EDSR: EXTREME DOMAIN SHIFT RESILIENCE VIA CAUSALITY-GUIDED ZERO-SHOT ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly detection is significantly challenging when multivariate time series are subjected to extreme domain shifts. These shifts can lead to magnitude variations of up to 100× between the training phase and deployment, occurring without any labeled anomaly examples for reference, in a zero-shot context. Existing models struggle with substantial domain shifts because they rely on statistical correlations that break down when major changes occur in the distribution. Therefore, we propose a hierarchical causality-guided framework (EDSR) that exploits the invariance of causal relationships; the physical laws remain constant regardless of magnitude scaling. EDSR integrates causal discovery, causality-aware contrastive encoding, gated attention refinement, adversarial manifold alignment, memory-based edge screening, and conditional domain adaptation into a unified pipeline that is resilient to 100× domain shifts. This hierarchical edge-cloud framework filters 85% of normal data at the edge in under 2 ms, and with end-to-end latency under 26 ms. Empirical evaluation demonstrates performance on synthetic benchmark datasets: VAR-100 and Lorenz-96. The EDSR under 100× magnitude shifts achieved F1 scores of 35.86% and 45.86%, respectively, outperforming baseline CAROTS method by 186% and 341%. Ablation studies revealed the critical role of adversarial refinement and attention mechanisms, particularly for nonlinear chaos. This study establishes causality as a foundational primitive for zero-shot anomaly detection in safety-critical deployments.

1 INTRODUCTION

Distributed networks record correlated signals to characterize normal operating behavior; for example, Multivariate Time Series (MTS) networks underlie several safety critical infrastructures, assuming statistical distribution during deployment and training are same. This assumption fails when operating scale of the system changes unpredictably. A distributed network calibrated under low-load conditions exposed to several orders of magnitude higher loads, named as *extreme domain shift*.

Extreme domain shifts are common whenever the infrastructure scales rapidly. A representative case of extreme domain shifts is crowd monitoring during mass gatherings where a system trained on sparsely occupied venues must operate once occupancy grows, without retraining. The Kumbh Mela pilgrimage Verma & Bains (2025) illustrated this challenge, attracting 660 million visitors over 45 days with peaks of more than 10 million on a single day. At such scale, labeled anomalies are unavailable, requiring models to generalize without supervision, referred as *zero-shot extreme domain shift*.

The authors (Su et al., 2019; Xu et al., 2022) proposed standard reconstruction-based methods to handle extreme domain shifts that memorize statistical patterns tied to a specific range and degrade once the input distribution moves outside its range. In (Shen et al., 2018; Zamanzadeh Draban et al., 2025), authors illustrated domain adaptation methods assuming source and target can be reconciled through feature normalization, which fails when scale changes without overlapping support.

In extreme domain shift conditions, statistical patterns rearrange without bound but the underlying causality explained by physics does not change (Pearl, 2009). Under a magnitude change $\mathbf{X}' = \alpha\mathbf{X}$,

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.

$\alpha \gg 1$, where $\mathbf{X}$, $\mathbf{X}'$ represent raw MTS data of source and target domain respectively and α represents magnitude scaling factor. Statistical patterns such as "variable i at 50 implies variable j at 75" may not generalize, while a causal relation "variable i increasing causes variable j to increase" can remain invariant to scale. For $Y = f(X) + \epsilon$, if f is homogeneous then $\Delta Y/\Delta X$ is preserved under scaling a property referred as *causal invariance*.

The proposed ESDR framework uses causal invariance to work under extreme domain shifts. The major contributions are as follows:

1. We propose a novel framework, that grounds anomaly detection in learned causal structure rather than absolute signal magnitude, enabling a single trained model to operate across deployment scales without target-domain labels (Section 4).
2. We handle magnitude shifts in multivariate causal systems by developing a formal notion of causal invariance, together with a structural argument establishing transitive causal closure as a requirement for robustness of a system under variable failure (Section 4.1.1).
3. We validate EDSR on synthetic benchmarks with known ground-truth causal graphs (VAR-100, Lorenz-96) which outperformed the baseline CAROTS method by 186% and 341% in F1 under 100× shift respectively (Section 5).
4. We also introduce a hierarchical edge-cloud architecture that filters 85% of normal data at the edge in under 2 ms with end-to-end latency under 26 ms (Section 5).
5. We identify the top-3 contributing variables via per-feature reconstruction divergence, providing actionable root-cause attribution for each detected anomaly (Section 5.3).

2 Related Work

In state-of-the-art methods for anomaly detection in time series data several techniques were proposed including Deep SVDD (Ruff et al., 2018) which defines anomalies as deviations from a learned hypersphere; however, an unstructured latent space provides weaker separability than a causally-organized embedding. A number of reconstruction-based methods (OmniAnomaly (Su et al., 2019), AnomalyTransformer (Xu et al., 2022) and NeuTraL AD (Qiu et al., 2025), Fusionfomer (Wang et al., 2025)) fail under distribution shift because learned patterns are tied to specific magnitude ranges, whereas GPT-AD (Li & Yu, 2025) provides zero-shot detection via LLM priors but lacks causal grounding and cannot attribute anomalies at the sensor level. In CAROTS (Kim et al., 2025), authors adapt causality-aware contrastive learning, but use single-stage cloud-only processing without extreme domain shift adaptation.

All domain adaptation methods assume *alignability*: the source and target distributions must have overlapping support after normalization. The authors of AdaTime (Ragab et al., 2023) provide a systematic evaluation benchmark for time-series domain adaptation. Exisiting methods such as CoTMix (Eldele et al., 2024) applies temporal mixup to bridge domains whereas VRADA (Shen et al., 2018) minimizes the wasserstein distance between source and target distributions; DACAD (Zamanzadeh Draban et al., 2025) uses KL-divergence gating for conditional feature alignment. Under 100× magnitude shifts, distributions become disjoint making forced normalization counterproductive.

The PC algorithm (Pearl, 2009) tests conditional independencies that remain valid under extreme domain shifts but scales as $\mathcal{O}(N^4)$ and struggles with non-Gaussian data. Existing improvements, such as NOTEARS (Zheng et al., 2018) optimize a continuous DAG structure but require strong identifiability assumptions and perform poorly on temporal data. Neural temporal methods, including TCDF (Nauta et al., 2023) use attention-based CNNs, whereas Granger causality transformers (Marcinkevičs & Vogt, 2024) offer interpretable self-supervised alternatives. CUTS+ (Cheng et al., 2024) achieves leading Transitive Causal Fidelity (TCF) on high-dimensional irregular time series via a differentiable Neural ODE formulation.

Recent methodologies in cloud computing systems used for domain shift adaptation include Broad Learning Systems (BLS) (Chen & Liu, 2018) which learn compact representations without deep hierarchies, enabling fast on-device inference while AdaMemBLS (Zhong et al., 2025) augments BLS with an adaptive memory for streaming scenarios. In the edge-cloud architecture, EdgeAD (Zhang et al., 2023) uses a split-inference architecture that partitions model computation between the edge

and cloud. TinyAD (Liu et al., 2024) compresses anomaly detectors via quantization in resource-limited hardware.

3 DATASETS

In this section, we discuss two synthetic environments tha provide controlled testbeds with a known ground-truth causal structure, enabling direct validation of the causal discovery stage independent of downstream detection performance.

VAR-100 A 100-variable dataset formed from a Vector Autoregressive model environment (Sims, 1980), $\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \boldsymbol{\epsilon}_t$. This linear generative process is used because it yields an exact known ground-truth adjacency $\mathbf{A}$. The dataset spans 100,000 time steps with an 80/20 train-test split. Anomalies are injected by swapping causal coefficients over 50-timestep windows (structural anomalies), with 5% contamination of the test set (see Algorithm 2 in Appendix A.2 for details).

Lorenz-96 A 40-variable chaotic atmospheric dataset formed from the Lorenz environment (Lorenz, 1996), $\frac{dX_i}{dt} = (X_{i+1} - X_{i-2})X_{i-1} - X_i + F$, with $F = 8.0$ (normal) and $F \in \{6.0, 10.0, 13.0\}$ (anomalous). This dataset introduces nonlinear, chaotic dynamics. It spans 50,000 time steps with an 80/20 split, integrated via 4th-order Runge-Kutta with $dt = 0.05$. Anomalies are injected by perturbing the forcing parameter from 8.0 to 15.0, with 5% contamination (see Algorithm 3 in Appendix A.2 for details).

Domain shift protocol To mimic extreme domain shifts condition found in real time scenarios, test data is scaled as $\mathbf{X}_{\text{test}} \leftarrow \alpha \cdot \mathbf{X}_{\text{test}}$, $\alpha \in \{1, 100\}$; training is fixed at $\alpha = 1$ to ensure no data leakage (see Algorithm 4 in Appendix A.2 for details).

4 METHODOLOGY

In this section, we propose EDSR framework which comprises of offline training with six phases and an online inference hierarchical edge-cloud pipeline. The complete architecture is depicted in Figure 1.

4.1 OFFLINE TRAINING

Algorithm 1 Offline training phase.

Require: Normal data $\mathbf{X} \in \mathbb{R}^{T \times N}$ (T timestamps), sparsity weight λ
Ensure: Weights $\{\theta_{\text{enc}}, \theta_{\text{fam}}, \mathcal{D}, \mathcal{G}, \mathcal{E}\}$ (critic $\mathcal{D}$, reconstructor $\mathcal{G}$, refiner $\mathcal{E}$), memory $\mathbf{M}$, statistics $(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s)$

1: **phase 1: Causal Discovery.** Learn logits L via MPNN with penalty $\lambda\|L\|_1$; set graph $\mathbf{A} \leftarrow (\sigma(L) > 0.5)$
2: **phase 2: Contrastive Encoding.** For window $\mathbf{W}$: form $\mathbf{W}_+$ (noise σ=0.2) and $\mathbf{W}_-$ (bias $\pm 3\sigma$); minimize $\mathcal{L}_{\text{enc}} = \text{SupCon}(\phi_\theta(\mathbf{W}), \phi_\theta(\mathbf{W}_+), \phi_\theta(\mathbf{W}_-))$. Output embeddings $\mathbf{Z}_{\text{carots}}$
3: **phase 3: Attention Refinement.** Freeze θ_{enc}; compute $\mathbf{Z}_{\text{fam}} \leftarrow \text{LN}(\mathbf{Z}_{\text{carots}} + \gamma \cdot \text{SpatioTemporalAttn}(\mathbf{Z}_{\text{carots}}))$ with learnable scalar γ; minimize $\mathcal{L}_{\text{fam}} = \text{SupCon}(\mathbf{Z}_{\text{fam}})$
4: **phase 4: Adversarial Refinement.** Freeze $\theta_{\text{enc}}, \theta_{\text{fam}}$. *Stage 1*: MSE warmup until $\mathcal{L}_{\text{MSE}} < 0.5$. *Stage 2*: Train WGAN-GP ($\mathcal{D}, \mathcal{G}, \mathcal{E}$): $\mathcal{L}_{\mathcal{G}} = 5\|\hat{\mathbf{X}} - \mathbf{X}\|^2 - \mathcal{D}(\hat{\mathbf{X}}) + 0.1\|\mathbf{Z}_{\text{carots}} - \mathbf{Z}_{\text{fam}}\|^2$
5: **phase 5: Memory Bank.** Compute edge memory bank $\mathbf{M} \leftarrow \text{KMeans}(\text{pool}(\mathbf{Z}_{\text{fam}}), K$=50)
6: **phase 6: Source Statistics.** Compute source domain parameters $\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s \leftarrow \text{mean}(\mathbf{Z}_{\text{fam}}), \text{std}(\mathbf{Z}_{\text{fam}})$

4.1.1 PHASE 1: CAUSAL DISCOVERY

We introduce Algorithm 1 that describes the six-phase offline pipeline. In phase 1, a neural ODE (Ordinary Differential Equation) based causal discovery algorithm (see Agorithm 5 in Appendix A.3

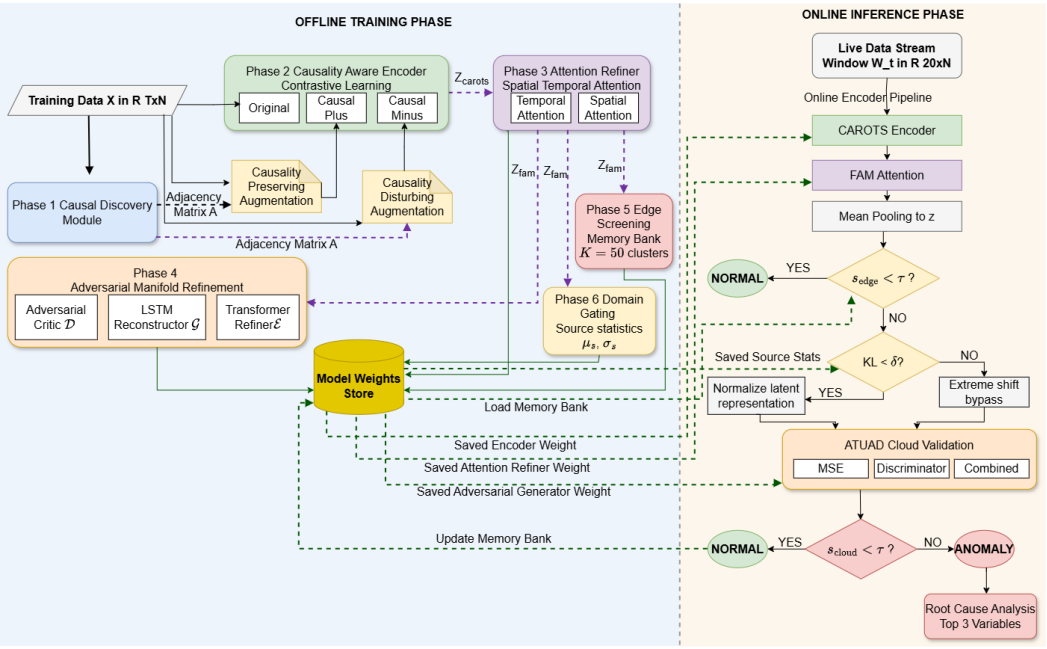


Figure 1: A six-phase of proposed hierarchical architecture of EDSR. **Offline:** phase 1 discovers $\mathbf{A}$ via Neural ODE; phase 2 and phase 3 learn causality-aware embeddings with gated attention; phase 4 trains the adversarial manifold refiner via WGAN-GP; phase 5 and phase 6 build the memory bank (K=50) and fit source statistics. **Online:** windows flow through CAROTS→FAM, edge screening (<2 ms, 85% clearance), KL-conditioned gating, and ATUAD cloud validation (<22 ms) with Top-3 root-cause attribution.

for details) learns a DAG $G{=}(V, E)$ from $\mathbf{X}$, where V, E represent vertices and edges respectively. Every edge of this graph is parameterized by a logit L_{ij} at each step also Gumbel noise is added and passed through a temperature-scaled sigmoid. This forms $\mathbf{A}_{\text{soft}} \in [0,1]^{N\times N}$ where N represents number of variables of dataset, which weights a message-passing forecasting network. The joint loss combines forecasting MSE with L_1 sparsity on $\mathbf{A}_{\text{soft}}$. At convergence, a adjacency matrix $\mathbf{A} = (\sigma(L) > 0.5)$ is formed.

Evaluation of causal discovery uses Transitive Causal Fidelity (TCF), which computes precision and recall against the transitive closure of the ground-truth graph. Transitive causal closure occures when an algorithm infers $A \to C$ while the true path is $A \to B \to C$ where A, B, C are variables. The inferred edge still carries valid information if B fails.

4.1.2 Phases 2–3: Causality-Aware Encoding and Attention Refinement

Causality-Aware Encoder: Sliding windows $\mathbf{W}_t \in \mathbb{R}^{L\times N}$ (L=20) are encoded via multi-scale parallel 1D convolutions (group normalization, N groups) followed by a 2-layer LSTM (128 units), producing base embeddings $\mathbf{Z}_{\text{carots}} \in \mathbb{R}^{B\times L\times 128}$. By perturbing ancestral nodes with $\mathcal{N}(0, 0.2)$, we construct causality-preserving augmentations, while causality-disturbing augmentations are generated by applying $\pm 3\sigma$ bias to root-descendant subgraphs (see Agorithm 6 in Appendix A.4 for details).

The supervised contrastive objective pulls the anchor $\mathbf{z}$ toward its causality-preserving view $\mathbf{z}_+$ and pushes it away from the causality-disturbing view $\mathbf{z}_-$:

$$\mathcal{L}_{\text{enc}} = \frac{1}{B}\sum_{b=1}^{B}\left(\max\left(0,\ \|\mathbf{z}_b - \mathbf{z}_b^+\|_2 - \|\mathbf{z}_b - \mathbf{z}_b^-\|_2 + m\right) + \varrho\|\mathbf{z}_b - \mathbf{z}_b^+\|_2^2\right), \tag{1}$$

where margin m=1.0 with compactness weight ϱ=0.5 is used for phase 2, while m=2.0 with ϱ=0.8 is adopted for phase 3, reflecting the tighter boundary separation and stronger cluster compactness required after attention calibration.

FAM attention Refiner: Embeddings are refined using dual spatial-temporal attention:

$$\mathbf{Z}_{\text{temp}} = \text{LayerNorm}\left(\mathbf{Z}_{\text{carots}} + \text{MultiHead}_{\text{temp}}(\mathbf{Z}_{\text{carots}}, 8 \text{ heads})\right), \tag{2}$$

$$\mathbf{Z}_{\text{spat}} = \text{LayerNorm}\left(\mathbf{Z}_{\text{temp}} + \text{MultiHead}_{\text{spat}}(\mathbf{Z}_{\text{temp}}, 8 \text{ heads})\right), \tag{3}$$

$$\mathbf{Z}_{\text{fam}} = \mathbf{Z}_{\text{carots}} + \gamma \cdot \text{LayerNorm}\left(\mathbf{Z}_{\text{spat}} + \text{MLP}(\mathbf{Z}_{\text{spat}})\right). \tag{4}$$

The scalar gate γ (initialized to 0) preserves an identity mapping at initialization, and contribute gradually as training progresses and a dropout of 0.2 is applied to the attention layers to prevent overfitting on frozen representations of $\mathbf{Z}_{\text{carots}}$ (see Algorithm 7 in Appendix A.5 for details).

4.1.3 Phase 4: Adversarial Manifold Refinement

A 3-layer Transformer refiner $\mathcal{E}$ (128-dim, 8 heads) maps $\mathbf{Z}_{\text{fam}} \to \mathbf{Z}_{\text{enc}}$; a 2-layer BiLSTM $\mathcal{G}$ reconstructs $\hat{\mathbf{X}}$; and a 3-layer MLP critic $\mathcal{D}$ distinguishes real from refined embeddings. Here, two-staged training is implemented, where, stage 1 warms up $\mathcal{G}$ on MSE until $\mathcal{L}_{\text{MSE}} < 0.5$ and then stage 2 applies WGAN-GP with 5D:1G updates. During training w_{adv} ramps from 0 to 0.5 over 50 epochs. The gradient penalty term in stage 2 of training enforces the Lipschitz constraint on $\mathcal{D}$:

$$\text{GP} = \lambda_{\text{gp}}\, \mathbb{E}_{\tilde{\mathbf{z}}}\left[\left(\|\nabla_{\tilde{\mathbf{z}}}\, \mathcal{D}(\tilde{\mathbf{z}})\|_2 - 1\right)^2\right], \tag{5}$$

where $\tilde{\mathbf{z}}$ is sampled uniformly along straight lines between real embeddings $\mathbf{Z}_{\text{fam}}$ and refined embeddings $\mathbf{Z}_{\text{enc}}$, with λ_{gp}=10.

Therefore the anomaly score :

$$s_{\text{cloud}} = 0.5 \cdot \text{MSE}(\hat{\mathbf{X}}, \mathbf{X}) + 0.5 \cdot (1 - \mathcal{D}(\mathbf{Z}_{\text{enc}})), \tag{6}$$

increases as reconstruction quality degrades and the discriminator flags the embedding as inconsistent with normal structure. A sustained elevation of the score above the calibrated threshold triggers the anomaly branch of the online inference pipeline (see Algorithm 8 in Appendix A.6 for details).

4.1.4 Phases 5–6: Edge Screening and Domain Gating

A memory bank $\mathbf{M} \in \mathbb{R}^{50\times128}$ is built via K-means on normal training embeddings. Now at inference:

$$s_{\text{edge}} = 1 - \max_k \frac{\mathbf{z} \cdot \mathbf{M}_k}{\|\mathbf{z}\|\|\mathbf{M}_k\|}, \tag{7}$$

is calculated, windows with $s_{\text{edge}} < 0.1$ trigger an EMA prototype update: $\mathbf{M}_{k^*} \leftarrow 0.95\mathbf{M}_{k^*} + 0.05\mathbf{z}$, and other windows are checked for KL divergence between the incoming batch and stored source statistics $\mathcal{N}(\boldsymbol{\mu}_s, \text{diag}(\boldsymbol{\sigma}_s^2))$. Utilizing KL divergence (δ=1.0) values, we calculate:

$$\mathbf{z}_{\text{adapted}} = \begin{cases} (\mathbf{z} - \boldsymbol{\mu}_s)/\boldsymbol{\sigma}_s & \text{if KL} < \delta \\ \mathbf{z} & \text{if KL} \geq \delta, \end{cases} \tag{8}$$

for normalizing latent representations. Under extreme shift (KL $\geq \delta$), z-score normalization is bypassed, therefore, causal embeddings are assumed to remain meaningful through structural invariance. The source embedding distribution is modeled as a diagonal Gaussian $\mathcal{N}(\boldsymbol{\mu}_s, \text{diag}(\boldsymbol{\sigma}_s^2))$, which is a reasonable approximation because the WGAN-GP training in phase 4 explicitly compacts normal embeddings toward a structured manifold; empirically, per-dimension histograms of $\mathbf{Z}_{\text{fam}}$ on both VAR-100 and Lorenz-96 training sets show unimodal, approximately symmetric distributions, consistent with the Gaussian assumption (see Algorithm 9 in Appendix A.7 and Algorithm 12 in Appendix A.10 for details).

4.2 Online Inference and Holistic Scoring

In inference, each window passes through causal aware encoder, attention refiner, and edge screening in under 2 ms. If flagged as an anomaly at the edge, the window passes through domain gating, and

cloud validation. Here, we fuse the five normalized anomaly signals as follows:

$$z_i = \frac{s_i - \mu_i^{\text{cal}}}{\sigma_i^{\text{cal}} + \epsilon} \quad \text{where } i \in \{\text{Z-distance}, \text{reconstruction}, \text{memory}, \text{discriminator}, \text{Z-velocity}\},$$

$$s_{\text{Z-distance}} \leftarrow \|\mathbf{z} - \bar{\mathbf{z}}_{\text{normal}}\|_2, \qquad s_{\text{reconstruction}} \leftarrow \|\hat{\mathbf{X}} - \mathbf{W}_{\text{norm}}\|^2,$$

$$s_{\text{memory}} \leftarrow 1 - \max_k \left(\frac{\mathbf{z} \cdot \mathbf{M}_k}{\|\mathbf{z}\|\|\mathbf{M}_k\|} \right), \quad s_{\text{discriminator}} \leftarrow \mathcal{D}(\mathbf{Z}_{\text{enc}}), \quad s_{\text{Z-velocity}} \leftarrow \|\mathbf{Z}_t - \mathbf{Z}_{t-1}\|_2, \tag{9}$$

$$\mathcal{S} = 1.5|z_{\text{Z-distance}}| + 1.0|z_{\text{reconstruction}}| + 0.5|z_{\text{memory}}| + 1.0\max(0, -z_{\text{discriminator}}) + 0.2|z_{\text{Z-velocity}}|.$$

The scalar weights $(1.5, 1.0, 0.5, 1.0, 0.2)$ were selected via grid search over the validation split, sweeping each weight independently in $\{0.2, 0.5, 1.0, 1.5, 2.0\}$ and retaining the configuration that maximized AUROC on held-out normal and anomalous windows. Z_{distance} captures global context drift (highest weight); $s_{\text{reconstrcution}}$ indicates causal structure shifts; s_{memory} indicates anomalousness against normal prototypes; $s_{\text{discriminator}}$ is applied when the critic detects a "fake" input; Z_{velocity} indicates temporal jitter and is weighted lowest to reduce sensitivity.

Here, the temporal smoothing filter suppresses single-window noise spikes where a window is flagged anomalous if $\mathcal{S}_t > \tau$ persists for at least 3 consecutive windows where threshold τ=0.85 is calibrated on the validation set (99th-percentile) (see Algorithm 11 in Appendix A.9 for details).

5 Results and Discussion

In this section, we analyze the results obtained when all the models share identical train-test splits, window length (l=20), latent dimension (128), and threshold calibration (99th-percentile validation) with hardware support of NVIDIA RTX 4050 (6 GB VRAM) with total training time of ≈12 h.

Table 1: Performance under 1× and 100× domain shift.

Dataset	Model	Scale	AUROC	F1 (%)	Latency (ms)
VAR-100	USAD	1×	0.48	6.69	0.65
	CAROTS	1×	0.64	**29.42**	1.39
	EDSR	1×	**0.75**	29.29	25.64
	USAD	100×	0.51	7.15	0.54
	CAROTS	100×	0.61	12.52	0.76
	EDSR	100×	**0.72**	**35.86**	25.77
Lorenz-96	USAD	1×	**0.77**	37.87	0.59
	CAROTS	1×	0.45	10.42	0.69
	EDSR	1×	0.75	**42.83**	10.69
	USAD	100×	0.77	38.79	0.56
	CAROTS	100×	0.47	10.39	0.70
	EDSR	100×	**0.79**	**45.86**	10.40

As shown in Table 1 the EDSR maintains an AUROC of 0.72 under 100× shift (3% drop from 1×), while F1 improves from 29.29% to 35.86% on VAR-100, whereas, CAROTS F1 collapses from 29.42% to 12.52%. On Lorenz-96, CAROTS fails under 100× shift with an AUROC score of 0.47, F1 10.39%. Our model achieves an AUROC of 0.79, F1 45.86%, a 341% F1 improvement over CAROTS. USAD shows numerical stability on Lorenz-96 but lacks causal grounding and root-cause attribution. The F1 improvement under 100× shift on VAR-100, from 29.29% to 35.86% is counterintuitive and warrants explanation. Anomalies in VAR-100 are injected as swaps of causal coefficients, which alter the structure of inter-variable relationships rather than their magnitude. Under α=100, signal magnitudes scale uniformly, but the additive noise floor $\boldsymbol{\epsilon}$ does not allow the signal-to-noise ratio to increase. Therefore, causal structure violations produce larger absolute deviations in the latent space relative to the noise baseline, making anomalous embeddings

more separable from normal ones. AUROC remains stable from 0.75 to 0.72 because the ranking of anomaly scores is largely preserved; F1 improves because the margin between normal and anomalous scores widens, allowing the fixed percentile threshold to capture more true positives. EDSR latencies (10–26 ms) remain within the 26 ms real-time requirement because edge filtering handles 85% of inferences in under 2 ms.

5.1 Ablation Study

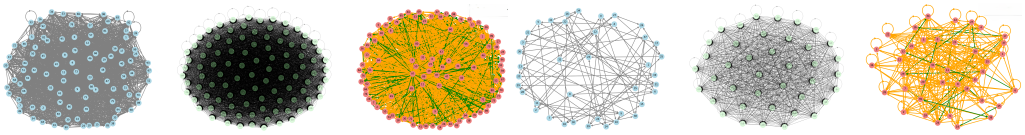
Table 2: Impact on AUROC with variation of EDSR using VAR-100 and Lorenz-96.

Configuration	**VAR-100**	**Lorenz-96**
Full EDSR	0.7539	0.7492
w/o Attention	0.7635 (+1.3%)	**0.4816** (−35.7%)
w/o Memory	0.7546 (+0.1%)	0.7492 (±0.0%)
w/o Adversarial Critic	**0.6243** (−17.2%)	**0.4782** (−36.2%)

We observed three key findings from Table 2. First, the adversarial critic is the most critical component: removing it causes the largest degradation observed (17.2% on VAR-100, 36.2% on Lorenz-96), which confirms that adversarial training sharpens the normal anomalous boundary beyond what is provided by reconstruction loss alone. Second, the attention is chaos-specific: removing the attention module is slightly beneficial for linear VAR-100 (+1.3%) but catastrophic for chaotic Lorenz-96 (−35.7%), indicating that nonlinear spatio-temporal dependencies require attention to resolve nonlinear spatio-temporal relationships that linear systems do not require. Third, removing the memory bank leaves AUROC unchanged because memory contributes to latency efficiency rather than detection accuracy.

5.2 Phase-wise Representation Quality

Causal discovery graphs, as shown in Figure 2, detect the majority of causal relationships for the Lorenz-96 dataset and successfully recover transitive closures. Here, the number of candidate edges is approximately 1,600, and the top 390 candidates are selected, where only 120 direct edges are present among 40 chaotic variables, which represents strong recovery performance. However, VAR-100 is a more challenging dataset because of its large number of edges: the model must detect 2,043 direct edges among 100 variables. CUTS+ nonetheless learns approximately 10,000 candidate edges and selects the top 2,043. These graphs serve as the structural foundation for the causality-guided contrastive augmentations in phase 2.

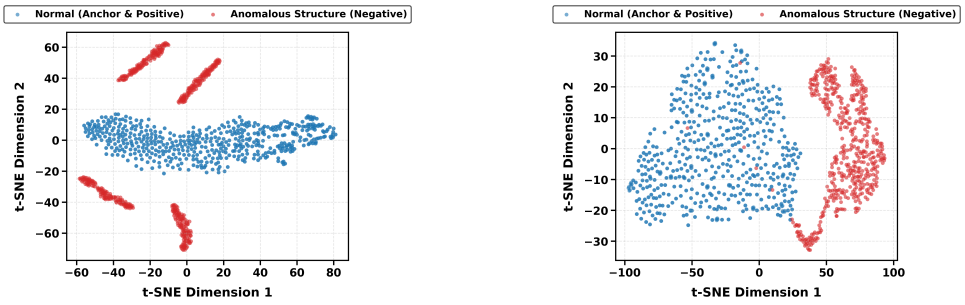


(a) VAR-100: 2043 true edges. (b) Lorenz-96: 390 true edges.

Figure 2: A representation of CUTS+ causal graph discovery in phase 1. Each panel (left to right): ground-truth graph, all learned edges, thresholded prediction with edge-type coloring (green = true positive, orange = valid transitive, red = false positive).

Figure 3 shows the t-SNE projections of VAR-100 and Lorenz-96 after being processed by causality aware encoder. For VAR-100, normal and abnormal time windows form distinct clusters, indicating that the linear causal relationship is well captured by the contrastive encoder whereas in Lorenz-96, separation occurs but mixing persists at the cluster boundary.

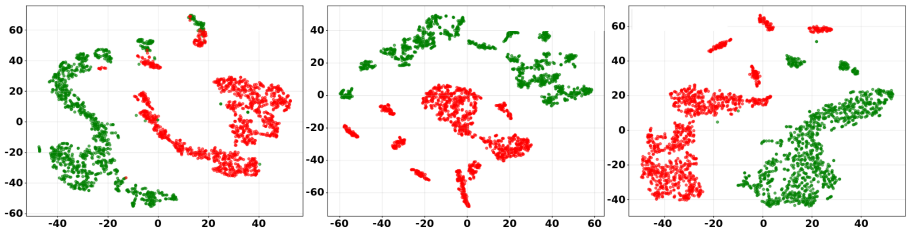
The latent-space quality across phases 2-4 is shown in Figure 4. Here, gated attention refinement in phase 3 improves silhouette scores from 0.367 $\rightarrow$ 0.508 on VAR-100 and 0.243 $\rightarrow$ 0.498 on Lorenz-96, confirming that nonlinear spatio-temporal dependencies require explicit attention to resolve the residual boundary. In phase 4, the adversarial acts on the manifold geometry, where, WGAN-GP



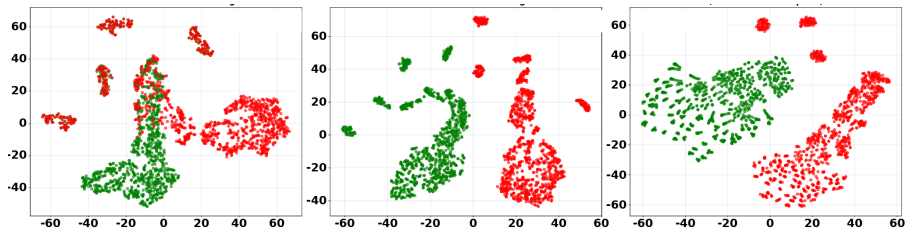
(a) t-SNE projection of phase 2 on VAR-100.

(b) t-SNE projection of phase 2 on Lorenz-96.

Figure 3: Representation of causal aware t-SNE embeddings of phase 2.



(a) VAR-100: phase 2→phase 3→phase 4.



(b) Lorenz-96: phase 2→phase 3→phase 4.

Figure 4: Representation of t-SNE embeddings across phases 2–4. Adversarial refinement compacts the normal manifold (green) and peripheralizes anomalies (red).

critic forces the transformer refiner $\mathcal{E}$ to map normal embeddings towards a compact region, while anomalous inputs are displaced to the periphery.

The loss convergence across all phases is indicated in Figure 5, where the contrastive loss of phase 2 converges within 30 epochs on VAR-100 and 45 epochs on Lorenz-96, consistent with the CosineAnnealingLR schedule. In phase 4, stage 1 MSE loss drops below 0.5 within the first 15 epochs and in stage 2 the WGAN-GP critic and generator losses then stabilize after approximately 10 adversarial epochs without mode collapse, indicating that the normal manifold is compactly aligned before threshold calibration begins.

5.3 End-to-End Pipeline: Anomaly Attribution Output

Figure 6 presents two representative anomaly cases from the online inference pipeline, where, Figure 6(a) show VAR-100 and Figure 6(b) show Lorenz-96. In both cases, the edge screening stage assigns $s_{\text{edge}} > \tau_{\text{edge}}$, correctly escalating the window to cloud validation. The cloud validator computes $\mathcal{S} > \tau_{\text{cloud}}$ (Section 4.2), confirming the anomaly decision. The per-feature reconstruction divergence is then decomposed to identify the top-3 contributing variables, providing actionable root-cause attribution alongside the binary detection output.

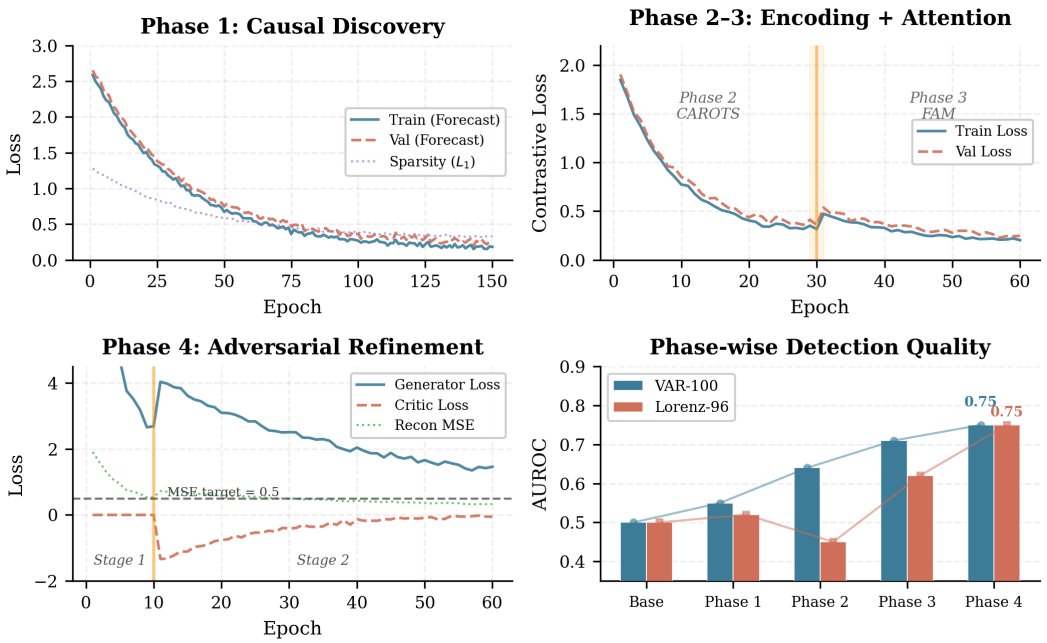
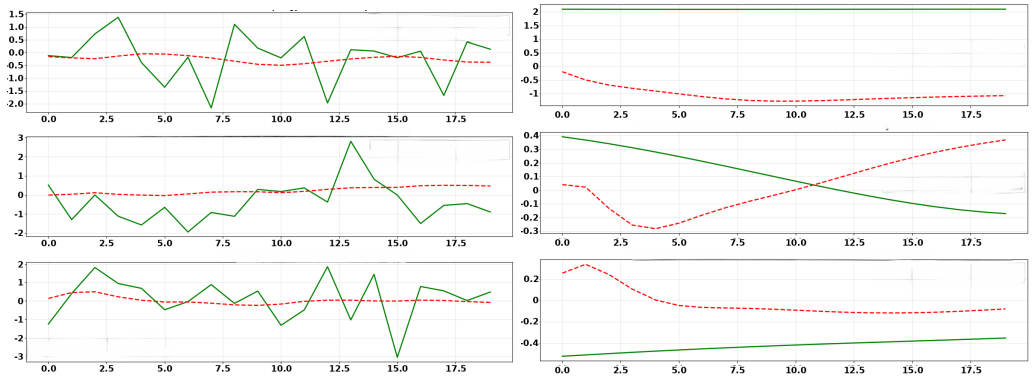


Figure 5: Training loss convergence across all offline phases.



(a) VAR-100 anomaly + Top-3 RCA.

(b) Lorenz-96 anomaly + Top-3 RCA.

Figure 6: End-to-end inference output indicating anomalous windows with Top-3 root-cause attribution. Green-Sensor data, Red-ATUAD reconstruction data.

6 Conclusion

We address the problem of anomaly detection under extreme domain shifts where the core contribution of this study is to ground the latent representation in a discovered causal graph rather than in magnitude-specific statistics by producing embeddings that preserve discriminability when signal scale shifts. The phase-wise analysis of our proposed framework EDSR establishes a clear progression, with causal contrastive encoding achieving clean separation on linear VAR-100 but leaving residual boundary mixing on chaotic Lorenz-96, which is then resolved using gated spatio-temporal attention, providing a quantitative signal that dual attention is specifically necessary for nonlinear dynamics. Further compaction of the latent space is achieved using adversarial manifold refinement, with an ablation study confirming its contribution and indicating that reconstruction loss alone is an insufficient discriminative signal. During inference, memory-based edge screening enables the identification of anomalous or safe windows in under 2 ms latency, while conditional domain gating uses KL divergence to distinguish normalizable windows from extreme shift windows, selectively routing latter to cloud based anomaly detection and root-cause attribution.

AI USE STATEMENT

Generative AI tools were used to assist with the implementation of the methods, brainstorming architectural variants, and polishing selected portions of the manuscript. These tools were not used to generate datasets, formulate theoretical models or mathematical claims, develop proofs, conduct or interpret experiments, perform literature review or information retrieval, or create or modify scientific figures.

All AI-assisted outputs were critically reviewed and independently verified by the authors. In particular, implementation-related suggestions were checked against the underlying methodology and experimental code, while generated or edited text was revised for technical accuracy and consistency with the reported results. No AI-generated claim was included without independent verification. The authors take full responsibility for the final content, claims, methodology, experimental results, and artifacts presented in this work.

ACKNOWLEDGMENTS

The authors sincerely acknowledge the Indian Institute of Information Technology Allahabad (IIIT-Allahabad) for providing the academic environment, resources, and support that enabled this research. We are grateful for the institutional guidance and facilities that contributed to the successful completion of this work and the preparation of this research paper.

REPRODUCIBILITY STATEMENT

To guarantee complete reproducibility of all experimental results, we have provided comprehensive documentation across the main text, appendix, and supplementary materials. Appendix A details exhaustive algorithmic pseudocode (Algorithms 2–13) and a unified notation table. Finally, a self-contained, turnkey code repository with fixed random seeds and, an automated checkpoint resumption, is provided in the supplementary material (`iclr_submission/`) to allow full one-click replication of all reported metrics.

REFERENCES

C. L. Philip Chen and Zhulin Liu. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):10–24, 2018.

Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts+: high-dimensional causal discovery from irregular time-series. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. Contrastive domain adaptation for time-series via temporal mixup. *IEEE Transactions on Artificial Intelligence*, 5(4):1185–1194, 2024.

HyunGi Kim, Jisoo Mok, Dongjun Lee, Jaihyun Lew, Sungjae Kim, and Sungroh Yoon. Causality-aware contrastive learning for robust multivariate time-series anomaly detection. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Wenhuan Li and Jiong Yu. Gptad: A generative pre-trained transformer for time-series anomaly detection via dual-path decomposition and statistical synthesis. In *2025 5th International Conference on Communication Technology and Information Technology (ICCTIT)*, pp. 508–512, 2025.

Yifan Liu, Hao Zhang, Dong Wang, Yi Zhou, and Zhenghua Chen. TinyAD: Quantization-aware training for edge anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, pp. 1–18. Reading, 1996.

Ričards Marcinkevičs and Julia E Vogt. Interpretable models for granger causality using self-explaining neural networks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning*, 112(7):2237–2290, 2023.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Chen Qiu, Marius Kloft, Stephan Mandt, and Maja Rudolph. Self-supervised anomaly detection with neural transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2170–2185, 2025.

Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli li. Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data*, 17:1–18, 05 2023.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 10–15 Jul 2018.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pp. 1–48, 1980.

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*, KDD '19, pp. 2828–2837, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

Ritika Verma and Harkirat Bains. Security and surveillance at large religious gatherings: Crowd management strategies for the 2025 maha kumbh mela. *Synergy: International Journal of Multidisciplinary Studies*, 2(3):47–57, Sep. 2025.

Chuang Wang, Zidong Wang, Hongli Dong, Stanislao Lauria, Weibo Liu, Yiming Wang, Futra Fadzil, and Xiaohui Liu. Fusionformer: A novel adversarial transformer utilizing fusion attention for multivariate anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):14479–14492, 2025.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022.

Zahra Zamanzadeh Draban, Yiyuan Yang, Geoffrey Webb, Charu 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*, 37:4485–4496, 01 2025.

Wei Zhang, Peng Zhao, Ao Li, Yizheng Jiang, and Feng Liu. EdgeAD: Edge-cloud collaborative anomaly detection for industrial IoT time series. *IEEE Internet of Things Journal*, 10(18):16102–16115, 2023.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric Xing. Dags with no tears: Continuous optimization for structure learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Zhijie Zhong, Zhiwen Yu, Ziwei Fan, C. L. Philip Chen, and Kaixiang Yang. Adaptive memory broad learning system for unsupervised time series anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 36(5):8331–8345, May 2025. ISSN 2162-2388.

A Appendix

This appendix provides the formal, self-contained mathematical and algorithmic specifications for all individual phases of the proposed framework, master training orchestration, offline calibration, and the unified online inference engine. It also visualizes the results of phase 4.

A.1 Notational Conventions and Dimensionality Table

To maintain rigorous mathematical consistency across all algorithmic definitions, Table 3 catalogs all primary symbols, operator definitions, and structural tensor dimensions.

Table 3: Comprehensive Notation and Tensor Dimensionality Catalog

Symbol	Domain / Shape	Physical / Mathematical Definition
T	$\mathbb{N}$	Total length of time-series observations in dataset
N	$\mathbb{N}$	Number of observed multivariate entities (sensor dimensions)
L	$\mathbb{N}$	Temporal sliding window length ($L = 20$ in standard baseline)
d_z	$\mathbb{N}$	Latent representation embedding dimension ($d_z = 128$)
B	$\mathbb{N}$	Mini-batch dimension
$\mathbf{X}$	$\mathbb{R}^{T\times N}$	Full multivariate time series continuous observations
$\mathbf{W}_t$	$\mathbb{R}^{L\times N}$	Sliding observation window $\mathbf{X}_{t-L+1:t}$
$\mathbf{A} \in \{0,1\}^{N\times N}$	$\{0,1\}^{N\times N}$	Discovered binary causal adjacency matrix ($A_{ij} = 1 \implies i \to j$)
$\mathcal{G}_{\text{DAG}}$	$(\mathcal{V}, \mathcal{E})$	Causal directed graph induced by adjacency matrix $\mathbf{A}$
$\mathbf{Z}_{\text{carots}}$	$\mathbb{R}^{L\times d_z}$	Latent sequence representation output by CAROTS encoder
$\mathbf{Z}_{\text{fam}}$	$\mathbb{R}^{L\times d_z}$	Spatially and temporally refined representation from FAM
$\bar{\mathbf{z}}$	$\mathbb{R}^{d_z}$	Temporally pooled latent vector: $\frac{1}{L}\sum_{\ell=1}^{L} \mathbf{Z}_{\text{fam}}[\ell, :]$
$\hat{\mathbf{W}}_t$	$\mathbb{R}^{L\times N}$	Reconstructed signal window from ATUAD generator $\mathcal{G}$
$(\hat{\boldsymbol{\mu}}, \log \hat{\boldsymbol{\sigma}}^2)$	$\mathbb{R}^{L\times N} \times \mathbb{R}^{L\times N}$	Parametric Gaussian reconstruction mean and log-variance
$\mathcal{D}(\cdot)$	$\mathbb{R}^{d_z} \to \mathbb{R}$	Adversarial critic score (Wasserstein manifold alignment logit)
$\mathbf{M}$	$\mathbb{R}^{K\times d_z}$	AdaMemBLS memory prototype dictionary (K nominal centroids)
$\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s$	$\mathbb{R}^{N}$	Empirical mean and standard deviation anchored on training data
$\mathbf{z}_{\text{cent}}$	$\mathbb{R}^{d_z}$	Empirical centroid of nominal training latents
$\boldsymbol{\mu}^{\text{cal}}, \boldsymbol{\sigma}^{\text{cal}}$	$\mathbb{R}^{5}$	Empirical mean and standard deviation per score component on calibration data
$s_{\text{Z-dist}}$	$\mathbb{R}_{\geq 0}$	Euclidean distance of sample latent to nominal centroid: $\|\bar{\mathbf{z}} - \mathbf{z}_{\text{cent}}\|_2$
s_{recon}	$\mathbb{R}_{\geq 0}$	Signal reconstruction negative log-likelihood or squared error
s_{mem}	$[0, 2]$	AdaMemBLS memory surprise: $1 - \max_k(\bar{\mathbf{z}} \cdot \mathbf{M}_k / (\|\bar{\mathbf{z}}\| \|\mathbf{M}_k\|))$
s_{disc}	$\mathbb{R}$	Adversarial discriminator authenticity score: $\mathcal{D}(\mathbf{Z}_{\text{ref}})$
$s_{\text{Z-vel}}$	$\mathbb{R}_{\geq 0}$	Latent trajectory velocity / temporal first-difference: $\frac{1}{L-1}\sum_{\ell} \|\mathbf{Z}[\ell] - \mathbf{Z}[\ell-1]\|_2$
z_i	$\mathbb{R}$	Calibrated standardized score: $(s_i - \mu_i^{\text{cal}})/(\sigma_i^{\text{cal}} + \epsilon)$
$\mathcal{S}_{\text{ens}}$	$\mathbb{R}_{\geq 0}$	Exact multi-criteria fused anomaly score with calibrated weights
τ	$\mathbb{R}$	Anomaly classification decision threshold
$\hat{y}_t$	$\{0, 1\}$	Predicted binary anomaly label at time index t

A.2 Synthetic Physical Environments and Anomaly Injection Suite

This section formalizes the generative physical processes for Vector Autoregression (VAR-100), Lorenz-96 nonlinear atmospheric chaos, and the 5-part academic anomaly injection suite.

Algorithm 2 Vector Autoregression (VAR-100) Generative Process

Require: Dimension $N = 100$, observation length $T = 100{,}000$, graph sparsity $s = 0.2$, noise magnitude $\sigma_\epsilon = 0.1$, spectral threshold $\rho_{\max} = 0.95$.
Ensure: Multivariate observation matrix $\mathbf{X} \in \mathbb{R}^{T \times N}$, ground-truth DAG adjacency $\mathbf{A} \in \{0,1\}^{N \times N}$.
1: Sample random Erdős-Rényi structure: $\mathbf{A}[i,j] \sim \text{Bernoulli}(s)$ for $i, j \in \{1, \ldots, N\}$.
2: Sample random coefficients: $\mathbf{W}[i,j] \sim \text{Uniform}(-0.5, 0.5) \cdot \mathbf{A}[i,j]$.
3: Evaluate spectral radius: $\rho(\mathbf{W}) = \max_k |\lambda_k(\mathbf{W})|$.
4: **if** $\rho(\mathbf{W}) \geq 1.0$ **then**
5: Rescale coefficient matrix for asymptotic stationarity: $\mathbf{W} \leftarrow \frac{\mathbf{W}}{\rho(\mathbf{W})+0.05} \cdot \rho_{\max}$.
6: **end if**
7: Initialize $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; execute burn-in iterations for $t = 1 \ldots 1{,}000$.
8: **for** $t = 1$ **to** T **do**
9: Sample independent Gaussian noise: $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$.
10: Propagate linear causal dynamics: $\mathbf{X}[t,:] = \mathbf{W}\mathbf{X}[t-1,:]^\top + \boldsymbol{\epsilon}_t$.
11: **end for**
12: **return** Data $\mathbf{X}$, causal adjacency matrix $\mathbf{A}$.

Algorithm 3 Lorenz-96 Atmospheric Chaotic System via 4th-Order Runge-Kutta (RK4)

Require: Number of variables $N = 40$, timesteps $T = 50{,}000$, forcing constant $F = 8.0$, step size $\Delta t = 0.01$.
Ensure: Chaotic time series $\mathbf{X} \in \mathbb{R}^{T \times N}$, cyclic ground-truth adjacency $\mathbf{A} \in \{0,1\}^{N \times N}$.
1: Construct ground-truth cyclic adjacency matrix:

$$\mathbf{A}[i, (i-1) \pmod N] = 1, \quad \mathbf{A}[i, (i-2) \pmod N] = 1, \quad \mathbf{A}[i, (i+1) \pmod N] = 1 \tag{10}$$

2: Define governing differential equation for node $i \in \{0, \ldots, N-1\}$:

$$f_i(\mathbf{x}) = (x_{(i+1) \pmod N} - x_{(i-2) \pmod N}) \cdot x_{(i-1) \pmod N} - x_i + F \tag{11}$$

3: Initialize state with small perturbation: $\mathbf{x}_0 = F + \boldsymbol{\delta},\ \boldsymbol{\delta} \sim \text{Uniform}(0,1)^N$.
4: Discard initial transient chaotic burn-in over 2,000 integration steps.
5: **for** $t = 1$ **to** T **do**
6: Compute RK4 intermediate tangent slopes:

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}_t) \tag{12}$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{x}_t + \frac{\Delta t}{2}\mathbf{k}_1\right) \tag{13}$$

$$\mathbf{k}_3 = \mathbf{f}\left(\mathbf{x}_t + \frac{\Delta t}{2}\mathbf{k}_2\right) \tag{14}$$

$$\mathbf{k}_4 = \mathbf{f}\left(\mathbf{x}_t + \Delta t\, \mathbf{k}_3\right) \tag{15}$$

7: Update continuous atmospheric state:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{16}$$

8: Assign $\mathbf{X}[t,:] \leftarrow \mathbf{x}_{t+1}$.
9: **end for**
10: **return** Data $\mathbf{X}$, ground-truth causal graph $\mathbf{A}$.

Algorithm 4 Multi-Type Academic Anomaly Injection Suite

Require: Nominal continuous time series $\mathbf{X} \in \mathbb{R}^{T \times N}$, target anomaly ratio $r = 0.01$, test split ratio $r_{\text{test}} = 0.30$.

Ensure: Anomaly-contaminated series $\mathbf{X}_{\text{mod}} \in \mathbb{R}^{T \times N}$, ground-truth binary label vector $\mathbf{y} \in \{0, 1\}^T$.

1: Initialize labels $\mathbf{y} \leftarrow \mathbf{0}_T$; copy observations $\mathbf{X}_{\text{mod}} \leftarrow \mathbf{X}$.

2: Restrict perturbations strictly to test split: $t_{\text{start}} \leftarrow \lfloor (1 - r_{\text{test}}) \cdot T \rfloor$.

3: Total anomaly budget: $M_{\text{anom}} \leftarrow \lfloor T \cdot r \rfloor$; allocate $M_k \leftarrow \lfloor M_{\text{anom}}/5 \rfloor$ timesteps per type.

4: **1. Point Anomalies:** Sample M_1 indices $t \in [t_{\text{start}}, T]$ and variables $v \in \{1, \ldots, N\}$:

$$\mathbf{X}_{\text{mod}}[t, v] \leftarrow \mathbf{X}_{\text{mod}}[t, v] + \text{Sign} \cdot 5 \cdot \text{std}(\mathbf{X}[:, v]), \quad y_t \leftarrow 1 \tag{17}$$

5: **2. Contextual Anomalies:** Sample segments $[s, s + \ell] \subset [t_{\text{start}}, T]$, variable v:

$$\mathbf{X}_{\text{mod}}[s : s + \ell, v] \leftarrow \mathbf{X}_{\text{mod}}[s : s + \ell, v] \cdot c, \; c \in \{0.1, 2.5\}, \quad \mathbf{y}[s : s + \ell] \leftarrow 1 \tag{18}$$

6: **3. Shapelet Anomalies:** Inject square-wave patterns over test segments $[s, s + \ell]$:

$$\mathbf{X}_{\text{mod}}[s : s + \ell, v] \leftarrow \mathbf{X}_{\text{mod}}[s : s + \ell, v] + 3 \cdot \text{std}(\mathbf{X}[:, v]), \quad \mathbf{y}[s : s + \ell] \leftarrow 1 \tag{19}$$

7: **4. Trend Anomalies:** Inject continuous linear drift over test segments $[s, s + \ell]$:

$$\mathbf{X}_{\text{mod}}[s : s+\ell, v] \leftarrow \mathbf{X}_{\text{mod}}[s : s+\ell, v] + \text{linspace}(0, 5 \cdot \text{std}(\mathbf{X}[:, v]), \ell), \quad \mathbf{y}[s : s+\ell] \leftarrow 1 \tag{20}$$

8: **5. Seasonal Anomalies:** Inject high-frequency sinusoidal harmonic disturbances:

$$\mathbf{X}_{\text{mod}}[s : s+\ell, v] \leftarrow \mathbf{X}_{\text{mod}}[s : s+\ell, v] + 2 \cdot \text{std}(\mathbf{X}[:, v]) \cdot \sin(2\pi \cdot (0 : \ell)/5), \quad \mathbf{y}[s : s+\ell] \leftarrow 1 \tag{21}$$

9: **return** $\mathbf{X}_{\text{mod}}, \mathbf{y}$.

A.3 Phase 1: Differentiable Causal Discovery (CUTS+)

Phase 1 discovers the invariant causal DAG $\mathbf{A}$ governing normal physical dynamics using Neural Ordinary Differential Equations and continuous Gumbel-Softmax graph sampling.

Algorithm 5 Phase 1: Causal Graph Discovery via CUTS+

Require: Training normal time series $\mathbf{X}_{\text{train}} \in \mathbb{R}^{T_{\text{train}} \times N}$, sequence window length W_s, sparsity coefficient λ_s, epochs E_1, initial temperature τ_{start}, final temperature τ_{end}.
Ensure: Binary causal adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$.
1: Initialize edge weight logits $\mathbf{L} \in \mathbb{R}^{N \times N}$ randomly; zero diagonal $\mathbf{L}_{ii} \leftarrow -\infty$.
2: Initialize Neural ODE forecasting parameters $\boldsymbol{\phi} = \{\mathbf{W}_{\text{enc}}, \mathbf{W}_{\text{dec}}\}$.
3: **for** epoch $= 1$ **to** E_1 **do**
4: Anneal temperature: $\tau \leftarrow \tau_{\text{start}} \cdot (\tau_{\text{end}}/\tau_{\text{start}})^{\text{epoch}/E_1}$.
5: **for** each mini-batch window $\mathbf{X}_b \in \mathbb{R}^{B \times W_s \times N}$ **do**
6: Sample i.i.d. Gumbel perturbations: $G_{ij}, G'_{ij} \sim \text{Gumbel}(0,1)$.
7: Compute continuous relaxed adjacency matrix via Gumbel-Softmax:

$$\mathbf{A}_{\text{soft}}[i,j] = \sigma\left(\frac{\mathbf{L}_{ij} + G_{ij} - G'_{ij}}{\tau}\right) \tag{22}$$

8: Forecast next-step temporal values $\hat{\mathbf{X}}_{t+1} \leftarrow \text{NeuralODE}(\mathbf{X}_b, \mathbf{A}_{\text{soft}}; \boldsymbol{\phi})$.
9: Compute multi-step prediction loss:

$$\mathcal{L}_{\text{pred}} = \frac{1}{B \cdot N} \sum_{b=1}^{B} \sum_{j=1}^{N} \left\| \mathbf{X}_{b,t+1}^{(j)} - \hat{\mathbf{X}}_{b,t+1}^{(j)} \right\|_2^2 \tag{23}$$

10: Formulate total causal discovery objective with L_1 DAG sparsity:

$$\mathcal{L}_{\text{CUTS}} = \mathcal{L}_{\text{pred}} + \lambda_s \sum_{i \neq j} |\mathbf{A}_{\text{soft}}[i,j]| \tag{24}$$

11: $\mathbf{L} \leftarrow \mathbf{L} - \eta_1 \nabla_{\mathbf{L}} \mathcal{L}_{\text{CUTS}}$
12: $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta_1 \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\text{CUTS}}$
13: **end for**
14: **end for**
15: Apply hard thresholding (or Top-K binarization if ground-truth density K is known):

$$\mathbf{A}[i,j] = \begin{cases} 1 & \text{if } \sigma(\mathbf{L}_{ij}) \geq 0.5 \quad (i \neq j) \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

16: **return** Binary causal adjacency matrix $\mathbf{A}$.

A.4 Phase 2: CAROTS Representation Learning

Phase 2 maps raw sensor traces to a causal-aware latent geometry using multi-scale convolutions and an asymmetric triplet contrastive formulation over DAG-guided augmentations.

Algorithm 6 Phase 2: CAROTS Representation Learning

Require: Training series $\mathbf{X}_{\text{train}}$, binary DAG $\mathbf{A}$, window length L, embed dim d_z, epochs E_2, learning rate η_2, margin m_1, varrho ϱ.
Ensure: Trained CAROTS encoder $\mathcal{F}_\theta : \mathbb{R}^{L\times N} \to \mathbb{R}^{L\times d_z}$.
Initialize CAROTS parameter set θ (grouped 1D Conv + Bidirectional LSTM).
Extract causal topological ordering and ancestor sets $\text{Anc}(v)$ from $\mathbf{A}$.
for epoch $= 1$ **to** E_2 **do**
  for each mini-batch of normal windows $\mathbf{W} \in \mathbb{R}^{B\times L\times N}$ **do**
    Generate Causality-Preserving Positive Augmentation $\mathbf{W}^+$:
    Select random subset of root causes $\mathcal{S}_{\text{root}} \subset \{1, \dots, N\}$.
    Perturb root causes: $\mathbf{W}^+[:,:,v] \leftarrow \mathbf{W}[:,:,v] + \boldsymbol{\epsilon}_v,\ \boldsymbol{\epsilon}_v \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\text{pos}}\mathbf{I})$.
    Propagate physical noise along graph: for each descendant $u \in \text{Desc}(v)$, $\mathbf{W}^+[:,:,u] \leftarrow \mathbf{W}^+[:,:,u] + 0.5 \cdot \boldsymbol{\epsilon}_v$.
    Generate Causality-Disturbing Negative Augmentation $\mathbf{W}^-$:
    Select root node v and its descendant set $\text{Desc}(v)$ from $\mathbf{A}$; sample random time steps $\{t_m\}$.

    Apply Relative Bias Transform violating physical invariant:
    For $u \in \{v\} \cup \text{Desc}(v)$: $\mathbf{W}^-[:,t_m,u] \leftarrow \mathbf{W}[:,t_m,u] + \beta \cdot \text{std}(\mathbf{W}[:,:,u]),\ \beta \sim \text{Uniform}(\{-3,-2,2,3\})$.
    Compute projected latent embeddings:

$$\mathbf{z} = \mathcal{F}_\theta(\mathbf{W}), \quad \mathbf{z}^+ = \mathcal{F}_\theta(\mathbf{W}^+), \quad \mathbf{z}^- = \mathcal{F}_\theta(\mathbf{W}^-) \tag{26}$$

    Compute normalized Euclidean pairwise distances:

$$d^+ = \left\| \frac{\mathbf{z}}{\|\mathbf{z}\|_2} - \frac{\mathbf{z}^+}{\|\mathbf{z}^+\|_2} \right\|_2, \quad d^- = \left\| \frac{\mathbf{z}}{\|\mathbf{z}\|_2} - \frac{\mathbf{z}^-}{\|\mathbf{z}^-\|_2} \right\|_2 \tag{27}$$

    Evaluate supervised contrastive loss:

$$\mathcal{L}_{\text{CAROTS}}(\theta) = \frac{1}{B}\sum_{b=1}^{B} \left(\max(0, d_b^+ - d_b^- + m_1) + \varrho(d_b^+)^2\right) \tag{28}$$

    Update parameters: $\theta \leftarrow \theta - \eta_2 \nabla_\theta \mathcal{L}_{\text{CAROTS}}(\theta)$.
  end for
  Decay learning rate via Cosine Annealing: $\eta_2 \leftarrow \eta_{\min} + \frac{1}{2}(\eta_2 - \eta_{\min})(1 + \cos(\pi \cdot \text{epoch}/E_2))$.
end for
return Optimized encoder $\mathcal{F}_\theta$.

A.5 Phase 3: FAM Cross-Sensor and Temporal Attention Calibration

Phase 3 freezes the base representation $\mathcal{F}_\theta$ and trains spatial-temporal attention with a learnable zero-initialized residual gate to model complex inter-sensor couplings.

Algorithm 7 Phase 3: Spatial-Temporal Attention Calibration (FAM)

Require: Frozen CAROTS encoder $\mathcal{F}_\theta$, training data $\mathbf{X}_{\text{train}}$, DAG $\mathbf{A}$, epochs E_3, attention heads $H = 8$, learning rate η_3, margin m_2, varrho ϱ.
Ensure: Calibrated Feature Attention Module Ψ_ω with gating parameter γ.
1: Initialize FAM parameters $\omega = \{\mathbf{W}_Q^t, \mathbf{W}_K^t, \mathbf{W}_V^t, \mathbf{W}_Q^s, \mathbf{W}_K^s, \mathbf{W}_V^s, \mathbf{W}_{\text{mlp}}\}$; initialize scalar $\gamma \leftarrow 0.0$.
2: Freeze θ: $\nabla_\theta \leftarrow \mathbf{0}$.
3: **for** epoch $= 1$ **to** E_3 **do**
4: **for** each mini-batch of normal windows $\mathbf{W} \in \mathbb{R}^{B\times L\times N}$ **do**
5: Extract base representations: $\mathbf{Z} \leftarrow \mathcal{F}_\theta(\mathbf{W}) \in \mathbb{R}^{B\times L\times d_z}$.
6: Compute temporal attention across windows:

$$\mathbf{Z}_{\text{temp}} = \text{LayerNorm}\left(\mathbf{Z} + \text{MHA}_{\text{temp}}(\mathbf{Z}, \mathbf{Z}, \mathbf{Z})\right) \tag{29}$$

7: Compute spatial attention across sensor dimensions:

$$\mathbf{Z}_{\text{spat}} = \text{LayerNorm}\left(\mathbf{Z}_{\text{temp}} + \text{MHA}_{\text{spat}}(\mathbf{Z}_{\text{temp}}, \mathbf{Z}_{\text{temp}}, \mathbf{Z}_{\text{temp}})\right) \tag{30}$$

8: Apply feed-forward transformation and gated residual connection:

$$\mathbf{Z}_{\text{fam}} = \mathbf{Z} + \gamma \cdot \text{LayerNorm}\left(\mathbf{Z}_{\text{spat}} + \text{MLP}(\mathbf{Z}_{\text{spat}})\right) \tag{31}$$

9: Compute temporally pooled vectors: $\bar{\mathbf{z}} \leftarrow \frac{1}{L}\sum_{\ell=1}^{L} \mathbf{Z}_{\text{fam}}[:, \ell, :]$.
10: Forward positive $\mathbf{W}^+$ and negative $\mathbf{W}^-$ windows to compute $\bar{\mathbf{z}}^+$ and $\bar{\mathbf{z}}^-$.
11: Compute contrastive loss:

$$\mathcal{L}_{\text{FAM}}(\omega, \gamma) = \frac{1}{B}\sum_{b=1}^{B}\left(\max(0, \|\bar{\mathbf{z}}_b - \bar{\mathbf{z}}_b^+\|_2 - \|\bar{\mathbf{z}}_b - \bar{\mathbf{z}}_b^-\|_2 + m_2) + \varrho\|\bar{\mathbf{z}}_b - \bar{\mathbf{z}}_b^+\|_2^2\right) \tag{32}$$

12: Update: $(\omega, \gamma) \leftarrow (\omega, \gamma) - \eta_3 \nabla_{(\omega,\gamma)} \mathcal{L}_{\text{FAM}}(\omega, \gamma)$.
13: **end for**
14: **end for**
15: **return** Calibrated module $\Psi_{\omega,\gamma}$.

A.6 Phase 4: ATUAD Adversarial Manifold Alignment

Phase 4 trains the multi-stage refiner: a Transformer encoder $\mathcal{E}_\psi$, a Bidirectional LSTM generator $\mathcal{G}_\rho$ producing Gaussian observation parameters $(\hat{\boldsymbol{\mu}}, \log \hat{\boldsymbol{\sigma}}^2)$, and a Wasserstein critic $\mathcal{D}_\xi$.

Algorithm 8 Phase 4: ATUAD Adversarial Manifold Alignment

Require: Frozen $\mathcal{F}_\theta$ and $\Psi_{\omega,\gamma}$, training data $\mathbf{X}_{\text{train}}$, epochs E_4, critic learning rate $\eta_{\mathcal{D}}$, generator learning rate $\eta_{\mathcal{G}}$, critic update ratio $n_{\text{critic}} = 5$.
Ensure: Optimized refiner $\mathcal{E}_\psi$, generator $\mathcal{G}_\rho$, and critic $\mathcal{D}_\xi$.

Compute training dataset empirical statistics $\boldsymbol{\mu}_s = \mathbb{E}[\mathbf{X}_{\text{train}}]$ and $\boldsymbol{\sigma}_s = \sqrt{\text{Var}(\mathbf{X}_{\text{train}}) + 10^{-8}}$.
Initialize Transformer refiner $\mathcal{E}_\psi$, BiLSTM generator $\mathcal{G}_\rho$, and Critic MLP $\mathcal{D}_\xi$.
for epoch $= 1$ **to** E_4 **do**
  Calculate dynamic adversarial weight: $\lambda_{\text{adv}} \leftarrow \min\left(0.5, \frac{\text{epoch}}{E_4} \cdot 0.5\right)$.
  for each mini-batch of normalized windows $\tilde{\mathbf{W}} = (\mathbf{W} - \boldsymbol{\mu}_s)/\boldsymbol{\sigma}_s$ **do**
    Extract frozen latent sequence: $\mathbf{Z}_{\text{fam}} \leftarrow \Psi_{\omega,\gamma}(\mathcal{F}_\theta(\mathbf{W}))$.
    — Critic Optimization Step —
    for $k = 1$ **to** n_{critic} **do**
      Generate refined latent sequence: $\mathbf{Z}_{\text{ref}} \leftarrow \mathcal{E}_\psi(\mathbf{Z}_{\text{fam}})$.
      Evaluate critic on nominal vs. refined representations:

$$\mathcal{L}_{\mathcal{D}}(\xi) = \mathbb{E}\left[\mathcal{D}_\xi(\bar{\mathbf{z}}_{\text{ref}})\right] - \mathbb{E}\left[\mathcal{D}_\xi(\bar{\mathbf{z}}_{\text{fam}})\right] + \lambda_{\text{gp}}\mathbb{E}\left[(\|\nabla_{\hat{\mathbf{z}}}\mathcal{D}_\xi(\hat{\mathbf{z}})\|_2 - 1)^2\right] \tag{33}$$

      $\xi \leftarrow \xi - \eta_{\mathcal{D}}\nabla_\xi\mathcal{L}_{\mathcal{D}}(\xi)$.
    end for
    — Generator and Refiner Optimization Step —
    $\mathbf{Z}_{\text{ref}} \leftarrow \mathcal{E}_\psi(\mathbf{Z}_{\text{fam}})$.
    Decode reconstructed sequence distribution: $(\hat{\boldsymbol{\mu}}, \log\hat{\boldsymbol{\sigma}}^2) \leftarrow \mathcal{G}_\rho(\mathbf{Z}_{\text{ref}})$.
    Compute reconstruction Negative Log-Likelihood (NLL):

$$\mathcal{L}_{\text{NLL}}(\psi,\rho) = \frac{1}{B \cdot L \cdot N}\sum_{b,\ell,n}\left(\frac{(\tilde{\mathbf{W}}_{b,\ell,n} - \hat{\boldsymbol{\mu}}_{b,\ell,n})^2}{\exp(\log\hat{\boldsymbol{\sigma}}^2_{b,\ell,n})} + \log\hat{\boldsymbol{\sigma}}^2_{b,\ell,n}\right) \tag{34}$$

    Formulate total generator adversarial objective:

$$\mathcal{L}_{\text{gen}}(\psi,\rho) = \mathcal{L}_{\text{NLL}}(\psi,\rho) - \lambda_{\text{adv}}\mathbb{E}\left[\mathcal{D}_\xi(\bar{\mathbf{z}}_{\text{ref}})\right] \tag{35}$$

    Update: $(\psi,\rho) \leftarrow (\psi,\rho) - \eta_{\mathcal{G}}\nabla_{(\psi,\rho)}\mathcal{L}_{\text{gen}}(\psi,\rho)$.
  end for
end for
return Refiner $\mathcal{E}_\psi$, Generator $\mathcal{G}_\rho$, Critic $\mathcal{D}_\xi$.

A.7 Phases 5–6: Memory Bank Construction and Domain Calibration

Phase 5 clusters nominal latent vectors into a dictionary of normal prototypes (AdaMemBLS), and phase 6 anchors domain calibration statistics and threshold boundaries.

Algorithm 9 Phases 5–6: Memory Bank Construction and Calibration

Require: Frozen pipeline models $\{\mathcal{F}_\theta, \Psi_{\omega,\gamma}\}$, nominal dataset $\mathbf{X}_{\text{train}}$, memory capacity $K = 50$, percentile cutoff $p_{\text{thresh}} = 99.5$.

Ensure: Memory prototype matrix $\mathbf{M} \in \mathbb{R}^{K \times d_z}$, normalization parameters $(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s)$, latent centroid $\mathbf{z}_{\text{cent}}$, threshold τ.

1: Compute empirical sensor-level parameters:

$$\boldsymbol{\mu}_s = \frac{1}{T_{\text{train}}} \sum_{t=1}^{T_{\text{train}}} \mathbf{X}_{\text{train}}[t,:], \quad \boldsymbol{\sigma}_s = \sqrt{\frac{1}{T_{\text{train}}} \sum_{t=1}^{T_{\text{train}}} (\mathbf{X}_{\text{train}}[t,:] - \boldsymbol{\mu}_s)^2 + 10^{-8}} \tag{36}$$

2: Initialize latent buffer $\mathcal{B} \leftarrow \emptyset$.
3: **for** each validation sliding window $\mathbf{W}_t \in \mathbb{R}^{L \times N}$ **do**
4: Compute pooled latent: $\bar{\mathbf{z}}_t = \frac{1}{L} \sum_{\ell=1}^{L} \Psi_{\omega,\gamma}(\mathcal{F}_\theta(\mathbf{W}_t))[\ell,:]$.
5: Add to buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup \{\bar{\mathbf{z}}_t\}$.
6: **end for**
7: Cluster latent representations: $\mathbf{M} \leftarrow \text{KMeans}(\mathcal{B}, \text{n_clusters} = K, \text{n_init} = 10)$.
8: Normalize memory prototypes: $\mathbf{M}[k,:] \leftarrow \frac{\mathbf{M}[k,:]}{\|\mathbf{M}[k,:]\|_2} \quad \forall k \in \{1, \ldots, K\}$.
9: Compute nominal latent centroid: $\mathbf{z}_{\text{cent}} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{z} \in \mathcal{B}} \mathbf{z}$.
10: Evaluate calibration scores over nominal validation set:
11: Collect scores $\{S_{\text{ens}}(t)\}_{t=1}^{T_{\text{val}}}$ using CompletePipeline (Algorithm 11).
12: Set decision threshold: $\tau \leftarrow \text{Percentile}(\{S_{\text{ens}}(t)\}, p_{\text{thresh}})$.
13: **return** $\mathbf{M}, \boldsymbol{\mu}_s, \boldsymbol{\sigma}_s, \mathbf{z}_{\text{cent}}, \tau$.

A.8 Master Orchestration Algorithm with Fault-Tolerant Checkpoint Resume

Algorithm 10 details the full orchestration sequence implemented in `main.py`, guaranteeing reproducibility and error recovery.

Algorithm 10 Master Training Orchestration with Resume Logic

Require: Dataset identifier $\mathcal{D}_{\text{name}} \in \{\text{var}, \text{lorenz}\}$, hyperparameter configuration Θ, checkpoint directory $\mathcal{P}_{\text{ckpt}}$.
Ensure: Fully trained and validated model suite ready for deployment.
Load execution journal: $\mathcal{J} \leftarrow \text{ReadStateJSON}(\mathcal{P}_{\text{ckpt}}/\text{pipeline_state.json})$.
if $\mathbf{X}_{\text{train}}$ does not exist at target path **then**
  Generate $\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{gt}}, \mathbf{A}_{\text{gt}} \leftarrow \text{DatasetGenerator}(\mathcal{D}_{\text{name}})$.
end if
— Execute phase 1: Causal Discovery —
if "phase1" $\notin \mathcal{J}$.completed_phases **then**
  $\mathbf{A} \leftarrow$ Algorithm 5$(\mathbf{X}_{\text{train}}, \Theta_{\text{cuts}})$.
  Save $\mathbf{A}$; append "phase1" to $\mathcal{J}$; save $\mathcal{J}$.
else
  Load cached graph $\mathbf{A} \leftarrow \text{Load}(\mathcal{P}_{\text{ckpt}}/\mathbf{A}.\text{npy})$.
end if
— Execute phase 2: CAROTS Encoder —
if "phase2" $\notin \mathcal{J}$.completed_phases **then**
  $\mathcal{F}_\theta \leftarrow$ Algorithm 6$(\mathbf{X}_{\text{train}}, \mathbf{A}, \Theta_{\text{carots}})$.
  Save $\mathcal{F}_\theta$; append "phase2" to $\mathcal{J}$; save $\mathcal{J}$.
else
  Load $\mathcal{F}_\theta \leftarrow \text{Load}(\mathcal{P}_{\text{ckpt}}/\text{carots.pth})$.
end if
— Execute phase 3: FAM Attention —
if "phase3" $\notin \mathcal{J}$.completed_phases **then**
  $\Psi_{\omega,\gamma} \leftarrow$ Algorithm 7$(\mathcal{F}_\theta, \mathbf{X}_{\text{train}}, \mathbf{A}, \Theta_{\text{fam}})$.
  Save $\Psi_{\omega,\gamma}$; append "phase3" to $\mathcal{J}$; save $\mathcal{J}$.
else
  Load $\Psi_{\omega,\gamma} \leftarrow \text{Load}(\mathcal{P}_{\text{ckpt}}/\text{fam.pth})$.
end if
— Execute phase 4: ATUAD Refinement —
if "phase4" $\notin \mathcal{J}$.completed_phases **then**
  $\mathcal{E}_\psi, \mathcal{G}_\rho, \mathcal{D}_\xi \leftarrow$ Algorithm 8$(\mathcal{F}_\theta, \Psi_{\omega,\gamma}, \mathbf{X}_{\text{train}}, \Theta_{\text{atuad}})$.
  Save weights; append "phase4" to $\mathcal{J}$; save $\mathcal{J}$.
else
  Load $\mathcal{E}_\psi, \mathcal{G}_\rho, \mathcal{D}_\xi \leftarrow \text{Load}(\mathcal{P}_{\text{ckpt}}/\text{atuad.pth})$.
end if
— Execute phases 5–6: Memory Bank & Calibration —
if "phase5" $\notin \mathcal{J}$.completed_phases **then**
  $\mathbf{M}, \boldsymbol{\mu}_s, \boldsymbol{\sigma}_s, \mathbf{z}_{\text{cent}}, \tau \leftarrow$ Algorithm 9$(\mathcal{F}_\theta, \Psi_{\omega,\gamma}, \mathbf{X}_{\text{train}})$.
  Save calibration buffers; append "phase5" to $\mathcal{J}$; save $\mathcal{J}$.
end if
return Complete pipeline artifacts.

A.9 Unified Multi-Criteria Online Inference Pipeline

Algorithm 11 defines the real-time scoring logic evaluated in `run_inference.py`.

Algorithm 11 Unified Multi-Criteria Inference Pipeline

Require: Test window observation $\mathbf{W}_t \in \mathbb{R}^{L\times N}$, trained modules $\{\mathcal{F}_\theta, \Psi_{\omega,\gamma}, \mathcal{E}_\psi, \mathcal{G}_\rho, \mathcal{D}_\xi\}$, memory bank $\mathbf{M}$, normalization buffers $(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s, \mathbf{z}_{\text{cent}})$, decision threshold τ.

Ensure: Anomaly flag $\hat{y}_t \in \{0,1\}$, ensemble score S_{ens}, and decomposed metric attribution.

1: **Normalization Anchoring:** Normalize incoming window using training-fitted parameters:

$$\tilde{\mathbf{W}}_t = (\mathbf{W}_t - \boldsymbol{\mu}_s) \oslash \boldsymbol{\sigma}_s \tag{37}$$

2: **Latent Feature Extraction:**

$$\mathbf{Z}_{\text{carots}} = \mathcal{F}_\theta(\tilde{\mathbf{W}}_t) \in \mathbb{R}^{L\times d_z} \tag{38}$$

$$\mathbf{Z}_{\text{fam}},\ (\mathbf{A}_{\text{temp}}, \mathbf{A}_{\text{spat}}) = \Psi_{\omega,\gamma}(\mathbf{Z}_{\text{carots}}) \tag{39}$$

$$\bar{\mathbf{z}}_t = \frac{1}{L}\sum_{\ell=1}^{L} \mathbf{Z}_{\text{fam}}[\ell,:] \in \mathbb{R}^{d_z} \tag{40}$$

3: **Component Score Extraction:**

$$s_{\text{Z-dist}} = \|\bar{\mathbf{z}}_t - \mathbf{z}_{\text{cent}}\|_2 \tag{41}$$

$$s_{\text{recon}} = \|\hat{\mathbf{W}}_t - \tilde{\mathbf{W}}_t\|_2^2 \quad \text{or} \quad \frac{1}{L\cdot N}\sum_{\ell,n}\left(\frac{(\tilde{\mathbf{W}}_{t,\ell,n} - \hat{\boldsymbol{\mu}}_{\ell,n})^2}{\exp(\log \hat{\boldsymbol{\sigma}}^2_{\ell,n})} + \log \hat{\boldsymbol{\sigma}}^2_{\ell,n}\right) \tag{42}$$

$$s_{\text{mem}} = 1.0 - \max_{k\in\{1,\dots,K\}}\left(\frac{\bar{\mathbf{z}}_t \cdot \mathbf{M}[k,:]^\top}{\|\bar{\mathbf{z}}_t\|_2 \|\mathbf{M}[k,:]\|_2}\right) \tag{43}$$

$$s_{\text{disc}} = \mathcal{D}_\xi(\mathbf{Z}_{\text{ref}}) \tag{44}$$

$$s_{\text{Z-vel}} = \frac{1}{L-1}\sum_{\ell=2}^{L} \|\mathbf{Z}_{\text{carots}}[\ell,:] - \mathbf{Z}_{\text{carots}}[\ell-1,:]\|_2 \tag{45}$$

4: **Standardized Calibration Z-Scores:** For each criterion $i \in \{\text{Z-dist}, \text{recon}, \text{mem}, \text{disc}, \text{Z-vel}\}$, normalize against empirical calibration statistics $(\mu_i^{\text{cal}}, \sigma_i^{\text{cal}})$ evaluated on nominal training data:

$$z_i = \frac{s_i - \mu_i^{\text{cal}}}{\sigma_i^{\text{cal}} + \epsilon} \tag{46}$$

5: **Multi-Criteria Score Fusion:** Penalize abnormal variance bidirectionally via absolute deviation and discriminator confidence directionally:

$$\mathcal{S}_{\text{ens}} = 1.5|z_{\text{Z-dist}}| + 1.0|z_{\text{recon}}| + 0.5|z_{\text{mem}}| + 1.0\max(0, -z_{\text{disc}}) + 0.2|z_{\text{Z-vel}}| \tag{47}$$

6: **Anomaly Decision:**

$$\hat{y}_t = \begin{cases} 1 & \text{if } \mathcal{S}_{\text{ens}} \geq \tau \quad \text{(Anomaly)} \\ 0 & \text{otherwise} \quad \text{(Nominal)} \end{cases} \tag{48}$$

7: **Root-Cause Sensor Attribution (Top-3 Anomalous Features):**

$$\mathbf{e}_n = \frac{1}{L}\sum_{\ell=1}^{L} |\tilde{\mathbf{W}}_{t,\ell,n} - \hat{\boldsymbol{\mu}}_{\ell,n}|, \quad \mathcal{R}_{\text{top3}} = \text{argTop3}_{n\in\{1,\dots,N\}}(\mathbf{e}_n) \tag{49}$$

8: **return** $\hat{y}_t, \mathcal{S}_{\text{ens}}, \mathcal{R}_{\text{top3}}$.

A.10 Domain Adaptation Gating via DACAD KL-Divergence

Algorithm 12 formalizes the statistical gating mechanism that detects extreme distribution divergence between the deployment regime and the source calibration manifold.

Algorithm 12 DACAD: KL-Divergence Domain Shift Gating

Require: Query batch of latent vectors $\mathbf{Z}_{\text{eval}} \in \mathbb{R}^{B\times d_z}$, source domain Gaussian parameters $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, divergence threshold τ_{KL}.
Ensure: Domain shift flag $\mathbb{I}_{\text{shift}} \in \{0, 1\}$, and KL divergence magnitude $\mathcal{D}_{\text{KL}}$.
1: Estimate empirical batch statistics under diagonal covariance assumption:

$$\boldsymbol{\mu}_1 = \frac{1}{B}\sum_{b=1}^{B} \mathbf{Z}_{\text{eval}}[b,:], \quad \boldsymbol{\sigma}_1^2 = \frac{1}{B}\sum_{b=1}^{B}(\mathbf{Z}_{\text{eval}}[b,:] - \boldsymbol{\mu}_1)^2 + 10^{-6} \tag{50}$$

2: Compute exact analytical multivariate Gaussian Kullback-Leibler divergence:

$$\mathcal{D}_{\text{KL}}(\mathcal{N}_1 \parallel \mathcal{N}_0) = \frac{1}{2}\sum_{j=1}^{d_z}\left(\log\frac{\boldsymbol{\sigma}_{0,j}^2}{\boldsymbol{\sigma}_{1,j}^2} + \frac{\boldsymbol{\sigma}_{1,j}^2 + (\boldsymbol{\mu}_{1,j} - \boldsymbol{\mu}_{0,j})^2}{\boldsymbol{\sigma}_{0,j}^2} - 1\right) \tag{51}$$

3: Determine extreme domain shift condition:

$$\mathbb{I}_{\text{shift}} = \begin{cases} 1 & \text{if } \mathcal{D}_{\text{KL}} > \tau_{\text{KL}} \quad \text{(Out-of-Domain Distribution)} \\ 0 & \text{otherwise} \quad \text{(In-Domain Nominal)} \end{cases} \tag{52}$$

4: **return** $\mathbb{I}_{\text{shift}}, \mathcal{D}_{\text{KL}}$.

A.11 Causal Evaluation Metric: Transitive Causal Fidelity (TCF)

Algorithm 13 outlines the calculation of Transitive Causal Fidelity, by evaluating recovered edges against the reachability paths in the ground-truth causal graph.

Algorithm 13 Transitive Causal Fidelity (TCF) Evaluation

Require: Predicted adjacency matrix $\mathbf{A}_{\text{pred}} \in \mathbb{R}^{N\times N}$, ground-truth DAG adjacency $\mathbf{A}_{\text{true}} \in \{0, 1\}^{N\times N}$, binarization threshold $\rho = 0.5$.
Ensure: Transitive precision $\mathcal{P}_{\text{TC}}$, transitive recall $\mathcal{R}_{\text{TC}}$, and balanced fidelity score TCF $\in [0, 1]$.
1: Binarize estimated causal graph: $\mathbf{B}_{\text{pred}} = (\mathbf{A}_{\text{pred}} \geq \rho) \in \{0, 1\}^{N\times N}$.
2: Compute Transitive Closure graph $\mathbf{R}_{\text{true}}$ of ground-truth DAG $\mathbf{A}_{\text{true}}$ via reachability:

$$\mathbf{R}_{\text{true}}[i, j] = \begin{cases} 1 & \text{if } \exists \text{ directed path } i \rightsquigarrow j \text{ in } \mathbf{A}_{\text{true}} \ (i \neq j) \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

3: Calculate Transitive Precision (fraction of predicted edges corresponding to valid causal paths):

$$\mathcal{P}_{\text{TC}} = \frac{\sum_{i\neq j} \mathbf{B}_{\text{pred}}[i, j] \cdot \mathbf{R}_{\text{true}}[i, j]}{\sum_{i\neq j} \mathbf{B}_{\text{pred}}[i, j] + 10^{-8}} \tag{54}$$

4: Calculate Transitive Recall (fraction of physical causal paths captured by predicted edges):

$$\mathcal{R}_{\text{TC}} = \frac{\sum_{i\neq j} \mathbf{B}_{\text{pred}}[i, j] \cdot \mathbf{R}_{\text{true}}[i, j]}{\sum_{i\neq j} \mathbf{R}_{\text{true}}[i, j] + 10^{-8}} \tag{55}$$

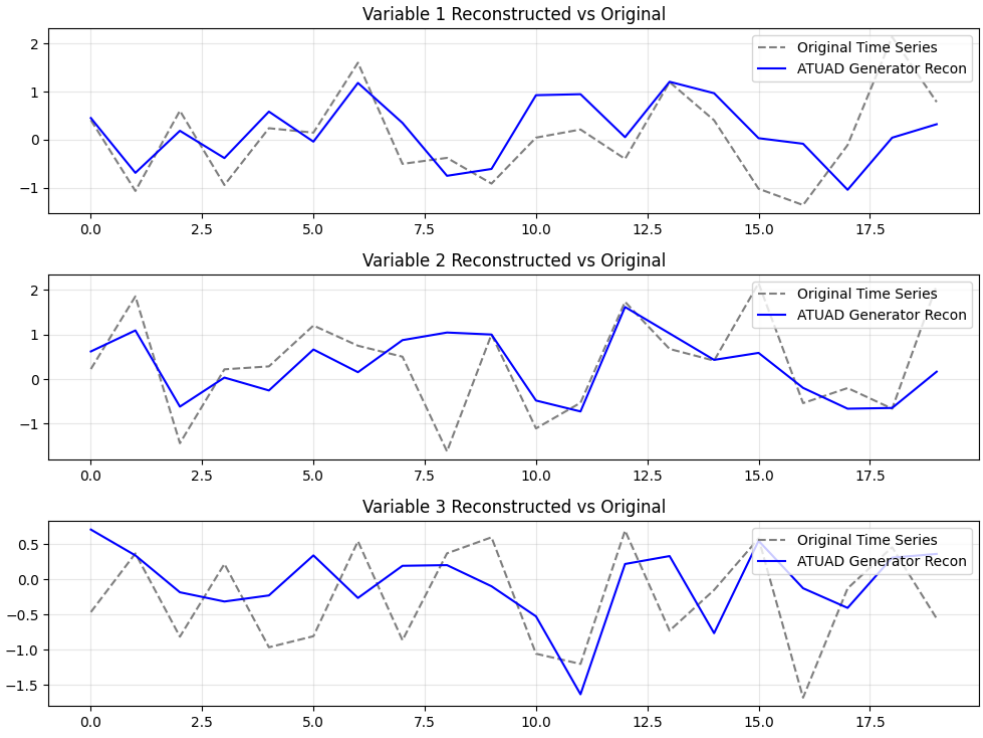
5: Formulate harmonic Transitive Causal Fidelity score:

$$\text{TCF} = 2 \cdot \frac{\mathcal{P}_{\text{TC}} \cdot \mathcal{R}_{\text{TC}}}{\mathcal{P}_{\text{TC}} + \mathcal{R}_{\text{TC}} + 10^{-8}} \tag{56}$$

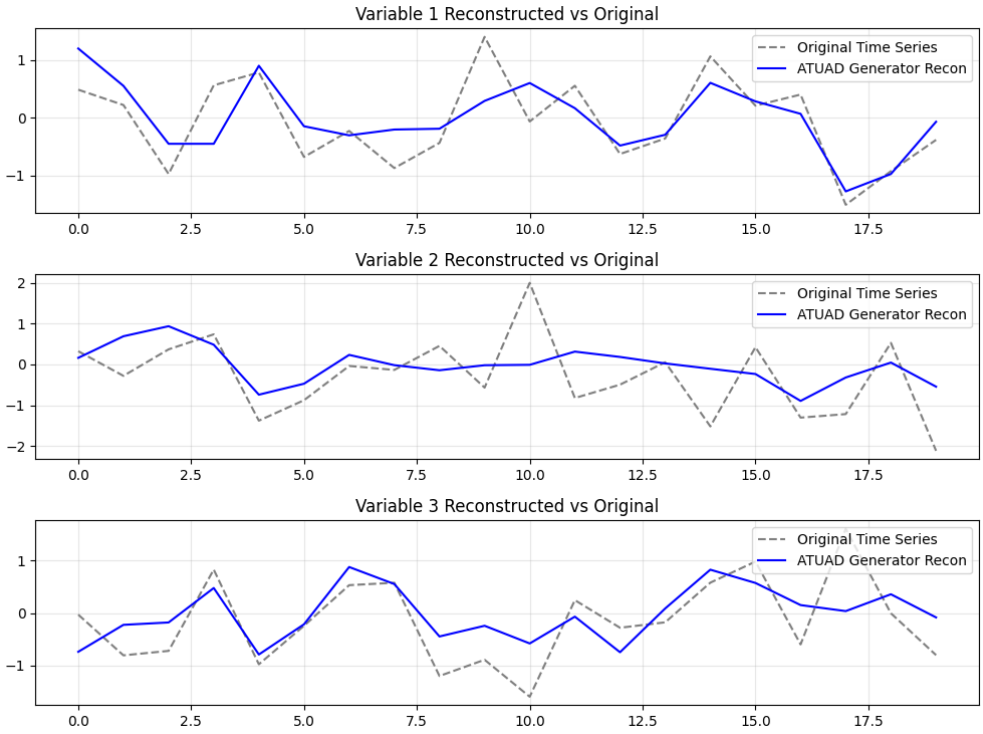
6: **return** $\mathcal{P}_{\text{TC}}, \mathcal{R}_{\text{TC}}, \text{TCF}$.

A.12 ADDITIONAL RESULTS

Figure 7 and Figure 8 illustrate the reconstruction convergence across the two training stages of phase 4. At stage 1, the generator produces smooth, low-frequency reconstructions that track the signal trend but fail to recover fine-grained temporal dynamics. The reconstruction error remains systematically elevated. Following stage 2 adversarial training (WGAN-GP), the generator matches both amplitude and temporal structure across all three representative variables, with the reconstructed signal closely tracking the original. This convergence is critical: anomalous windows, which encode causal structure violations, are invisible to the encoder and fail to reconstruct with comparable fidelity, producing elevated s_{cloud} scores that drive detection.

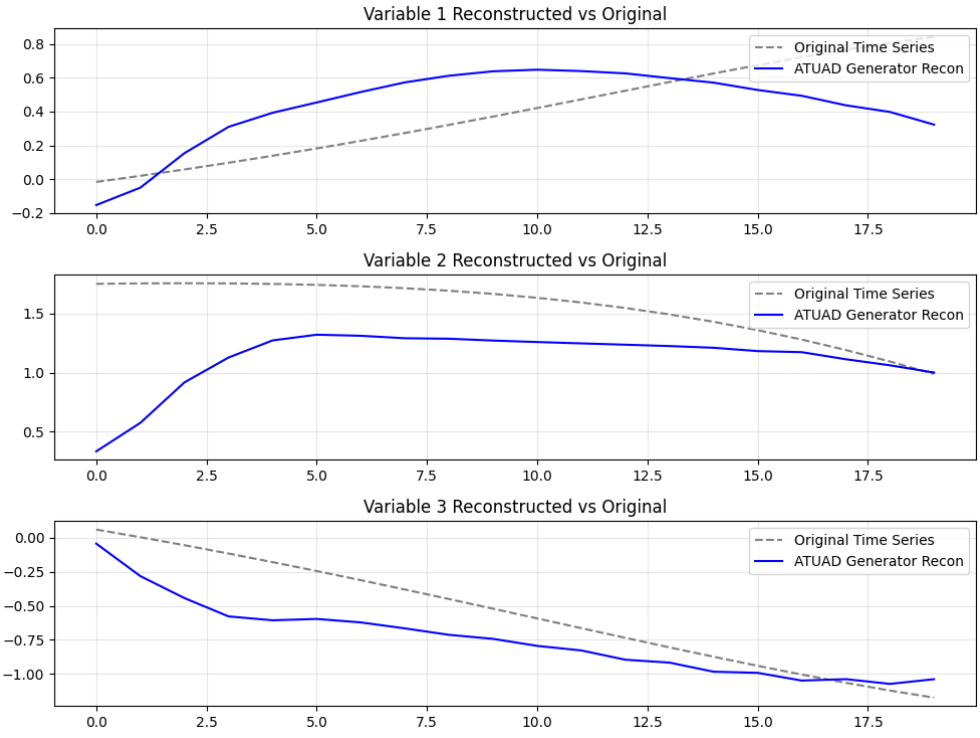


(a) Stage 1 (MSE warmup, epoch 40): smooth but inaccurate.

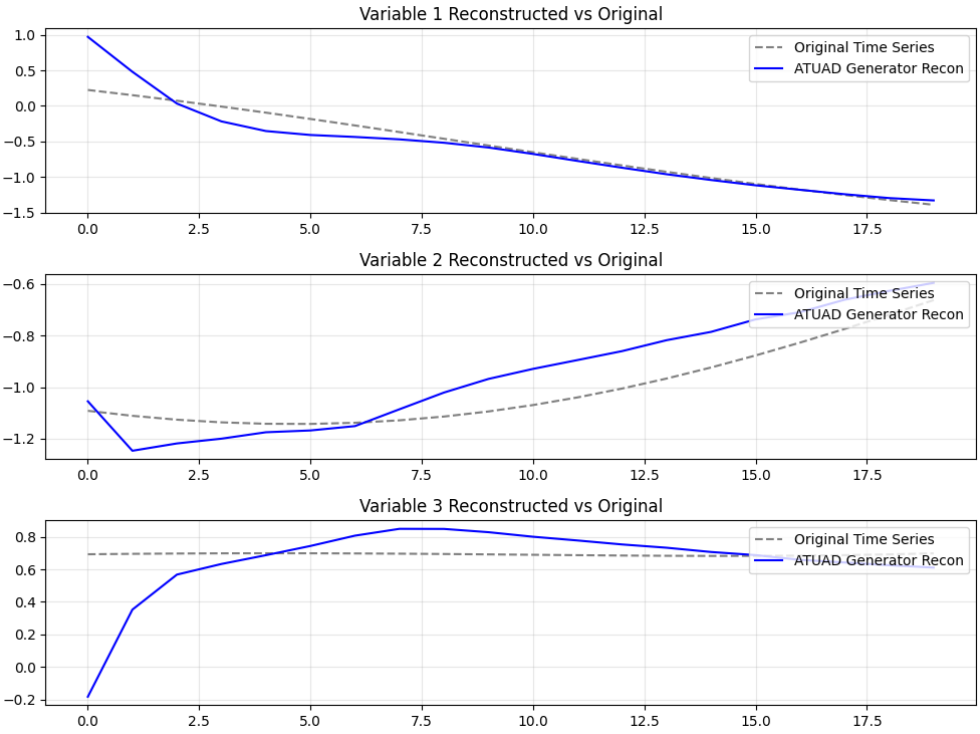


(b) Stage 2 (WGAN-GP, epoch 20): tight temporal alignment.

Figure 7: ATUAD reconstruction behavior on VAR-100 across the two training stages.



(a) Stage 1 (MSE warmup, epoch 40): smooth but inaccurate.



(b) Stage 2 (WGAN-GP, epoch 20): tight temporal alignment.

Figure 8: ATUAD reconstruction behavior on Lorenz-96 across the two training stages.