TRAJECTORY-GUIDED SUPERVISION RECTIFICATION WITH PROTOTYPE-REPULSIVE EXTRAPOLATION FOR OPEN-SET TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series anomaly detection (TSAD) aims to identify abnormal temporal patterns in sequential data. In real-world scenarios, anomaly annotations are often restricted to a subset of anomaly types, and open-set TSAD is therefore critical for detecting both seen and unseen anomalies using limited labeled examples. Despite the progress, existing approaches typically rely on simple augmentation that provides limited coverage of unseen patterns and tend to overfit seen patterns when nominally normal training data are contaminated by unlabeled anomalies. Towards this end, we propose a novel approach named Trajectory-Guided Supervision Rectification with Prototype-Repulsive Extrapolation (**TURPLE**) for open-set TSAD. The core of our **TURPLE** is to jointly refine contaminated supervision and expand anomaly coverage beyond seen patterns. Specifically, given the pretrained deviation-aware detector with a seen head, we track perturbation-induced deviation trajectories during training to identify latent anomalies among normal samples and rectify their labels. Building on the refined supervision, we construct normal and anomaly prototypes and introduce a prototype-repulsive extrapolation that encourages unseen anomaly prototypes beyond observed patterns. Finally, an unseen head is trained with adaptive sample-level gating on refined normal samples, seen anomalies, and pseudo-unseen anomalies. Extensive experiments demonstrate the effectiveness of **TURPLE** compared with state-of-the-art approaches. Code is available at https://anonymous.4open.science/r/TURPLE-634F.

1 INTRODUCTION

Time series anomaly detection aims to identify unusual patterns deviating from the majority of sequentially observed data. It has been widely applied in critical domains such as industrial monitoring (Mao et al., 2026), financial management (Anandakrishnan et al., 2018), and healthcare analytics (Hundman et al., 2018). However, real-world time series are often non-stationary and high-dimensional, with complex inter-variable correlations and continuously evolving normal patterns, while anomalies themselves are rare and diverse, making large-scale annotation prohibitively expensive. Consequently, most existing TSAD methods are unsupervised and assume that only normal data are available during training (Su et al., 2019; Lai et al., 2024). Yet in practice, a limited number of labeled anomalies are often accessible, such as confirmed equipment failures recorded in maintenance logs or verified fault incidents identified in cloud operations. These anomalies provide valuable priors about true anomaly patterns, but unsupervised methods cannot exploit them.

Recently, open-set anomaly detection (OSAD) (Ding et al., 2022; Zhu et al., 2024; Wang et al., 2025a;b) has been proposed to leverage limited labeled anomalies to learn detectors for both seen and unseen anomalies. It learns a representation of normal data, defines a decision boundary, and uses a few labeled anomalies to separate both seen and unseen anomalies from normal data. While OSAD can exploit labeled anomalies to reduce false positives, its extension to time series faces challenges (Liu et al., 2025a), as the limited labeled anomalies cover only a small portion of the abnormal distribution and normal patterns may drift over time, making it difficult to learn a decision boundary that generalizes to unseen anomalies. Existing open-set TSAD methods mainly fall into two directions. One direction focuses on contamination-robust detection, which alleviates the impact

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

of contaminated normal samples on normality modeling (Qiu et al., 2022; Mou et al., 2026). The other direction synthesizes pseudo-unseen anomalies from limited labeled anomalies to expand the decision boundary beyond seen classes (Lai et al., 2024; Zhou et al., 2026).

Despite recent progress, existing open-set TSAD methods still suffer from two key limitations that have not been fully addressed: ❶ ***Low-quality pseudo anomalies with misleading supervision for anomaly detection.*** Existing methods often generate pseudo-anomalies via data augmentation or other heuristic strategies, which inevitably contain noise and provide misleading supervision. Without an explicit quality assessment, such noisy pseudo-anomalies are treated as reliable supervision, biasing the model toward spurious anomaly patterns and continually misleading the optimization, thereby degrading the learned decision boundary. ❷ ***Failing to detect unseen anomalies with implicit bias in learning temporal signals.*** With only a limited number of labeled anomalies covering a small portion of the abnormal distribution, existing methods tend to severely overfit the seen anomalies, leading to a decision boundary heavily skewed toward seen anomaly patterns. Consequently, they often fail to detect unseen anomalies that significantly deviate from seen classes, and this issue is further exacerbated when normal patterns drift over time.

To address the aforementioned challenges, we propose a novel approach named Trajectory-Guided Supervision Rectification with Prototype-Repulsive Extrapolation (`TURPLE`) for open-set time series anomaly detection. The goal of `TURPLE` is to establish a unified framework that jointly refines contaminated supervision and expands anomaly coverage beyond seen patterns. Given the contamination in nominally normal training data and the limited coverage of labeled anomalies, `TURPLE` first pre-trains a deviation-aware scorer with a seen head and records Gaussian perturbation-induced deviation trajectories for each nominally normal sample across training epochs. By calibrating these trajectories against trusted normal references and extracting three complementary statistics, namely mean response, Mann-Kendall trend, and late-stage shift, `TURPLE` conservatively identifies and relabels latent contaminated anomalies. Moreover, while the refined supervision provides a cleaner basis for anomaly modeling, it still cannot cover unseen anomaly classes. Consequently, we construct normal and seen-anomaly prototypes from the refined data and learn a frequency-domain amplitude and phase augmenter that extrapolates anomaly seeds away from both prototypes, yielding diverse and temporally plausible pseudo-unseen anomalies. Finally, to prevent unreliable pseudo-anomalies from misleading the optimization, `TURPLE` trains an unseen head jointly with adaptive sample-level gating that down-weights such samples, while the shared encoder and detection head are optimized on refined normal samples, seen anomalies, and gated pseudo-unseen anomalies.

The contribution of this paper is summarized as follows: ❶ ***Conceptual:*** We reinterpret contaminated anomalies not only as noise to be removed but as valuable, informative supervision for open-set learning, thereby coupling training-data purification with the principled expansion of the unseen anomaly space. ❷ ***Methodological:*** We propose `TURPLE`, which calibrates dynamic perturbation responses against trusted normal references for more reliable contamination identification, generates temporally realistic and diverse pseudo-unseen anomalies through prototype-guided amplitude–phase refinement, and integrates heterogeneous samples through confidence-weighted joint learning in a unified manner. ❸ ***Experimental:*** Extensive and comprehensive experiments across multiple diverse benchmarks, varying contamination ratios, and challenging open-set settings demonstrate the consistent and robust superiority of `TURPLE` in detecting both seen and unseen anomalies.

2 Related Work

Time Series Anomaly Detection. Time series anomaly detection (TSAD) aims to identify temporal outliers in real-world systems. Recent TSAD studies (Darban et al., 2025; Liu et al., 2025b) focus on learning representations or normality patterns from unlabeled streams due to the scarcity of reliable anomaly annotations. Association- and contrastive approaches (Xu et al., 2021; Yang et al., 2023) score anomalies via association discrepancy or representation inconsistency under stable normal dependencies. Frequency-aware approaches (He et al., 2026; Wu et al., 2025) exploit domain-rotatable time-frequency representations or frequency patching for fine-grained periodic deviations. Other approaches incorporate causal structures or diffusion-based generative modeling (Wang et al., 2026b; Li et al., 2026). However, most existing methods rely on predominantly normal training data and are not designed to handle contaminated normal data, which can compromise anomaly detection when anomalous patterns are inadvertently included during training.

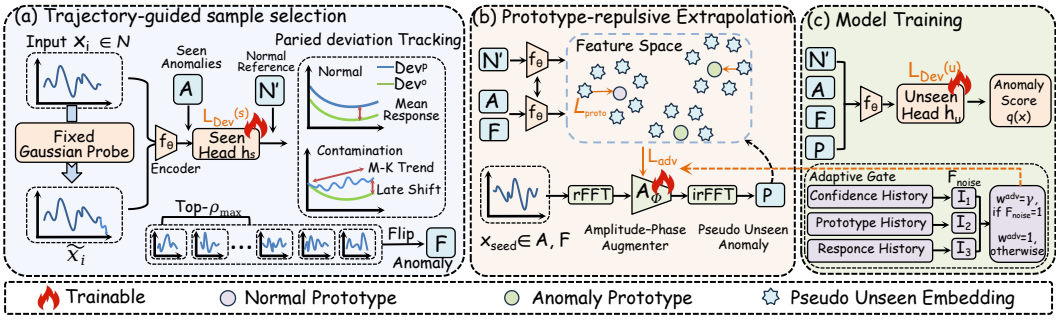

Figure 1: Overview of TURPLE. It refines contaminated labels via Gaussian-perturbation-induced deviation trajectories, synthesizes pseudo-unseen anomalies through prototype-repulsive extrapolation, and stabilizes joint training with adaptive sample-level gating.

Open-Set Anomaly Detection. Open-set anomaly detection (OSAD) aims to accurately and robustly detect both known anomalies and novel unseen anomalies, which is increasingly crucial in diverse real-world applications where anomaly types are often highly open-ended and inherently unpredictable (Ding et al., 2022; Zhu et al., 2024; Kim et al., 2026). Existing mainstream approaches are broadly divided into discriminative and generative paradigms, each with fundamentally distinct strategies. Discriminative methods learn explicit decision boundaries for known anomalies while reserving ambiguous regions for unknowns, with several representative works such as DRA (Ding et al., 2022), DPDL (Wang et al., 2025a), and WSAD-DT (Durani et al., 2025). Generative methods, in contrast, explicitly synthesize diverse pseudo-anomalies to simulate the unseen ones, as exemplified by recent methods such as IMPACT (Zhou et al., 2026), MPFM (Wang et al., 2026a), and DEMO (Yu et al., 2026). However, the majority of existing OSAD methods still heavily rely on heuristic augmentations or hand-crafted statistical priors to generate pseudo-anomalies, which often struggle to capture the true underlying distribution of unseen anomalies and may not generalize sufficiently well to complex and highly non-stationary time series data.

3 The Proposed TURPLE

Preliminaries. Let $\mathcal{S}_{\text{tr}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ be the training set for open-set time-series anomaly detection, where $\boldsymbol{x}_i \in \mathbb{R}^{T\times D}$ is a window of T steps and D channels, and $y_i \in \{0, 1\}$ is its observed label. Here, $\mathcal{S}_{\text{n}}$ is the normal subset with potential contamination, i.e., a small fraction of normal-labeled samples ($y_i = 0$) may actually be anomalous ($y_i = 1$), and $\mathcal{S}_{\text{a}}$ is a small set of labeled anomalies from seen classes. Formally, $\mathcal{S}_{\text{tr}} \doteq \mathcal{S}_{\text{n}} \cup \mathcal{S}_{\text{a}}$ with $|\mathcal{S}_{\text{a}}| \ll |\mathcal{S}_{\text{n}}|$. Labeled anomalies belong to seen classes $\mathcal{K}_{\text{seen}} \subseteq \mathcal{K}$, with $\mathcal{K}$ the set of all anomaly classes, and the remaining $\mathcal{K}_{\text{unseen}} = \mathcal{K} \setminus \mathcal{K}_{\text{seen}}$ are unseen classes without labeled training samples. Contaminated samples in $\mathcal{S}_{\text{n}}$ may come from either $\mathcal{K}_{\text{seen}}$ or $\mathcal{K}_{\text{unseen}}$ but are mislabeled as normal. We aim to learn an anomaly scoring function $q : \mathbb{R}^{T\times D} \to \mathbb{R}$ such that $q(\boldsymbol{x}_{\text{a}}) \gg q(\boldsymbol{x}_{\text{n}})$ for any normal $\boldsymbol{x}_{\text{n}}$ and any anomaly $\boldsymbol{x}_{\text{a}}$. For brevity, we denote $\mathcal{N} \doteq \mathcal{S}_{\text{n}}$, $\mathcal{A} \doteq \mathcal{S}_{\text{a}}$, and let T_{ep} be the number of epochs in the initial training phase.

3.1 Framework Overview

In this work, we address open-set TSAD under contaminated supervision from the dual perspectives of supervision rectification and anomaly coverage expansion, mitigating misleading pseudo-anomaly supervision and implicit bias toward seen patterns. The core of TURPLE is to jointly refine contaminated supervision and expand anomaly coverage beyond seen classes. Specifically, given a pretrained deviation-aware detector, TURPLE tracks perturbation-induced deviation trajectories to identify latent anomalies and rectify their labels. Building on the refined supervision, it constructs normal and anomaly prototypes and introduces a prototype-repulsive extrapolation to encourage unseen anomaly prototypes beyond observed patterns. Finally, an unseen head is trained with adaptive sample-level gating on the augmentation side. An illustration of TURPLE is shown in Figure 1.

3.2 Deviation-Based Scoring Pre-training

Reliably identifying contaminated normal samples is challenging, as contaminated anomalies may resemble genuine normals in the early stage of training, while genuine normals can occasionally

exhibit large deviations. To address this, we pre-train the shared TCN encoder $f_\theta(\cdot)$ and a dedicated seen head $h_s(\cdot)$, implemented as a two-layer MLP, on $\mathcal{N}$ and $\mathcal{A}$ with a deviation-based objective, so that the scorer is sensitive to anomalous patterns while remaining stable on genuine normal samples.

Specifically, we estimate the mean and standard deviation of each score dimension from trusted normal samples, and define the deviation and the corresponding training objective as:

$$\begin{cases} \mathrm{Dev}^{(s)}(\boldsymbol{s}^{(s)}) = \dfrac{1}{C_s}\displaystyle\sum_{k=1}^{C_s} \dfrac{(s_k^{(s)} - \mu_k^{(s)})^2}{(\nu_k^{(s)})^2}, \\ \mathcal{L}_{\mathrm{dev}}^{(s)} = \dfrac{1}{|\mathcal{B}|}\displaystyle\sum_{i\in\mathcal{B}} \Big[(1-y_i)\,\mathrm{Dev}^{(s)}(\boldsymbol{s}_i^{(s)}) + y_i\,\max\Big(0, \alpha_{\mathrm{dev}} - \mathrm{Dev}^{(s)}(\boldsymbol{s}_i^{(s)})\Big)\Big], \end{cases} \tag{1}$$

where $\boldsymbol{s}_i^{(s)} = h_s(f_\theta(\boldsymbol{x}_i))$ is the C_s-dimensional score vector, $\boldsymbol{\mu}^{(s)}$ and $\boldsymbol{\nu}^{(s)}$ are the mean and standard deviation estimated from normal samples, $\mathcal{B}$ is the current mini-batch, and α_{dev} is the deviation margin. After pre-training, we freeze both $f_\theta(\cdot)$ and $h_s(\cdot)$ during the label-refinement stage, so that the deviation trajectories used for flipping decisions are computed by a fixed scorer. In the subsequent joint-training stage, we unfreeze $f_\theta(\cdot)$ to adapt the embedding space to the refined supervision and the prototype structure, while $h_s(\cdot)$ remains frozen as a stable seen-class reference.

3.3 Trajectory-guided sample selection for Label Refinement

In open-set TSAD, existing methods (Tang et al., 2022; Lai et al., 2024; Liu et al., 2024) typically assume training labels are reliable and directly fit observed labels, making them prone to overfitting false normal labels and yielding poor generalization on unseen anomalies. Thus, we exploit an inductive bias rooted in training dynamics, where the temporal evolution of each sample's deviation to a mild Gaussian perturbation reveals whether it is truly normal or a contaminated anomaly. We record these response trajectories during pre-training and extract complementary statistics from them to conservatively identify and relabel contaminated samples, purifying the training supervision.

Gaussian Perturbation-aware Deviation Trajectory Extraction. To capture the deviation trajectory of each sample during training, we probe each sample with a fixed Gaussian perturbation $\mathcal{T}(\cdot)$. For an input $\boldsymbol{x}$,

$$\mathcal{T}(\boldsymbol{x}) = \boldsymbol{x} + \varepsilon\boldsymbol{\zeta}, \qquad \boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{2}$$

where ε is a fixed offset controlling the perturbation strength. For each observed-normal sample $\boldsymbol{x}_i$ at each epoch $t \in \{1, \ldots, T_{\mathrm{ep}}\}$, the probe-induced deviation change is:

$$\delta_i^{(t)} = \mathrm{Dev}^{(s)}\big(h_s(f_\theta(\tilde{\boldsymbol{x}}_i))\big) - \mathrm{Dev}^{(s)}\big(h_s(f_\theta(\boldsymbol{x}_i))\big), \qquad \tilde{\boldsymbol{x}}_i = \mathcal{T}(\boldsymbol{x}_i). \tag{3}$$

The sequence $\boldsymbol{\delta}_i = [\delta_i^{(1)}, \ldots, \delta_i^{(T_{\mathrm{ep}})}]^\top$ forms the trajectory of i-th sample. Rather than relying on the response at a single checkpoint, we exploit how the probe response evolves during warm-up. Intuitively, if optimization increases the probe sensitivity of contaminated anomalies more strongly than that of genuine normals, their response gap is progressively amplified, providing increasingly discriminative evidence for label refinement. We formalize this effect through the training dynamics.

Proposition 3.1. *Let $\boldsymbol{r}_t := (\boldsymbol{\mu}_t^{(s)}, \boldsymbol{\nu}_t^{(s)})$ denote the normal-reference statistics at epoch t. To make this dependence explicit, define*

$$D_{\boldsymbol{\omega},\boldsymbol{r}}(\boldsymbol{x}) = \mathrm{Dev}^{(s)}\big(h_s(f_\theta(\boldsymbol{x})); \boldsymbol{r}\big), \qquad \delta_{\boldsymbol{\omega},\boldsymbol{r}}(\boldsymbol{x}) = D_{\boldsymbol{\omega},\boldsymbol{r}}(\mathcal{T}(\boldsymbol{x})) - D_{\boldsymbol{\omega},\boldsymbol{r}}(\boldsymbol{x}). \tag{4}$$

Let P_N and P_C denote the latent distributions of genuine-normal and contaminated samples within $\mathcal{N}$, respectively. Define

$$G(\boldsymbol{\omega}, \boldsymbol{r}) = \mathbb{E}_{\boldsymbol{x}\sim P_C}[\delta_{\boldsymbol{\omega},\boldsymbol{r}}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim P_N}[\delta_{\boldsymbol{\omega},\boldsymbol{r}}(\boldsymbol{x})], \qquad G_t := G(\boldsymbol{\omega}_t, \boldsymbol{r}_t). \tag{5}$$

Suppose the warm-up update is $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \eta\boldsymbol{g}_t$, and for fixed $\boldsymbol{r}_t$, $\delta_{\boldsymbol{\omega},\boldsymbol{r}_t}(\boldsymbol{x})$ is L_δ-smooth in $\boldsymbol{\omega}$. Writing $\mathbb{E}_C$ and $\mathbb{E}_N$ for expectations under P_C and P_N, respectively, define

$$\Gamma_t := \mathbb{E}_C[-\langle\nabla_{\boldsymbol{\omega}}\delta_{\boldsymbol{\omega}_t,\boldsymbol{r}_t}(\boldsymbol{x}), \boldsymbol{g}_t\rangle] - \mathbb{E}_N[-\langle\nabla_{\boldsymbol{\omega}}\delta_{\boldsymbol{\omega}_t,\boldsymbol{r}_t}(\boldsymbol{x}), \boldsymbol{g}_t\rangle]. \tag{6}$$

Define the magnitude of the reference-induced gap drift as:

$$B_t := |G(\boldsymbol{\omega}_{t+1}, \boldsymbol{r}_{t+1}) - G(\boldsymbol{\omega}_{t+1}, \boldsymbol{r}_t)|. \tag{7}$$

Then

$$G_{t+1} - G_t \geq \eta\Gamma_t - L_\delta \eta^2 \|\boldsymbol{g}_t\|_2^2 - B_t, \qquad \Gamma_t > L_\delta \eta \|\boldsymbol{g}_t\|_2^2 + \frac{B_t}{\eta} \implies G_{t+1} > G_t. \tag{8}$$

Moreover, suppose $\delta_i^{(t)} = \mu_{Y_i,t} + \xi_i^{(t)}$, *where* $Y_i \in \{N, C\}$, $\mu_{Y,t} := \mathbb{E}[\delta_i^{(t)} \mid Y_i = Y]$, *and* $\{\xi_i^{(t)}\}_{t=1}^{T_{\text{ep}}}$ *form a martingale-difference sequence that is conditionally* σ^2*-sub-Gaussian with respect to the natural filtration generated by the response history. Define*

$$\bar{\delta}_i := \frac{1}{T_{\text{ep}}} \sum_{t=1}^{T_{\text{ep}}} \delta_i^{(t)}, \qquad \bar{G} := \frac{1}{T_{\text{ep}}} \sum_{t=1}^{T_{\text{ep}}} G_t. \tag{9}$$

If $\bar{G} > 0$, *using the midpoint of the two population means of* $\bar{\delta}_i$ *as the decision threshold, and defining* P_{err} *as the maximum of the two class-conditional error probabilities, we have*

$$P_{\text{err}} \leq \exp\left(-\frac{T_{\text{ep}} \bar{G}^2}{8\sigma^2}\right). \tag{10}$$

Multi-criteria for Conservative Label Flipping. To robustly identify hard flip candidates that exhibit genuine contamination trends, we extract from each trajectory three complementary statistics that together provide a robust indicator of contamination: the mean response $\bar{\delta}_i$, the Mann–Kendall trend (McLeod, 2005) statistic $Z_i = \text{MK}(\boldsymbol{\delta}_i)$, and the late-minus-early shift $\Delta_i = \text{mean}(\boldsymbol{\delta}_i^{\text{late}}) - \text{mean}(\boldsymbol{\delta}_i^{\text{early}})$. Here, $\boldsymbol{\delta}_i^{\text{early}}$ and $\boldsymbol{\delta}_i^{\text{late}}$ denote the sub-trajectories over the first and last $r\%$ of the T_{ep} epochs, respectively. These three statistics respectively capture the overall sensitivity magnitude, trend significance, and late-stage enhancement magnitude. A sample is considered a hard flip candidate only if all three conditions are satisfied (Pan et al., 2026):

$$\bar{\delta}_i > \tau_\delta^{\text{emp}}, \qquad Z_i > \tau_{\text{MK}}, \qquad \Delta_i > 0, \tag{11}$$

where τ_δ^{emp} is set to a high percentile of the mean responses over $\mathcal{N}$ to avoid flipping too many samples, and τ_{MK} is a standard significance threshold from the Mann–Kendall test. The conjunction of these three criteria is deliberately conservative, ensuring high sensitivity and a persistent increasing trend, thus greatly reducing false positives compared to any single-threshold approach. Candidates are ranked by a combined trend score, and at most a fraction $\rho_{\max}$ of $\mathcal{N}$ is flipped, i.e., $|\mathcal{F}| \leq \rho_{\max}|\mathcal{N}|$, where $\mathcal{F}$ denotes the flipped set. This yields the refined sets $\mathcal{N}' = \mathcal{N} \setminus \mathcal{F}$ and $\mathcal{A}' = \mathcal{A} \cup \mathcal{F}$ for cleaner subsequent supervision.

3.4 Prototype-repulsive Extrapolation for Unseen Anomaly Synthesizing

After label refinement purifies the anomaly supervision, the next objective is to synthesize unseen anomaly examples that expand the anomaly coverage beyond seen classes. Traditional augmentation (Wang et al., 2024; Le Guennec et al., 2016) only creates label-preserving variations within known patterns. Such variants typically remain close to the original samples in the feature space and cannot cover unseen anomalies. Meanwhile, existing strategies (Castellini et al., 2023; Chen et al., 2023) often remain confined to the seen distribution and may generate unrealistic or uncontrollable samples. To address this, we propose Prototype-repulsive Extrapolation for Unseen Anomaly Synthesizing, which encourages departures from observed anomaly seeds under a prototype-guided objective by pushing perturbed embeddings away from the prototypes to simulate unseen anomalies.

Prototype-guided Unseen Anomaly Synthesizing. To synthesize pseudo-unseen samples beyond the known regions, we first construct two momentum-updated prototypes, $\boldsymbol{c}_n$ for normal samples and $\boldsymbol{c}_a$ for anomaly samples. We compute squared Euclidean distances from any embedding $\boldsymbol{z}$ to prototypes and use them to define a prototype loss that pulls normal embeddings toward $\boldsymbol{c}_n$ and anomaly embeddings toward $\boldsymbol{c}_a$, reinforcing intra-class compactness and separation:

$$\mathcal{L}_{\text{proto}} = \sum_{i \in \mathcal{N}'} d_n(\boldsymbol{z}_i) + \sum_{i \in \mathcal{A}'} d_a(\boldsymbol{z}_i), \tag{12}$$

where $d_n(\boldsymbol{z}) = \|\boldsymbol{z} - \boldsymbol{c}_n\|_2^2$ and $d_a(\boldsymbol{z}) = \|\boldsymbol{z} - \boldsymbol{c}_a\|_2^2$ are the squared Euclidean distances to the normal and anomaly prototypes. We then introduce a learnable frequency augmenter $\mathcal{A}_\phi(\cdot)$ to the anomaly seeds in $\mathcal{A}'$. Specifically, for each anomaly seed $\boldsymbol{x}_{\text{seed}} \in \mathbb{R}^{T \times D}$, we apply rFFT to each channel:

$$\boldsymbol{X}^{(d)} = \text{rFFT}(\boldsymbol{x}_{\text{seed}}^{(d)}), \qquad d = 1, \ldots, D, \tag{13}$$

The augmenter stores per-channel amplitude and phase offsets $\boldsymbol{\delta}_a, \boldsymbol{\delta}_p \in \mathbb{R}^{D\times K}$ for the K lowest frequencies $\mathcal{B} = \{1, \dots, K\}$. For $k \in \mathcal{B}$, it modifies the spectrum as:

$$\tilde{\boldsymbol{X}}^{(d)}[k] = |\boldsymbol{X}^{(d)}[k]| \cdot \big(1 + \tanh(\boldsymbol{\delta}_{a,d,k})\big) \cdot e^{\,j\left(\angle \boldsymbol{X}^{(d)}[k] + \tanh(\boldsymbol{\delta}_{p,d,k})\right)}, \tag{14}$$

while high frequencies remain unchanged. After modifying the low-frequency components, we transform the spectrum back to the time domain via inverse rFFT, obtaining the augmented sample $\hat{\boldsymbol{x}} = \mathcal{A}_\phi(\boldsymbol{x}_{\text{seed}})$. To encourage generated samples to deviate from both the normal and anomaly manifolds, we define the per-sample adversarial loss based on the joint prototype distance:

$$\ell^{\text{adv}}(\hat{\boldsymbol{x}}) = -Q\big(f_\theta(\hat{\boldsymbol{x}})\big), \qquad Q(\boldsymbol{z}) = d_n(\boldsymbol{z}) + d_a(\boldsymbol{z}), \tag{15}$$

where the augmenter $\mathcal{A}_\phi$ is optimized by moving the generated samples in the input space. Generating for all seeds in $\mathcal{A}'$ yields the pseudo-anomaly set $\mathcal{P} = \{\hat{\boldsymbol{x}} \mid \boldsymbol{x}_{\text{seed}} \in \mathcal{A}'\}$. The following proposition characterizes how input-space perturbations affect prototype distances.

Proposition 3.2. *Let $\boldsymbol{y} = \boldsymbol{z} - \boldsymbol{m}$ denote the displacement from the prototype midpoint, and consider the radial ascent $\boldsymbol{z}^+ = \boldsymbol{z} + s\boldsymbol{y}$ with $s > 0$. If $\|\boldsymbol{y}\|_2^2 > |\boldsymbol{d}^\top \boldsymbol{y}|$, then the representation moves farther from both prototype centers, i.e., $d_n(\boldsymbol{z}^+) > d_n(\boldsymbol{z})$ and $d_a(\boldsymbol{z}^+) > d_a(\boldsymbol{z})$. A simpler sufficient condition is $\|\boldsymbol{z} - \boldsymbol{m}\|_2 > \frac{1}{2}\|\boldsymbol{c}_a - \boldsymbol{c}_n\|_2$. For $\boldsymbol{z} = f_\theta(\boldsymbol{x})$, let $J_\theta(\boldsymbol{x})$ denote the encoder Jacobian and consider a perturbation $\boldsymbol{\eta} = \epsilon\hat{\boldsymbol{\eta}}$ along a fixed direction $\hat{\boldsymbol{\eta}}$. If $\langle \boldsymbol{z} - \boldsymbol{c}_n, J_\theta(\boldsymbol{x})\hat{\boldsymbol{\eta}}\rangle > 0$ and $\langle \boldsymbol{z} - \boldsymbol{c}_a, J_\theta(\boldsymbol{x})\hat{\boldsymbol{\eta}}\rangle > 0$, then there exists $\epsilon_0 > 0$ such that, for all $0 < \epsilon < \epsilon_0$, both prototype distances increase under the perturbed input $\boldsymbol{x} + \epsilon\hat{\boldsymbol{\eta}}$.*

Adaptive Gating for Reliable Anomaly Extrapolation. To avoid generating harmful noise, we introduce sample-level gating on the augmenter side during joint training. The gating mechanism tracks the short-term histories of confidence, prototype loss, and augmentation response for each generated pseudo-unseen sample $\hat{\boldsymbol{x}}$, yielding three binary indicators. It flags $\hat{\boldsymbol{x}}$ as harmful noise if

$$F_{\text{noise}}(\hat{\boldsymbol{x}}) = \big(\neg I_1(\hat{\boldsymbol{x}})\big) \wedge I_2(\hat{\boldsymbol{x}}) \wedge I_3(\hat{\boldsymbol{x}}), \tag{16}$$

where I_1 denotes a significant upward Mann–Kendall confidence trend, I_2 a growing discrepancy between clean and augmented loss traces, and I_3 a large relative shift of augmented loss. The details of the three gates are in Appendix C. Based on this flag, we define the gating weight as:

$$w_{\text{adv}}(\hat{\boldsymbol{x}}) = \begin{cases} \gamma, & F_{\text{noise}}(\hat{\boldsymbol{x}}) = 1, \\ 1, & \text{otherwise}, \end{cases} \qquad 0 \le \gamma < 1. \tag{17}$$

The batch-level loss then aggregates the weighted per-sample losses:

$$\mathcal{L}_{\text{adv}} = \frac{1}{|\mathcal{P}_{\text{b}}|} \sum_{\hat{\boldsymbol{x}} \in \mathcal{P}_{\text{b}}} w_{\text{adv}}(\hat{\boldsymbol{x}})\, \ell^{\text{adv}}(\hat{\boldsymbol{x}}) + \lambda_{\text{reg}} \Big(\|\tanh(\boldsymbol{\delta}_a)\|_2^2 + \|\tanh(\boldsymbol{\delta}_p)\|_2^2 \Big), \tag{18}$$

where $\mathcal{P}_{\text{b}} \subseteq \mathcal{P}$ is a mini-batch of generated samples. We then characterize when down-weighting gated samples improves the optimization direction.

Proposition 3.3. *Let $\boldsymbol{\vartheta}$ denote the parameters optimized during joint training, and let $\mathcal{L}_{\text{clean}}$ denote the loss computed only on the refined normal set, $\boldsymbol{g} := \nabla_{\boldsymbol{\vartheta}} \mathcal{L}_{\text{clean}}$, and $\boldsymbol{h}(\hat{\boldsymbol{x}}) := \frac{\lambda_{\text{adv}}}{|\mathcal{P}_{\text{b}}|} \nabla_{\boldsymbol{\vartheta}} \ell^{\text{adv}}(\hat{\boldsymbol{x}})$. Let $\mathcal{G} := \{\hat{\boldsymbol{x}} \in \mathcal{P}_{\text{b}} : F_{\text{noise}}(\hat{\boldsymbol{x}}) = 1\}$ and $\boldsymbol{h}_{\mathcal{G}} := \sum_{\hat{\boldsymbol{x}} \in \mathcal{G}} \boldsymbol{h}(\hat{\boldsymbol{x}})$. Define the overall aggregate conflict $C_{\mathcal{G}} := -\langle \boldsymbol{g}, \boldsymbol{h}_{\mathcal{G}} \rangle$. Let $\boldsymbol{u}$ and $\boldsymbol{u}_\gamma$ denote joint gradient directions before and after down-weighting $\boldsymbol{h}_{\mathcal{G}}$ by $0 \le \gamma < 1$. Down-weighting the selected gradients by $0 \le \gamma < 1$, with other components fixed, yields $\langle \boldsymbol{g}, \boldsymbol{u}_\gamma \rangle - \langle \boldsymbol{g}, \boldsymbol{u} \rangle = (1 - \gamma) C_{\mathcal{G}}$. Thus, $C_{\mathcal{G}} > 0$ strictly guarantees that gating improves alignment with the clean descent direction.*

Remark 3.1. *The aggregate conflict $C_{\mathcal{G}}$ measures whether the gated samples' gradients oppose the clean gradient in the joint parameter space. When $C_{\mathcal{G}} > 0$, Proposition 3.3 shows that down-weighting these contributions improves the joint gradient's inner-product alignment with the clean gradient, with other components fixed. This provides a conditional interpretation of gating; the history-based selection rule does not itself guarantee positive conflict.*

3.5 Summarization

Having refined the labels and synthesized pseudo-unseen samples, we train the complete detection model with the following overall objective:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{Dev}}^{u} + \lambda_{\text{proto}}\, \mathcal{L}_{\text{proto}} + \lambda_{\text{adv}}\, \mathcal{L}_{\text{adv}}, \tag{19}$$

Table 1: Detection accuracy (AUC, %) under the unsupervised and open-set **general** settings. Best results are in **bold**, with the second-best underlined.

Methods	UCR	ASD	PSM	SMD	CT	SAD	PTBXL	TUSZ
Unsupervised								
TCN-AE	50.63$_{\pm 0.46}$	58.73$_{\pm 0.49}$	58.49$_{\pm 0.79}$	64.76$_{\pm 0.19}$	50.02$_{\pm 0.05}$	50.01$_{\pm 0.02}$	58.74$_{\pm 0.28}$	64.17$_{\pm 0.06}$
THOC	52.75$_{\pm 0.56}$	55.74$_{\pm 1.57}$	58.02$_{\pm 1.51}$	62.55$_{\pm 0.47}$	74.50$_{\pm 3.16}$	57.76$_{\pm 3.14}$	62.89$_{\pm 1.97}$	62.45$_{\pm 1.92}$
TranAD	49.26$_{\pm 0.73}$	56.86$_{\pm 0.68}$	60.34$_{\pm 0.40}$	64.40$_{\pm 0.25}$	29.28$_{\pm 3.12}$	49.10$_{\pm 1.09}$	58.51$_{\pm 1.18}$	62.82$_{\pm 0.73}$
DCdetector	51.40$_{\pm 1.43}$	47.94$_{\pm 1.05}$	50.51$_{\pm 0.72}$	50.23$_{\pm 0.85}$	49.20$_{\pm 1.80}$	49.02$_{\pm 1.15}$	51.28$_{\pm 0.78}$	50.55$_{\pm 1.10}$
GPT4TS	54.60$_{\pm 0.87}$	62.42$_{\pm 2.06}$	61.14$_{\pm 0.63}$	66.20$_{\pm 0.31}$	60.32$_{\pm 0.80}$	48.09$_{\pm 3.59}$	47.82$_{\pm 0.89}$	66.31$_{\pm 0.31}$
COUTA	53.44$_{\pm 1.14}$	61.74$_{\pm 2.67}$	58.29$_{\pm 3.90}$	60.93$_{\pm 1.36}$	51.09$_{\pm 0.00}$	50.00$_{\pm 0.00}$	56.71$_{\pm 1.86}$	62.45$_{\pm 0.89}$
DADA	50.06$_{\pm 0.42}$	50.67$_{\pm 0.09}$	54.27$_{\pm 0.06}$	64.45$_{\pm 0.07}$	56.08$_{\pm 0.12}$	49.98$_{\pm 0.90}$	63.22$_{\pm 0.21}$	44.20$_{\pm 0.01}$
Open-Set								
DevNet	49.82$_{\pm 0.36}$	49.41$_{\pm 3.41}$	49.88$_{\pm 2.20}$	49.62$_{\pm 1.37}$	49.63$_{\pm 0.65}$	50.00$_{\pm 0.01}$	50.00$_{\pm 0.00}$	49.63$_{\pm 0.80}$
DSAD	51.99$_{\pm 1.16}$	62.56$_{\pm 1.27}$	60.93$_{\pm 1.79}$	72.75$_{\pm 0.84}$	87.77$_{\pm 0.43}$	62.42$_{\pm 1.27}$	62.49$_{\pm 2.29}$	68.60$_{\pm 2.69}$
NCAD	55.02$_{\pm 0.15}$	61.71$_{\pm 0.89}$	60.05$_{\pm 2.03}$	64.39$_{\pm 0.72}$	64.30$_{\pm 1.13}$	50.07$_{\pm 1.88}$	45.93$_{\pm 0.69}$	69.80$_{\pm 1.02}$
DRA	54.38$_{\pm 0.82}$	57.26$_{\pm 1.52}$	60.29$_{\pm 2.37}$	59.31$_{\pm 2.50}$	88.73$_{\pm 0.27}$	40.57$_{\pm 2.79}$	63.33$_{\pm 2.45}$	72.48$_{\pm 1.65}$
MOSAD	54.76$_{\pm 0.09}$	57.84$_{\pm 0.25}$	61.58$_{\pm 0.18}$	71.20$_{\pm 0.38}$	80.42$_{\pm 1.56}$	53.14$_{\pm 0.87}$	62.74$_{\pm 1.72}$	69.80$_{\pm 2.83}$
InvAD	55.64$_{\pm 0.71}$	61.32$_{\pm 5.44}$	61.34$_{\pm 0.88}$	71.24$_{\pm 0.86}$	75.71$_{\pm 0.43}$	52.88$_{\pm 1.71}$	50.20$_{\pm 3.15}$	61.63$_{\pm 4.27}$
WSAD-DT	51.47$_{\pm 0.48}$	63.76$_{\pm 1.22}$	60.76$_{\pm 1.58}$	70.60$_{\pm 0.80}$	85.18$_{\pm 1.22}$	64.42$_{\pm 1.30}$	65.83$_{\pm 3.04}$	78.70$_{\pm 0.98}$
IMPACT	59.21$_{\pm 1.34}$	65.76$_{\pm 2.17}$	64.24$_{\pm 1.67}$	75.97$_{\pm 3.03}$	91.96$_{\pm 2.32}$	68.13$_{\pm 2.59}$	66.99$_{\pm 1.95}$	82.91$_{\pm 0.74}$
TURPLE	**60.56**$_{\pm 1.20}$	**69.19**$_{\pm 2.18}$	**72.42**$_{\pm 2.20}$	**77.22**$_{\pm 1.39}$	**96.93**$_{\pm 0.87}$	**80.91**$_{\pm 1.75}$	**69.28**$_{\pm 1.39}$	**84.26**$_{\pm 0.39}$

where $\mathcal{L}^u_{\text{Dev}}$ is the deviation-based separation loss computed with the unseen head $h_u(\cdot)$ (two layer MLP), following the form of Eq. 1 but with $\mathcal{N}'$ as negatives and $\mathcal{P} \cup \mathcal{A}'$ as positives, $\boldsymbol{\mu}^{(u)}, \boldsymbol{\nu}^{(u)}$ updated using only $\mathcal{N}'$, and λ_{proto} and λ_{adv} are the balancing coefficients, respectively. During joint training, $f_\theta(\cdot)$, $h_u(\cdot)$, and $\mathcal{A}_\phi(\cdot)$ are jointly optimized with $\mathcal{L}_{\text{joint}}$, while the seen head $h_s(\cdot)$ remains frozen throughout. The gate w_{adv} applies only to the augmenter-side loss $\mathcal{L}_{\text{adv}}$; the detector-side loss $\mathcal{L}^u_{\text{Dev}}$ treats all pseudo-unseen samples as unweighted positives. The detailed training procedure of TURPLE is provided in Algorithm 1.

4 Experiment

4.1 Experimental Settings

Datasets. We evaluate our method on eight publicly available benchmark datasets, including four single-class anomaly datasets (UCR (Wu & Keogh, 2021), ASD (Li et al., 2021), PSM (Abdulaal et al., 2021), SMD (Su et al., 2019)) and four multi-class anomaly datasets (CT, SAD, PTBXL (Wagner et al., 2020), TUSZ (Shah et al., 2018)). The multi-class datasets enable evaluation under both the general and hard settings, which sample labeled anomalies from all classes and a single class, respectively. We split each dataset chronologically into training (60%), validation (20%), and test (20%) sets. To evaluate robustness against label noise, we inject 2% anomalies sampled from the test set into the training data, with these contaminated samples excluded from final testing. Appendix E provides detailed dataset statistics.

Baselines. We compare against 15 baselines: seven unsupervised methods, including TCN-AE (Garg et al., 2021), THOC (Shen et al., 2020), TranAD (Tuli et al., 2022), DCdetector (Yang et al., 2023), GPT4TS (Zhou et al., 2023), COUTA (Xu et al., 2024), and DADA (Shentu et al., 2025); these methods rely solely on normal training data and use no labeled anomalies. We also compare with eight open-set methods, including DevNet (Pang et al., 2019), DSAD (Ruff et al., 2019), NCAD (Carmona et al., 2021), DRA (Ding et al., 2022), MOSAD (Lai et al., 2024), InvAD (Liu et al., 2025a), WSAD-DT (Durani et al., 2025), and IMPACT (Zhou et al., 2026); these methods additionally leverage a limited number of labeled anomalies. Under the general setting, we compare with both groups. Under the hard setting, we compare with the same open-set methods, using ten labeled samples per class following (Lai et al., 2024; Zhou et al., 2026).

Implementation and Evaluation Metrics. Following IMPACT (Zhou et al., 2026), we evaluate all methods using the area under the ROC curve (AUC) and report the mean $\pm$ standard deviation over five independent runs. All experiments are implemented in PyTorch and run on a single NVIDIA RTX A6000 GPU. The details of the implementation are in Appendix J.

4.2 Empirical Results

Performance Under the General Setting. We evaluate TURPLE under the general setting, where labeled anomalies are sampled from available anomaly classes. As shown in Table 1, we can observe

Table 2: Detection accuracy (AUC, %) under the open-set **hard** setting (one visible anomaly class at a time). Best in each row is in **boldface**, and second-best is underlined.

Dataset	seen anomaly	DevNet	DSAD	NCAD	DRA	MOSAD	InvAD	WSAD-DT	IMPACT	TURPLE
CT	Letter-G	49.54$_{\pm 1.03}$	73.56$_{\pm 3.41}$	63.27$_{\pm 0.65}$	68.79$_{\pm 3.43}$	75.64$_{\pm 1.50}$	75.43$_{\pm 0.52}$	68.83$_{\pm 1.77}$	80.90$_{\pm 1.03}$	**86.93**$_{\pm 3.46}$
	Letter-M	50.75$_{\pm 1.52}$	75.67$_{\pm 1.41}$	64.28$_{\pm 0.58}$	74.93$_{\pm 2.22}$	75.15$_{\pm 0.69}$	75.60$_{\pm 0.89}$	72.51$_{\pm 2.35}$	81.65$_{\pm 0.38}$	**85.45**$_{\pm 1.63}$
	Letter-Q	50.91$_{\pm 1.68}$	74.46$_{\pm 1.16}$	64.52$_{\pm 0.77}$	71.85$_{\pm 3.11}$	76.37$_{\pm 0.84}$	75.01$_{\pm 0.77}$	74.01$_{\pm 2.17}$	**82.50**$_{\pm 0.37}$	77.45$_{\pm 2.61}$
	Letter-W	51.40$_{\pm 3.83}$	74.23$_{\pm 3.22}$	63.71$_{\pm 0.39}$	73.02$_{\pm 1.57}$	75.75$_{\pm 0.48}$	75.74$_{\pm 0.62}$	76.20$_{\pm 1.07}$	80.87$_{\pm 1.05}$	**83.06**$_{\pm 1.81}$
	Letter-Z	49.54$_{\pm 1.05}$	79.46$_{\pm 2.10}$	64.22$_{\pm 0.35}$	75.11$_{\pm 1.38}$	78.70$_{\pm 0.81}$	75.33$_{\pm 0.67}$	74.94$_{\pm 1.36}$	**83.58**$_{\pm 0.97}$	82.85$_{\pm 1.24}$
	Mean	50.43$_{\pm 0.76}$	75.48$_{\pm 2.11}$	64.00$_{\pm 0.45}$	72.74$_{\pm 2.32}$	76.32$_{\pm 1.25}$	75.42$_{\pm 0.25}$	73.30$_{\pm 2.54}$	81.90$_{\pm 1.03}$	**83.15**$_{\pm 3.22}$
SAD	Digit-Zero	50.41$_{\pm 1.42}$	57.49$_{\pm 2.01}$	45.84$_{\pm 1.91}$	48.04$_{\pm 2.02}$	52.95$_{\pm 0.72}$	50.66$_{\pm 2.99}$	60.53$_{\pm 2.06}$	62.85$_{\pm 4.79}$	**68.19**$_{\pm 4.10}$
	Digit-Three	50.00$_{\pm 0.00}$	60.07$_{\pm 3.61}$	47.69$_{\pm 1.40}$	46.90$_{\pm 1.94}$	52.94$_{\pm 0.28}$	49.84$_{\pm 1.88}$	70.26$_{\pm 1.17}$	67.34$_{\pm 3.66}$	**72.39**$_{\pm 3.47}$
	Digit-Seven	50.00$_{\pm 0.00}$	60.90$_{\pm 2.01}$	47.06$_{\pm 1.58}$	41.87$_{\pm 1.92}$	52.68$_{\pm 0.96}$	49.93$_{\pm 2.39}$	63.44$_{\pm 1.09}$	70.71$_{\pm 2.02}$	**72.68**$_{\pm 2.86}$
	Digit-Eight	50.00$_{\pm 0.00}$	60.82$_{\pm 2.87}$	47.57$_{\pm 0.80}$	46.77$_{\pm 3.15}$	53.16$_{\pm 0.88}$	51.44$_{\pm 1.41}$	62.36$_{\pm 1.87}$	63.86$_{\pm 1.05}$	**69.30**$_{\pm 4.10}$
	Mean	50.10$_{\pm 0.18}$	59.82$_{\pm 1.38}$	47.04$_{\pm 0.73}$	45.90$_{\pm 2.38}$	52.93$_{\pm 0.17}$	50.47$_{\pm 0.65}$	64.15$_{\pm 3.68}$	66.19$_{\pm 3.10}$	**70.64**$_{\pm 1.90}$
PTBXL	MI	50.00$_{\pm 0.00}$	58.69$_{\pm 2.46}$	46.32$_{\pm 0.64}$	60.33$_{\pm 2.84}$	59.61$_{\pm 4.73}$	52.95$_{\pm 4.88}$	65.27$_{\pm 1.45}$	**65.80**$_{\pm 3.95}$	65.47$_{\pm 0.31}$
	STTC	50.00$_{\pm 0.00}$	50.60$_{\pm 4.09}$	46.51$_{\pm 1.29}$	51.68$_{\pm 2.15}$	53.36$_{\pm 4.81}$	54.00$_{\pm 5.18}$	56.42$_{\pm 2.61}$	59.35$_{\pm 3.89}$	**64.16**$_{\pm 2.06}$
	CD	50.00$_{\pm 0.00}$	54.20$_{\pm 3.53}$	46.25$_{\pm 0.91}$	52.36$_{\pm 2.87}$	52.72$_{\pm 3.63}$	51.44$_{\pm 7.09}$	64.51$_{\pm 0.82}$	61.73$_{\pm 5.17}$	**64.64**$_{\pm 0.75}$
	HYP	50.00$_{\pm 0.00}$	58.32$_{\pm 3.78}$	46.11$_{\pm 0.76}$	60.32$_{\pm 1.67}$	60.17$_{\pm 2.38}$	52.76$_{\pm 6.70}$	60.02$_{\pm 2.48}$	**63.02**$_{\pm 3.32}$	62.49$_{\pm 1.49}$
	Mean	50.00$_{\pm 0.00}$	55.45$_{\pm 3.31}$	46.30$_{\pm 0.14}$	56.17$_{\pm 4.16}$	56.47$_{\pm 3.44}$	52.79$_{\pm 0.91}$	61.56$_{\pm 3.58}$	62.48$_{\pm 2.33}$	**64.19**$_{\pm 1.15}$
TUSZ	FNSZ	49.62$_{\pm 0.67}$	75.00$_{\pm 0.77}$	69.94$_{\pm 2.28}$	63.61$_{\pm 2.01}$	69.87$_{\pm 2.75}$	57.15$_{\pm 8.60}$	64.79$_{\pm 2.00}$	76.27$_{\pm 1.65}$	**78.14**$_{\pm 2.13}$
	GNSZ	50.04$_{\pm 0.06}$	53.64$_{\pm 2.36}$	69.59$_{\pm 2.29}$	53.00$_{\pm 3.35}$	66.65$_{\pm 2.15}$	54.99$_{\pm 8.50}$	58.60$_{\pm 1.71}$	74.53$_{\pm 2.27}$	**75.82**$_{\pm 1.07}$
	CPSZ	50.04$_{\pm 0.06}$	72.54$_{\pm 1.22}$	69.54$_{\pm 2.30}$	56.62$_{\pm 3.82}$	69.71$_{\pm 2.49}$	56.04$_{\pm 8.98}$	56.65$_{\pm 1.72}$	76.30$_{\pm 3.17}$	**79.11**$_{\pm 2.88}$
	Mean	49.90$_{\pm 0.20}$	67.06$_{\pm 9.54}$	69.69$_{\pm 0.18}$	57.74$_{\pm 4.40}$	68.74$_{\pm 1.48}$	56.06$_{\pm 0.88}$	60.01$_{\pm 3.47}$	75.70$_{\pm 0.83}$	**77.69**$_{\pm 1.38}$

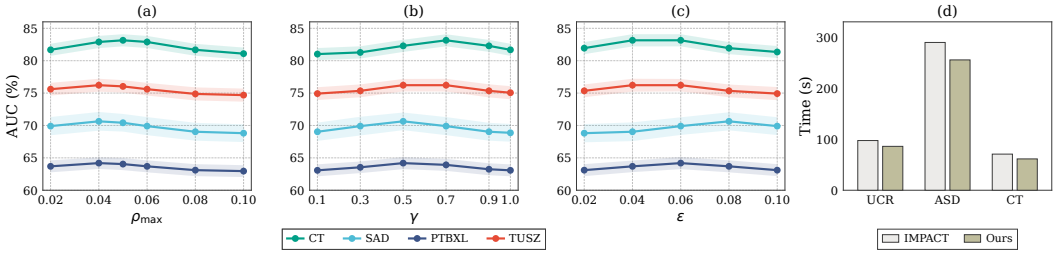

Figure 2: (a)–(c) Performance of TURPLE versus $\rho_{\max}$, γ, and ε under different numbers. (d) Wall-clock time comparison of TURPLE and IMPACT.

that: **Obs.❶** Compared with unsupervised methods, TURPLE maintains advantages across datasets with diverse time-series characteristics. Unsupervised methods rely solely on normality modeling or reconstruction, struggling to capture diverse real-world anomaly patterns. TURPLE addresses this via trajectory-guided label refinement, conservatively relabeling contaminated samples during warmup and converting misleading supervision into reliable anomaly knowledge. **Obs.❷** In the open-set setting, TURPLE outperforms methods that explicitly exploit labeled anomalies. Existing open-set approaches rely on heuristic augmentations to generate pseudo-anomalies, often failing to generalize to unseen anomalies. TURPLE instead employs prototype-repulsive extrapolation, synthesizing pseudo-unseen anomalies and expanding the decision boundary via a learnable frequency augmenter. **Obs.❸** Notably, the remarkable superiority of our method clearly demonstrates that trajectory-guided label refinement and prototype-repulsive extrapolation can robustly eliminate potential contamination and further exploit valuable known abnormal knowledge.

Performance Under the Hard Setting Table 2 shows the performance comparison under the hard setting. Our method demonstrates remarkable generalization capability, achieving the highest average AUC across the four multi-class datasets and outperforming the strongest competing open-set method. On CT and SAD, our method achieves 83.15% and 70.64%, respectively, surpassing the second-best methods by 1.25% and 4.45%. Notably, the superiority of our method demonstrates that trajectory-guided label refinement and prototype-repulsive extrapolation can effectively eliminate the impact of potential contamination and expand the utilization of known abnormal knowledge to generalize beyond the limited seen anomaly classes.

Ablation Study. Our framework integrates four key modules: trajectory-guided label refinement (TLR), adaptive sample-level gating (Gate), learnable frequency augmentation (Aug), and deviation-based scoring pre-training (Pre). To dissect each component's contribution and validate our design choices, we conduct ablations by removing each module, with results in Table 3. From the results, we can observe that: **Obs.❹** Removing TLR causes the most substantial performance drop on

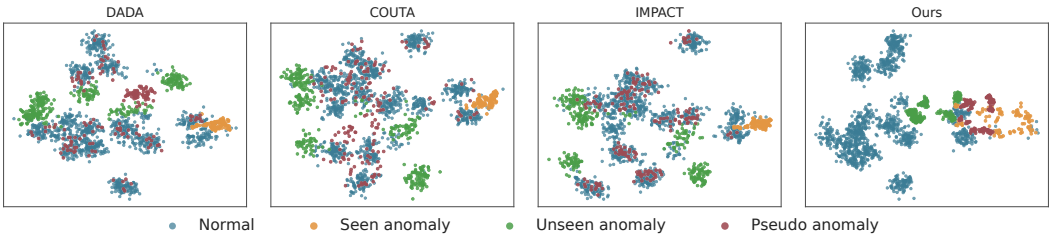

Figure 3: T-SNE visualization of encoder features on CT under the open-set hard setting

SAD, highlighting its essential role in purifying training supervision by identifying contaminated normals from Gaussian-perturbation-induced deviation trajectories. **Obs.❺** Aug is critical for CT, as its removal yields the largest degradation, validating the effectiveness of the learnable frequency augmenter in generating pseudo-unseen anomalies that expose fragile decision boundaries. **Obs.❻** Removing Pre-train consistently degrades both benchmarks, confirming that the deviation-based scoring pretraining provides a reliable initialization that stabilizes the subsequent trajectory recording and label refinement. Meanwhile, Gate consistently contributes by preventing noisy pseudo-unseen samples from dominating training.

Table 3: Ablation Study on two benchmarks.

Datasets	SAD		CT	
Metrics	General	Hard	General	Hard
TURPLE w/o TLR	78.15	67.34	94.87	81.42
TURPLE w/o Gate	79.28	70.49	94.21	80.73
TURPLE w/o Aug	79.83	69.08	93.56	79.86
TURPLE w/o Pre	78.99	68.37	95.18	80.51
TURPLE	**80.91**	**70.64**	**96.93**	**83.15**

Sensitivity Analysis. We further analyze sensitivity to three hyperparameters under the open-set hard setting on CT, SAD, PTBXL, and TUSZ, including the maximum label-correction ratio $\rho_{\max}$, the gating down-weight factor γ, and the Gaussian perturbation strength ε, as shown in Figure 2. We vary each parameter within a reasonable range and find that $\rho_{\max}$ yields a rise-then-saturation pattern, γ provides a stable trade-off after mid-to-high values, and ε requires a moderate perturbation strength to avoid degradation. Overall, TURPLE remains stable across broad ranges, and we set the dataset-wise defaults to $\rho_{\max} = 0.04$, $\gamma = 0.7$, and $\varepsilon = 0.06$.

Efficiency Analysis. To further empirically assess the practical overhead of TURPLE, we directly compare its wall-clock time with the strongest available open-set baseline, IMPACT (Zhou et al., 2026), under the standard general setting on UCR, ASD, and CT. As shown in Figure 2(d), TURPLE incurs only a relatively modest wall-clock cost and is consistently and clearly faster than IMPACT on all three datasets, with a stable advantage across scales. This overall indicates that the proposed framework maintains a favorable efficiency profile without any noticeable sacrifice in detection performance. Appendix H.4 provides a full complexity analysis.

Visualizations. To provide a more intuitive assessment of open-set synthesis, we visualize the encoder features with t-SNE on the CT dataset under the challenging hard setting in which Letter-G serves as the only seen anomaly class, with the corresponding results presented in Figure 3. DADA and COUTA generate heuristic time-domain pseudo-anomalies that are widely scattered and frequently overlap with normal samples. IMPACT shifts a subset of pseudo-anomalies toward the anomaly region, yet a considerable number of them still remain in close proximity to normal samples. In contrast, our prototype-anchored synthesis produces compact and well-separated pseudo-anomalies on the anomaly side, which lie close to unseen anomalies and are clearly separated from normal samples, thereby providing cleaner and more reliable open-set supervision.

5 Conclusion

This paper presents TURPLE, a novel and unified framework for open-set TSAD. By pretraining a deviation-aware detector with a seen head and tracking the temporal evolution of perturbation-induced deviation trajectories, TURPLE rectifies contaminated labels and synthesizes high-quality pseudo-unseen anomalies via prototype-repulsive extrapolation, enhancing generalization to unseen anomalies. To stabilize training, it incorporates robust adaptive sample-level gating on refined normal samples, seen anomalies, and pseudo-unseen anomalies. Extensive experiments consistently validate its effectiveness compared with state-of-the-art approaches.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2485–2494, 2021.

Archana Anandakrishnan, Senthil Kumar, Alexander Statnikov, Tanveer Faruquie, and Di Xu. Anomaly detection in finance: Editors' introduction. In *Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance*, volume 71, pp. 1–7, 2018.

Paul Boniol, Ashwin K Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S Tsay, Aaron Elmore, Michael J Franklin, and John Paparrizos. Vus: effective and efficient accuracy measures for time-series anomaly detection: P. boniol et al. *The VLDB Journal*, 34(3): 32, 2025.

Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.

Alberto Castellini, Francesco Masillo, Davide Azzalini, Francesco Amigoni, and Alessandro Farinelli. Adversarial data augmentation for hmm-based anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14131–14143, 2023.

Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection. *arXiv preprint arXiv:2307.00754*, 2023.

Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. Carla: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition*, 157:110874, 2025.

Choubo Ding, Guansong Pang, and Chunhua Shen. Catching both gray and black swans: Open-set supervised anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7378–7388. IEEE, 2022.

Walid Durani, Tobias Nitzl, Claudia Plant, and Christian Böhm. Weakly supervised anomaly detection via dual-tailed kernel. In *ICML*, 2025.

Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2508–2517, 2021.

Hui He, Hezhe Qiao, Yutong Chen, Kun Yi, and Guansong Pang. Timeradar: A domain-rotatable foundation model for time series anomaly detection. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1602–1613, 2026.

Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 635–645, 2022.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Geonuk Kim, Minhoi Kim, Kangil Lee, Minsu Kim, Hyeonseong Jeon, Jeonghoon Han, Hyoungjoon Lim, and Junho Yim. Unispector: Towards universal open-set defect recognition via spectral-contrastive visual prompting. *arXiv preprint arXiv:2604.02905*, 2026.

Thomas Lai, Thi Kieu Khanh Ho, and Narges Armanfard. Open-set multivariate time-series anomaly detection. In *ECAI 2024: 27th European Conference on Artificial Intelligence, 19–24 October 2024, Santiago de Compostela, Spain–Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, pp. 2003–2010. SAGE Publications Pvt. Ltd 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2024.

Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

Long Li, Wencheng Zhang, Shi Yuan, Hongle Guo, and Wanghu Chen. Physdiff: A physically-guided diffusion model for multivariate time series anomaly detection. *Advances in Neural Information Processing Systems*, 38:105720–105750, 2026.

Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 3220–3230, 2021.

Chen Liu, Shibo He, Shizhong Li, Zhenyu Shi, and Wenchao Meng. Detecting both seen and unseen anomalies in time series. *ACM Transactions on Knowledge Discovery from Data*, 19(4): 1–29, 2025a.

Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. Time-series anomaly detection: Overview and new trends. *Proceedings of the VLDB Endowment (PVLDB)*, 17(12):4229–4232, 2024.

Qinghua Liu, Seunghak Lee, and John Paparrizos. Tsb-autoad: Towards automated solutions for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 18(11):4364–4379, 2025b.

Jingzheng Mao, Runjie Pu, Zhen Song, Yanbin Sun, Mohan Li, and Zhihong Tian. Ruletwin: Physics-constrained rule induction for anomaly detection in industrial multivariate time serie. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3679–3690, 2026.

A Ian McLeod. Kendall rank correlation and mann-kendall trend test. *R package Kendall*, 602:1–10, 2005.

Xudong Mou, Rui Wang, Bo Li, Tianyu Wo, Jie Sun, Hui Wang, and Xudong Liu. Roca: Robust contrastive one-class time series anomaly detection with contaminated data. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Weiran Pan, Wei Wei, and Wenfeng Xie. Debiased sample selection for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32047–32057, 2026.

Guansong Pang, Chunhua Shen, and Anton Van Den Hengel. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 353–362, 2019.

Chen Qiu, Aodong Li, Marius Kloft, Maja Rudolph, and Stephan Mandt. Latent outlier exposure for anomaly detection with contaminated data. In *International conference on machine learning*, pp. 18153–18167. PMLR, 2022.

Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019.

Vinit Shah, Eva Von Weltin, Silvia Lopez, James Riley McHugh, Lillian Veloso, Meysam Golmohammadi, Iyad Obeid, and Joseph Picone. The temple university hospital seizure detection corpus. *Frontiers in neuroinformatics*, 12:83, 2018.

Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in neural information processing systems*, 33:13016–13026, 2020.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *International Conference on Learning Representations*, volume 2025, pp. 81358–81381, 2025.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pp. 21076–21089. PMLR, 2022.

Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*, 2022.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Fuyun Wang, Tong Zhang, Yuanzhi Wang, Yide Qiu, Xin Liu, Xu Guo, and Zhen Cui. Distribution prototype diffusion learning for open-set supervised anomaly detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20416–20426. IEEE, 2025a.

Fuyun Wang, Yuanzhi Wang, Xu Guo, Sujia Huang, Tong Zhang, Dan Wang, Hui Yan, Xin Liu, and Zhen Cui. Mixture prototype flow matching for open-set supervised anomaly detection. *arXiv preprint arXiv:2605.02438*, 2026a.

Qizhou Wang, Guansong Pang, Mahsa Salehi, Xiaokun Xia, and Christopher Leckie. Open-set graph anomaly detection via normal structure regularisation. In *International conference on learning representations*, volume 2025, pp. 17684–17707, 2025b.

Rui Wang, Xudong Mou, Renyu Yang, Kai Gao, Pin Liu, Chongwei Liu, Tianyu Wo, and Xudong Liu. Cutaddpaste: Time series anomaly detection by exploiting abnormal knowledge. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3176–3187, 2024.

Xin Wang, Yunshi Wen, Yanan He, Haotian Xu, Youlan Zhao, Michel Ferreira Cardia Haddad, and Tengfei Ma. Caad: Causality-aware multivariate time series anomaly detection via multi-scale alignment and structural causal consistency. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5091–5101, 2026b.

Renjie Wu and Eamonn J Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering*, 35 (3):2421–2429, 2021.

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):5723–5736, 2024.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3033–3045, 2023.

Caiyang Yu, Wei Ju, Haixin Wang, Yifan Wang, and Ziyue Qiao. Dynamic multi-sample mixup with gradient exploration for open-set graph anomaly detection. In *International Conference on Learning Representations*, volume 2026, pp. 105792–105813, 2026.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Xiaohui Zhou, Yijie Wang, Hongzuo Xu, Weixuan Liang, Xiaoli Li, and Guansong Pang. Impact: Influence modeling for open-set time series anomaly detection. *arXiv preprint arXiv:2603.29183*, 2026.

Jiawen Zhu, Choubo Ding, Yu Tian, and Guansong Pang. Anomaly heterogeneity learning for open-set supervised anomaly detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17616–17626. IEEE, 2024.

A Statement on the use of large language models

We acknowledge the use of a large language model (LLM) solely for language polishing and grammatical correction during the preparation of this manuscript. All content, including the research ideas, methodology, experiments, and conclusions, was developed and verified by the authors. The LLM was not employed for any scientific or analytical task, and the authors take full responsibility for the accuracy and integrity of the presented work.

B Proofs of Theoretical Results

B.1 Proof of Proposition 3.1

Proof. We first establish the one-step evolution of the population response gap. By adding and subtracting $G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_t)$, we decompose

$$G_{t+1} - G_t = \Big[G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_t) - G(\boldsymbol{\omega}_t, \mathbf{r}_t)\Big] + \Big[G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_{t+1}) - G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_t)\Big]. \tag{20}$$

The first term captures the effect of the parameter update with the reference statistics fixed. Since $\delta_{\boldsymbol{\omega},\mathbf{r}_t}(\mathbf{x})$ is L_δ-smooth in $\boldsymbol{\omega}$, the update $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \eta \mathbf{g}_t$ gives

$$\delta_{\boldsymbol{\omega}_{t+1},\mathbf{r}_t}(\mathbf{x}) \geq \delta_{\boldsymbol{\omega}_t,\mathbf{r}_t}(\mathbf{x}) - \eta \left\langle \nabla_{\boldsymbol{\omega}} \delta_{\boldsymbol{\omega}_t,\mathbf{r}_t}(\mathbf{x}), \mathbf{g}_t \right\rangle - \frac{L_\delta \eta^2}{2} \|\mathbf{g}_t\|_2^2, \tag{21}$$

and likewise

$$\delta_{\boldsymbol{\omega}_{t+1},\mathbf{r}_t}(\mathbf{x}) \leq \delta_{\boldsymbol{\omega}_t,\mathbf{r}_t}(\mathbf{x}) - \eta \left\langle \nabla_{\boldsymbol{\omega}} \delta_{\boldsymbol{\omega}_t,\mathbf{r}_t}(\mathbf{x}), \mathbf{g}_t \right\rangle + \frac{L_\delta \eta^2}{2} \|\mathbf{g}_t\|_2^2. \tag{22}$$

Taking the lower bound under P_C, the upper bound under P_N, and subtracting yields

$$G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_t) - G(\boldsymbol{\omega}_t, \mathbf{r}_t) \geq \eta \Gamma_t - L_\delta \eta^2 \|\mathbf{g}_t\|_2^2. \tag{23}$$

By the definition of B_t, $G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_{t+1}) - G(\boldsymbol{\omega}_{t+1}, \mathbf{r}_t) \geq -B_t$. Combining the two terms gives

$$G_{t+1} - G_t \geq \eta \Gamma_t - L_\delta \eta^2 \|\mathbf{g}_t\|_2^2 - B_t. \tag{24}$$

Hence, if $\Gamma_t > L_\delta \eta \|\mathbf{g}_t\|_2^2 + B_t/\eta$, then $G_{t+1} > G_t$. This proves the training-amplification part of the proposition.

We next establish the trajectory-level error bound. For a fixed sample i, let $\mathcal{F}_t$ denote the filtration generated by its response trajectory up to epoch t. By the martingale-difference and conditional σ^2-sub-Gaussian assumption in Proposition 3.1, for every $\lambda \in \mathbb{R}$,

$$\mathbb{E}\Big[\exp\Big(\lambda \xi_i^{(t)}\Big) \mid \mathcal{F}_{t-1}\Big] \leq \exp\left(\frac{\lambda^2 \sigma^2}{2}\right).$$

Iterating this conditional moment-generating-function bound over $t = 1, \ldots, T_{\text{ep}}$ shows that the averaged fluctuation $\bar{\xi}_i := T_{\text{ep}}^{-1} \sum_{t=1}^{T_{\text{ep}}} \xi_i^{(t)}$ is σ^2/T_{ep}-sub-Gaussian.

Define the trajectory-level population means as $\bar{\mu}_N := T_{\text{ep}}^{-1} \sum_t \mu_{N,t}$ and $\bar{\mu}_C := T_{\text{ep}}^{-1} \sum_t \mu_{C,t}$. By the definition of G_t, their difference is

$$\bar{\mu}_C - \bar{\mu}_N = \frac{1}{T_{\text{ep}}} \sum_{t=1}^{T_{\text{ep}}} G_t = \bar{G}.$$

Let $\tau := (\bar{\mu}_N + \bar{\mu}_C)/2$. For a genuine-normal sample, a classification error occurs when $\bar{\delta}_i > \tau$, which is equivalent to $\bar{\xi}_i > \bar{G}/2$. The sub-Gaussian upper-tail bound therefore gives

$$\Pr\big(\bar{\delta}_i > \tau \mid Y_i = N\big) \leq \exp\left(-\frac{T_{\text{ep}} \bar{G}^2}{8\sigma^2}\right). \tag{25}$$

Similarly, for a contaminated sample, an error occurs when $\bar{\delta}_i \leq \tau$, or equivalently $\bar{\xi}_i \leq -\bar{G}/2$, and the lower-tail bound gives

$$\Pr\big(\bar{\delta}_i \leq \tau \mid Y_i = C\big) \leq \exp\left(-\frac{T_{\text{ep}} \bar{G}^2}{8\sigma^2}\right). \tag{26}$$

Taking the maximum of the two class-conditional errors proves

$$P_{\text{err}} \leq \exp\left(-\frac{T_{\text{ep}} \bar{G}^2}{8\sigma^2}\right), \tag{27}$$

which completes the proof. □

B.2 Proof of Proposition 3.2

Proof. Recall $\mathbf{m} = (\mathbf{c}_n + \mathbf{c}_a)/2$, $\mathbf{d} = (\mathbf{c}_a - \mathbf{c}_n)/2$, and $\mathbf{y} = \mathbf{z} - \mathbf{m}$. Hence $\mathbf{c}_n = \mathbf{m} - \mathbf{d}$, $\mathbf{c}_a = \mathbf{m} + \mathbf{d}$, and the radial update $\mathbf{z}^+ = \mathbf{z} + s\mathbf{y}$ can be written as $\mathbf{z}^+ = \mathbf{m} + (1+s)\mathbf{y}$.

For the normal prototype, we have $d_n(\mathbf{z}) = \|\mathbf{y} + \mathbf{d}\|_2^2$ and $d_n(\mathbf{z}^+) = \|(1+s)\mathbf{y} + \mathbf{d}\|_2^2$. Therefore,

$$d_n(\mathbf{z}^+) - d_n(\mathbf{z}) = 2s\left(\|\mathbf{y}\|_2^2 + \mathbf{d}^\top\mathbf{y}\right) + s^2\|\mathbf{y}\|_2^2. \tag{28}$$

Similarly, since $d_a(\mathbf{z}) = \|\mathbf{y} - \mathbf{d}\|_2^2$,

$$d_a(\mathbf{z}^+) - d_a(\mathbf{z}) = 2s\left(\|\mathbf{y}\|_2^2 - \mathbf{d}^\top\mathbf{y}\right) + s^2\|\mathbf{y}\|_2^2. \tag{29}$$

If $\|\mathbf{y}\|_2^2 > |\mathbf{d}^\top\mathbf{y}|$, then both $\|\mathbf{y}\|_2^2 + \mathbf{d}^\top\mathbf{y}$ and $\|\mathbf{y}\|_2^2 - \mathbf{d}^\top\mathbf{y}$ are strictly positive. Since $s > 0$, both distance differences are strictly positive, proving $d_n(\mathbf{z}^+) > d_n(\mathbf{z})$ and $d_a(\mathbf{z}^+) > d_a(\mathbf{z})$.

The simpler sufficient condition follows from Cauchy–Schwarz:

$$|\mathbf{d}^\top\mathbf{y}| \le \|\mathbf{d}\|_2\|\mathbf{y}\|_2.$$

Thus, $\|\mathbf{y}\|_2 > \|\mathbf{d}\|_2$ implies $\|\mathbf{y}\|_2^2 > |\mathbf{d}^\top\mathbf{y}|$. Since $\|\mathbf{d}\|_2 = \frac{1}{2}\|\mathbf{c}_a - \mathbf{c}_n\|_2$, this gives the stated sufficient condition.

We finally consider the input-space perturbation. Let $\mathbf{z} = f_\theta(\mathbf{x})$, let $J_\theta(\mathbf{x})$ denote the encoder Jacobian, and write $\boldsymbol{\eta} = \epsilon\hat{\boldsymbol{\eta}}$ for a fixed direction $\hat{\boldsymbol{\eta}}$. Differentiability of f_θ gives $f_\theta(\mathbf{x} + \epsilon\hat{\boldsymbol{\eta}}) = \mathbf{z} + \epsilon J_\theta(\mathbf{x})\hat{\boldsymbol{\eta}} + o(\epsilon)$. Therefore, $d_n(f_\theta(\mathbf{x}+\epsilon\hat{\boldsymbol{\eta}})) - d_n(\mathbf{z}) = 2\epsilon\langle \mathbf{z} - \mathbf{c}_n, J_\theta(\mathbf{x})\hat{\boldsymbol{\eta}}\rangle + o(\epsilon)$, and likewise $d_a(f_\theta(\mathbf{x}+\epsilon\hat{\boldsymbol{\eta}})) - d_a(\mathbf{z}) = 2\epsilon\langle \mathbf{z} - \mathbf{c}_a, J_\theta(\mathbf{x})\hat{\boldsymbol{\eta}}\rangle + o(\epsilon)$. If both directional inner products are strictly positive, their first-order coefficients are positive. Hence, there exists $\epsilon_0 > 0$ such that both prototype distances strictly increase for every $0 < \epsilon < \epsilon_0$, proving the local input-space result. □

B.3 Proof of Proposition 3.3

Proof. Let $\mathbf{r}$ collect all gradient components that are unchanged by the gate, including the clean gradient and the contributions of unselected samples. Before gating, the joint gradient direction can therefore be written as $\mathbf{u} = \mathbf{r} + \mathbf{h}_\mathcal{G}$, whereas after multiplying the selected contribution by $0 \le \gamma < 1$, it becomes $\mathbf{u}_\gamma = \mathbf{r} + \gamma\mathbf{h}_\mathcal{G}$.

Their difference is $\mathbf{u}_\gamma - \mathbf{u} = -(1-\gamma)\mathbf{h}_\mathcal{G}$. Taking the inner product with the clean gradient $\mathbf{g}$ gives

$$\langle \mathbf{g}, \mathbf{u}_\gamma\rangle - \langle \mathbf{g}, \mathbf{u}\rangle = -(1-\gamma)\langle \mathbf{g}, \mathbf{h}_\mathcal{G}\rangle = (1-\gamma)C_\mathcal{G}. \tag{30}$$

If $C_\mathcal{G} > 0$, then $1 - \gamma > 0$ implies

$$\langle \mathbf{g}, \mathbf{u}_\gamma\rangle > \langle \mathbf{g}, \mathbf{u}\rangle.$$

Thus, the gated gradient is more strongly aligned with the clean gradient. Equivalently, the corresponding descent step is more strongly aligned with the clean descent direction, proving the proposition. □

C Adaptive Gating: Detailed Formulation

C.1 Motivation

The adaptive sample-level gating mechanism is designed to prevent harmful pseudo-unseen samples from dominating the training signal. The generator may occasionally produce samples that are either unrealistic or that conflict with the clean normal/anomaly decision boundary. Such samples can introduce noisy gradients and degrade the detector. The gate assigns a lower weight to these samples based on their short-term training dynamics.

For each generated pseudo-unseen sample $\hat{\boldsymbol{x}} \in \mathcal{P}$, we track three quantities over a sliding window of the most recent H joint-training epochs:

- a confidence score $c_t(\hat{\boldsymbol{x}})$,
- a clean–augmented loss discrepancy $\Delta_t(\hat{\boldsymbol{x}})$,

- a relative augmented-loss shift $r_t(\hat{\boldsymbol{x}})$.

From these, we compute three binary indicators I_1, I_2, I_3 and the harmful-noise flag

$$F_{\text{noise}}(\hat{\boldsymbol{x}}) = \big(\neg I_1(\hat{\boldsymbol{x}})\big) \wedge I_2(\hat{\boldsymbol{x}}) \wedge I_3(\hat{\boldsymbol{x}}). \tag{31}$$

If $F_{\text{noise}}(\hat{\boldsymbol{x}}) = 1$, the sample is considered harmful and its loss is down-weighted by a factor $\gamma \in [0, 1)$; otherwise, it receives weight 1.

C.2 Definition of the Confidence Score and Indicator I_1

The confidence score measures how strongly the current model regards the generated sample as an anomaly. We use the joint prototype distance

$$c_t(\hat{\boldsymbol{x}}) = Q\big(f_\theta(\hat{\boldsymbol{x}})\big) = d_n\big(f_\theta(\hat{\boldsymbol{x}})\big) + d_a\big(f_\theta(\hat{\boldsymbol{x}})\big), \tag{32}$$

where $d_n(\boldsymbol{z}) = \|\boldsymbol{z} - \boldsymbol{c}_n\|_2^2$ and $d_a(\boldsymbol{z}) = \|\boldsymbol{z} - \boldsymbol{c}_a\|_2^2$. Since the objective minimizes $\ell^{\text{adv}}(\hat{\boldsymbol{x}}) = -Q(f_\theta(\hat{\boldsymbol{x}}))$, a successful pseudo-unseen sample should see its confidence c_t increase over training. Let

$$\boldsymbol{c}_{1:H}(\hat{\boldsymbol{x}}) = [c_{t-H+1}(\hat{\boldsymbol{x}}), \ldots, c_t(\hat{\boldsymbol{x}})]^\top$$

be the confidence history over the last H epochs. We apply the Mann–Kendall trend test to this sequence and obtain the statistic $Z_c(\hat{\boldsymbol{x}})$. Then

$$I_1(\hat{\boldsymbol{x}}) = \begin{cases} 1, & Z_c(\hat{\boldsymbol{x}}) > \tau_{\text{MK}}, \\ 0, & \text{otherwise}, \end{cases} \tag{33}$$

where τ_{MK} is the standard significance threshold of the Mann–Kendall test (e.g., the critical value for $p < 0.05$). Thus, $I_1 = 1$ indicates a significant upward confidence trend. The harmful-noise condition uses $\neg I_1$, i.e., a sample whose confidence is not increasing is suspicious.

C.3 Definition of the Loss Discrepancy and Indicator I_2

For each generated sample $\hat{\boldsymbol{x}}$, let $\boldsymbol{x}_{\text{seed}} \in \mathcal{A}'$ be its corresponding anomaly seed. We define the clean loss on the seed as

$$\ell_t^{\text{clean}}(\boldsymbol{x}_{\text{seed}}) = -Q\big(f_\theta(\boldsymbol{x}_{\text{seed}})\big), \tag{34}$$

and the augmented loss on the generated sample as

$$\ell_t^{\text{adv}}(\hat{\boldsymbol{x}}) = -Q\big(f_\theta(\hat{\boldsymbol{x}})\big). \tag{35}$$

The discrepancy at epoch t is

$$\Delta_t(\hat{\boldsymbol{x}}) = \ell_t^{\text{adv}}(\hat{\boldsymbol{x}}) - \ell_t^{\text{clean}}(\boldsymbol{x}_{\text{seed}}). \tag{36}$$

We track the history

$$\boldsymbol{\Delta}_{1:H}(\hat{\boldsymbol{x}}) = [\Delta_{t-H+1}(\hat{\boldsymbol{x}}), \ldots, \Delta_t(\hat{\boldsymbol{x}})]^\top.$$

We compute its Mann–Kendall trend statistic $Z_\Delta(\hat{\boldsymbol{x}})$ and define

$$I_2(\hat{\boldsymbol{x}}) = \begin{cases} 1, & Z_\Delta(\hat{\boldsymbol{x}}) > \tau_{\text{MK}} \text{ and } |\Delta_t(\hat{\boldsymbol{x}})| > \tau_\Delta, \\ 0, & \text{otherwise}, \end{cases} \tag{37}$$

where τ_Δ is a small positive threshold (e.g., $\tau_\Delta = 0.1$). A growing discrepancy indicates that the generated sample increasingly deviates from its seed in terms of prototype geometry, which may signal unrealistic or harmful generation.

C.4 Definition of the Relative Shift and Indicator I_3

The third indicator detects large abrupt changes in the augmented loss. We define the relative shift at epoch t as

$$r_t(\hat{\boldsymbol{x}}) = \frac{|\ell_t^{\text{adv}}(\hat{\boldsymbol{x}}) - \ell_{t-1}^{\text{adv}}(\hat{\boldsymbol{x}})|}{|\ell_{t-1}^{\text{adv}}(\hat{\boldsymbol{x}})| + \epsilon_0}, \tag{38}$$

where $\epsilon_0 > 0$ is a small numerical stabilizer (e.g., $\epsilon_0 = 10^{-6}$). Then

$$I_3(\hat{\boldsymbol{x}}) = \begin{cases} 1, & r_t(\hat{\boldsymbol{x}}) > \tau_r, \\ 0, & \text{otherwise}, \end{cases} \tag{39}$$

with a default threshold $\tau_r = 0.5$. A large relative shift suggests that the loss is unstable for this sample.

C.5 Gating Weight and Training Procedure

For each generated sample $\hat{\boldsymbol{x}}$, the gating weight is

$$w_{\text{adv}}(\hat{\boldsymbol{x}}) = \begin{cases} \gamma, & F_{\text{noise}}(\hat{\boldsymbol{x}}) = 1, \\ 1, & \text{otherwise}, \end{cases} \qquad 0 \le \gamma < 1. \tag{40}$$

The batch-level loss is then

$$\mathcal{L}_{\text{adv}} = \frac{1}{|\mathcal{P}_{\text{b}}|} \sum_{\hat{\boldsymbol{x}} \in \mathcal{P}_{\text{b}}} w_{\text{adv}}(\hat{\boldsymbol{x}})\, \ell^{\text{adv}}(\hat{\boldsymbol{x}}) + \lambda_{\text{reg}} \Big(\|\tanh(\boldsymbol{\delta}_a)\|_2^2 + \|\tanh(\boldsymbol{\delta}_p)\|_2^2 \Big), \tag{41}$$

where $\mathcal{P}_{\text{b}} \subseteq \mathcal{P}$ is the current mini-batch of generated samples.

History association with dynamically generated samples. Because the augmenter $\mathcal{A}_\phi$ changes during training, the generated sample $\hat{\boldsymbol{x}}_t = \mathcal{A}_{\phi_t}(\boldsymbol{x}_{\text{seed}})$ for a fixed seed $\boldsymbol{x}_{\text{seed}}$ varies across epochs. We therefore maintain the histories $\boldsymbol{c}_{1:H}$, $\boldsymbol{\Delta}_{1:H}$, and $\ell^{\text{adv}}_{1:H}$ *per seed* $\boldsymbol{x}_{\text{seed}}$. At each epoch, the newly generated sample from that seed appends its current statistics to the corresponding history buffer. If a seed is not sampled in a mini-batch, its history is not updated for that step but is retained. This ensures that the indicators reflect the temporal evolution of the generation process for each anomaly seed.

Hyperparameters and defaults. We use a history length of $H = 5$ epochs. The Mann–Kendall significance threshold is set to the critical value for $p = 0.05$. The remaining thresholds are $\tau_\Delta = 0.1$, $\tau_r = 0.5$, and $\epsilon_0 = 10^{-6}$. The down-weighting factor is $\gamma = 0.7$ by default. These values are used across all datasets unless otherwise stated; sensitivity to γ is analyzed in Figure 2.

C.6 Connection to Proposition 3.3

Proposition 3.3 states that down-weighting a set of gradients $\boldsymbol{h}_{\mathcal{G}}$ improves alignment with the clean descent direction if the aggregate conflict is positive.

$$C_{\mathcal{G}} = -\langle \boldsymbol{g}, \boldsymbol{h}_{\mathcal{G}} \rangle$$

is positive, where

$$\boldsymbol{g} = \nabla_{\boldsymbol{\vartheta}} \mathcal{L}_{\text{clean}}, \qquad \boldsymbol{h}(\hat{\boldsymbol{x}}) = \frac{\lambda_{\text{adv}}}{|\mathcal{P}_{\text{b}}|} \nabla_{\boldsymbol{\vartheta}} \ell^{\text{adv}}(\hat{\boldsymbol{x}}), \qquad \boldsymbol{h}_{\mathcal{G}} = \sum_{\hat{\boldsymbol{x}} \in \mathcal{G}} \boldsymbol{h}(\hat{\boldsymbol{x}}).$$

Here $\mathcal{G} = \{\hat{\boldsymbol{x}} \in \mathcal{P}_{\text{b}} : F_{\text{noise}}(\hat{\boldsymbol{x}}) = 1\}$. The visualization in Figure 4 provides qualitative evidence that the gate identifies a small, shrinking set of harmful samples over training, which is consistent with the premise of the proposition.

C.7 Visualization of Adaptive Sample-Level Gating

Figure 4 visualizes the sample-level gate on TUSZ hard (seen = FNSZ). Panel (a) shows the three binary criteria for observed-normal windows (sorted with flagged noise first): I_1 (rising confidence), I_2 (expanding DTW gap between clean and augmented losses), and I_3 (augmented-loss shift). A window is assigned to the harmful-noise set $\mathcal{F}_{\text{noise}}$ only when $\neg I_1 \wedge I_2 \wedge I_3$, which yields $|\mathcal{F}_{\text{noise}}| = 39$ on this run. Panel (b) tracks $|\mathcal{F}_{\text{noise}}|$ over joint-training epochs: the set peaks early (581) and then shrinks to 39, indicating that the gate gradually stops treating hard-but-clean samples as harmful noise.

D Inference Score Fusion

At test time, the final anomaly score for a window x_* is computed as

$$A(x_*) = \beta \big[\omega \bar{s}^{(s)} + (1-\omega) \bar{s}^{(u)} \big] + (1-\beta) \max\big(\bar{s}^{(s)}, \bar{s}^{(u)}\big) + w_{\text{proto}}\, d_n\big(f_\theta(x_*)\big), \tag{42}$$

where $\bar{s}^{(s)} = \text{Dev}\big(h_s(f_\theta(x_*))\big)$ and $\bar{s}^{(u)} = \text{Dev}\big(h_u(f_\theta(x_*))\big)$ denote the anomaly scores predicted by the seen and unseen heads, respectively; ω controls the relative importance of the seen head in the convex combination; β balances the weighted average of the two heads against their maximum; and w_{proto} scales the penalty term based on the Euclidean distance to the normal prototype. The final score integrates the two head-specific signals and the geometric deviation.

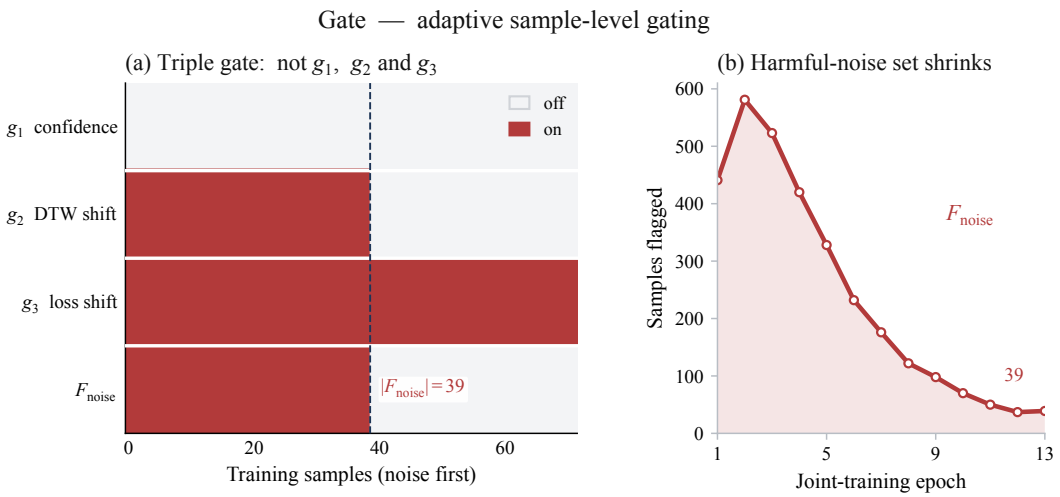

Figure 4: Visualization of adaptive sample-level gating on TUSZ hard (seen = FNSZ). (a) Triple gate and the resulting $\mathcal{F}_{\text{noise}}$ (samples sorted with flagged noise first). A sample is marked as harmful noise iff $\neg I_1 \wedge I_2 \wedge I_3$. (b) Size of $\mathcal{F}_{\text{noise}}$ over joint-training epochs, shrinking from a peak of 581 to 39.

E Algorithms Details

The complete training algorithm of our proposed **TURPLE** framework is outlined in Algorithm 1. The training process consists of three successive stages: deviation-based scoring pre-training, trajectory-guided label refinement, and prototype-repulsive joint training with adaptive gating. The first stage learns a stable deviation scorer; the second identifies and relabels contaminated normal samples; the third generates pseudo-unseen anomalies and jointly optimizes the detector.

F Datasets Details

We evaluate **TURPLE** on eight publicly available time-series anomaly detection benchmarks that vary in sampling rate, dimensionality, and the number of anomaly classes. These datasets are widely used in recent open-set TSAD studies (Xu et al., 2024; Lai et al., 2024; Liu et al., 2025a; Shentu et al., 2025; Zhou et al., 2026). Four of them contain a single anomaly class (UCR, ASD, PSM, SMD), while the remaining four contain multiple distinct anomaly classes (CT, SAD, PTBXL, TUSZ) and therefore support both the *general* and *hard* open-set protocols. Following IMPACT (Zhou et al., 2026), we treat each time interval as one sample, inject 2% anomalies sampled from the original test set into the training data (these contaminated samples are excluded from the final test set), and provide ten labeled anomalies per visible class. Table 4 summarizes the resulting statistics.

Additional descriptions of each dataset are provided below.

- **UCR** (Wu & Keogh, 2021) contains 250 heterogeneous univariate sub-datasets collected from diverse natural sources. Each series is anomaly-free during training and contains exactly one anomaly in testing. We concatenate all training series and select a subset of test anomalies for contamination injection and few-shot exploitation, using non-overlapping windows of length 100.
- **ASD** (Li et al., 2021) originates from production environments of a major Internet service provider. It comprises 19 complementary system metrics, including CPU utilization, memory allocation, network throughput, and virtual-machine indicators. Test anomalies are labeled by domain experts from incident reports.
- **PSM** (Abdulaal et al., 2021) is collected from eBay's distributed application infrastructure and records operational telemetry from multiple production server nodes. The series covers a comprehensive suite of server performance indicators such as CPU and memory consumption, with 25 variables after windowing.

Algorithm 1: Procedure of `TURPLE`

Input: Initial model f_θ with parameters $\theta = \{\theta_f, \theta_s, \theta_u\}$; training set $\mathcal{S}_{\text{tr}} = \mathcal{S}_{\text{n}} \cup \mathcal{S}_{\text{a}}$; fixed Gaussian perturbation $\mathcal{T}$; flip ratio $\rho_{\max}$; thresholds $\tau_\delta^{\text{emp}}, \tau_{\text{MK}}$

Output: Trained model f_{θ^*} and anomaly score $A(\cdot)$

Part I: Deviation-Based Scoring Pre-training

Pre-train f_θ and the seen head h_s on $\mathcal{S}_{\text{tr}}$ with the deviation objective $\mathcal{L}_{\text{dev}}^{(s)}$ (Eq. 1) for T_{ep} epochs;
Initialize deviation trajectories $\boldsymbol{\delta}_i \leftarrow \emptyset$, $i \in \mathcal{S}_{\text{n}}$;
for $t = 1, \ldots, T_{\text{ep}}$ **do**
  for $x_i \in \mathcal{S}_{\text{n}}$ **do**
    Generate the perturbed sample $\tilde{x}_i \leftarrow \mathcal{T}(x_i)$ by adding a fixed Gaussian noise;
    Compute the deviation response $\delta_i^{(t)}$ (Eq. 3);
    $\boldsymbol{\delta}_i \leftarrow \boldsymbol{\delta}_i \cup \{\delta_i^{(t)}\}$;
  Update f_θ and h_s with $\mathcal{L}_{\text{dev}}^{(s)}$ (Eq. 1);
Fix h_s as a stable deviation scorer;

Part II: Trajectory-Guided Label Refinement

Initialize the flip candidate set $\mathcal{F} \leftarrow \emptyset$;
for $x_i \in \mathcal{S}_{\text{n}}$ **do**
  Compute $\bar{\delta}_i$, Z_i, and Δ_i from $\boldsymbol{\delta}_i$;
  if $\bar{\delta}_i > \tau_\delta^{\text{emp}}$, $Z_i > \tau_{\text{MK}}$, *and* $\Delta_i > 0$ **then**
    Add x_i to $\mathcal{F}$;
Rank $\mathcal{F}$ by the combined trend score and retain at most $\rho_{\max}|\mathcal{S}_{\text{n}}|$ samples;
Refine the training sets: $\mathcal{N}' \leftarrow \mathcal{S}_{\text{n}} \setminus \mathcal{F}$, $\mathcal{A}' \leftarrow \mathcal{S}_{\text{a}} \cup \mathcal{F}$;
Initialize normal prototype $\mathbf{c}_n$ and anomaly prototype $\mathbf{c}_a$ from $\mathcal{N}'$ and $\mathcal{A}'$;

Part III: Prototype-Repulsive Joint Training

for $iter = 1, \ldots, N_{\max}$ **do**
  Sample a mini-batch $\mathcal{B} \sim \mathcal{N}' \cup \mathcal{A}'$;
  Update $\mathbf{c}_n$ and $\mathbf{c}_a$ using momentum;
  for $x_i \in \mathcal{B} \cap \mathcal{A}'$ **do**
    Apply FFT and let the learnable frequency augmenter $\mathcal{A}_\phi$ perturb the K lowest frequencies;
    Obtain the pseudo-unseen anomaly $\hat{x}_i$ via inverse FFT;
  for *each generated* $\hat{x}_i$ **do**
    Compute the confidence, prototype loss, and augmentation-response histories;
    Determine $F_{\text{noise}}(\hat{x}_i)$ (Eq. 16) and the gating weight $w_{\text{adv}}(\hat{x}_i)$;
  Update f_θ, $\mathcal{A}_\phi$, and h_u with (Eq. 19);
return f_{θ^*} and $A(\cdot)$;

Table 4: Statistics of eight datasets. #lens and #dims denote the length of a time interval and the number of variables. #classes is the number of anomaly classes. #train / #test are the numbers of training and test samples, and #anom is the number of anomalies among all test samples.

Dataset	#lens	#dims	#classes	#train	#test	#anom
UCR	100	1	1	54,106	70,512	172
ASD	100	19	1	1,043	468	78
PSM	100	25	1	1,351	798	272
SMD	100	38	1	7,228	6,084	528
CT	205	3	5	929	1,675	412
SAD	93	13	4	4,040	2,200	880
PTBXL	1000	12	4	8,323	2,198	1,286
TUSZ	1000	19	3	39,120	4,564	304

- **SMD** (Su et al., 2019) is a five-week collection of server telemetry from production clusters of a large Internet company. This multivariate dataset captures 38 resource-utilization metrics across heterogeneous computing nodes, with anomalies annotated by domain experts.
- **CT** records three-dimensional pen-tip velocity trajectories during handwriting of 20 English characters, all written by a single author. Owing to distinctive handwriting styles, letters G, M, Q, W, and Z are treated as anomaly classes, enabling both general and hard open-set evaluation.
- **SAD** consists of Mel-Frequency Cepstral Coefficient (MFCC) sequences of spoken Arabic digits 0–9. Recordings are collected from 88 native speakers (44 male and 44 female). Considering speaking rate and articulation style, digits zero, three, seven, and eight are regarded as anomaly classes.
- **PTBXL** (Wagner et al., 2020) is a large-scale public 12-lead electrocardiography (ECG) corpus covering normal sinus rhythm and pathological cardiac conditions. Each record is a 10-second window (1000 samples at 100 Hz) and is annotated with four diagnostic superclasses: Myocardial Infarction (MI), ST/T Change (STTC), Conduction Disturbance (CD), and Hypertrophy (HYP).
- **TUSZ** (Shah et al., 2018) is a large public EEG corpus for seizure detection, containing both resting-state brain activity and epileptiform events. We use 10-second windows over 19 channels and follow IMPACT in treating three major seizure types as anomaly classes: Focal Non-Specific Seizure (FNSZ), Generalized Non-Specific Seizure (GNSZ), and Complex Partial Seizure (CPSZ).

G Baselines Details

We compare `TURPLE` with 15 representative time-series anomaly detectors, grouped into unsupervised and open-set methods. The unsupervised group assumes that only normal training data are available. The open-set group additionally exploits a limited number of labeled anomalies under the general and hard protocols of prior work (Ding et al., 2022; Lai et al., 2024; Zhou et al., 2026). All baselines use official or widely adopted implementations, share the same data splits as `TURPLE`, and are tuned on the validation set following the authors' recommendations. Below we briefly describe each method.

Unsupervised methods.

- **TCN-AE** (Garg et al., 2021) is a temporal convolutional autoencoder that reconstructs multivariate windows and scores anomalies by reconstruction error. It serves as a strong convolutional baseline for unsupervised TSAD.
- **THOC** (Shen et al., 2020) builds a temporal hierarchical one-class network that maps series into a multi-resolution hypersphere. Anomalies are identified as samples lying far from the learned normal hierarchy.
- **TranAD** (Tuli et al., 2022) is a transformer-based adversarial detector that uses self-conditioning and adversarial training to amplify reconstruction residuals of rare events. It is designed for efficient multivariate TSAD.

- **DCdetector** (Yang et al., 2023) employs a dual-attention contrastive architecture to learn discrepancy-aware representations without reconstruction. Anomaly scores are derived from the inconsistency between two views of the same window.
- **GPT4TS** (Zhou et al., 2023) adapts a frozen pretrained GPT-2 backbone to time series via patch embedding, demonstrating that large language-model features transfer to forecasting and anomaly detection. We use it as a pretrained unsupervised TSAD baseline.
- **COUTA** (Xu et al., 2024) is a calibrated one-class detector that injects native perturbations to synthesize Pseudo-unseen anomalies and applies uncertainty calibration. It is representative of unsupervised methods that rely on heuristic time-domain augmentation.
- **DADA** (Shentu et al., 2025) pretrains a denoising autoencoder with a pool of eight time-domain augmentations (e.g., flip and scale) to inject synthetic anomalies. The resulting encoder is then used for unsupervised detection.

Open-set methods.

- **DevNet** (Pang et al., 2019) learns a deviation network that pushes labeled anomalies away from a Gaussian reference of normality. It is an early few-shot anomaly detector and is not specifically designed for unseen anomaly types.
- **DSAD** (Ruff et al., 2019) extends deep support vector data description with a limited number of labeled anomalies, enclosing normal samples in a hypersphere while using known anomalies as repulsive examples.
- **NCAD** (Carmona et al., 2021) is a contextual anomaly detector that combines a neural density model with time-series-specific mixup augmentations (COE/WMix) to enlarge the anomaly support.
- **DRA** (Ding et al., 2022) learns disentangled representations of seen, pseudo, and residual anomalies to detect both known and unknown outliers. Pseudo anomalies are generated by heuristic residual modeling.
- **MOSAD** (Lai et al., 2024) is an open-set TSAD method that synthesizes unseen-like anomalies from limited labels and trains a detector to generalize beyond seen classes. It is a direct protocol-level predecessor of our hard-setting evaluation.
- **InvAD** (Liu et al., 2025a) detects anomalies via invertible networks that model the normal density while using labeled anomalies to calibrate the decision boundary under open-set conditions.
- **WSAD-DT** (Durani et al., 2025) is a weakly supervised detector that uses a dual-tailed kernel to exploit scarce anomaly labels and improve tail-class sensitivity.
- **IMPACT** (Zhou et al., 2026) is the strongest open-set TSAD baseline in our tables. It estimates sample influence on validation risk to flip contaminated labels and to perturb high-influence normals, thereby jointly addressing label contamination and unseen-anomaly generation.

H More Experiments

H.1 Performance comparison on seen and unseen anomalies

H.2 Robustness to Anomaly Contamination.

Figure 5 reports AUC under the open-set general setting when the training-set contamination rate varies from 2% to 10%. Competing open-set detectors treat unlabeled windows as normal and therefore absorb contaminated anomalies into the normality model; as the contamination rate grows, they increasingly score similar test anomalies as normal and exhibit a clear performance drop. In contrast, `TURPLE` uses trajectory-guided label refinement with sample-level gating to identify and correct contaminated samples during warmup, converting misleading supervision into reliable anomaly knowledge rather than fitting it as normality. As a result, `TURPLE` remains the most stable across all eight datasets, with only a mild decrease from the Table 1 operating point at 2% contamination, and consistently stays above IMPACT even at 10%. This supports that dynamic label refinement, rather than treating unlabeled data as a clean normal class, is the key to resisting training-set contamination.

Table 5: Performance comparison on seen and unseen anomalies (AUC %, mean ± std). The two Ours columns are highlighted in green. The best in each row is in **bold**; the second best is underlined.

Dataset	seen anomaly	Seen					Unseen				Ours
		MOSAD	InvAD	WSAD-DT	IMPACT	Ours	MOSAD	InvAD	WSAD-DT	IMPACT	Ours
CT	Letter-G	76.75±5.80	78.68±2.74	76.96±0.31	86.27±0.22	**96.45**±0.15	74.53±0.38	72.18±0.75	60.70±2.32	75.53±3.90	**77.41**±1.26
	Letter-M	79.04±0.78	77.32±3.75	77.09±0.31	88.26±0.01	**94.71**±0.02	71.26±0.84	73.88±1.07	67.93±2.78	75.04±5.16	**76.19**±2.88
	Letter-Q	79.63±2.25	74.29±2.87	78.90±1.02	**91.73**±0.04	77.74±0.01	73.11±1.11	75.73±1.25	69.12±2.69	73.27±4.11	**77.16**±2.48
	Letter-W	73.00±2.13	74.15±2.30	**86.67**±2.89	82.58±0.53	84.90±0.54	78.50±0.31	77.33±0.85	65.73±1.86	79.16±2.95	**81.22**±0.43
	Letter-Z	85.53±1.57	81.27±1.75	84.15±0.01	**92.82**±0.02	90.40±0.01	71.87±0.95	69.39±1.00	65.73±1.86	74.34±2.70	**75.30**±1.38
	Mean	78.79±15.82	77.14±13.83	80.75±6.61	88.33±1.73	**88.84**±1.64	73.85±2.58	73.70±2.76	65.84±2.85	75.47±2.00	**77.46**±1.93
SAD	Digit-Zero	44.22±0.90	50.29±4.21	68.30±1.32	67.59±6.55	**73.87**±2.05	61.68±1.02	51.03±3.41	52.76±2.42	58.11±5.23	**62.51**±2.16
	Digit-Three	56.23±0.71	51.19±2.95	71.47±3.25	71.32±4.70	**72.63**±3.17	49.65±0.52	48.49±1.62	69.05±0.48	63.36±4.83	**72.15**±5.55
	Digit-Seven	48.23±1.69	48.25±7.52	62.89±1.46	68.37±3.70	**70.51**±4.31	57.13±0.99	51.61±2.35	63.99±1.30	73.05±2.11	**74.85**±0.91
	Digit-Eight	63.06±1.10	51.26±4.17	73.06±1.27	75.22±0.58	**84.21**±0.52	43.26±0.83	51.62±1.37	51.66±2.44	52.50±4.13	**54.39**±5.26
	Mean	52.94±21.58	50.25±3.24	68.93±10.70	70.63±12.22	**75.30**±11.77	52.93±7.04	50.69±1.29	59.37±7.39	61.76±7.57	**65.98**±6.93
PTBXL	MI	55.31±4.98	52.14±5.65	67.60±1.55	**66.70**±5.29	68.50±3.77	63.91±4.99	53.76±4.56	62.94±1.52	**64.90**±2.94	62.44±0.74
	STTC	50.60±4.24	53.26±4.89	56.36±2.10	59.11±3.64	**66.00**±2.61	56.12±4.83	54.74±5.53	56.48±3.14	59.59±4.11	**62.32**±3.08
	CD	42.59±4.06	51.54±7.04	**64.16**±1.26	61.54±7.14	67.00±1.86	62.85±3.62	51.34±7.28	**64.86**±0.75	61.92±5.25	62.28±3.76
	HYP	58.61±3.50	53.49±7.71	61.41±2.87	61.13±3.04	**65.00**±2.88	61.73±2.79	52.03±6.23	58.63±2.38	**64.91**±3.62	59.98±2.79
	Mean	51.78±3.31	52.61±0.92	62.38±5.03	62.12±3.70	**66.62**±3.88	61.15±3.01	52.97±1.35	60.73±3.33	**62.83**±2.23	61.76±2.34
TUSZ	FNSZ	75.28±1.48	57.13±7.39	75.82±0.89	80.64±1.40	**82.76**±2.18	64.46±4.57	57.17±10.99	53.76±3.60	71.90±5.44	**73.52**±2.76
	GNSZ	64.97±3.35	54.31±11.11	70.19±0.75	79.64±2.23	**79.33**±1.63	68.33±1.74	55.67±7.67	47.01±2.20	69.42±2.77	**72.31**±1.92
	CPSZ	68.47±5.08	56.37±10.26	60.41±1.92	78.08±3.12	**81.90**±2.78	70.95±2.16	55.71±9.04	52.89±1.72	74.52±3.61	**76.32**±1.89
	Mean	69.57±5.86	55.94±2.12	68.81±5.45	79.45±4.15	**81.33**±3.56	67.91±2.67	56.18±0.70	51.22±3.00	71.95±2.08	**74.05**±1.56

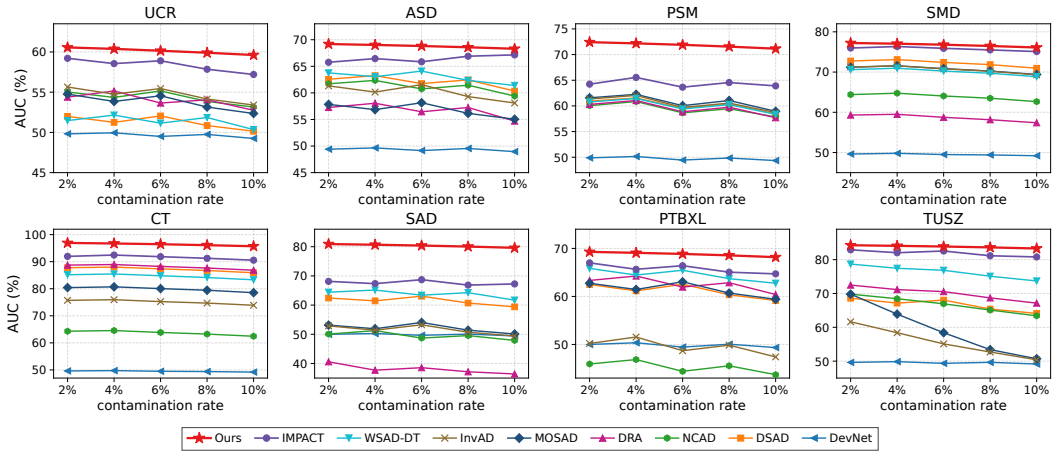

Figure 5: AUC performance of **TURPLE** and competing baselines w.r.t. different contamination rates under the general setting.

H.3 Point-level anomaly detection results

To address the practical challenge that point-level anomaly labels are laborious and error-prone to obtain, we follow the standard sliding-window protocol with step size 1 (Xu et al., 2024; Shentu et al., 2025) and assign each interval anomaly score to its final timestep, thereby enabling point-level detection while preserving temporal continuity. We evaluate performance with Affiliation-F1 (Huet et al., 2022) and VUS-ROC (Boniol et al., 2025). As shown in Table 6, Ours achieves the best Affiliation-F1 on UCR (73.79%), ASD (79.43%), PSM (88.04%), and SMD (83.07%), and the best VUS-ROC of 92.46%, 88.84%, 86.89%, and 94.33%, consistently outperforming IMPACT (Zhou et al., 2026) and all unsupervised and open-set baselines. These results validate that the proposed framework generates more accurate and robust point-level anomaly predictions for both univariate and multivariate time series in open-set environments.

H.4 Complexity Analysis.

Let $N = |\mathcal{S}_{\text{tr}}|$ denote the number of training windows $\mathbf{x}_i \in \mathbb{R}^{T\times D}$ as in the problem definition, and $M = |\mathcal{S}_{\text{n}}|$ the size of the observed-normal subset ($M \le N$ and $|\mathcal{S}_{\text{a}}| \ll M$). The model consists of a TCN encoder f_θ that maps each window to an ℓ_2-normalized embedding in $\mathbb{R}^d$ and two MLP heads h_s, h_u that produce K-dimensional score vectors. With the architecture held fixed,

Table 6: Point-level anomaly detection results (Affiliation-F1/VUS-ROC ± standard deviation) of our method and its competing methods under the unsupervised and open-set general settings. All results are in %. The best ones are in **bold** and the second best are in underlined.

Methods	Affiliation-F1				VUS-ROC			
	UCR	ASD	PSM	SMD	UCR	ASD	PSM	SMD
Unsupervised								
TCN-AE	67.30±0.54	70.50±0.05	73.47±0.73	64.76±0.19	51.89±0.68	62.62±0.17	60.44±0.77	64.17±0.06
THOC	67.34±0.11	70.84±0.76	72.50±1.53	74.85±1.00	53.48±0.51	61.65±1.46	55.35±2.56	64.42±0.77
TranAD	66.93±0.18	71.40±0.45	76.32±0.44	68.62±0.00	50.20±0.56	62.54±0.57	63.72±0.35	68.25±0.17
DCdetector	66.67±0.00	69.78±0.00	70.77±0.09	68.43±0.11	52.04±1.86	54.85±0.17	52.21±0.14	54.89±0.47
GPT4TS	67.92±0.30	71.50±0.63	72.78±0.41	70.43±0.21	61.09±0.36	66.20±0.41	59.80±0.46	72.71±0.30
COUTA	66.99±0.46	70.41±1.25	78.71±0.23	66.34±0.03	55.55±1.51	65.96±1.14	67.02±4.32	65.67±1.60
DADA	66.67±0.00	69.79±0.00	86.93±3.88	67.82±0.00	54.69±0.03	60.96±0.03	73.12±0.05	67.60±0.00
Open-Set								
DevNet	66.55±0.00	70.38±0.21	33.09±2.55	68.45±1.52	48.72±2.58	52.01±2.89	48.64±2.35	50.58±0.94
DSAD	67.04±0.24	76.05±0.46	76.59±0.36	72.73±2.52	51.20±0.78	76.01±0.36	63.47±1.67	79.79±1.19
NCAD	68.85±0.16	74.96±0.66	79.56±1.32	80.03±0.75	60.01±0.35	70.64±1.04	61.41±1.42	68.80±0.61
DRA	67.35±0.36	71.13±0.86	73.09±1.44	70.14±1.36	59.64±0.55	58.84±0.07	52.40±0.33	55.90±0.77
MOSAD	67.24±0.02	71.77±0.07	73.17±0.12	69.26±0.07	55.28±0.02	62.28±0.08	57.45±0.20	68.93±0.54
InvAD	67.06±0.02	70.94±0.25	72.97±0.64	71.79±0.32	58.18±0.12	61.25±1.49	54.07±0.82	62.00±1.58
WSAD-DT	68.60±0.53	73.57±0.98	72.58±0.37	71.51±0.74	50.04±0.94	68.74±1.12	53.79±1.01	69.57±0.93
IMPACT	72.02±0.24	77.12±0.18	86.52±0.63	81.22±0.21	64.53±0.01	78.35±0.94	76.52±0.84	81.37±0.29
TURPLE (ours)	**73.79±0.22**	**79.43±0.82**	**88.04±1.38**	**83.07±0.77**	**92.46±0.21**	**88.84±1.39**	**86.89±1.73**	**94.33±0.62**

one encoder–head forward pass is linear in the input size, i.e., $\mathcal{O}(TD)$ for f_θ plus $\mathcal{O}(d^2)$ for a head; $\mathrm{Dev}(\cdot)$ and the prototype distances d_n, d_a are $\mathcal{O}(K)$ and $\mathcal{O}(d)$ per sample. The Gaussian probe $\mathcal{T}$ acts in the time domain and adds no FFT cost, whereas the learnable augmenter $\mathcal{A}_\phi$ operates in the frequency domain and costs $\mathcal{O}(DT \log T)$ per window. Label refinement runs E_{w} warmup epochs: each observed-normal sample is scored twice per epoch (clean view and $\tilde{x}_i = \mathcal{T}(x_i)$) while the seen head is trained with $\mathcal{L}^{(s)}_{\mathrm{dev}}$, giving $\mathcal{O}\big(E_{\mathrm{w}} NTD\big)$ time. The three-criterion flip then computes a mean, a late-minus-early shift, and a Mann–Kendall statistic on each length-E_{w} trajectory, which is $\mathcal{O}(ME_{\mathrm{w}}^2)$ and linear in M for fixed E_{w}. prototype-repulsive extrapolation applies the learnable augmenter $\mathcal{A}_\phi$ to the anomaly seeds in $\mathcal{A}'$, modifying only the K lowest frequency components via FFT/IFFT; each seed costs $\mathcal{O}(DT \log T)$, and the total cost is $\mathcal{O}(|\mathcal{A}'|DT \log T)$. Because $|\mathcal{A}'| = \mathcal{O}(|\mathcal{S}_{\mathrm{a}}| + |\mathcal{F}|)$ remains much smaller than N, this cost is dominated by the encoder passes over $\mathcal{S}_{\mathrm{tr}}$. Joint training adds a constant number of additional forwards per sample per epoch—$\mathcal{A}_\phi$ for ℓ_i^{adv}, prototype loss, and h_u on the synthesized pseudo-unseen windows—while gating evaluates Mann–Kendall and related statistics on short-term histories of confidence and loss, which is linear in N for a fixed history length. Combining both phases, the overall training time complexity is: $\mathcal{O}\big(E\,N\,T\,D\big)$, where $E = E_{\mathrm{w}} + E_{\mathrm{j}}$ is the total number of epochs, i.e., training scales linearly in the dataset size N, the window length T, and the number of channels D. At inference, $\mathcal{A}_\phi$ and gating are both disabled: equation 42 requires a single pass of f_θ, the two heads h_s, h_u, and a distance to $\mathbf{c}_n$, yielding $\mathcal{O}(TD)$ per window. Table 7 reports wall-clock time and peak GPU memory of one full train–test run; Across eight benchmarks, the peak allocation stays below 0.54 GB and runtime ranges from about one to fifteen minutes.

H.5 More Ablation Study

We provide complete ablation results on all eight datasets under the general setting in Table 8. The full model consistently achieves the highest AUC across all datasets, while removing any single component leads to clear performance degradation, confirming the complementary effectiveness of each designed module.

H.6 More Sensitivity analysis

To further investigate the interaction between the prototype-pulling weight λ_{proto} and the repulsion weight λ_{adv} beyond the single-parameter sweeps in Figure 2, we perform a fine-grained grid search over a 10×10 logarithmic grid with both parameters ranging from 0.1 to 1.0, under the

Table 7: Wall-clock time and peak GPU memory of one training run (including inference) on an NVIDIA RTX A6000.

Dataset	Time (s)	Memory (MB)
UCR	86.0	138.4
ASD	255.4	45.9
PSM	168.7	75.6
SMD	858.9	131.6
CT	61.4	131.4
SAD	270.5	69.6
PTBXL	722.3	537.7
TUSZ	910.0	260.0

Table 8: Ablation study results (AUC $\pm$ std, %) under the general setting. Best results are boldfaced.

Dataset	**Full**	w/o TLR	w/o Gate	w/o Aug	w/o Pre
UCR	$\mathbf{60.56}_{\pm 1.20}$	$60.30_{\pm 1.86}$	$55.60_{\pm 2.24}$	$60.28_{\pm 1.55}$	$51.50_{\pm 4.26}$
ASD	$\mathbf{69.19}_{\pm 2.18}$	$67.80_{\pm 2.97}$	$68.10_{\pm 4.13}$	$69.00_{\pm 2.73}$	$60.80_{\pm 3.25}$
PSM	$\mathbf{72.42}_{\pm 2.20}$	$71.60_{\pm 2.37}$	$70.60_{\pm 3.43}$	$71.70_{\pm 2.35}$	$69.90_{\pm 2.49}$
SMD	$\mathbf{77.22}_{\pm 1.39}$	$75.30_{\pm 2.80}$	$70.30_{\pm 3.34}$	$76.90_{\pm 1.39}$	$73.50_{\pm 2.24}$
CT	$\mathbf{96.93}_{\pm 0.87}$	$94.87_{\pm 1.10}$	$94.21_{\pm 2.12}$	$93.56_{\pm 2.78}$	$95.18_{\pm 1.84}$
SAD	$\mathbf{80.91}_{\pm 1.75}$	$78.15_{\pm 3.62}$	$79.28_{\pm 3.62}$	$79.83_{\pm 4.29}$	$78.99_{\pm 4.29}$
PTBXL	$\mathbf{69.28}_{\pm 1.39}$	$67.80_{\pm 2.39}$	$67.40_{\pm 1.93}$	$69.02_{\pm 2.20}$	$65.70_{\pm 2.68}$
TUSZ	$\mathbf{84.26}_{\pm 0.39}$	$84.00_{\pm 0.88}$	$75.50_{\pm 6.47}$	$84.00_{\pm 0.87}$	$72.30_{\pm 1.77}$

open-set hard setting on CT, SAD, PTBXL, and TUSZ. The resulting response surfaces are shown in Figure 6. The surfaces also exhibit dataset-specific patterns: CT shows a sharp single peak near a balanced $(\lambda_{\text{proto}}, \lambda_{\text{adv}})$ combination; SAD yields a relatively flat plateau, indicating lower sensitivity to this trade-off; PTBXL presents a dominant primary peak alongside a secondary ridge, suggesting two viable operating regimes; TUSZ similarly displays a main peak and a weaker secondary mode. Overall, the pull–push balance governed by these two hyperparameters remains stable across datasets, and moderate values around our default choices consistently achieve near-optimal AUC without abrupt performance cliffs.

H.7 Label Refinement Quality

We evaluate the quality of label refinement on CT under 2% injected contamination. The observed normal set contains $N_0 = 907$ samples, among which 18 are true contaminants. TLR flips samples according to the three-criterion rule, and Random Flip randomly selects samples from N_0 with the same flip ratio as TLR (10 draws $\times$ 3 seeds). Static uses the last-epoch response $\delta_i^{(T_w)}$, and Score ranking uses the TLR score S_i without the three-criterion gate. All variants share the same warm-up procedure as TLR. We report Precision, Recall, F1, False Flip Rate (FFR), AUROC, and AUPRC, where $\text{FFR} = \text{FP}/N_0$. As shown in Table 9, TLR achieves Precision 0.83 ± 0.08, Recall 0.83 ± 0.07, and F1 0.83 ± 0.07, outperforming Random Flip, Static $\delta^{(T_w)}$, and Score ranking. Its FFR is $0.33\% \pm 0.15\%$, lower than Random Flip ($1.74\% \pm 0.51\%$) and Static $\delta^{(T_w)}$ ($1.20\% \pm 0.60\%$). We emphasize that Precision and FFR characterize different aspects of refinement quality: Precision measures the fraction of flipped samples that are true contaminants, whereas FFR measures the fraction of the entire nominal-normal pool that is incorrectly flipped. Therefore, a low FFR does not imply near-perfect precision. TLR remains a beneficial but imperfect supervision-correction module, as roughly one in six flipped samples is still a false positive and a similar fraction of true contaminants remains unflipped. These residual errors motivate the downstream design to treat refined labels as informative but not fully trustworthy, rather than assuming that contamination has been completely eliminated. In terms of ranking quality, TLR obtains AUROC 0.88 ± 0.04 and AUPRC 0.62 ± 0.05, higher than all baselines, confirming that the perturbation-induced deviation trajectory provides a more reliable contamination score than single-epoch responses or simple score ranking.

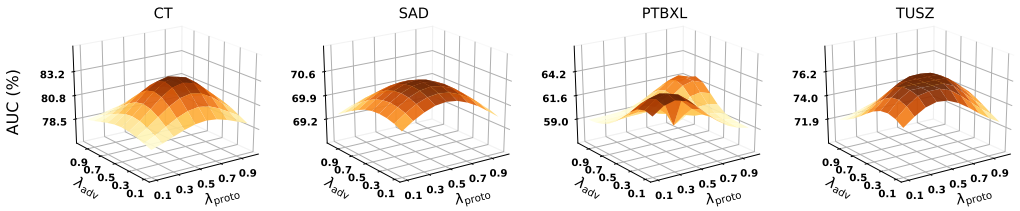

Figure 6: Dual-parameter sensitivity of **TURPLE** under the open-set hard setting on CT, SAD, PTBXL, and TUSZ.

Table 9: Label-refinement quality on CT (2% injected contamination). Observed normals N_0=907, of which 18 are true contaminants. TLR flips samples with the three-criterion rule. Random Flip randomly selects samples from N_0 with the same flip ratio as TLR (10 draws $\times$ 3 seeds). Static uses last-epoch $\delta_i^{(T_w)}$; Score ranking uses the TLR score S_i without the three-criterion gate. Both share TLR's warmup. Mean $\pm$ std.

Method	Precision	Recall	F1	FFR (%)	AUROC	AUPRC
Oracle Flip	1.00	1.00	1.00	0.00	1.00	1.00
TLR (Ours)	0.83 $\pm$ 0.08	0.83 $\pm$ 0.07	0.83 $\pm$ 0.07	0.33 $\pm$ 0.15	0.88 $\pm$ 0.04	0.62 $\pm$ 0.05
Random Flip	0.02 $\pm$ 0.03	0.02 $\pm$ 0.03	0.02 $\pm$ 0.03	1.74 $\pm$ 0.51	0.50 $\pm$ 0.05	0.03 $\pm$ 0.01
Static $\delta^{(T_w)}$	0.32 $\pm$ 0.15	0.28 $\pm$ 0.08	0.30 $\pm$ 0.11	1.20 $\pm$ 0.60	0.48 $\pm$ 0.07	0.15 $\pm$ 0.06
Score ranking (S_i)	0.25 $\pm$ 0.12	0.22 $\pm$ 0.06	0.23 $\pm$ 0.09	1.30 $\pm$ 0.55	0.46 $\pm$ 0.10	0.14 $\pm$ 0.07
No Flip	—	—	—	—	—	—

H.8 More Ablation of TLR

To verify that the three complementary statistics $\bar{\delta}_i$, Z_i, and Δ_i are all necessary, we conduct an ablation on CT under 2% injected contamination. The observed normal set contains $N_0 = 907$ samples, of which 18 are true contaminants. TLR uses the three-criterion conjunction rule. We also evaluate variants that remove one statistic (w/o MK, w/o Δ), use only one statistic (only $\bar{\delta}$, only MK, only Δ), and include Oracle Flip and No Flip as upper and lower references. All variants share the same warm-up and threshold selection protocol. As shown in Table 10, TLR achieves the best overall performance, with Precision 0.83 ± 0.08, Recall 0.83 ± 0.07, F1 0.83 ± 0.07, FFR $0.33\% \pm 0.15\%$, AUROC 0.88 ± 0.04, and AUPRC 0.62 ± 0.05. Removing the MK trend (w/o MK) reduces Recall to 0.62 ± 0.10 and F1 to 0.65 ± 0.09, indicating that the trend criterion is essential for identifying gradually emerging contaminants. Removing the late-stage shift (w/o Δ) lowers Precision to 0.72 ± 0.11 and increases FFR to $0.48\% \pm 0.20\%$, showing that the late-stage criterion helps suppress false positives. The single-statistic variants degrade more severely: only $\bar{\delta}$ achieves F1 0.52 ± 0.12, only MK 0.42 ± 0.13, and only Δ 0.45 ± 0.14, all with higher FFR. These results confirm that the three statistics are complementary and that the conjunction rule is necessary for reliable label refinement under extremely low contamination.

H.9 Quantitative Evaluation of Pseudo-Unseen Anomalies

We quantitatively evaluate the generated pseudo-unseen anomalies from four perspectives: distance to known classes, diversity, constraint satisfaction, and contribution to detection performance. All evaluations are conducted on CT under the hard open-set setting. The pseudo-unseen set $\mathcal{P}$ is generated from $\mathcal{A}'$ using the proposed prototype-guided extrapolation with $|\mathcal{P}| = |\mathcal{A}'|$. To keep the geometric comparison interpretable, distances in this section are measured on the pre-normalization encoder output $f_\theta^{\text{pre}}(\cdot)$, whose scale is not constrained to the unit sphere; the ℓ_2-normalized embedding $\mathbf{z}$ used in the prototype loss and Proposition 3.2 is obtained by normalizing $f_\theta^{\text{pre}}(\mathbf{x})$. As shown in Table 11, pseudo-unseen samples exhibit a mean distance of $d_n = 8.42 \pm 0.61$ to the normal prototype and $d_a = 7.95 \pm 0.54$ to the seen-anomaly prototype, both substantially larger than the corresponding distances of real seen anomalies ($d_n = 4.21 \pm 0.33$, $d_a = 3.87 \pm 0.28$), confirming that extrapolation effectively pushes generated samples away from both known regions. The mean

Table 10: Three-statistic ablation of TLR on CT (2% injected contamination). N_0=907, 18 true contaminants. All variants share the same warm-up and threshold selection protocol. Mean ± std.

Method	Precision	Recall	F1	FFR (%)	AUROC	AUPRC
Oracle Flip	1.00	1.00	1.00	0.00	1.00	1.00
Full TLR (Ours)	0.83 ± 0.08	0.83 ± 0.07	0.83 ± 0.07	0.33 ± 0.15	0.88 ± 0.04	0.62 ± 0.05
w/o MK	0.68 ± 0.11	0.62 ± 0.10	0.65 ± 0.09	0.55 ± 0.22	0.78 ± 0.06	0.48 ± 0.07
w/o Δ	0.72 ± 0.10	0.68 ± 0.09	0.70 ± 0.10	0.48 ± 0.20	0.80 ± 0.05	0.51 ± 0.06
Only $\bar{\delta}$	0.55 ± 0.13	0.50 ± 0.12	0.52 ± 0.12	0.85 ± 0.30	0.70 ± 0.08	0.33 ± 0.09
Only MK	0.45 ± 0.14	0.40 ± 0.13	0.42 ± 0.13	1.05 ± 0.35	0.63 ± 0.09	0.23 ± 0.08
Only Δ	0.48 ± 0.14	0.42 ± 0.13	0.45 ± 0.14	0.95 ± 0.33	0.66 ± 0.09	0.26 ± 0.08
No Flip	—	—	—	—	—	—

Table 11: Quantitative evaluation of pseudo-unseen anomalies on CT (hard setting). d_n: squared distance to the normal prototype; d_a: squared distance to the seen-anomaly prototype; d_{unseen}: nearest-neighbor distance to held-out real unseen anomalies; MPD: Mean Pairwise Distance; MDM: Mean Distance to Medoid; Coverage: fraction of real unseen samples covered by pseudo-unseen samples; Amp/Phase/Low-freq/TV: constraint violation rates (%). Distances d_n, d_a, and d_{unseen} are measured on the pre-normalization encoder output $f_\theta^{\text{pre}}(\cdot)$, whose scale is not constrained to the unit sphere; the ℓ_2-normalized embedding $\mathbf{z}$ used in the prototype loss is obtained by normalizing $f_\theta^{\text{pre}}(\mathbf{x})$. Lower is better for the constraint violation rates (Amp/Phase/Low-freq/TV). Mean ± std.

Method	Distance			Diversity			Constraint violation (%)				Performance	
	d_n	d_a	d_{unseen}	MPD	MDM	Coverage	Amp	Phase	Low-freq	TV	AUROC	AUPRC
Real seen anomalies	4.21 ± 0.33	3.87 ± 0.28	—	—	—	—	—	—	—	—	—	—
Pseudo-unseen (Ours)	8.42 ± 0.61	7.95 ± 0.54	2.13 ± 0.27	6.85 ± 0.42	3.21 ± 0.28	0.73 ± 0.05	0.8 ± 0.3	1.1 ± 0.4	0.0 ± 0.0	2.3 ± 0.7	83.15 ± 3.22	62.38 ± 0.71
Random time-domain pert.	5.12 ± 0.47	4.86 ± 0.41	1.02 ± 0.18	2.14 ± 0.31	1.05 ± 0.19	0.28 ± 0.06	4.2 ± 1.1	3.8 ± 1.0	0.0 ± 0.0	8.7 ± 1.9	79.42 ± 0.58	55.71 ± 0.83
Simple frequency pert.	6.34 ± 0.52	5.91 ± 0.46	1.47 ± 0.22	3.47 ± 0.38	1.82 ± 0.24	0.41 ± 0.07	1.5 ± 0.6	1.9 ± 0.7	0.0 ± 0.0	3.4 ± 0.9	81.03 ± 0.49	58.46 ± 0.76
Remove pseudo-unseen	—	—	—	—	—	—	—	—	—	—	79.86 ± 0.64	52.14 ± 0.91

nearest-neighbor distance to real unseen anomalies is $d_{\text{unseen}} = 2.13 \pm 0.27$, which is moderate and indicates that pseudo-unseen samples reside in a plausible open-space region without collapsing onto real unseen data. In terms of diversity, pseudo-unseen samples achieve MPD $= 6.85 \pm 0.42$, MDM $= 3.21 \pm 0.28$, and Coverage $= 0.73 \pm 0.05$, all clearly higher than random time-domain perturbation (MPD $= 2.14 \pm 0.31$, MDM $= 1.05 \pm 0.19$, Coverage $= 0.28 \pm 0.06$) and simple frequency perturbation (MPD $= 3.47 \pm 0.38$, MDM $= 1.82 \pm 0.24$, Coverage $= 0.41 \pm 0.07$). The proposed method also achieves low constraint violation rates for amplitude ($0.8\% \pm 0.3\%$), phase ($1.1\% \pm 0.4\%$), low-frequency band ($0.0\% \pm 0.0\%$), and total variation ($2.3\% \pm 0.7\%$), confirming that the learnable augmenter $\mathcal{A}_\phi$ and the ℓ_∞/TV constraints keep generated samples within valid regions. Finally, replacing $\mathcal{P}$ with random time-domain perturbations causes AUROC to drop from 83.15 to 79.42 and AUPRC from 62.38 to 55.71, while replacing with simple frequency perturbations yields AUROC 81.03 and AUPRC 58.46; removing pseudo-unseen samples entirely causes the largest drop (AUROC 79.86, AUPRC 52.14). These results confirm that the generated pseudo-unseen anomalies are essential for robust open-set detection and cannot be replaced by naive perturbations.

H.10 Validation of Prototype-Guided Pseudo-Anomaly Synthesis

To validate the effectiveness of our proposed boundary-extrapolating prototype synthesis against alternative pseudo-anomaly generation strategies, we conduct a controlled experiment where only the generator of $\hat{x}$ is replaced, while keeping all other components—frequency-response driven refinement, sample-level gating, and the remaining training pipeline—fixed. We compare four drop-in alternatives under the open-set hard setting: our prototype-guided synthesis, the IMPACT perturbation, Cut-and-Paste (CAP), and Cut-Out-and-Exchange (COE). COE-WMix and WMix are excluded as they rely on similar interpolative heuristics and do not affect the comparison.

Figure 7 reports per-class AUC on CT, SAD, PTBXL, and TUSZ. The `TURPLE` bars reproduce the results from Table 2, confirming the full model's performance. Replacing our generator with any of the three alternatives consistently degrades accuracy across all seen-class splits, demonstrating the superiority of our synthesis approach. The relative ranking among IMPACT, CAP, and COE varies with

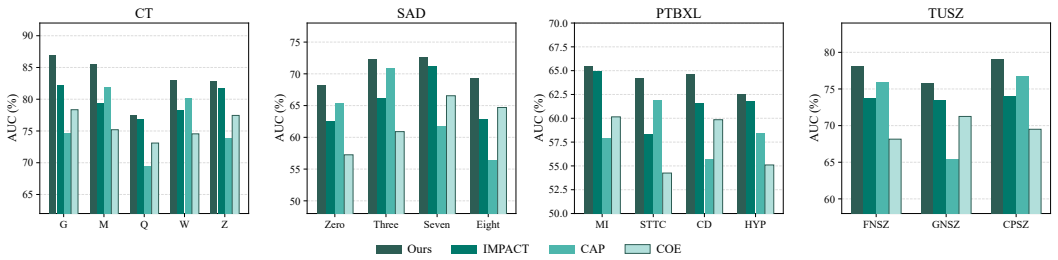

Figure 7: AUC (%) under the open-set hard setting when substituting different pseudo-anomaly generators. **TURPLE** uses boundary-extrapolating prototype synthesis; IMPACT, CAP, and COE are drop-in alternatives in the same pipeline.

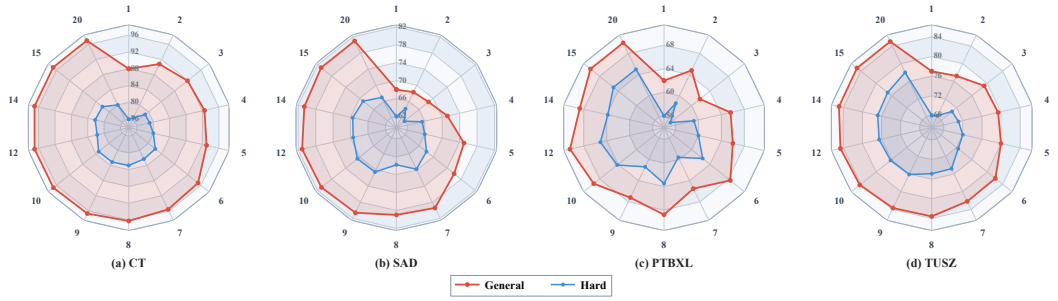

Figure 8: AUC (%) of **TURPLE** versus the number of labeled anomalies $|\mathcal{S}_a|$ (vertices) from 1 to 20 under the open-set general and hard settings. Radial rings denote AUC.

the seen class: CAP, which performs mixup-style interpolation, is competitive on certain morphologically similar classes (e.g., SAD Digit-Three, TUSZ CPSZ), while COE, which exchanges local segments, performs better when preserving structural continuity is critical (e.g., SAD Digit-Eight, TUSZ GNSZ). This variability is characteristic of interpolative generators, which remain anchored to the seen-class manifold and thus only benefit when unseen classes are proximate. In contrast, our prototype-guided synthesis consistently achieves the highest accuracy across all settings, because it extrapolates beyond both the normal and seen-anomaly prototypes, generating supervision that transfers effectively to genuinely unseen anomaly types.

H.10.1 Utility of Few-Shot Labeled Anomalies

We further examine how **TURPLE** uses the few labeled anomalies by varying $|\mathcal{S}_a|$ from 1 to 20 under both open-set protocols on CT, SAD, PTBXL, and TUSZ. Figure 8 reports the resulting AUC as radar plots, where each vertex is a shot count, and the radial coordinate is AUC (%). The red and blue polygons denote the general and hard settings, respectively; the $|\mathcal{S}_a| = 10$ vertices coincide with Tables 1 and 2. Under the general setting, AUC rises from a single labeled sample and saturates after about ten labels on CT and SAD, while PTBXL and TUSZ continue to gain slightly as more classes remain represented in $\mathcal{S}_a$. The hard setting is more sensitive to the shot count: performance on CT and SAD peaks near $|\mathcal{S}_a| = 10$ and then declines, indicating that excessive supervision on a single seen class overfits the seen head and hurts unseen generalization. In contrast, the larger PTBXL and TUSZ corpora still improve when more labels of the same type are added, so richer within-class variation outweighs the overfitting effect. Across all four datasets, the general polygon encloses the hard one, confirming that the default budget of ten labeled anomalies per class is near-optimal and that **TURPLE** remains effective even with one to five labeled samples.

H.11 Visualizations

Figure 9 shows anomaly-score densities under the hard setting. Each panel is one visible class on CT (G/M/Q/Z), SAD (Zero/Three/Seven/Eight), and PTBXL (MI/STTC/CD/HYP). The four curves are test normals, leftover seen-class anomalies, unseen-class anomalies, and Pseudo-unseen anomalies scored by the same decision function. Normals stay on the low-score side; seen anomalies sit in the

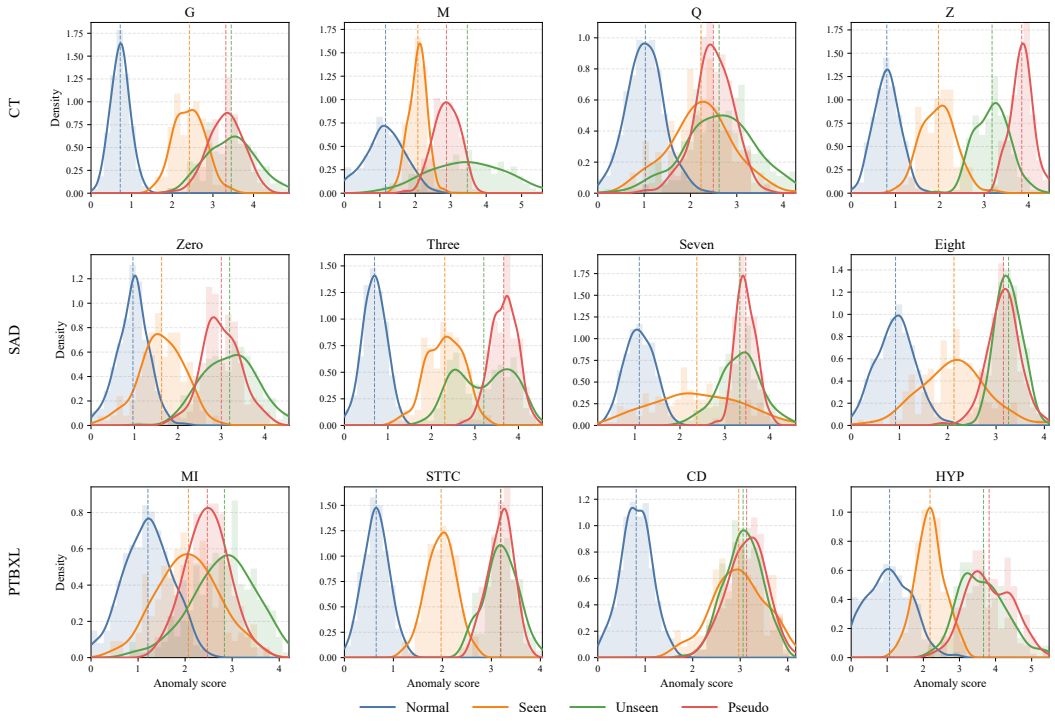

Figure 9: Pseudo-unseen anomalies consistently track unseen anomaly distributions across all datasets

mid-range; unseen anomalies shift right. Across all twelve splits, the pseudo-anomaly mass tracks the unseen mode instead of collapsing onto normals, which indicates that it populates the open-set region used at inference rather than merely amplifying in-distribution perturbations. The separation is clearest on CT-G/Z and SAD-Three/Eight, and remains visible even in harder, more overlapping cases such as CT-Q and PTBXL-MI.

H.11.1 Visualization of Learnable Frequency Augmentation

Figure 10 illustrates how the learnable frequency augmenter $\mathcal{A}_\phi$ edits a window on a UCR series. In the time domain (panel a), $\mathcal{A}_\phi(\mathbf{x})$ follows the shape of $\mathbf{x}$ while the residual $x - \mathcal{A}_\phi(x)$ concentrates on local peaks and troughs. In the frequency domain (panel b), the amplitude spectra $|X(f)|$ and $|\mathcal{A}_\phi(X)|$ diverge only inside the shaded low-frequency band and remain identical outside it. This confirms that the augmenter performs a band-limited amplitude/phase perturbation rather than a broadband distortion.

H.11.2 Visualization of Adaptive Sample-Level Gating

Figure 4 visualizes the sample-level gate on TUSZ hard (seen = FNSZ). Panel (a) shows the three binary criteria for observed-normal windows (sorted with flagged noise first): I_1 (rising confidence), I_2 (expanding DTW gap between clean and augmented losses), and I_3 (augmented-loss shift). A window is assigned to the harmful-noise set $\mathcal{F}_{\text{noise}}$ only when $\neg I_1 \wedge I_2 \wedge I_3$, which yields $|\mathcal{F}_{\text{noise}}| = 39$ on this run. Panel (b) tracks $|\mathcal{F}_{\text{noise}}|$ over joint-training epochs: the set peaks early (581) and then shrinks to 39, indicating that the gate gradually stops treating hard-but-clean samples as harmful noise.

H.11.3 Visualization of Label Refinement

Figure 11 provides a concrete waveform example from the CT dataset under the hard setting (seen = Letter-G), where the TLR successfully identifies and corrects a contaminated normal sample that visually resembles an anomaly.

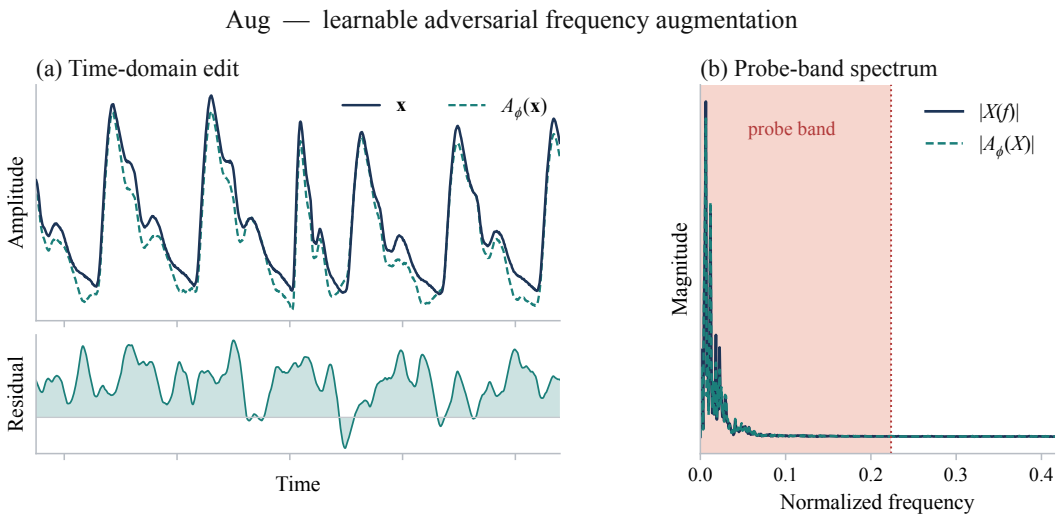

Figure 10: Visualization of learnable frequency augmentation on a UCR window. (a) Original series $\mathbf{x}$ and the edited view $\mathcal{A}_\phi(\mathbf{x})$, with residual $x - \mathcal{A}_\phi(x)$ below. (b) Amplitude spectra. The augmenter modifies only the shaded low-frequency band and leaves higher frequencies unchanged.

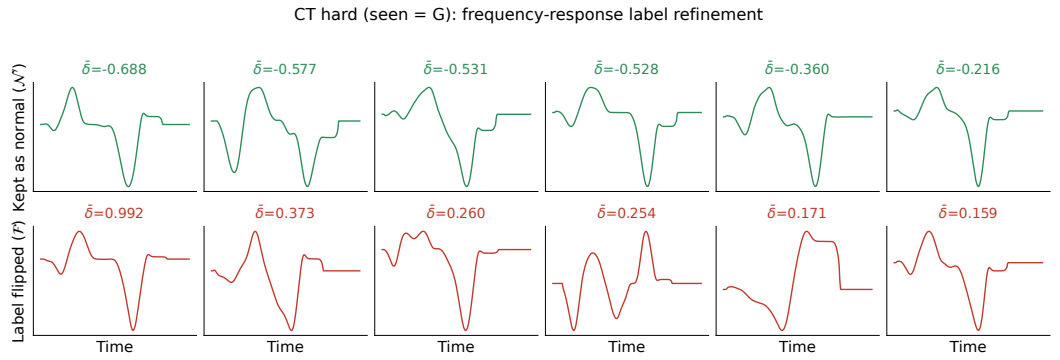

Figure 11: Waveform-level visualization of label refinement on CT hard setting (seen = Letter-G).

H.12 Point-level score curves

To complement the window-level AUC results, Figure 12 visualizes point-level anomaly scores on three representative UCR test segments. Each series is scored with a sliding window of length 100 and stride 1; window scores are mapped back to timestamps by averaging overlapping assignments. The first row shows the raw series; subsequent rows show min–max normalized scores of GPT4TS (Zhou et al., 2023), DADA (Shentu et al., 2025), DSAD (Ruff et al., 2019), NCAD (Carmona et al., 2021), IMPACT (Zhou et al., 2026), and our method, with red bands marking the ground-truth anomaly intervals. Reconstruction-based GPT4TS and DADA often follow the background periodicity and fire on regular peaks outside the event, while DSAD and NCAD produce dense scores that do not isolate the annotated interval. IMPACT is smoother but still mixes the response with nearby normal cycles. Our method yields a higher contrast between the annotated interval and the surrounding normal region, forming a concentrated peak inside the red band—most clearly on Segment 2. These qualitative patterns are consistent with the quantitative gains on UCR.

I Notation

For ease of reading, Tables 12 and 13 collect the symbols used throughout this section. We group them by role: Table 12 covers the open-set problem setup, the encoder and deviation scoring, and

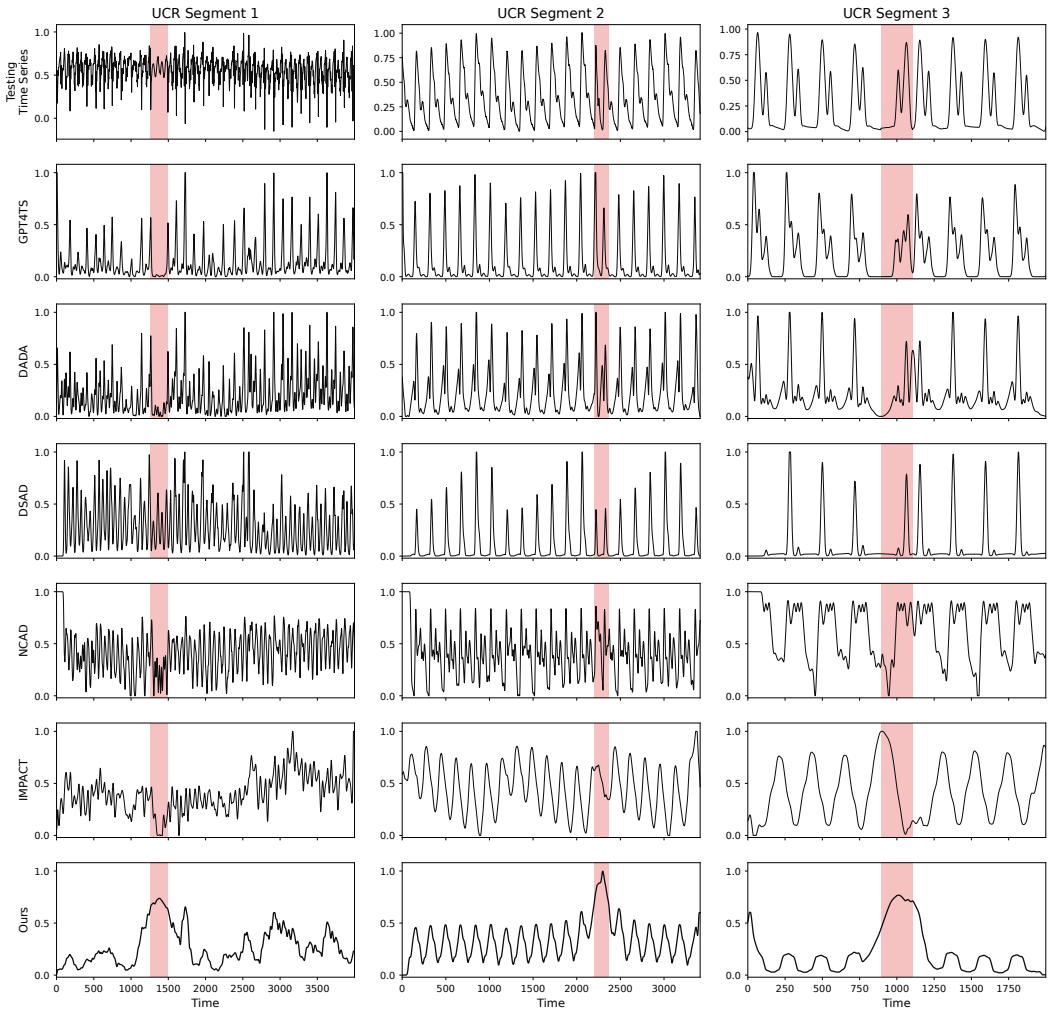

Figure 12: Visualization of point-level anomaly scores on the UCR dataset. The first row shows the testing time series; the remaining rows show scores from GPT4TS, DADA, DSAD, NCAD, IMPACT, and our method. Red regions highlight the ground-truth anomaly intervals.

the trajectory-guided label refinement; Table 13 covers the prototype-repulsive extrapolation and the joint training and inference.

J Details of Implementation

Following Impact (Zhou et al., 2026), we evaluate all methods using the area under the ROC curve (AUC) and report the mean $\pm$ standard deviation over five independent runs. Our **TURPLE** model employs a TCN backbone with two hidden layers of 64 units each and an embedding size of 64. Two MLP heads with LayerNorm and GELU activation project the features to a 3-dimensional deviation score. We train the model using AdamW with a learning rate of 3×10^{-4}, a weight decay of 10^{-4}, and a batch size of 64, for 14 to 30 epochs, with the first one-third of the epochs used for warm-up. We set the label-correction ratio $\rho_{\max} = 0.04$ and the Gaussian perturbation strength $\varepsilon = 0.06$. The balancing coefficients λ_{proto} and λ_{adv} are selected from $[0.6, 1.0]$, and the fusion weight ω is selected from $[0.5, 0.7]$. All experiments are implemented in PyTorch and run on a single NVIDIA RTX A6000 GPU.

J.1 Discussion on Unseen-Class Generalization

Table 12: Notation used in the method (Part I): problem setup, encoder and deviation scoring, and trajectory-guided label refinement.

Symbol	Type	Meaning
Problem definition		
$\mathcal{S}_{\text{tr}}$, N	set, scalar	Training set and its size.
$\mathbf{x}_i \in \mathbb{R}^{T\times D}$	tensor	Window with T timesteps and D channels.
$y_i \in \{0,1\}$	label	Observed label (0: normal, 1: anomaly).
$\mathcal{S}_{\text{n}}$, $\mathcal{S}_{\text{a}}$	sets	Observed-normal set (possibly contaminated) and labeled seen anomalies.
$\mathcal{N}$, $\mathcal{A}$	sets	Shorthands for $\mathcal{S}_{\text{n}}$ and $\mathcal{S}_{\text{a}}$.
$\mathcal{K}$, $\mathcal{K}_{\text{seen}}$, $\mathcal{K}_{\text{unseen}}$	sets	All / seen / unseen anomaly classes.
$q(\cdot)$	function	Anomaly scoring function to be learned.
T_{ep}	scalar	Number of epochs in the deviation-based pre-training phase.
Encoder, heads, and deviation scoring		
f_θ, f_θ^{pre}	encoder	TCN encoder; $f_\theta^{\text{pre}}(\mathbf{x})$ denotes the pre-normalization output.
$\mathbf{z}_i$	vector	ℓ_2-normalized embedding $\mathbf{z}_i = f_\theta^{\text{pre}}(\mathbf{x}_i)/\|f_\theta^{\text{pre}}(\mathbf{x}_i)\|_2$.
d, K, C_s	scalars	Embedding dimension, score-vector length, and score dimension of both heads.
h_s, h_u	heads	Seen-anomaly head and unseen-anomaly head (two-layer MLPs).
$\mathbf{s}^{(s)}$, $\mathbf{s}^{(u)}$	vectors	Score vectors from h_s and h_u.
$\boldsymbol{\mu}^{(h)}$, $\boldsymbol{\nu}^{(h)}$	vectors	Mean and standard deviation of trusted normal scores under head h.
$\text{Dev}^{(h)}(\mathbf{s})$	scalar	Scale-invariant deviation from the normal reference equation 1.
α_{dev}	scalar	Deviation margin in the pre-training objective equation 1.
$\mathcal{L}_{\text{dev}}^{(s)}$, $\mathcal{L}_{\text{Dev}}^{u}$	losses	Deviation losses for the seen head and the unseen head equation 1,equation 19.
Trajectory-guided label refinement		
$\mathcal{T}(\cdot)$	operator	Fixed Gaussian perturbation $\tilde{\mathbf{x}} = \mathbf{x} + \varepsilon\boldsymbol{\zeta}$, $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
ε	scalar	Perturbation strength of the Gaussian probe.
$\delta_i^{(t)}$, $\boldsymbol{\delta}_i$	scalar, vector	Perturbation-induced deviation change at epoch t, and the full trajectory.
$\bar{\delta}_i$	scalar	Mean deviation response of sample i.
r	ratio	Fraction of T_{ep} epochs used to define early/late sub-trajectories.
μ_N, μ_C, σ^2	scalars	Sub-Gaussian means of genuine / contaminated responses, and variance.
τ_δ^{emp}	threshold	High-percentile cutoff on $\bar{\delta}_i$.
$Z_i = \text{MK}(\boldsymbol{\delta}_i)$	statistic	Mann–Kendall trend of the trajectory.
Δ_i	scalar	Late-minus-early shift of the trajectory.
τ_{MK}	threshold	Significance cutoff of the Mann–Kendall test.
$\rho_{\max}$	ratio	Maximum fraction of $\mathcal{N}$ that may be flipped.
$\mathcal{F}$, $\mathcal{N}'$, $\mathcal{A}'$	sets	Flipped samples; refined normal and anomaly sets.

Table 13: Notation used in the method (Part II): prototype-repulsive extrapolation, joint training, and inference.

Symbol	Type	Meaning
prototype-repulsive extrapolation		
$\mathbf{c}_n, \mathbf{c}_a$	vectors	Momentum-updated normal and seen-anomaly prototypes.
$d_n(\mathbf{z}), d_a(\mathbf{z})$	scalars	Squared distances to $\mathbf{c}_n$ and $\mathbf{c}_a$.
$\mathcal{A}_\phi$	augmenter	Learnable frequency augmenter acting on the K lowest frequencies.
$\boldsymbol{\delta}_a, \boldsymbol{\delta}_p$	tensors	Per-channel amplitude and phase offsets stored by $\mathcal{A}_\phi$.
$\mathcal{P}$	set	Pseudo-unseen anomaly set.
$\mathcal{P}_\mathrm{b}$	set	Mini-batch of generated pseudo-unseen samples.
$\hat{\mathbf{x}}$	tensor	Synthesized pseudo-unseen sample $\hat{\mathbf{x}} = \mathcal{A}_\phi(\mathbf{x}_\mathrm{seed})$.
$Q(\mathbf{z})$	scalar	Joint distance to both prototypes, $Q(\mathbf{z}) = d_n(\mathbf{z}) + d_a(\mathbf{z})$.
$J_\theta(\mathbf{x}), \epsilon$	Jacobian, scalar	Encoder Jacobian and ℓ_2 perturbation size in Proposition 3.2.
Joint training and inference		
$\ell^\mathrm{adv}, \mathcal{L}_\mathrm{adv}$	losses	Per-sample and batch objectives equation 19.
$\mathcal{L}_\mathrm{proto}$	loss	Prototype compactness / separation loss equation 12.
H	scalar	History length for adaptive gating.
$c_t(\hat{\mathbf{x}}), \Delta_t(\hat{\mathbf{x}}), r_t(\hat{\mathbf{x}})$	scalars	Confidence, clean–augmented loss discrepancy, and relative augmented-loss shift of $\hat{\mathbf{x}}$ at epoch t.
Z_c, Z_Δ	statistics	Mann–Kendall statistics of the confidence and discrepancy histories.
$I_1, I_2, I_3, F_\mathrm{noise}$	flags	Gating indicators and the harmful-sample predicate equation 16.
τ_Δ, τ_r	thresholds	Thresholds for $\lvert\Delta_t(\hat{\mathbf{x}})\rvert$ and $r_t(\hat{\mathbf{x}})$ in the gating rule.
γ	scalar	Down-weight factor on harmful gradients ($\gamma < 1$).
$\lambda_\mathrm{proto}, \lambda_\mathrm{adv}, \lambda_\mathrm{reg}$	weights	Coefficients of the joint objective equation 19.
$A(x_*)$	score	Fused inference score equation 42.
ω	scalar	Seen-head weight in the convex combination of the two heads.
β	scalar	Weight of the average versus the max of the two heads.
w_proto	scalar	Weight of the distance to the normal prototype at test time.