LEARNING TIME-SERIES ANOMALY GENERATORS FROM SYNTHETIC SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic anomalies are widely used to alleviate the scarcity of labeled anomalies in time-series anomaly detection. Existing anomaly injection methods use predefined transformations to construct injected anomalies, which may limit data diversity and fail to sufficiently capture the temporal characteristics of the target data. This work proposes a generative framework for multivariate time-series anomaly synthesis based on normal-pattern pre-training and anomaly-oriented fine-tuning. The generator is first pre-trained exclusively on normal time-series data to capture target-specific temporal patterns. Then, multi-type injected anomalies are constructed to provide *synthetic supervision*, and a dimension-adaptive reconstruction objective is introduced during fine-tuning to adjust relative reconstruction penalties according to channel-wise statistics. After fine-tuning, latent perturbations broaden the sampling region of the learned latent distribution, enabling more diverse anomaly generation. Unlike conventional anomaly injection, the injected anomalies are not directly used as the final generated anomalies, but instead guide the generator in learning heterogeneous deviations from normal temporal patterns. Experiments on five multivariate time-series benchmarks and three downstream anomaly detectors show consistent improvements, with average AUPR gains of 13.28%, 11.28%, and 16.85% for CARLA, COUTA, and DACAD, respectively. Representation analysis further shows that the generated anomalies achieve stronger separability from normal patterns and greater diversity than direct anomaly injection, supporting their effectiveness for downstream training.

1 INTRODUCTION

Time-series anomaly detection (TSAD) aims to identify observations or temporal segments that deviate from expected system behavior, with applications in industrial monitoring, cyber-physical systems, and other safety-critical scenarios (Zamanzadeh Darban et al., 2024; Schmidl et al., 2022; Goh et al., 2017; Hundman et al., 2018). Recent deep TSAD methods have explored diverse architectures. OmniAnomaly models stochastic temporal dynamics using stochastic recurrent neural networks (Su et al., 2019), while MSCRED and USAD adopt reconstruction-based architectures (Zhang et al., 2019; Audibert et al., 2020). THOC and InterFusion learn hierarchical temporal or inter-metric representations (Shen et al., 2020; Li et al., 2021). Graph-based methods model cross-variable dependencies through graph attention or graph neural networks (Zhao et al., 2020; Deng & Hooi, 2021), whereas Anomaly Transformer and TranAD leverage attention-based Transformer architectures (Xu et al., 2021; Tuli et al., 2022). Despite substantial progress, multivariate TSAD remains challenging because real anomalies are typically rare, heterogeneous, and expensive to label (Zamanzadeh Darban et al., 2024; Carmona et al., 2021).

Data augmentation has been widely explored to alleviate limited supervision in time-series learning (Wen et al., 2020), and recent TSAD studies increasingly incorporate synthetic anomalies into anomaly-aware representation learning. Methods such as NCAD (Carmona et al., 2021), COUTA (Xu et al., 2024), CutAddPaste (Wang et al., 2024), CARLA (Darban et al., 2025b), DACAD (Darban et al., 2025c), CAROTS (Kim et al., 2025), and CAAE (Xie et al., 2026) construct synthetic anomalies through different perturbation or structural mechanisms to improve normal–abnormal discrimination. These developments highlight the importance of informative synthetic anomalies for detector training. Throughout this work, we use *synthetic anomalies* as a general

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.

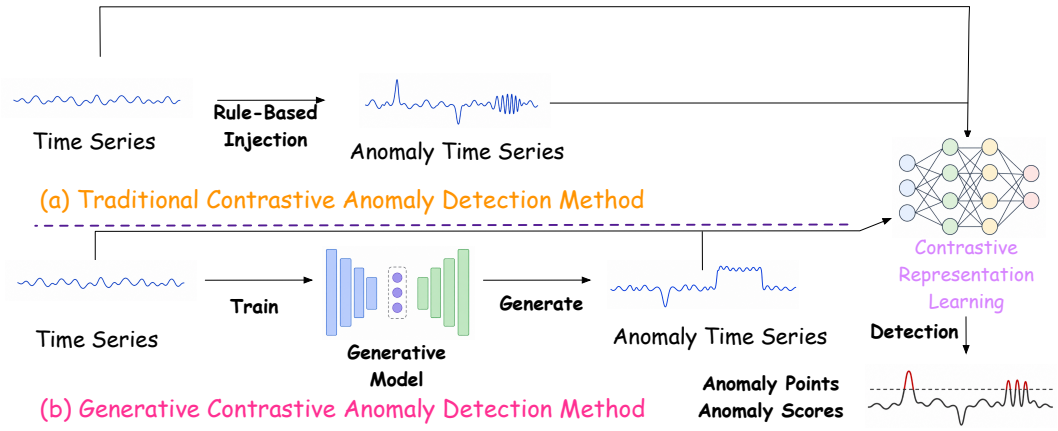


Figure 1: Comparison of direct anomaly injection and our generative anomaly synthesis paradigm for anomaly-aware detector training.

term for artificially constructed anomalies, while distinguishing *injected anomalies* produced by rule-based anomaly injection from *generated anomalies* produced by our adapted generator.

However, many anomaly construction methods rely on predefined transformations, including point perturbations, trend changes, contextual modifications, seasonal disturbances, and patch-based manipulations (Carmona et al., 2021; Wang et al., 2024; Xu et al., 2024; Darban et al., 2025b;c). Such rule-based mechanisms constrain anomalies to predefined operators, potentially limiting pattern diversity. Generative anomaly synthesis offers a promising yet underexplored alternative by learning to generate anomalies from data. GenIAS (Darban et al., 2025a) jointly learns a compact normal latent distribution and a learnable variance perturbation in a TCN-VAE, followed by deviation-based patching to construct localized anomalies. However, its anomalies are generated primarily from representations learned from normal data, which may limit its ability to model heterogeneous anomalous patterns. This motivates introducing explicit synthetic supervision into generative model training to learn deviations from normal temporal patterns. Moreover, channel-wise statistical heterogeneity in multivariate time series remains insufficiently explored in generative anomaly learning. Figure 1 contrasts direct anomaly injection with our generative synthesis paradigm for downstream detector training.

Therefore, we propose a generative framework for multivariate time-series anomaly synthesis based on normal-pattern pre-training and anomaly-oriented fine-tuning. The generator is first pre-trained on normal time series to capture target-specific temporal patterns, and is then fine-tuned with multi-type injected anomalies as *synthetic supervision*, enabling it to explicitly learn heterogeneous deviations from normal behavior rather than directly using injected anomalies as final downstream training samples. During fine-tuning, a dimension-adaptive objective adjusts reconstruction penalties according to channel-wise statistics while bounding extreme weight differences, improving robustness to statistical heterogeneity across dimensions. After fine-tuning, latent perturbations broaden the sampling region of the learned latent distribution, enabling more diverse anomaly generation. The resulting framework is detector-agnostic and can be integrated into different anomaly-aware detection models.

Our main contributions are summarized as follows:

- We propose a generative framework for multivariate time-series anomaly synthesis based on normal-pattern pre-training and anomaly-oriented fine-tuning. During pre-training, the generator learns target-specific temporal patterns from normal data; during fine-tuning, multi-type injected anomalies provide *synthetic supervision* to guide the generator in learning heterogeneous abnormal patterns.
- We introduce a dimension-adaptive fine-tuning objective that adjusts reconstruction penalties according to channel-wise statistics while limiting extreme weight differences. We

further apply latent perturbations after fine-tuning to broaden the sampling region of the learned latent distribution and improve anomaly diversity.

- Extensive experiments on five multivariate time-series benchmarks and three downstream anomaly detectors show average AUPR gains of 13.28%, 11.28%, and 16.85% for CARLA, COUTA, and DACAD, respectively. Representation analysis and ablation studies further validate the separability, diversity, and downstream utility of the generated anomalies. Our code is available at `https://anonymous.4open.science/r/code-18E5`.

2 Related Work

2.1 Anomaly-Aware Representation Learning

Deep anomaly detection has increasingly benefited from representation learning and explicit anomaly-aware supervision. Deep SVDD (Ruff et al., 2018) learns compact one-class representations of normal data, while Deep SAD (Ruff et al., 2019) and Outlier Exposure (Hendrycks et al., 2018) demonstrate that even limited or auxiliary anomalous examples can substantially improve anomaly discrimination. In time-series learning, TS-TCC (Eldele et al., 2021) and TS2Vec (Yue et al., 2022) learn temporal representations through self-supervised contrastive objectives. Recent TSAD methods further incorporate synthetic anomalies into detector training: NCAD (Carmona et al., 2021), COUTA (Xu et al., 2024), CARLA (Darban et al., 2025b), and DACAD (Darban et al., 2025c) introduce synthetic anomalies or perturbations into one-class, contrastive, or domain-adaptive objectives. These studies primarily focus on how synthetic anomalies are exploited by the detector, whereas our work focuses on synthesizing informative anomalies independently of a specific downstream architecture.

2.2 Synthetic Anomaly Construction

Time-series data augmentation encompasses both transformation-based and generative strategies for increasing training diversity (Wen et al., 2020). In TSAD, rule-based approaches typically perturb normal time series through predefined transformations, such as point, contextual, trend, seasonal, shape-based, or patch-based modifications (Carmona et al., 2021; Wang et al., 2024; Xu et al., 2024; Darban et al., 2025b;c). Beyond these generic transformations, CAROTS (Kim et al., 2025) constructs anomalies by disturbing inter-variable causal relationships, while CAAE (Xie et al., 2026) employs a multi-lineage anomaly injector to produce diverse abnormal patterns. GenIAS (Darban et al., 2025a) instead learns anomaly generation through latent-space perturbation in a variational autoencoder. More broadly, generative models such as TimeGAN (Yoon et al., 2019), COT-GAN (Xu et al., 2020), and GT-GAN (Jeon et al., 2022) learn temporal distributions for general time-series synthesis. While these methods primarily aim to reproduce the underlying data distribution, our method generates informative deviations from the normal distribution by using injected anomalies as synthetic supervision to adapt the generator to model heterogeneous anomalous behaviors.

3 Problem Formulation

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ denote a multivariate time series of length T, where $\mathbf{x}_t \in \mathbb{R}^d$ is the d-dimensional observation at time step t. Let

$$\mathcal{D}_{\text{normal}} = \{\mathbf{X}_i\}_{i=1}^{N} \tag{1}$$

denote the normal training set. Since real anomalies are scarce or unavailable during training, we aim to produce synthetic anomalies from the normal training data while retaining the temporal characteristics of the target system. In practice, each time series is divided into fixed-length windows $\mathbf{x} \in \mathbb{R}^{L \times d}$, where L denotes the input window length. Anomaly synthesis is performed at the window level, allowing the generator to model local temporal patterns and their abnormal deviations. Our goal is to learn a generative model G_θ that produces a set of generated anomalies

$$\widehat{\mathcal{D}}_{\text{anomaly}} = \{\widehat{\mathbf{x}}_k\}_{k=1}^{K}, \qquad \widehat{\mathbf{x}}_k \sim G_\theta(\cdot \mid \mathbf{x}), \quad \mathbf{x} \in \mathcal{D}_{\text{normal}}^{(L)}, \tag{2}$$

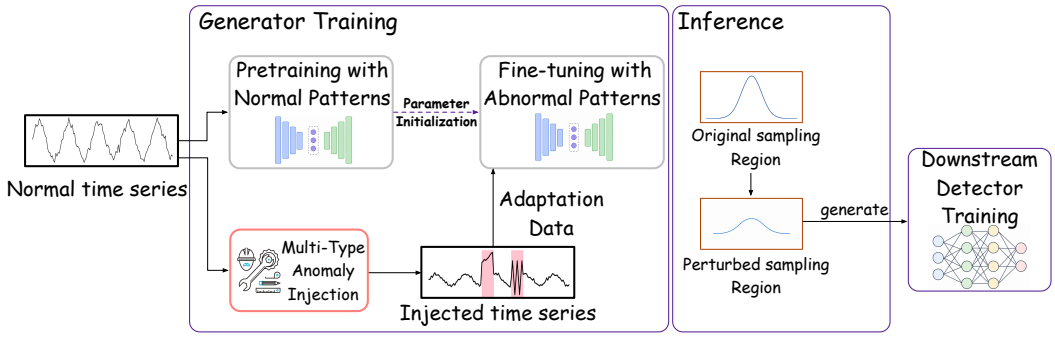


Figure 2: Overview of the proposed generative anomaly synthesis framework. The generator is first pre-trained on normal time series and then fine-tuned using multi-type injected anomalies as synthetic supervision. During inference, latent perturbations broaden the sampling region of the learned latent distribution to produce a diverse set of generated anomalies for downstream detector training.

where $\mathcal{D}^{(L)}_{\text{normal}}$ denotes the set of fixed-length windows extracted from the normal training sequences. The generated anomalies are expected to preserve target-specific temporal characteristics while exhibiting meaningful deviations from normal behavior. These generated anomalies are subsequently used to augment the training data of downstream anomaly detectors.

4 METHOD

4.1 OVERVIEW

As illustrated in Figure 2, our framework follows a pre-training and anomaly-oriented fine-tuning paradigm. A generative model is first trained on normal time series to capture target-specific temporal patterns. We then inject multiple anomaly types into normal samples and use the resulting injected anomalies as *synthetic supervision*, rather than as final downstream training samples, to adapt the generator to model heterogeneous anomalous behaviors through a dimension-adaptive objective. Finally, latent perturbations are applied to the adapted generator to produce a diverse set of generated anomalies for downstream detector training. The complete training and generation procedure is summarized in Algorithm 1 in Appendix A.1.

4.2 NORMAL PATTERN PRE-TRAINING

We instantiate the generator as a temporal convolutional variational autoencoder (TCN-VAE) (Bai et al., 2018; Kingma & Welling, 2013) with encoder q_ϕ and decoder p_θ. Given a normal segment $\mathbf{x} \in \mathbb{R}^{L\times d}$, the encoder parameterizes a Gaussian posterior with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$. The latent representation is sampled as

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

followed by reconstruction $\hat{\mathbf{x}} = p_\theta(\mathbf{z})$.

The generator is pre-trained exclusively on normal data using the standard reconstruction loss

$$\mathcal{L}^{\text{pre}}_{\text{rec}} = \frac{1}{Ld} \sum_{t=1}^{L} \sum_{r=1}^{d} (\hat{x}_{t,r} - x_{t,r})^2 . \tag{4}$$

To preserve structurally inactive observations, we additionally penalize artificial responses at zero-valued positions:

$$\mathcal{L}_{\text{zero}}(\mathbf{x}) = \frac{1}{Ld} \sum_{t=1}^{L} \sum_{r=1}^{d} \mathbb{I}[x_{t,r} = 0] \, |\hat{x}_{t,r}|. \tag{5}$$

The complete pre-training objective is

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{rec}}^{\text{pre}} + \beta D_{\text{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right) + \lambda_0 \mathcal{L}_{\text{zero}}(\mathbf{x}). \tag{6}$$

Pre-training is performed separately for each entity to capture its specific temporal and channel-wise characteristics.

4.3 Multi-type Anomaly Injection

Normal-pattern pre-training alone does not characterize how abnormal behaviors deviate from normality. We therefore construct an anomaly-oriented fine-tuning set using five anomaly operators: trend, contextual, shapelet, global, and seasonal perturbations (Darban et al., 2025b). For a normal segment $\mathbf{x}$, an injected sample is generated as

$$\mathbf{x}^a = \mathcal{A}_k\left(\mathbf{x}; \mathcal{I}, \mathcal{S}\right), \tag{7}$$

where $\mathcal{I}$ denotes the affected temporal region and $\mathcal{S} \subseteq \{1, \ldots, d\}$ denotes the set of perturbed dimensions. By varying the operator, temporal region, and dimensions, we obtain $\mathcal{D}_{\text{ft}} = \{\mathbf{x}_i^a\}_{i=1}^{N_{\text{ft}}}$. These injected anomalies provide synthetic supervision for generator adaptation rather than serving as the final generated anomalies.

4.4 Dimension-Adaptive Fine-tuning

The pre-trained generator is then optimized on $\mathcal{D}_{\text{ft}}$. Multivariate channels can exhibit substantially different variability and activity patterns, causing some dimensions to affect the reconstruction objective disproportionately. We therefore introduce channel-wise weighting to adjust their relative reconstruction penalties while limiting extreme differences between dimension weights.

We assign each dimension a weight using its variance v_r and zero-value ratio ρ_r, both estimated from normal training data:

$$\begin{aligned} \bar{w}_r &= \frac{1}{\sqrt{v_r + \epsilon}\,(\rho_r + \epsilon)}, \\ w_r &= \frac{\text{clip}(\bar{w}_r, w_{\min}, w_{\max})}{\frac{1}{d}\sum_{j=1}^{d} \text{clip}(\bar{w}_j, w_{\min}, w_{\max})}. \end{aligned} \tag{8}$$

Here, the variance and zero-value ratio jointly characterize the statistical behavior of each dimension. Clipping limits extreme differences between channel weights, while normalization sets their mean to one. Consequently, the weighting mechanism adjusts relative reconstruction penalties without allowing a small number of dimensions to dominate the objective through excessively large weights.

For an injected sample $\mathbf{x}^a$, the dimension-adaptive reconstruction loss is

$$\mathcal{L}_{\text{rec}}^{\text{ft}} = \frac{1}{Ld} \sum_{t=1}^{L} \sum_{r=1}^{d} w_r \left(\hat{x}_{t,r}^a - x_{t,r}^a\right)^2. \tag{9}$$

We retain the same zero-preserving constraint defined during pre-training, which operates at individual time steps and complements the dimension-level weighting. The complete fine-tuning objective is

$$\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{rec}}^{\text{ft}} + \beta D_{\text{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}^a) \| p(\mathbf{z})\right) + \lambda_0 \mathcal{L}_{\text{zero}}(\mathbf{x}^a). \tag{10}$$

This stage adapts the generator from modeling the normal temporal distribution to modeling heterogeneous anomalous patterns while accounting for channel-wise statistical differences.

4.5 Anomaly Synthesis with Latent Perturbation

After fine-tuning, we further diversify anomaly generation through latent perturbations. Given a normal input window $\mathbf{x}$, we first sample a latent representation $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})$ using the fine-tuned encoder. We then apply an additional perturbation $\boldsymbol{\delta}$ using Gaussian perturbations of different strengths, directional shifts, or partial-dimension perturbations, and generate

$$\widetilde{\mathbf{z}} = \mathbf{z} + \boldsymbol{\delta}, \qquad \boldsymbol{\delta} \sim \mathcal{P}_\delta, \qquad \widetilde{\mathbf{x}} = p_\theta(\widetilde{\mathbf{z}}). \tag{11}$$

These perturbations broaden the region of latent space accessed during inference, enabling the fine-tuned generator to produce a more diverse set of generated anomalies while retaining target-specific temporal characteristics. The resulting samples are used to augment downstream anomaly detector training.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate on five multivariate TSAD benchmarks: MSL and SMAP (Hundman et al., 2018), SMD (Su et al., 2019), GECCO (Moritz et al., 2018), and SWAN (Angryk et al., 2020). They cover both multi-entity and single-entity settings with diverse dimensionalities and anomaly ratios. Dataset statistics are summarized in Table 1.

Table 1: Statistics of the benchmark datasets.

Benchmark	# Entities	# Dims	Train Samples	Test Samples	Anomaly (%)
MSL	27	55	58,317	73,729	10.72
SMAP	55	25	140,825	444,035	13.13
SMD	28	38	708,405	708,420	4.16
GECCO	1	9	69,260	69,261	1.10
SWAN	1	38	60,000	60,000	32.60

Downstream Detectors. We evaluate the generated anomalies using CARLA (Darban et al., 2025b), COUTA (Xu et al., 2024), and DACAD (Darban et al., 2025c). For each detector, we retain the original architecture and training procedure and modify only the anomaly samples used during training. DACAD is evaluated on MSL, SMAP, and SMD only, as it is not applicable to the single-entity GECCO and SWAN benchmarks.

Baselines. We compare against the original detector-specific anomaly injection strategy and three synthesis baselines: GenIAS (Darban et al., 2025a), CAAE (Xie et al., 2026), and CAROTS (Kim et al., 2025). For all alternatives, only the anomaly synthesis component is replaced while the downstream detector and training configuration remain unchanged.

Implementation Details. The TCN-VAE uses a latent dimension of 100 and a hidden dimension of 32. Its input window length is matched to the original setting of each downstream detector, while other detector-specific settings follow the original implementations. Detailed training hyperparameters are provided in Appendix A.5.

Evaluation Metrics. We primarily report AUPR, which is suitable for highly imbalanced data (Saito & Rehmsmeier, 2015). All AUPR and F1 scores are computed from the original point-wise predictions without point adjustment or post-processing. For multi-entity benchmarks, mean AUPR and standard deviation are reported where applicable. F1 results are provided in Appendix A.3. We additionally evaluate anomaly separability and diversity in the representation space, with definitions given in Appendix A.2. Computational efficiency is reported in Appendix A.4.

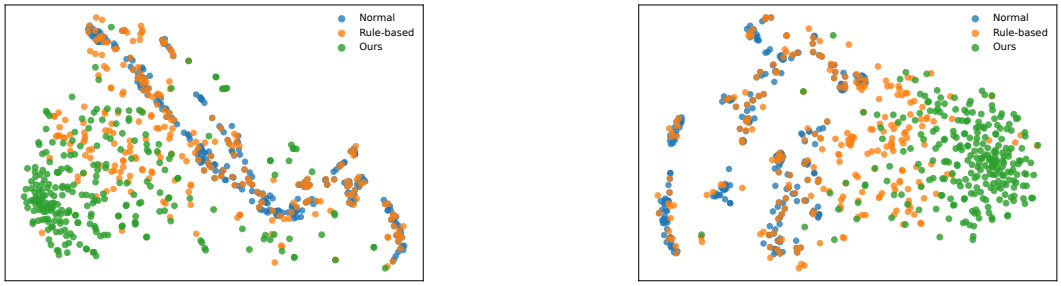
5.2 OVERALL DETECTION PERFORMANCE

Table 2 compares the downstream detection performance of different anomaly synthesis methods in terms of AUPR.

Our method consistently improves all three downstream detectors, with average AUPR gains of 13.28%, 11.28%, and 16.85% for CARLA, COUTA, and DACAD, respectively. It achieves the largest average gain for all three detectors and the best AUPR in 9 of the 13 applicable detector–benchmark combinations, indicating that the generated anomalies generalize across different downstream architectures.

Table 2: AUPR comparison across downstream detectors and benchmarks. Avg. Gain denotes the average improvement over the corresponding base detector. "Base" uses the detector's original anomaly generation or augmentation mechanism, and "M" denotes an inapplicable setting.

Model	Method	Avg. Gain	MSL	SMAP	SMD	GECCO	SWAN
CARLA	Base	–	0.499 ± 0.247	0.462 ± 0.309	0.478 ± 0.200	0.228	0.840
	GenIAS	+4.75%	0.529 ± 0.251	0.467 ± 0.315	0.512 ± 0.195	0.236	0.882
	CAAE	−3.03%	0.543 ± 0.254	0.455 ± 0.328	0.458 ± 0.192	0.182	0.793
	CAROTS	+10.57%	0.554 ± 0.229	0.499 ± 0.333	0.501 ± 0.207	**0.315**	**0.903**
	Ours	+13.28%	**0.577** ± 0.241	**0.573** ± 0.331	**0.530** ± 0.183	0.271	0.889
COUTA	Base	–	0.319 ± 0.265	0.289 ± 0.304	0.353 ± 0.235	0.278	0.579
	GenIAS	+2.81%	0.325 ± 0.251	0.319 ± 0.339	**0.385** ± 0.270	0.213	0.627
	CAAE	−0.94%	0.319 ± 0.260	0.292 ± 0.335	0.367 ± 0.235	0.260	0.563
	CAROTS	−0.88%	0.314 ± 0.253	0.291 ± 0.335	0.375 ± 0.223	0.244	0.578
	Ours	+11.28%	**0.333** ± 0.261	**0.326** ± 0.339	0.381 ± 0.236	**0.332**	**0.651**
DACAD	Base	–	0.332 ± 0.261	0.380 ± 0.325	0.457 ± 0.237	M	M
	GenIAS	+12.32%	0.386 ± 0.313	0.371 ± 0.305	0.556 ± 0.218	M	M
	CAAE	+3.85%	0.322 ± 0.270	**0.422** ± 0.333	0.470 ± 0.233	M	M
	CAROTS	+10.52%	0.326 ± 0.262	0.406 ± 0.328	0.560 ± 0.148	M	M
	Ours	+16.85%	**0.390** ± 0.284	0.412 ± 0.312	**0.564** ± 0.238	M	M



(a) M-4 (b) D-15

Figure 3: t-SNE visualization of normal, injected, and generated anomaly representations on two MSL entities.

In contrast, competing synthesis methods provide less consistent gains and occasionally degrade the corresponding base detector. Although they remain competitive on several individual benchmarks, our method delivers the most consistent overall improvement across detectors and datasets.

5.3 Representation Analysis of Generated Anomalies

We compare normal samples, injected anomalies, and our generated anomalies on two MSL entities, M-4 and D-15. All samples are encoded by the same TCN-VAE trained on normal data and projected to two dimensions using t-SNE (Van der Maaten & Hinton, 2008).

As shown in Figure 3, injected anomalies substantially overlap with normal samples, whereas our generated anomalies are more clearly separated from the normal region.

To quantify this difference beyond the two-dimensional visualization, we measure *separability* and *diversity* in the original 100-dimensional latent space. Separability reflects the distance between anomalous and normal representations, while diversity measures the dispersion among anomalous representations. Formal definitions are provided in Appendix A.2.

As shown in Table 3, our method achieves substantially higher separability and diversity than anomaly injection on both entities. As a negative control, Shuffle-Amplify randomly disrupts temporal order and sparsely amplifies selected observations. Despite these strong perturbations, it yields separability comparable to anomaly injection and substantially lower diversity, indicating that trivial

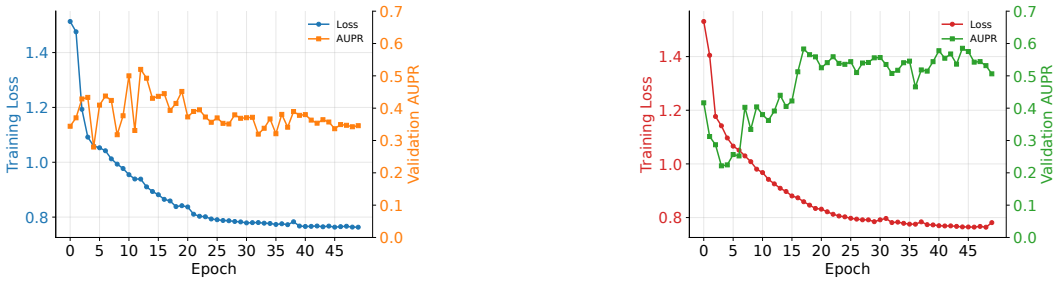
Table 3: Separability and diversity on two MSL entities. Shuffle-Amplify is a deliberately corrupted control.

Entity	Method	Separability ↑	Diversity ↑
M-4	Injected	1.146	3.052
	Shuffle-Amplify	1.098	0.655
	Ours	**3.254**	**4.489**
D-15	Injected	1.108	1.940
	Shuffle-Amplify	1.158	0.296
	Ours	**3.359**	**3.176**

numerical corruption does not reproduce the representation characteristics of our generated anomalies. This suggests that the observed gains arise not merely from increasing perturbation magnitude, but from adapting the generator to learn structured deviations from the normal temporal distribution, yielding anomalies that are both distinguishable from normal patterns and diverse within the abnormal representation space.

5.4 Effect on Downstream Model Training

We further examine how generated anomalies affect downstream training dynamics. Figure 4 compares DACAD with and without our generated anomalies on the D-15 entity of MSL.



(a) DACAD

(b) DACAD + Ours

Figure 4: Training loss and validation AUPR over 50 epochs on the D-15 entity of MSL.

As shown in Figure 4, both settings exhibit steadily decreasing training loss. However, the original DACAD reaches its peak validation AUPR relatively early and then gradually declines, whereas DACAD with our generated anomalies maintains a higher and more stable validation AUPR in later epochs.

This divergence between decreasing training loss and declining validation AUPR suggests that continued optimization of the original DACAD does not necessarily improve anomaly discrimination. In contrast, the generated anomalies provide additional abnormal patterns and are associated with more stable downstream validation performance.

5.5 Ablation Study

We evaluate three key components of the proposed framework: normal-pattern pre-training, dimension-adaptive fine-tuning, and latent perturbation, using CARLA and COUTA on MSL, SMAP, and SMD. "w/o Dim.-Adaptive" replaces the dimension-weighted reconstruction loss with an unweighted reconstruction loss, "w/o Pre-training" removes normal-pattern pre-training, and "w/o Latent Perturb." removes the additional latent-space perturbations during anomaly generation.

As shown in Table 4, removing any of the three components consistently degrades performance. The effect is particularly evident for CARLA, where removing pre-training reduces AUPR from 0.573 to 0.532 on SMAP, while removing dimension-adaptive reconstruction lowers AUPR from 0.530

Table 4: Ablation study of the main components of the proposed framework.

Model	Benchmark	Full		w/o Dim.-Adaptive		w/o Pre-training		w/o Latent Perturb.	
		F1	AUPR	F1	AUPR	F1	AUPR	F1	AUPR
CARLA	MSL	0.601	0.577	0.590	0.558	0.575	0.560	0.585	0.557
	SMAP	0.729	0.573	0.677	0.560	0.652	0.532	0.683	0.556
	SMD	0.549	0.530	0.516	0.492	0.519	0.501	0.503	0.508
COUTA	MSL	0.417	0.333	0.412	0.322	0.409	0.321	0.415	0.330
	SMAP	0.392	0.326	0.379	0.312	0.388	0.322	0.389	0.324
	SMD	0.423	0.381	0.421	0.372	0.421	0.375	0.419	0.374

to 0.492 on SMD. Latent perturbation also provides consistent gains, with smaller but systematic improvements across both detectors.

Overall, the results confirm that the three components play complementary roles: pre-training captures target-specific normal patterns, dimension-adaptive fine-tuning accounts for heterogeneous channel statistics, and latent perturbation broadens the latent sampling region during inference.

5.6 Effect of Multiple Anomaly Types

We further examine whether the anomaly types used to construct the fine-tuning set provide complementary supervision. Starting from trend, contextual, shapelet, global, and seasonal anomalies, we remove one type at a time and repeat the anomaly generation stage. The resulting samples are evaluated using CARLA on MSL.

Table 5: Ablation of anomaly types using CARLA on the MSL benchmark.

Anomaly Type	Prec	Rec	F1	AUPR
Full types	0.492	0.772	**0.601**	**0.577**
w/o trend	0.443	0.803	0.571	0.538
w/o contextual	0.468	0.773	0.583	0.554
w/o shapelet	0.458	0.753	0.569	0.555
w/o global	0.456	0.769	0.573	0.545
w/o seasonal	0.474	0.780	0.590	0.569

As shown in Table 5, using all five anomaly types achieves the best AUPR of 0.577, while removing any individual type degrades performance. The largest drops occur when trend and global anomalies are removed, with AUPR decreasing to 0.538 and 0.545, respectively. These results indicate that different anomaly types provide complementary abnormal patterns during fine-tuning, enabling the generator to capture a broader range of deviations from the normal temporal distribution.

6 Conclusion

This work proposes a generative framework for multivariate time-series anomaly synthesis based on normal-pattern pre-training and anomaly-oriented fine-tuning. The generator first learns target-specific temporal patterns from normal data and is then fine-tuned with multi-type injected anomalies as synthetic supervision, enabling it to learn heterogeneous deviations from normal behavior. A dimension-adaptive objective accounts for channel-wise statistical heterogeneity, while latent perturbations broaden the sampling region of the learned latent distribution to improve anomaly diversity. Experiments on five multivariate time-series benchmarks and three downstream detectors show consistent improvements over the original detectors and the largest mean AUPR gains among the compared synthesis strategies. Representation analysis further demonstrates stronger separability and diversity, while ablation studies validate the contributions of the proposed components. These results highlight the potential of generative anomaly synthesis as a flexible approach for providing informative generated anomalies to downstream models that learn from anomalous samples.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and improving the clarity, grammar, and presentation of the manuscript. The technical content, methodology, experimental design, results, and conclusions were developed and verified by the authors. All AI-assisted text was carefully reviewed and revised by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide an anonymous implementation at `https://anonymous.4open.science/r/code-18E5`. The proposed synthesis framework, including normal-pattern pre-training, multi-type anomaly injection, dimension-adaptive fine-tuning, and latent perturbation, is described in Section 4. The evaluated benchmarks, downstream detectors, baselines, and evaluation protocol are specified in Section 5.1. Additional evaluation details and results are provided in Appendix A.2, Appendix A.3, and Appendix A.4, while the generator architecture, training hyperparameters, and latent perturbation configurations are documented in Appendix A.5.

REFERENCES

Rafal A Angryk, Petrus C Martens, Berkay Aydin, Dustin Kempton, Sushant S Mahajan, Sunitha Basodi, Azim Ahmadzadeh, Xumin Cai, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, et al. Multivariate time series dataset for space weather data analytics. *Scientific data*, 7(1):227, 2020.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

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

Zahra Zamanzadeh Darban, Qizhou Wang, Geoffrey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. Genias: Generator for instantiating anomalies in time series. *arXiv preprint arXiv:2502.08262*, 2025a.

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, 2025b.

Zahra Zamanzadeh Darban, Yiyuan Yang, Geoffrey I Webb, Charu C Aggarwal, Qingsong Wen, Shirui Pan, and Mahsa Salehi. Dacad: Domain adaptation contrastive learning for anomaly detection in multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, 37 (8):4485–4496, 2025c.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4027–4035, 2021.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 140–145. IEEE, 2017.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

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.

Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. *Advances in Neural Information Processing Systems*, 35:36999–37010, 2022.

HyunGi Kim, Jisoo Mok, Dongjun Lee, Jaihyun Lew, Sungjae Kim, and Sungroh Yoon. Causality-aware contrastive learning for robust multivariate time-series anomaly detection. *arXiv preprint arXiv:2506.03964*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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.

Steffen Moritz, Frederik Rehbach, Sowmya Chandrasekaran, Margarita Rebolledo, and Thomas Bartz-Beielstein. Gecco industrial challenge 2018 dataset: A water quality dataset for the 'internet of things: Online anomaly detection for drinking water quality'competition at the genetic and evolutionary computation conference 2018, kyoto, japan. *Kyoto, Japan*, 2018.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

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.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022. doi: 10.14778/3538598.3538602. URL https://doi.org/10.14778/3538598.3538602.

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.

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.

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.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Xin Xie, Kexuan Liu, Ying Wang, Mengqi Wu, Huichaoyi Zhang, and Tao Wan. Caae: Contrastive adversarial autoencoder for multivariate time series anomaly detection. *Pattern Recognition*, pp. 113687, 2026.

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.

Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. Cot-gan: Generating sequential data via causal optimal transport. *Advances in neural information processing systems*, 33: 8798–8809, 2020.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1):1–42, 2024.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1409–1416, 2019.

Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, pp. 841–850. IEEE, 2020.

A Appendix

A.1 Overall Algorithm

Algorithm 1 Learning Time-Series Anomaly Generators from Synthetic Supervision

Require: Normal training set $\mathcal{D}_{\text{normal}}$, anomaly operators $\mathcal{A}$, latent perturbation distribution $\mathcal{P}_\delta$
Ensure: Generated anomaly set $\widehat{\mathcal{D}}_{\text{anomaly}}$
1: Pre-train generator G_θ on $\mathcal{D}_{\text{normal}}$ using Eq. 6
2: Construct injected anomalies $\mathcal{D}_{\text{ft}}$ from $\mathcal{D}_{\text{normal}}$ using multi-type anomaly operators $\mathcal{A}$
3: Compute dimension weights $\{w_r\}_{r=1}^{d}$ according to Eq. 8
4: Fine-tune G_θ on $\mathcal{D}_{\text{ft}}$ using synthetic supervision and Eq. 10
5: Initialize $\widehat{\mathcal{D}}_{\text{anomaly}} \leftarrow \emptyset$
6: **for** each normal window $\mathbf{x}$ selected for generation **do**
7: Sample $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})$
8: Sample $\boldsymbol{\delta} \sim \mathcal{P}_\delta$
9: Generate $\widetilde{\mathbf{x}} \leftarrow p_\theta(\mathbf{z} + \boldsymbol{\delta})$
10: $\widehat{\mathcal{D}}_{\text{anomaly}} \leftarrow \widehat{\mathcal{D}}_{\text{anomaly}} \cup \{\widetilde{\mathbf{x}}\}$
11: **end for**
12: **return** $\widehat{\mathcal{D}}_{\text{anomaly}}$

A.2 Definition of Representation Metrics

To complement the qualitative t-SNE visualization in the main text, we quantitatively evaluate the representation characteristics of generated anomalies using two metrics: *separability* and *diversity*. Both metrics are computed directly in the original 100-dimensional latent representation space produced by the TCN-VAE encoder trained on normal data, rather than in the two-dimensional t-SNE space.

Let $\{\mathbf{z}_j^n\}_{j=1}^{N_n}$ and $\{\mathbf{z}_i^a\}_{i=1}^{N_a}$ denote the latent representations of normal and anomalous samples, respectively.

Separability. Separability measures how distinguishable the anomalous representations are from normal patterns. For each anomalous representation, we compute its Euclidean distance to the nearest normal representation and average these distances over all anomalous samples:

$$\text{Sep} = \frac{1}{N_a} \sum_{i=1}^{N_a} \min_{1 \le j \le N_n} \left\| \mathbf{z}_i^a - \mathbf{z}_j^n \right\|_2 . \tag{12}$$

A higher separability indicates that the anomalous representations are, on average, farther from their nearest normal representations.

Diversity. Diversity measures the dispersion of anomalous representations within the latent space. It is computed as the average pairwise Euclidean distance among all anomalous representations:

$$\text{Div} = \frac{2}{N_a(N_a - 1)} \sum_{i<j} \left\| \mathbf{z}_i^a - \mathbf{z}_j^a \right\|_2 . \tag{13}$$

A higher diversity indicates that the anomalous representations cover a broader region of the latent space rather than concentrating within a narrow region.

For a fair comparison, normal samples, injected anomalies, and generated anomalies from our method are encoded using the TCN-VAE encoder trained on normal data. Separability and diversity are calculated before applying t-SNE, thereby avoiding potential distortions introduced by two-dimensional projection.

A.3 Additional F1-Score Results

In addition to the AUPR results reported in the main text, Table 6 presents the corresponding F1 scores across different downstream detectors and benchmarks. Our method improves upon the base

Table 6: F1-score comparison on time-series anomaly detection benchmarks. "Base" denotes each detector using its original anomaly generation or augmentation mechanism, without incorporating any additional anomaly synthesis method. "M" indicates that the detector is not applicable to the benchmark.

Model	Method	Avg. Gain	MSL	SMAP	SMD	GECCO	SWAN
CARLA	Base	–	0.542	0.600	0.467	0.293	0.908
	GenIAS	+6.51%	0.579	0.620	0.528	**0.329**	**0.937**
	CAAE	+0.57%	0.578	0.623	0.468	0.220	**0.937**
	CAROTS	+6.90%	0.569	0.660	0.514	0.326	0.935
	Ours	+10.75%	**0.601**	**0.729**	**0.549**	0.296	**0.937**
COUTA	Base	–	0.408	0.370	0.403	0.389	0.544
	GenIAS	+2.51%	0.414	0.388	0.425	0.360	0.580
	CAAE	+0.71%	0.407	0.365	0.422	**0.424**	0.511
	CAROTS	+1.47%	0.409	0.364	**0.427**	0.398	0.547
	Ours	+6.53%	**0.417**	**0.392**	0.423	0.412	**0.608**
DACAD	Base	–	0.424	0.472	0.541	M	M
	GenIAS	+7.86%	0.456	0.470	0.624	M	M
	CAAE	+0.97%	0.417	0.494	0.540	M	M
	CAROTS	+7.03%	0.422	0.491	0.625	M	M
	Ours	+11.41%	**0.464**	**0.509**	**0.628**	M	M

detector in all applicable detector–benchmark combinations and achieves the largest relative improvement in mean F1 for each detector. Avg. Gain denotes the relative improvement in mean F1 over the corresponding base model, computed across its applicable benchmarks.

The improvements vary across datasets. For example, CARLA gains 0.129 in F1 on SMAP, while COUTA gains 0.064 on SWAN. Our method also achieves the highest F1 on all three benchmarks evaluated with DACAD. Although it does not outperform every competing method on every benchmark, these results show that its benefits extend beyond AUPR to the precision–recall balance captured by F1.

A.4 Efficiency Analysis

Table 7: Efficiency comparison averaged over five runs. Runtime is reported in seconds, with relative runtime to the corresponding original detector shown in parentheses. For "+ Ours", total runtime includes both anomaly synthesis and downstream detector training.

Method	MSL	SMAP	SMD	GECCO	SWAN
CARLA	1993 (1.00×)	6310 (1.00×)	8866 (1.00×)	3608 (1.00×)	2534 (1.00×)
CARLA + Ours	2590 (1.30×)	7230 (1.15×)	12063 (1.36×)	3759 (1.04×)	2802 (1.11×)
COUTA	562 (1.00×)	855 (1.00×)	1659 (1.00×)	70 (1.00×)	104 (1.00×)
COUTA + Ours	629 (1.12×)	1098 (1.28×)	4323 (2.61×)	202 (2.89×)	305 (2.93×)

Table 7 reports the total runtime averaged over five runs, including anomaly synthesis and downstream detector training. CARLA incurs moderate overhead ($1.04\times$–$1.36\times$), while COUTA exhibits a wider range ($1.12\times$–$2.93\times$). The larger relative overhead for COUTA on GECCO and SWAN partly reflects its short baseline runtime: the absolute increases are 132 and 201 seconds, respectively.

The overhead is more substantial on SMD, where COUTA's runtime increases from 1659 to 4323 seconds. Thus, the computational cost depends on both the detector and the benchmark. Reporting absolute runtime alongside relative ratios makes this variation explicit and allows the detection improvements to be assessed together with their additional training cost.

A.5 Implementation Details

The anomaly generator uses a lightweight TCN-VAE with a 100-dimensional latent space, a hidden dimension of 32, one temporal convolutional layer, and a kernel size of 3. The input window length follows the original setting of each downstream detector. We set the KL weight to 10^{-3} and the zero-dimension coefficient to 0.5. For dimension-adaptive fine-tuning, $\epsilon = 10^{-8}$ and the dimension weights are clipped to $[0.5, 5.0]$ and normalized to unit mean. Clipping limits extreme differences between dimension weights, while normalization preserves their average scale.

Detector- and benchmark-specific training schedules are reported in Table 8. Pretraining and fine-tuning use separate epoch budgets and learning rates, with a lower learning rate during fine-tuning in all listed settings. These schedules refer to the anomaly generator rather than the downstream detector.

Table 8: Training hyperparameters of the anomaly generator.

Detector	Benchmark	Pre. Ep.	FT Ep.	Pre. LR	FT LR
COUTA	MSL / SMAP / GECCO / SWAN	150	100	10^{-3}	5×10^{-4}
	SMD	150	100	10^{-3}	10^{-4}
CARLA	MSL / SMAP / GECCO / SWAN	100	50	10^{-3}	10^{-4}
	SMD	100	50	10^{-3}	5×10^{-4}
DACAD	MSL / SMAP / SMD	100	50	10^{-3}	10^{-4}

Latent Perturbation. During anomaly generation, one of five perturbations is uniformly sampled with base strength $\alpha = 0.5$, as summarized in Table 9. The Gaussian variants introduce random deviations at different scales, whereas the directional variant applies a fixed shift to all latent coordinates. The partial-latent variant modifies only the first 20 coordinates, leaving the others unchanged. The perturbed latent code is then decoded into a synthetic anomaly window for downstream training.

Table 9: Latent perturbation configurations.

Type	Dimensions	Perturbation
Gaussian weak	All	$\mathcal{N}(\mathbf{0}, (0.5\alpha)^2\mathbf{I})$
Gaussian medium	All	$\mathcal{N}(\mathbf{0}, (0.7\alpha)^2\mathbf{I})$
Gaussian strong	All	$\mathcal{N}(\mathbf{0}, (1.2\alpha)^2\mathbf{I})$
Directional	All	$0.8\alpha\mathbf{1}$
Partial latent	First 20	$\mathcal{N}(\mathbf{0}, (0.9\alpha)^2\mathbf{I})$