 (4.0)	0.54 ± 0.05 (4.0)	2.00 ± 0.00 (1.5)
	w/o time	0.75 (2.0)	0.53 ± 0.37 (1.0)	0.67 ± 0.13 (1.0)	4.50 ± 3.69 (3.0)
Rank Mean	CTCF	1.94	1.96	2.06	2.27
	w/o proj	2.85	2.35	2.44	2.19
	w/o steer	3.17	3.60	3.44	2.98
	w/o time	2.04	2.10	2.06	2.56
Number of Wins	CTCF	19	13	13	12
	w/o proj	7	9	9	11
	w/o steer	8	4	5	10
	w/o time	17	9	9	8

H Limitations

This study has several limitations. First, our experiments focus on univariate time series; evaluating the method on multivariate time series is left for future work. For multivariate time series, interpretation of counterfactual explanations is more complex, and there are fewer datasets with known evidence for evaluating interpretability. Second, this paper targets time-series classification and does not address other tasks such as time-series forecasting or clustering. Third, our experiments use public datasets and do not include deployment results on real manufacturing equipment or infrastructure diagnostics.

I Broader impacts

The proposed method may have positive impacts beyond counterfactual explanations. First, because it provides clues about which waveform patterns a classifier relies on, it may help design engineering-based features. When applied to anomaly detection, it can generate a corresponding normal time series and thus may also support anomaly repair. Furthermore, by using a preassigned label as the desired class instead of the classifier's predicted class, the method can be used to analyze the rationale behind that labeling. In other words, it can help examine which waveform patterns underlie human annotations or existing diagnostic criteria.

The method also has potential for misuse. Because it searches for input changes that alter a classifier's decision to the desired class, a malicious user could use it to find input modifications that evade anomaly detectors or classifiers. This risk is especially relevant in one-class settings, where the desired class often corresponds to the normal class; generated counterfactuals could be misused as changes that make anomalous inputs look normal to the detector. When counterfactual explanations are provided to external users in real deployments, access control, query-rate limits, log auditing, and restrictions on the disclosure scope of high-risk explanation results should be considered.

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 Section 1 state the paper's scope and main contributions for constrained FM-based CF generation in time-series classification. The theoretical and empirical claims are supported by Section 3.2, Section 4, and Section 5.

   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 H discusses the main scope limitations, including univariate time series, classification-only tasks, and the absence of industrial deployment studies. Computational cost is reported in Appendix G and Figure 5.

   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: [Yes]

Justification: Theorem 1 states the marginal-matching assumption, and Proposition 1 states the assumptions for the linearized projection step. Complete proofs are provided in Appendix B and Appendix D.

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: The experimental setup is specified at the level needed to interpret the results. Section 5.1 describes the classifiers, UCR datasets, split adjustment, evaluated test instances, baselines, and ablations. Appendix E gives the inference-time choices for K, $\mathcal{H}_\eta$, and $\mathcal{H}_\gamma$, and Appendix F reports the optimizers, batch sizes, epochs, training stages, and architectures for $\boldsymbol{u}_{\boldsymbol{\theta}}$ and the desired-class region learner $(\boldsymbol{\psi}_{\boldsymbol{w}}, \boldsymbol{c}_t, r^2_{\boldsymbol{\nu}}(t))$. Appendix G.1 further reports software versions and baseline-specific settings.

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: [No]

   Justification: The UCR datasets are publicly available, and data access and preprocessing details are described in Section 5.1 and Appendix G.1. The paper does not provide executable scripts for reproducing the main experimental results.

   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 5.1 and Appendix G describe the classifiers, datasets, data splits, baselines, software versions, and evaluation protocol. Appendix E and Appendix F provide model architectures, hyperparameters, and training details.

   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: Validity comparisons use Friedman and Nemenyi tests, and pairwise metric comparisons use two-sided Wilcoxon signed-rank tests, as described in Section 5.2, Section 5.3, and Appendix G.3. Dataset-wise tables report means and standard deviations in Appendix G.4.

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 G reports that all experiments were run on an Amazon AWS g6e.12xlarge instance with Ubuntu OS, four NVIDIA L40S GPUs, and 384 GiB RAM. Figure 5 reports dataset-wise training time and per-sample inference time for CTCF and its ablations.

   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 study uses public benchmark data and does not conduct human-subject data collection. Limitations and broader impacts are discussed in Appendix H and Appendix I, and we are not aware of any deviation from the NeurIPS Code of Ethics.

   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 I discusses positive and negative uses. It also discusses operational mitigations.

    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 does not release high-risk pretrained models, scraped datasets, or a public high-risk generative system. Potential deployment-time misuse and operational mitigations are discussed in Appendix I.

    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: [No]

Justification: The paper cites the UCR Archive and software assets, and reports aeon and PyOD versions in Appendix G.1. However, the paper does not explicitly list the licenses or terms of use for all existing assets.

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 proposes a new method but does not release a new dataset, model checkpoint, or code asset as part of the current submission.

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 experiments or new human-subject studies. It uses publicly available benchmark datasets, as described in Section 5.1 and Appendix G.1.

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 crowdsourcing experiments or new human-subject studies, so IRB approval or equivalent review is not applicable.

    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: The core method does not use LLMs as an important, original, or non-standard algorithmic component.

    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.