Supervision Recovery for Time Series Anomaly Detection via Context-Anchored Pairing

Yifei Gao[1] **Tian Lan**[1] **Yimeng Lu**[1] **Xuming An**[1]
Meng Wang[2] **Wenjun He**[2] **Yijie Li**[2] **Chen Zhang**[1*]
[1]Department of Industrial Engineering, Tsinghua University
[2]Huawei
gao-yf@mail.tsinghua.edu.cn
{lant23,luym25,axm24}@mails.tsinghua.edu.cn
{wangmeng71,hewenjun8,liyijie5}@huawei.com
zhangchen01@tsinghua.edu.cn

Abstract

Time series anomaly detection (TSAD) remains challenging not only because anomaly labels are scarce, but also because temporal anomalies are highly context-dependent. Existing methods often rely on unsupervised objectives or surrogate abnormal patterns, providing limited supervision for context-dependent normal–anomalous distinctions. We propose Context-Anchored Pair Supervision (CAPS), a supervision-recovery framework for TSAD. CAPS views ideal anomaly supervision as a matched comparison between normal and anomalous outcomes under the same temporal context, and seeks to recover such supervision without target-domain anomaly labels. Using simulated normal–anomalous pairs, CAPS learns structure and anomaly-semantic representations through reconstruction, background consistency, and within-pair counterfactual recombination. The resulting anomaly representations form a continuous semantic space with coarse modes and induce a sampleable multimodal prior. CAPS conditionally realizes sampled semantics as residual-form effects on target reference trajectories. The resulting context-anchored normal–anomalous counterparts provide temporal supervision for discriminative detector learning. Experiments on nine datasets show that CAPS achieves the strongest aggregate performance across all four evaluation metrics among the compared methods, while complementary ablations and transfer analyses support the roles of context anchoring, semantic disentanglement, and conditional realization.

1 Introduction

Time series anomaly detection (TSAD) aims to identify abnormal events in sequential data, with broad applications in industrial monitoring [1, 2], system operations [3, 4], and diverse sensor-driven systems [5–7]. Despite its importance, effective TSAD remains challenging not only because anomaly labels are scarce, but also because temporal anomalies are highly context-dependent [8, 9]. The same local pattern may be benign in one temporal regime but abnormal in another; for example, a sudden increase can be normal during system start-up but anomalous during steady operation. To address these challenges, existing TSAD methods often rely on indirect or surrogate training signals, including reconstruction [10], forecasting [11], density estimation [12], representation learning [13], self-supervised or contrastive learning, pseudo anomalies, synthetic labels, and anomaly injection [14, 15]. Although effective in many cases, these methods either infer anomaly evidence indirectly from proxy objectives or obtain abnormal supervision from surrogate patterns. The resulting training

*Corresponding author.

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

signal may not always align with the normal–anomalous distinction relevant to the target temporal context, particularly when surrogate abnormal patterns are constructed without accounting for that context. This raises a key question: **can we recover better supervision for TSAD without target-domain anomaly labels?**

Addressing this question first requires clarifying **what such supervision should represent**. Inspired by a causal perspective, we view an anomaly as the effect of an anomaly mechanism acting on temporal evolution. Let $X^{(0)}$ denote the normal outcome of a sequence under a given temporal context and $X^{(\alpha)}$ the alternative outcome when anomaly mechanism α acts under the same context. Their observation-level difference $R^{(\alpha)} := X^{(\alpha)} - X^{(0)}$ represents the observable anomaly effect relative to the matched normal outcome. This contrast does not specify a structural model of how the anomaly is physically generated or assume that the underlying mechanism is additive. The desired supervision is therefore a matched normal–anomalous comparison under the same temporal context, such that the discriminative contrast is concentrated on anomaly-induced variation. Such paired outcomes, however, are not jointly observable in real target-domain data. Simulated source data provide a practical route because controllable normal–anomalous correspondences can be constructed by design, without assuming that target anomalies follow the same predefined categories. Yet directly using simulated anomalous observations as target positives may mix anomaly effects with source–target background differences, since these observations retain source-specific temporal structure. Moreover, the same anomaly semantics may manifest differently across temporal contexts, as their effects depend on the reference dynamics. We therefore need to extract transferable anomaly semantics from simulation and re-realize them under target temporal contexts. This raises the central methodological question: **how can transferable anomaly semantics be learned from simulated correspondences and converted into matched target-domain supervision?**

Building on this perspective, we propose Context-Anchored Pair Supervision (CAPS), a framework for recovering target-domain supervision in TSAD. CAPS first learns structure and anomaly-semantic representations from matched simulated pairs through reconstruction, background consistency, and counterfactual recombination. Specifically, the recombination objective requires an anomalous observation to remain reconstructible after replacing its structure code with that of its matched normal reference, while retaining its anomaly-semantic code and support. This encourages anomaly information to remain usable with structure extracted from a normal observation. The learned anomaly representations form a shared continuous semantic space and induce a sampleable multimodal prior. A context-conditioned residual generator then realizes sampled semantics as anomaly effects on target reference trajectories. Pairing each generated counterpart with its target reference yields context-anchored temporal supervision for discriminative detector learning. Our main contributions are summarized as follows:

- We formulate TSAD as a supervision-recovery problem, seeking matched normal–anomalous supervision under a shared temporal context without target-domain anomaly labels.
- We propose CAPS, which learns anomaly-semantic representations from simulated pairs through reconstruction, background consistency, and counterfactual recombination, organizes them into a continuous semantic prior, and conditionally realizes sampled semantics on target references to construct context-anchored matched supervision.
- We conduct extensive experiments on nine datasets, where CAPS achieves the strongest aggregate performance among the compared methods. Complementary ablations and transfer analyses support the components of supervision recovery, while theoretical results characterize the roles of background mismatch, semantic coverage, and contextual realization in supervision transfer.

2 Related Work

Time Series Anomaly Detection. In label-scarce TSAD, methods typically learn from unlabeled or normal-dominated data. A common paradigm is to model regular temporal behavior and identify anomalies as deviations from learned normality, including reconstruction-based [16, 17], forecasting-based [18], density-estimation [19], and one-class methods [20]. Self-supervised, contrastive, and representation-learning approaches further improve temporal representations through auxiliary objectives or transformed views [21, 22]. While effective, these methods primarily derive anomaly evidence

indirectly from learned normality or surrogate objectives. To obtain more explicit anomaly-oriented supervision, recent studies exploit limited anomaly labels or contextual anomalies [23–26], construct pseudo anomalies or synthetic labels [15], apply degradation-based or contrastive transformations [27, 14], or introduce artificial anomalies into normal sequences [28]. Such approaches reduce dependence on anomaly annotations, but their induced abnormal patterns are often governed by predefined perturbations or transformation rules, which may limit diversity and adaptability to temporal contexts and do not explicitly learn transferable anomaly semantics from matched normal–anomalous correspondences.

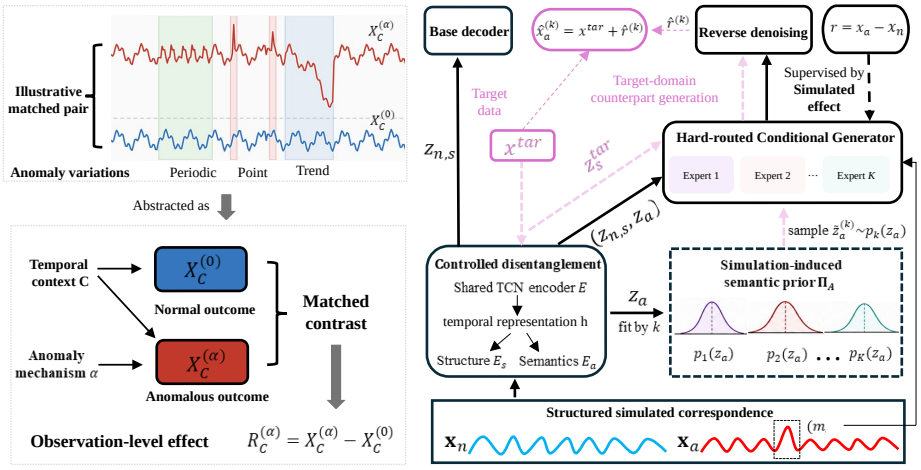
Beyond rule-based anomaly construction, learning-based generative approaches using GANs, diffusion models, and flow models [29–31] provide flexible mechanisms for modeling time-series distributions and generating time-series samples. However, directly modeling target-domain anomalous distributions typically requires observed anomaly examples, anomaly-conditioned data, or externally specified abnormal conditions, which may be unavailable in the target-label-scarce setting considered here. Using anomalous data from simulated or external domains reduces dependence on target anomaly examples, but transferring such observations or distributions may entangle anomaly characteristics with source-specific temporal structure rather than isolate transferable knowledge. In visual anomaly detection, mature vision-language and text-to-image models provide rich semantic priors for synthesizing unseen defects from normal images [32]. Constructing a comparable prior for temporal anomalies is challenging because their observable effects depend on temporal dynamics, operating states, and the surrounding reference context. Transferring anomaly knowledge therefore requires distinguishing reusable anomaly semantics from their context-dependent realization. This raises a more fundamental question for TSAD: what can provide an appropriate prior for temporal anomaly variation, and how can transferable anomaly knowledge be extracted from such a prior and adapted to target temporal contexts?

3 TSAD as Target-Domain Supervision Recovery

Setting and available information. Let $x = (x_1, \ldots, x_T) \in \mathbb{R}^T$ denote a time-series window. We consider target-domain TSAD where the training set $\mathcal{D}^{\mathrm{tr}}_{\mathrm{tar}} = \{x_i^{\mathrm{tar}}\}_{i=1}^{N_t}$ is unlabeled and normal-dominated, and the goal is to learn a detector D_θ without target-domain anomaly labels. We additionally assume access to simulated normal–anomalous data with controllable correspondences, including matched normal and anomalous outcomes, anomaly supports, and coarse simulated anomaly-family labels. Such data provide structured information about anomaly-induced variation, but constitute neither target-domain anomaly labels nor target-domain anomalous observations. The simulated family labels provide a coarse organization of anomaly variation without prescribing exhaustive anomaly categories in the target domain. Given our focus on temporal anomaly behavior, we primarily consider anomaly effects over time without explicitly modeling inter-variable dependencies.

Desired supervision under temporal context. To characterize the supervision missing in the target domain, let $X_C^{(0)}$ denote the normal outcome under temporal context C, and $X_C^{(\alpha)}$ the outcome when a conceptual anomaly mechanism $\alpha \in \mathcal{A}$ acts under the same context. Their observation-level contrast $R_C^{(\alpha)} := X_C^{(\alpha)} - X_C^{(0)}$ describes the realized anomaly effect relative to the matched normal outcome. Here, α denotes the conceptual mechanism, whereas $R_C^{(\alpha)}$ is its observable manifestation under context C; hence, the same mechanism may manifest differently across contexts, with $R_{C_1}^{(\alpha)} \neq R_{C_2}^{(\alpha)}$ in general. This difference is an outcome-level representation of the matched contrast: it neither assumes an additive underlying mechanism nor requires causal identification of α. The desired anomaly-oriented supervision is therefore the matched comparison $(X_C^{(0)}, X_C^{(\alpha)})$, whose shared temporal context isolates anomaly-induced variation from unrelated background differences.

Why simulated supervision is not directly transferable. Although simulation provides such matched correspondences, simulated anomalies are realized under simulated rather than target temporal contexts. For a simulated pair $(x_{n,j}^{\mathrm{sim}}, x_{a,j}^{\mathrm{sim}})$ and an unrelated target reference x_i^{tar}, $x_{a,j}^{\mathrm{sim}} - x_i^{\mathrm{tar}} = (x_{a,j}^{\mathrm{sim}} - x_{n,j}^{\mathrm{sim}}) + (x_{n,j}^{\mathrm{sim}} - x_i^{\mathrm{tar}})$, where the two terms correspond to the source-context anomaly effect and source–target background mismatch. Directly using a simulated anomaly as a target positive therefore confounds anomaly-induced variation with reference mismatch. Moreover, even $x_{a,j}^{\mathrm{sim}} - x_{n,j}^{\mathrm{sim}}$ is a realization tied to its simulated context and need not match how the same anomaly knowledge manifests on a target trajectory. Thus, the transferable object is neither the source



(a) Matched-supervision view of temporal anomalies

(b) CAPS framework

Figure 1: Overall illustration of CAPS. Panel (a) illustrates the matched-supervision view of temporal anomalies. Panel (b) presents the CAPS framework: black arrows denote learning from simulated correspondences, while purple arrows denote target-domain counterpart generation.

anomalous observation nor its realized effect itself. CAPS instead extracts anomaly semantics that remain meaningful across contexts and re-realizes them according to target temporal dynamics.

Goal: target-domain supervision recovery. Given the simulated correspondences and $\mathcal{D}^{\mathrm{tr}}_{\mathrm{tar}}$, our goal is to construct target-matched pairs $\widehat{\mathcal{P}}_{\mathrm{tar}} = \{(x_i^{\mathrm{tar}}, \widehat{x}_{i,a}, m_i)\}$, where $\widehat{x}_{i,a}$ preserves the temporal background of x_i^{tar} while realizing learned anomaly semantics on support m_i. Under the normal-dominated assumption, target windows serve as nominal reference anchors rather than being assumed strictly anomaly-free. The reference and its constructed counterpart provide temporal supervision $\mathbf{0}$ and m_i, respectively, where m_i is known by construction rather than observed from the target domain; the resulting supervision is therefore temporal rather than merely window-level. Accordingly, *supervision recovery* means constructing target-compatible normal–anomalous training supervision, not recovering unknown ground-truth labels, exact anomalous outcomes, or physical anomaly mechanisms. This formulation yields two complementary requirements: extracting transferable anomaly semantics from structured simulation and realizing them according to target temporal contexts.

4 Context-Anchored Pair Supervision

Sec. 3 formulates TSAD as target-domain supervision recovery, requiring transferable anomaly semantics and their realization under target temporal contexts. CAPS addresses these requirements in a unified framework. First, it learns structure and anomaly-semantic representations from simulated normal–anomalous pairs through reconstruction, background consistency, and counterfactual recombination. In particular, replacing the structure code of an anomalous sample with that of its matched normal reference encourages the anomaly-semantic code to remain usable with the normal structure. The resulting semantic representations induce a sampleable multimodal prior. Second, CAPS organizes this prior into coarse semantic modes and employs a hard-routed residual MoE to realize sampled semantics on target reference trajectories, allowing experts to specialize in heterogeneous modes. Given a target reference x_i^{tar}, the generator produces a context-dependent anomaly effect $\hat{r}_i^{(k)}$ and constructs its matched counterpart as $\hat{x}_{i,a}^{(k)} = x_i^{\mathrm{tar}} + \hat{r}_i^{(k)}$. Together with the sampled anomaly support, each reference–counterpart pair provides recovered temporal supervision for discriminative detector learning. CAPS therefore transfers anomaly knowledge through learned semantic representations and target-conditioned realization.

4.1 Structured Disentanglement for Transferable Anomaly Semantics

Learning from matched temporal contrasts. Sec. 3 distinguishes transferable anomaly semantics from their context-dependent observation-level effects. CAPS learns this factorization from simulated normal–anomalous pairs. Within each pair, the temporal reference is shared, while the anomaly introduces a controlled change. This correspondence provides a reference for learning which temporal

structure should be preserved and which variation is associated with the anomaly. We exploit it through complementary constraints on reconstruction, background consistency, and recombination of the learned factors.

Following the simulated anomaly generation procedure of [33], we construct correspondences (x_n, x_a, m, k), where x_n is the normal reference, x_a its anomalous counterpart, $m \in \{0,1\}^T$ denotes anomaly support, and k indexes a coarse simulated anomaly family. This procedure is used to construct the simulated source correspondences; CAPS subsequently learns representations for semantic transfer and target-conditioned realization. We learn two complementary representations: z_s captures temporal structure to be preserved, while $z_a \in \mathcal{Z}_A$ captures anomaly information in a shared continuous semantic space. The family index k provides a coarse source-side organization of this space, allowing continuous variation within each family without prescribing exhaustive target anomaly categories. The observable realization of anomaly information is jointly determined by its semantic representation and the temporal structure on which it acts.

We adopt a shared TCN-based encoder E to capture multi-scale temporal context through hierarchical receptive fields while retaining local anomaly-related patterns. As shown in Fig. 1(b), E is followed by a structure head E_s and an anomaly-semantic head E_a. Auxiliary base and context-conditioned residual decoders provide reconstruction constraints. Given a matched pair, $h_n = E(x_n)$ and $h_a = E(x_a)$, from which we obtain $z_{n,s} = E_s(h_n)$, $z_{a,s} = E_s(h_a)$, $z_{n,a} = E_a(h_n)$, and $z_a = E_a(h_a)$.

Reconstruction and background consistency. The base decoder reconstructs temporal structure, while the residual decoder reconstructs the anomaly effect conditioned on both structure and anomaly semantics. Let

$$\hat{x}_n = \mathrm{base}(z_{n,s}), \qquad \hat{x}_a = \mathrm{base}(z_{a,s}) + \mathrm{res}(z_{a,s}, z_a, m).$$

The reconstruction loss is

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{2}\left[\mathrm{MSE}(\hat{x}_n, x_n) + \mathrm{MSE}(\hat{x}_a, x_a)\right]. \tag{1}$$

Conditioning residual reconstruction on temporal structure allows the observable effect to depend on the reference context.

To constrain the role of the structure branch, we define

$$\mathrm{maskedL1}(u, v, q) = \frac{\sum_t q_t |u_t - v_t|}{\sum_t q_t + \eta},$$

where $\eta > 0$ is a small constant, and use

$$\mathcal{L}_{\mathrm{base}} = \mathrm{MSE}(\mathrm{base}(z_{n,s}), x_n) + \mathrm{maskedL1}\big(\mathrm{base}(z_{a,s}), x_a, 1 - m\big). \tag{2}$$

This objective reconstructs the normal reference and preserves the observed background outside the anomaly support. Because the two members of a pair share the same temporal reference, their structure representations should remain consistent. We also suppress anomaly-semantic activation for the normal member:

$$\mathcal{L}_{\mathrm{dis}} = \|z_{n,s} - z_{a,s}\|_2^2 + \|z_{n,a}\|_2^2. \tag{3}$$

Counterfactual recombination through structure swapping. To further constrain how the two representations are used, CAPS requires the anomalous outcome to remain reconstructible after swapping its structure code with that of the matched normal reference. Specifically, we replace $z_{a,s}$ with $z_{n,s}$ while retaining the anomaly-semantic code z_a and support m:

$$\mathrm{Dec}(z_{n,s}, z_a, m) = \mathrm{base}(z_{n,s}) + \mathrm{res}(z_{n,s}, z_a, m).$$

The matched anomalous observation x_a provides a reconstruction target for this swapped combination, yielding

$$\mathcal{L}_{\mathrm{cf}} = \mathrm{MSE}\big(\mathrm{Dec}(z_{n,s}, z_a, m), x_a\big). \tag{4}$$

This loss directly trains the decoder to combine structure extracted from a normal observation with anomaly information extracted from its matched anomalous counterpart. It complements $\mathcal{L}_{\mathrm{dis}}$: the latter encourages agreement between the paired structure codes, whereas $\mathcal{L}_{\mathrm{cf}}$ requires the swapped factors to reconstruct the anomalous outcome. The recombination constraint thus encourages z_a to carry anomaly information that remains usable with the normal structure code, while $z_{n,s}$ supplies

the context for its realization. This within-pair training provides a basis for the subsequent generation stage, where sampled anomaly semantics are combined with target-reference structure.

The complete representation-learning objective is

$$\mathcal{L}_{\text{disen}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{cf}}.$$

Together, these objectives encourage accurate reconstruction, background consistency, and recomposability of the learned factors. The resulting anomaly-semantic representations form the basis of the sampleable prior used for target-conditioned anomaly-effect generation.

4.2 Context-Conditioned Anomaly-Effect Generation

Simulation-induced semantic prior and expert specialization. The paired representation learning in Sec. 4.1 yields anomaly-semantic codes z_a in a shared continuous space $\mathcal{Z}_A$. To make these semantics sampleable without target anomaly codes, we fit a diagonal Gaussian $p_k(z_a) = \mathcal{N}(\mu_k, \Sigma_k)$ for each coarse simulated anomaly family k, yielding

$$\Pi_A(z_a) = \sum_{k=1}^{K} \pi_k p_k(z_a), \qquad p_k(z_a) = \mathcal{N}(\mu_k, \Sigma_k). \tag{5}$$

We use uniform weights $\pi_k = 1/K$. The K components provide a coarse multimodal approximation within the same semantic space rather than defining separate or exhaustive target anomaly categories. Continuous variation within each mode is represented by z_a, with samples $\tilde{z}_a^{(k)} \sim p_k(z_a)$ serving as mode-specific semantic conditions. CAPS employs a hard-routed residual MoE: during simulated training, family index k routes each pair to expert U_k, while z_a conditions residual generation. A lightweight router, trained with the same family supervision, maps $(z_{n,s}, z_a)$ from simulated correspondences to normalized expert-association scores for optional target-guided mode selection. Standard CAPS instead uses all modes (Sec. 4.3).

Context-conditioned residual realization. The semantic prior captures transferable anomaly variation but does not determine its context-specific manifestation. For a simulated pair (x_n, x_a), we define its realized effect as $r = x_a - x_n$, an observation-level contrast rather than an additive assumption on the underlying anomaly mechanism. CAPS applies conditional diffusion directly in this observation-level residual space. At diffusion step τ, we form $r_\tau = \sqrt{\bar{\alpha}_\tau} r + \sqrt{1-\bar{\alpha}_\tau}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_\tau$ denotes the cumulative signal-retention coefficient of the forward diffusion process. Expert U_k predicts

$$\hat{\epsilon}^{(k)} = U_k(r_\tau, \tau, z_{n,s}, z_a, m). \tag{6}$$

Here, z_a, $z_{n,s}$, and m specify anomaly semantics, reference context, and intended support, respectively. Thus, z_a specifies what is transferred, while $z_{n,s}$ determines how it is realized under the reference dynamics. For Eq. 7, the corresponding clean-residual estimate is computed as $\hat{r}^{(k)} = (r_\tau - \sqrt{1-\bar{\alpha}_\tau}\hat{\epsilon}^{(k)})/\sqrt{\bar{\alpha}_\tau}$.

For a target reference x_i^{tar}, CAPS extracts $z_{i,s}^{\text{tar}} = E_s(E(x_i^{\text{tar}}))$. For each selected mode k, it samples $\tilde{z}_a^{(k)} \sim p_k(z_a)$ and mask guidance m from the corresponding empirical mask pool. Conditioned on $(z_{i,s}^{\text{tar}}, \tilde{z}_a^{(k)}, m)$, expert U_k produces $\hat{r}_i^{(k)}$ through reverse denoising rather than transferring a source residual. Here, m specifies the intended support rather than an observed target label and subsequently provides point-wise supervision for the generated counterpart. Conditioning on $z_{i,s}^{\text{tar}}$ allows the same semantics to realize differently across target temporal contexts.

Residual localization objective. Simulated pairs provide both $r = x_a - x_n$ and support m. With maskedMSE defined analogously to maskedL1, we optimize

$$\mathcal{L}_{\text{gen}} = \underbrace{\text{maskedMSE}\big(\hat{\epsilon}^{(k)}, \epsilon, m\big)}_{\mathcal{L}_{\text{main}}} + \underbrace{\text{maskedMSE}\big(\hat{r}^{(k)}, 0, 1-m\big)}_{\mathcal{L}_{\text{out}}}. \tag{7}$$

Only expert U_k is optimized for a family-k sample. The first term supervises noise prediction within anomaly support, while the second suppresses the clean-residual estimate outside it, keeping unrelated regions anchored to the reference. Thus, $\mathcal{L}_{\text{gen}}$ learns localized, context-dependent realizations of sampled anomaly semantics.

4.3 Target-Reference Pairing for Detector Supervision

Constructing target-matched counterparts. Given the target-conditioned effect $\hat{r}_i^{(k)}$, CAPS constructs the anomalous counterpart

$$\hat{x}_{i,a}^{(k)} = x_i^{\mathrm{tar}} + \hat{r}_i^{(k)}. \tag{8}$$

The resulting pair shares the same target temporal background, with its controlled difference determined by the generated effect. Without target anomaly-mode guidance, standard CAPS instantiates all K semantic modes for each reference, exposing the detector to diverse effects represented by the learned semantic prior.

Targeted mode selection with limited anomaly guidance. When a small amount of anomalous target data is available, CAPS(Diagnosis) uses it only to narrow the modes used for counterpart generation. For each known anomalous sequence $x \in \mathcal{D}_{\mathrm{known}}$, the router input is constructed as $(E_s(E(x)), E_a(E(x)))$, without requiring a matched normal counterpart, and the router produces normalized mode-association scores $q_k(x)$. We aggregate these scores as $\bar{q}_k = |\mathcal{D}_{\mathrm{known}}|^{-1} \sum_{x \in \mathcal{D}_{\mathrm{known}}} q_k(x)$ and rank the modes accordingly. Let $k_{(1)}, \ldots, k_{(K)}$ denote this ordering. We select the smallest subset

$$\mathcal{K}_\rho = \{k_{(1)}, \ldots, k_{(J_\rho)}\}, \qquad J_\rho = \min \left\{ J : \sum_{j=1}^{J} \bar{q}_{k_{(j)}} \geq \rho \right\}, \tag{9}$$

where $\rho \in (0, 1]$ is a cumulative-association threshold. Accordingly, $\mathcal{K}_i = \{1, \ldots, K\}$ for standard CAPS and $\mathcal{K}_i = \mathcal{K}_\rho$ for CAPS(Diagnosis). The known anomalies are used only for mode selection, not detector supervision, while cumulative selection avoids forcing the observed target anomaly variation into a single coarse mode.

Recovered temporal supervision. With $m_i^{(k)}$ denoting the support sampled during generation, the recovered target pair set is

$$\widehat{\mathcal{P}}_{\mathrm{tar}} = \{(x_i^{\mathrm{tar}}, \hat{x}_{i,a}^{(k)}, m_i^{(k)}) : x_i^{\mathrm{tar}} \in \mathcal{D}_{\mathrm{tar}}^{\mathrm{tr}}, k \in \mathcal{K}_i\}.$$

Because $m_i^{(k)}$ is known by construction, CAPS provides temporal rather than window-level supervision: x_i^{tar} is paired with label sequence $\mathbf{0}$, while $\hat{x}_{i,a}^{(k)}$ is paired with $m_i^{(k)}$. We train an independent lightweight TCN detector $D_\theta : \mathbb{R}^T \to [0, 1]^T$,

$$D_\theta(x) = \sigma(h_\theta(E_D(x))), \tag{10}$$

using point-wise binary cross-entropy over these constructed labels. They constitute target-compatible supervision by construction rather than recovered ground-truth annotations. We keep the detector lightweight so that performance gains primarily reflect the recovered supervision; the complete procedure is provided in Algorithm 1 of Appendix B.1.

4.4 Analysis of Supervision Transfer

Reference matching. For a simulated pair, define its observation-level effect as $r^s := x_a^s - x_n^s$. Relative to an unrelated target reference x,

$$x_a^s - x = \underbrace{r^s}_{\text{source-context effect}} + \underbrace{(x_n^s - x)}_{\text{background mismatch}}. \tag{11}$$

This identity does not assume an additive anomaly mechanism. CAPS constructs $\hat{x}_a = x + \hat{r}$, removing the cross-reference term from the matched contrast. Whether this provides useful supervision still depends on which effects are covered and how they are realized under the target context.

Contextual realization. Consider an idealized effect $R = g(C, Z)$, where C denotes temporal context and Z denotes anomaly semantics and support guidance. If $\mathbb{E}\|R\|_2^2 < \infty$, then

$$\inf_h \mathbb{E}\|R - h(Z)\|_2^2 = \mathbb{E}[\mathrm{tr}\,\mathrm{Cov}(R \mid Z)], \tag{12}$$

with optimum $h^*(Z) = \mathbb{E}[R \mid Z]$. Thus, effect prediction without context incurs irreducible error when the same semantics produce different effects across contexts. The oracle predictor $g(C, Z)$

attains zero approximation error. This motivates target-conditioned realization, but does not establish a detection advantage or a distributional lower bound for stochastic residual transfer.

Detector risk. Let $\xi = (r, m) \in \Xi$ and $\ell_D(x, \xi) = \ell(D(x + r), m)$, where ℓ is the temporal classification loss. Given references $x \sim P_X$, denote ideal matched effect–support supervision by $P^{\Xi}_{\mathrm{tar}}(\cdot \mid x)$ and CAPS supervision by $Q^{\Xi}_{\Pi_A}(\cdot \mid x)$. Define

$$\mathcal{R}_\nu(D) = \frac{1}{2}\mathbb{E}_x \left[\ell(D(x), \mathbf{0}) + \mathbb{E}_{\xi\sim\nu(\cdot|x)}\ell_D(x, \xi)\right]. \tag{13}$$

These distributions respectively induce $\mathcal{R}_{\mathrm{ideal}}$ and $\mathcal{R}_{\mathrm{CAPS}}$, sharing the same reference component.

Introduce $Q^{\Xi,*}_{\Pi_A}$, the analytical distribution obtained with CAPS's semantic prior and support sampling but ideal contextual realization. Using a common metric d_Ξ, define

$$\begin{aligned} \Delta_{\mathrm{cov}} &:= \mathbb{E}_x W_1\Big(P^{\Xi}_{\mathrm{tar}}(\cdot \mid x), Q^{\Xi,*}_{\Pi_A}(\cdot \mid x)\Big), \\ \Delta_{\mathrm{real}} &:= \mathbb{E}_x W_1\Big(Q^{\Xi,*}_{\Pi_A}(\cdot \mid x), Q^{\Xi}_{\Pi_A}(\cdot \mid x)\Big). \end{aligned} \tag{14}$$

The first captures coverage and sampling mismatch; the second contextual realization error.

Proposition 1 (Supervision-transfer risk)**.** *Assume the risks are finite, the conditional distributions have integrable first moments, and $\ell_D(x, \cdot)$ is L-Lipschitz under d_Ξ, uniformly over $D \in \mathcal{H}$ and almost every x. Set $\Delta = \Delta_{\mathrm{cov}} + \Delta_{\mathrm{real}}$. Then*

$$\sup_{D\in\mathcal{H}} |\mathcal{R}_{\mathrm{CAPS}}(D) - \mathcal{R}_{\mathrm{ideal}}(D)| \le \frac{L}{2}\Delta. \tag{15}$$

If $\hat{D} \in \mathcal{H}$ satisfies $\mathcal{R}_{\mathrm{CAPS}}(\hat{D}) \le \inf_{D\in\mathcal{H}} \mathcal{R}_{\mathrm{CAPS}}(D) + \epsilon_{\mathrm{learn}}$, then

$$\mathcal{R}_{\mathrm{ideal}}(\hat{D}) - \inf_{D\in\mathcal{H}} \mathcal{R}_{\mathrm{ideal}}(D) \le L\Delta + \epsilon_{\mathrm{learn}}. \tag{16}$$

Reference matching removes the cross-reference contrast term, while coverage and realization govern the remaining supervision discrepancy. Their reduction by CAPS requires empirical validation. The bounds compare matched training risks with shared reference labels, so they do not account for contamination from anomalous references. Proofs, sufficient conditions for BCE, and the interpretation of $\epsilon_{\mathrm{learn}}$ are provided in Appendix A.

5 Experiments

We evaluate CAPS through benchmark comparisons, component ablations, and analyses of supervision transfer. The main text summarizes overall detection performance, selected ablations and controlled comparisons, and qualitative observations of the learned representations and cross-context semantic recombination. Detailed per-dataset results are provided in Appendix C.5. Appendix D provides detailed analyses of these comparisons and additional investigations of semantic transfer, coverage, realization mechanisms, robustness, and efficiency.

5.1 Experimental Setup

Datasets and evaluation. We evaluate CAPS on nine public TSAD datasets spanning diverse domains and anomaly characteristics: IOPS, MGAB, NAB, Power, SED, UCR, NEK, TODS, and YAHOO. Performance is reported using Affiliation-F, $\mathrm{F1_T}$, Standard-F1, and VUS-PR, covering complementary event-level, point-wise, and range-aware evaluation perspectives.

Baselines. We compare CAPS with classical methods LOF [34] and IForest [35], and deep models OmniAnomaly [5], USAD [3], TranAD [36], Anomaly Transformer [6], TimesNet [37], and FITS [38]. TCN-based references include TCN-AE, TCN-Forecast, TCN-Injection [28], and TCN-Supervised. Appendix C.5 additionally reports CAPS(Diagnosis), which uses limited known anomalies only for expert selection, without using them as detector-training labels. Baseline descriptions and experimental protocols are provided in Appendix C.

Implementation details. Unless otherwise specified, CAPS uses the TCN-based encoder–decoder and independent lightweight TCN detector described in Sec. 4, with sequence length 256 and

Table 1: Overall performance across nine TSAD benchmarks. Score and Rank denote the unweighted mean score and average dataset-wise rank, respectively. Avg.R averages the ranks across all four metrics. T1 and T2 count first-place and top-two results across the 36 dataset–metric pairs. Best and second-best aggregate results are bold and underlined. Detailed per-dataset results are reported in Appendix C.5.

Method	Aff.-F		$F1_T$		Std.-F1		VUS-PR		Avg.R	T1	T2
	Score↑	Rank↓	Score↑	Rank↓	Score↑	Rank↓	Score↑	Rank↓			
Unsupervised TSAD baselines											
LOF	72.26	8.94	28.19	7.44	23.11	8.33	23.19	8.06	8.19	0	2
IForest	47.86	10.67	14.15	10.94	15.65	9.72	18.72	8.67	10.00	0	0
OmniAnomaly	73.48	7.78	28.35	5.56	23.82	6.67	25.04	6.28	6.57	1	6
TranAD	73.39	8.67	21.46	7.67	19.00	8.44	22.54	7.28	8.01	0	0
USAD	67.84	8.89	26.64	5.28	25.23	4.44	25.92	6.22	6.21	3	5
AnomTrans.	69.10	9.00	7.60	11.56	11.76	11.11	12.53	11.78	10.86	0	0
TimesNet	77.90	6.22	16.77	10.72	15.99	10.33	20.05	9.72	9.25	1	1
FITS	77.02	6.22	15.64	10.94	15.53	9.67	18.68	10.22	9.26	0	1
TCN-based references											
TCN-AE	78.19	6.00	30.63	5.22	25.10	5.78	23.96	5.89	5.72	0	1
TCN-Injection	<u>80.30</u>	<u>3.89</u>	<u>36.66</u>	<u>4.00</u>	<u>32.36</u>	<u>4.00</u>	<u>33.87</u>	<u>3.11</u>	<u>3.75</u>	<u>5</u>	<u>16</u>
TCN-Supervised	78.72	6.00	31.34	4.33	27.50	4.67	30.72	4.44	4.86	0	3
TCN-Forecast	77.07	7.44	29.02	5.89	24.43	6.61	23.05	8.00	6.99	0	2
CAPS	**85.33**	**1.28**	**46.53**	**1.44**	**42.91**	**1.22**	**46.82**	**1.33**	**1.32**	**26**	**35**

Table 2: Ablations and controlled comparisons across nine datasets. Each entry is the unweighted average of the corresponding dataset-level scores. Higher is better; the best results are bolded. Detailed analyses are provided in Appendix D.

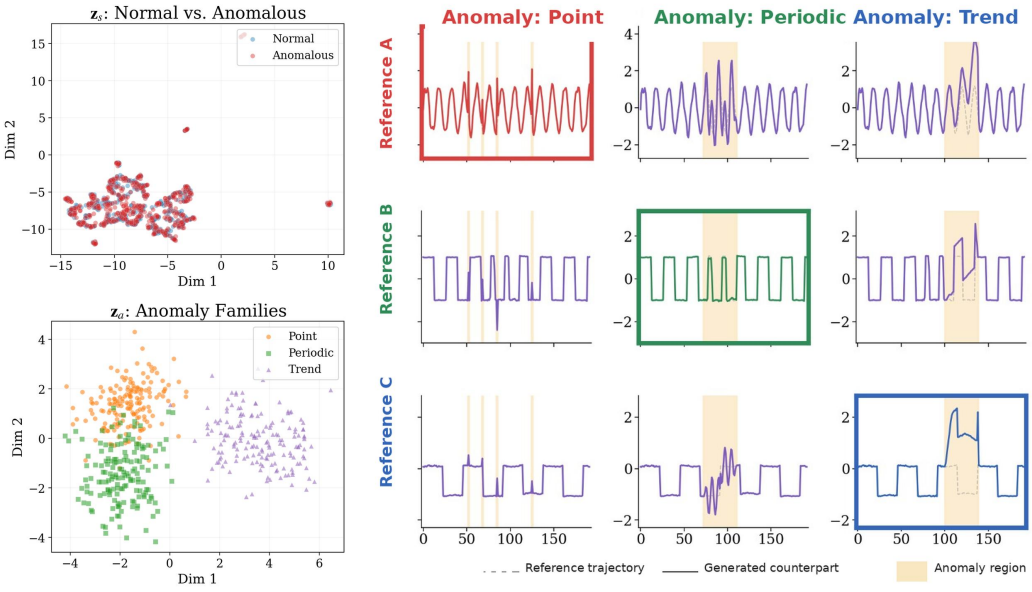
Variant	Affiliation-F	$F1_T$	Standard-F1	VUS-PR
Shuffled residuals	79.27	41.21	37.78	41.30
w/o residual parameterization	82.76	40.96	38.35	42.27
Direct source-residual injection	81.58	43.16	39.96	42.58
w/o counterfactual loss	83.65	42.76	39.11	43.11
CAPS	**85.33**	**46.53**	**42.91**	**46.82**

structure/anomaly latent dimensions of 128/48. We use $K = 3$ coarse semantic modes organized by simulated point, periodic, and trend anomaly families. Alternative mode constructions and different choices of K are examined in Appendix D. Implementation details are provided in Appendix C.

Ablations and Controlled Comparisons. Table 2 summarizes four comparisons using the same TCN detector and evaluation protocol: *(1) Shuffled residuals* applies each generated residual to a different target reference, shuffling its anomaly support together with it while preserving the generator and number of generated counterparts; *(2) Without residual parameterization* replaces residual-form construction with direct anomalous-sequence generation while retaining background-preservation supervision outside the intended anomaly support; *(3) Direct source-residual injection* adds the observation-level differences of the same simulated normal–anomalous pairs directly to target references, jointly bypassing semantic learning and target-conditioned realization; *(4) Without counterfactual loss* removes $\mathcal{L}_{\mathrm{cf}}$ in Eq. 4 while retaining $\mathcal{L}_{\mathrm{rec}}$, $\mathcal{L}_{\mathrm{base}}$, and $\mathcal{L}_{\mathrm{dis}}$, examining the contribution of within-pair counterfactual recomposition. Detailed analyses of these comparisons are provided in Appendix D.

5.2 Experimental Results

Overall detection performance. Table 1 shows that CAPS achieves the strongest aggregate performance across all four evaluation metrics among the compared methods. Its gains over conventional unsupervised methods support the value of explicitly learning normal–anomalous distinctions through recovered supervision. The controlled TCN comparisons further examine this benefit while keeping the detector backbone consistent. In particular, the improvement over TCN-Injection highlights the importance of how synthetic supervision is constructed. By pairing each generated counterpart with



(a) Structure and Anomaly Representations

(b) Cross-Context Semantic Recombination

Figure 2: Learned representations and cross-context semantic recombination. (a) Projections of z_s and z_a. (b) Each column reuses a fixed anomaly-semantic code across different references; colored diagonal cells indicate the original pairings.

its target reference, CAPS provides a shared temporal background against which the detector can learn anomaly-induced differences. CAPS also outperforms the label-supervised TCN reference in aggregate under the evaluated protocol. Although this reference does not represent an upper bound on supervised detection performance, the comparison illustrates the practical value of recovering useful training supervision from simulated anomaly knowledge when target anomaly labels are unavailable.

Contributions of the supervision-recovery components. Table 2 shows that disrupting context–effect matching, removing residual parameterization, or omitting the counterfactual loss reduces aggregate detection performance. These comparisons support matching generated effects to their reference contexts, explicitly retaining the reference during counterpart construction, and using within-pair recomposition to learn representations for transfer. The improvement over direct source-residual injection further supports the combined value of semantic learning and target-conditioned realization under the same simulated anomaly source. Fig. 2 provides complementary qualitative observations. Panel (a) shows substantial overlap between normal and anomalous structure representations z_s, together with organization by anomaly family in the anomaly-semantic representations z_a. Panel (b) illustrates how a fixed anomaly-semantic code can be recombined with different reference structures, producing counterparts that preserve the reference background while exhibiting context-dependent anomaly effects. These examples illustrate the intended distinction between transferable anomaly semantics and their realization under a particular temporal context.

6 Conclusion

We proposed CAPS, a supervision-recovery framework for time series anomaly detection without target-domain anomaly labels. CAPS learns transferable anomaly semantics from simulated normal–anomalous pairs, using within-pair counterfactual recomposition to encourage their separation from temporal structure. It conditionally realizes these semantics as residual effects on target references, constructing context-anchored pairs for discriminative detector learning. Experiments on nine benchmarks and controlled comparisons support the effectiveness of this approach and the contributions of its core components. Performance depends on how well the simulated semantic prior covers target-relevant anomaly variation, motivating richer and adaptive priors. The current framework focuses on temporal anomaly effects; explicit modeling of cross-variable dependencies remains an important direction for future work.

Acknowledgments and Disclosure of Funding

This work was supported by NSFC Grant 72271138, the Tsinghua University Initiative Scientific Research Program 20243080039, and the Tsinghua–Huawei Collaboration Project 20252001894. We also acknowledge computing support from the Tsinghua High-Performance Computing Center and the KaiTuo 1000 supercomputer system.

Competing Interests. The authors declare no additional competing interests.

References

[1] Hussain Nizam, Samra Zafar, Zefeng Lv, Fan Wang, and Xiaopeng Hu. Real-time deep anomaly detection framework for multivariate time-series data in industrial iot. *IEEE Sensors Journal*, 22(23):22836–22849, 2022.

[2] Wenchao Chen, Long Tian, Bo Chen, Liang Dai, Zhibin Duan, and Mingyuan Zhou. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In *International conference on machine learning*, pages 3621–3633. PMLR, 2022.

[3] 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*, pages 3395–3404, 2020.

[4] Hongcheng Guo, Jian Yang, Jiaheng Liu, Jiaqi Bai, Boyang Wang, Zhoujun Li, Tieqiao Zheng, Bo Zhang, Junran Peng, and Qi Tian. Logformer: A pre-train and tuning pipeline for log anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 135–143, 2024.

[5] 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*, pages 2828–2837, 2019.

[6] 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.

[7] 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, pages 4027–4035, 2021.

[8] Andreas Mueller. Open challenges in time series anomaly detection: An industry perspective. *arXiv preprint arXiv:2502.05392*, 2025.

[9] Fengling Wang, Yiyue Jiang, Rongjie Zhang, Aimin Wei, Jingming Xie, and Xiongwen Pang. A survey of deep anomaly detection in multivariate time series: Taxonomy, applications, and directions. *Sensors*, 25 (1):190, 2025.

[10] Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T Kwok. Time series anomaly detection with multiresolution ensemble decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9567–9575, 2021.

[11] Quan Zhou, Changhua Pei, Fei Sun, Han Jing, Zhengwei Gao, Haiming Zhang, Gaogang Xie, Dan Pei, and Jianhui Li. Kan-ad: Time series anomaly detection with kolmogorov–arnold networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 79136–79149. PMLR, 2025.

[12] Qihang Zhou, Jiming Chen, Haoyu Liu, Shibo He, and Wenchao Meng. Detecting multivariate time series anomalies with zero known label. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4963–4971, 2023.

[13] Jinju Park and Seokho Kang. Paano: Patch-based representation learning for time-series anomaly detection. In *International Conference on Learning Representations*, 2026.

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

[15] Kohei Obata, Yasuko Matsubara, and Yasushi Sakurai. Robust and explainable detector of time series anomaly via augmenting multiclass pseudo-anomalies. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2198–2209, 2025.

[16] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[17] Dongmin Kim, Sunghyun Park, and Jaegul Choo. When model meets new normals: Test-time adaptation for unsupervised time-series anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 13113–13121, 2024.

[18] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE access*, 7:1991–2005, 2018.

[19] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[20] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.

[21] Aitor Sánchez-Ferrera, Borja Calvo, and Jose A Lozano. A review on self-supervised learning in time series anomaly detection: Recent advances and open challenges. *ACM Computing Surveys*, 58(5):1–35, 2025.

[22] 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, pages 8980–8987, 2022.

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

[24] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

[25] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2843–2851, 2022.

[26] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. Arc: A generalist graph anomaly detector with in-context learning. *Advances in Neural Information Processing Systems*, 37: 50772–50804, 2024.

[27] Yungi Jeong, Eunseok Yang, Jung Hyun Ryu, Imseong Park, and Myungjoo Kang. Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme. *arXiv preprint arXiv:2305.04468*, 2023.

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

[29] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks*, pages 703–716. Springer, 2019.

[30] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, 2024.

[31] Yang Hu, Xiao Wang, Zezhen Ding, Lirong Wu, Huatian Zhang, Stan Z Li, Sheng Wang, Jiheng Zhang, Ziyun Li, and Tianlong Chen. Flowts: Time series generation via rectified flow. *arXiv preprint arXiv:2411.07506*, 2024.

[32] Han Sun, Yunkang Cao, Hao Dong, and Olga Fink. Unseen visual anomaly generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25508–25517, 2025.

[33] Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. Towards foundation models for zero-shot time series anomaly detection: Leveraging synthetic data and relative context discrepancy. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

[34] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[35] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[36] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, 2022.

[37] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

[38] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, 2024.

[39] Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

[40] M. Saquib Sarfraz, Mei-Yen Chen, Lukas Layer, Kunyu Peng, and Marios Koulakis. Position: Quo vadis, unsupervised time series anomaly detection? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 43461–43476. PMLR, 2024.

[41] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 635–645, 2022.

[42] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron J Elmore, and Michael J Franklin. Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *Proc. VLDB Endow.*, 15(11):2774–2787, 2022.

A Theoretical Details and Proofs

This section provides additional details for the theoretical analysis in the main text. We first formalize the limitation of predicting anomaly effects without temporal context, then prove Proposition 1, and finally clarify the regularity conditions, learning-error term, and scope of the resulting bound.

A.1 Contextual Realization Gap

We first elaborate on why transferable anomaly semantics alone are generally insufficient to determine their observable effects. Let C denote temporal context, Z denote transferable anomaly information including semantic identity and support, and R denote the corresponding observation-level anomaly effect. Consider predictors $h(Z)$ that predict the effect from transferable anomaly information without observing the temporal context.

Assuming finite second moments, the conditional mean

$$h^*(Z) = \mathbb{E}[R \mid Z]$$

minimizes the mean squared prediction error. By the conditional-variance decomposition,

$$\begin{aligned}\mathbb{E}\big[\|R - h(Z)\|_2^2\big] &= \mathbb{E}\big[\|R - \mathbb{E}[R \mid Z]\|_2^2\big] \\ &\quad + \mathbb{E}\big[\|\mathbb{E}[R \mid Z] - h(Z)\|_2^2\big]. \end{aligned} \tag{17}$$

Therefore,

$$\inf_h \mathbb{E}\big[\|R - h(Z)\|_2^2\big] = \mathbb{E}[\operatorname{tr}\operatorname{Cov}(R \mid Z)]. \tag{18}$$

Eq. (18) characterizes the minimum squared error of predicting the corresponding effect from Z alone. In particular, when the same transferable anomaly information admits different conditional mean effects under different temporal contexts, context-independent prediction cannot fully recover the corresponding realized effect.

Allowing the predictor to additionally observe C, the optimal predictor becomes

$$f^*(C, Z) = \mathbb{E}[R \mid C, Z],$$

with minimum error

$$\inf_f \mathbb{E}\big[\|R - f(C, Z)\|_2^2\big] = \mathbb{E}[\operatorname{tr}\operatorname{Cov}(R \mid C, Z)].$$

The reduction in optimal squared error is

$$\begin{aligned}&\inf_h \mathbb{E}\big[\|R - h(Z)\|_2^2\big] - \inf_f \mathbb{E}\big[\|R - f(C, Z)\|_2^2\big] \\ &\quad = \mathbb{E}\Big[\|\mathbb{E}[R \mid C, Z] - \mathbb{E}[R \mid Z]\|_2^2\Big] \geq 0. \end{aligned} \tag{19}$$

Hence, contextual conditioning improves the optimal squared-error prediction whenever temporal context changes the conditional mean realization of the same transferable anomaly information.

In the deterministic idealization used for intuition in the main text, where

$$R = g(C, Z),$$

a predictor with access to both C and Z can recover the corresponding effect exactly:

$$\mathbb{E}\big[\|R - g(C, Z)\|_2^2\big] = 0.$$

This motivates realizing sampled anomaly semantics conditionally on the target reference trajectory rather than treating semantic identity as a context-independent residual template.

This analysis concerns matched-effect prediction under squared loss. It does not establish a distributional lower bound for stochastic residual transfer or, by itself, imply an advantage in downstream anomaly detection.

A.2 Proof of Proposition 1

Let P_X denote the target-reference distribution and let

$$\xi = (r, m) \in \Xi$$

represent an anomaly-effect instance consisting of an observation-level residual r and its temporal support m. For a target reference x, define

$$\ell_D(x, \xi) = \ell(D(x + r), m), \qquad D \in \mathcal{H},$$

where $\mathcal{H}$ is the detector hypothesis class and ℓ is the temporal detection loss.

Let $P^{\Xi}_{\mathrm{tar}}(\cdot \mid x)$ denote the ideal target-domain conditional distribution of anomaly effects and supports under reference x, and let $Q^{\Xi}_{\Pi_A}(\cdot \mid x)$ denote the conditional distribution induced by CAPS. We additionally introduce an analytical intermediate distribution $Q^{\Xi,*}_{\Pi_A}(\cdot \mid x)$, which uses the same simulation-induced semantic prior and support sampling as CAPS but realizes them through an ideal target-context realization mechanism. Define

$$\Delta_{\mathrm{cov}} = \mathbb{E}_{x \sim P_X}\left[W_1\Big(P^{\Xi}_{\mathrm{tar}}(\cdot \mid x), Q^{\Xi,*}_{\Pi_A}(\cdot \mid x)\Big)\right], \tag{20}$$

$$\Delta_{\mathrm{real}} = \mathbb{E}_{x \sim P_X}\left[W_1\Big(Q^{\Xi,*}_{\Pi_A}(\cdot \mid x), Q^{\Xi}_{\Pi_A}(\cdot \mid x)\Big)\right]. \tag{21}$$

The first measures semantic-coverage and sampling mismatch relative to the ideal target effect distribution, while the second measures realization discrepancy relative to this ideal mechanism. This decomposition is analytical and depends on the chosen intermediate distribution $Q^{\Xi,*}_{\Pi_A}$; it is not intended as an identifiable decomposition of the two error sources from observed data.

For any conditional effect distribution $\nu(\cdot \mid x)$, define the matched training risk

$$\mathcal{R}_\nu(D) = \frac{1}{2}\mathbb{E}_{x \sim P_X}\left[\ell(D(x), \mathbf{0}) + \mathbb{E}_{\xi \sim \nu(\cdot|x)}\ell_D(x, \xi)\right]. \tag{22}$$

We write

$$\mathcal{R}_{\mathrm{ideal}} = \mathcal{R}_{P^{\Xi}_{\mathrm{tar}}}, \qquad \mathcal{R}_{\mathrm{CAPS}} = \mathcal{R}_{Q^{\Xi}_{\Pi_A}}.$$

Assume that, for P_X-almost every x, the mapping

$$\xi \mapsto \ell_D(x, \xi)$$

is L-Lipschitz on (Ξ, d_Ξ), uniformly over $D \in \mathcal{H}$. Since the nominal reference term $\ell(D(x), \mathbf{0})$ is identical in the two risks,

$$\begin{aligned} &|\mathcal{R}_{\mathrm{CAPS}}(D) - \mathcal{R}_{\mathrm{ideal}}(D)| \\ &\quad \le \frac{1}{2}\mathbb{E}_{x \sim P_X}\left|\mathbb{E}_{\xi \sim Q^{\Xi}_{\Pi_A}(\cdot|x)}\ell_D(x, \xi) - \mathbb{E}_{\xi \sim P^{\Xi}_{\mathrm{tar}}(\cdot|x)}\ell_D(x, \xi)\right|. \end{aligned} \tag{23}$$

By the Kantorovich–Rubinstein inequality,

$$|\mathcal{R}_{\mathrm{CAPS}}(D) - \mathcal{R}_{\mathrm{ideal}}(D)| \le \frac{L}{2}\mathbb{E}_{x \sim P_X} W_1\big(P^{\Xi}_{\mathrm{tar}}(\cdot \mid x), Q^{\Xi}_{\Pi_A}(\cdot \mid x)\big). \tag{24}$$

For P_X-almost every x, the triangle inequality for W_1 gives

$$\begin{aligned} &W_1\big(P^{\Xi}_{\mathrm{tar}}(\cdot \mid x), Q^{\Xi}_{\Pi_A}(\cdot \mid x)\big) \\ &\quad \le W_1\Big(P^{\Xi}_{\mathrm{tar}}(\cdot \mid x), Q^{\Xi,*}_{\Pi_A}(\cdot \mid x)\Big) \\ &\qquad + W_1\Big(Q^{\Xi,*}_{\Pi_A}(\cdot \mid x), Q^{\Xi}_{\Pi_A}(\cdot \mid x)\Big). \end{aligned} \tag{25}$$

Taking expectation over $x \sim P_X$ therefore yields

$$\mathbb{E}_{x \sim P_X} W_1\big(P^{\Xi}_{\mathrm{tar}}(\cdot \mid x), Q^{\Xi}_{\Pi_A}(\cdot \mid x)\big) \le \Delta_{\mathrm{cov}} + \Delta_{\mathrm{real}}.$$

Consequently,

$$\sup_{D \in \mathcal{H}} |\mathcal{R}_{\mathrm{CAPS}}(D) - \mathcal{R}_{\mathrm{ideal}}(D)| \le \frac{L}{2}\left(\Delta_{\mathrm{cov}} + \Delta_{\mathrm{real}}\right). \tag{26}$$

Let

$$\Delta = \Delta_{\mathrm{cov}} + \Delta_{\mathrm{real}},$$

and suppose that the learned detector $\hat{D} \in \mathcal{H}$ satisfies

$$\mathcal{R}_{\mathrm{CAPS}}(\hat{D}) \leq \inf_{D \in \mathcal{H}} \mathcal{R}_{\mathrm{CAPS}}(D) + \epsilon_{\mathrm{learn}}. \tag{27}$$

Using Eq.(26),

$$\begin{aligned} \mathcal{R}_{\mathrm{ideal}}(\hat{D}) &\leq \mathcal{R}_{\mathrm{CAPS}}(\hat{D}) + \frac{L}{2}\Delta \\ &\leq \inf_{D \in \mathcal{H}} \mathcal{R}_{\mathrm{CAPS}}(D) + \epsilon_{\mathrm{learn}} + \frac{L}{2}\Delta \\ &\leq \inf_{D \in \mathcal{H}} \mathcal{R}_{\mathrm{ideal}}(D) + L\Delta + \epsilon_{\mathrm{learn}}. \end{aligned} \tag{28}$$

Hence,

$$\mathcal{R}_{\mathrm{ideal}}(\hat{D}) - \inf_{D \in \mathcal{H}} \mathcal{R}_{\mathrm{ideal}}(D) \leq L\left(\Delta_{\mathrm{cov}} + \Delta_{\mathrm{real}}\right) + \epsilon_{\mathrm{learn}}, \tag{29}$$

which proves Proposition 1.

A.3 Additional Assumptions and Learning Error

We finally clarify the regularity conditions and scope of the preceding analysis.

Wasserstein regularity. The conditional distributions $P^{\Xi}_{\mathrm{tar}}(\cdot \mid x)$, $Q^{\Xi,*}_{\Pi_A}(\cdot \mid x)$, and $Q^{\Xi}_{\Pi_A}(\cdot \mid x)$ are assumed to be measurable probability kernels with finite first moments under d_{Ξ}. We additionally assume that

$$\Delta_{\mathrm{cov}} + \Delta_{\mathrm{real}} < \infty$$

and that the matched risks in Eq. (22) are finite. These conditions are sufficient for the Wasserstein quantities and expectations used in the proof to be well defined.

A sufficient Lipschitz condition for binary cross-entropy. The proposition requires the composed loss

$$\xi \mapsto \ell(D(x + r), m)$$

to admit a common Lipschitz constant uniformly over $D \in \mathcal{H}$. Binary cross-entropy is not globally Lipschitz when predicted probabilities approach 0 or 1. A sufficient condition is that, uniformly over $D \in \mathcal{H}$, detector outputs on all relevant inputs satisfy

$$D(u) \in [\delta, 1 - \delta]^T, \qquad 0 < \delta < \frac{1}{2},$$

and that there exists a common constant $K_D < \infty$ such that

$$\frac{1}{T}\|D(u) - D(v)\|_1 \leq K_D \|u - v\|_2 \tag{30}$$

for all relevant inputs u and v.

Consider the averaged binary cross-entropy

$$\ell_{\mathrm{BCE}}(p, m) = -\frac{1}{T}\sum_{t=1}^{T}\left[m_t \log p_t + (1 - m_t)\log(1 - p_t)\right].$$

On $p_t \in [\delta, 1 - \delta]$, its variation with respect to prediction satisfies

$$\left|\frac{\partial \ell_{\mathrm{BCE}}}{\partial p_t}\right| \leq \frac{1}{T\delta},$$

while changing the binary label at a fixed prediction is bounded by

$$\frac{1}{T}\left|\log\frac{1 - p_t}{p_t}\right| \leq \frac{1}{T}\log\frac{1 - \delta}{\delta}.$$

Under the product metric

$$d_{\Xi}((r,m),(r',m')) = \|r-r'\|_2 + \frac{1}{T}\|m-m'\|_1, \tag{31}$$

Eq. (30) therefore implies

$$\begin{aligned} &|\ell_{\mathrm{BCE}}(D(x+r),m) - \ell_{\mathrm{BCE}}(D(x+r'),m')| \\ &\quad \le \frac{K_D}{\delta}\|r-r'\|_2 + \log\frac{1-\delta}{\delta}\frac{1}{T}\|m-m'\|_1 \\ &\quad \le L\, d_{\Xi}((r,m),(r',m'))\,, \end{aligned} \tag{32}$$

where one valid common constant is

$$L = \max\left\{\frac{K_D}{\delta}, \log\frac{1-\delta}{\delta}\right\}. \tag{33}$$

These conditions are only sufficient conditions for the theoretical bound; they need not be interpreted as additional constraints imposed by the CAPS training procedure.

Interpretation of $\epsilon_{\mathrm{learn}}$. The term $\epsilon_{\mathrm{learn}}$ represents ordinary detector-learning error under the CAPS-induced population risk rather than semantic-transfer discrepancy. To make this interpretation explicit, let $\widehat{\mathcal{R}}_{\mathrm{CAPS}}$ denote the empirical risk on the recovered training set, and suppose that

$$\widehat{\mathcal{R}}_{\mathrm{CAPS}}(\hat{D}) \le \inf_{D\in\mathcal{H}} \widehat{\mathcal{R}}_{\mathrm{CAPS}}(D) + \epsilon_{\mathrm{opt}}.$$

Then

$$\begin{aligned} &\mathcal{R}_{\mathrm{CAPS}}(\hat{D}) - \inf_{D\in\mathcal{H}} \mathcal{R}_{\mathrm{CAPS}}(D) \\ &\quad \le 2\sup_{D\in\mathcal{H}} \left|\widehat{\mathcal{R}}_{\mathrm{CAPS}}(D) - \mathcal{R}_{\mathrm{CAPS}}(D)\right| + \epsilon_{\mathrm{opt}}. \end{aligned} \tag{34}$$

Thus, the right-hand side of Eq. (34) can be taken as $\epsilon_{\mathrm{learn}}$, separating finite-sample estimation error from optimization error.

This deterministic decomposition does not itself specify a finite-sample convergence rate. Establishing such a rate would additionally require assumptions accounting for temporal dependence and for dependence among generated counterparts constructed from shared references. Moreover, because Proposition 1 compares detectors within the same hypothesis class $\mathcal{H}$, approximation error relative to an unrestricted Bayes-optimal detector is not included in $\epsilon_{\mathrm{learn}}$.

Reference contamination. The bound compares ideal and CAPS matched-training risks under the same target-reference distribution and the same nominal reference label $\mathbf{0}$. It therefore isolates discrepancy associated with semantic coverage, sampling, and contextual realization. If a nominal target reference already contains an unobserved anomaly, assigning it the reference label $\mathbf{0}$ may introduce an additional supervision bias. Such contamination is not represented by Δ_{cov} or Δ_{real} and is outside the scope of Proposition 1; its empirical effect is examined separately in Appendix D.1.

B Training and Implementation Details

B.1 Overall Training and Inference Procedure

Algorithm 1 summarizes the training, supervision-recovery, and inference procedure of CAPS. CAPS first learns structure and anomaly-semantic representations from matched simulated normal–anomalous pairs using the objective in Sec. 4.1. Within-pair counterfactual recomposition reconstructs each anomalous sample from its anomaly-semantic code and its matched normal reference's structure code. The representation modules are then fixed, mode-wise semantic priors are estimated, and residual experts are trained. During target-domain supervision recovery, sampled anomaly semantics are conditionally realized on target references to construct matched reference–counterpart pairs. The detector is then trained with point-wise supervision on these pairs and directly applied at inference. In the standard setting, all modes are used without target anomaly labels. With Diagnosis, limited known anomalies are used only to select associated modes for targeted counterpart generation.

Algorithm 1 Overall Training, Supervision Recovery, and Inference Procedure of CAPS

Require: Simulated correspondences $\mathcal{D}_{\text{sim}}$ containing (x_n, x_a, m, k), unlabeled target training data $\mathcal{D}_{\text{tar}}^{\text{tr}}$, optional known anomalies $\mathcal{D}_{\text{known}}$, Diagnosis threshold $\rho = 0.8$ if enabled
Ensure: Semantic priors $\{p_k\}_{k=1}^{K}$, residual experts $\{U_k\}_{k=1}^{K}$, detector D_θ

Paired representation learning
for each training batch sampled from $\mathcal{D}_{\text{sim}}$ **do**
  Encode simulated normal–anomalous pairs to obtain $z_{n,s}, z_{a,s}, z_{n,a}, z_a$
  Compute the within-pair recomposition loss $\mathcal{L}_{\text{cf}} = \text{MSE}(\text{Dec}(z_{n,s}, z_a, m), x_a)$
  Update the encoder, representation heads, and auxiliary decoders using $\mathcal{L}_{\text{disen}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{cf}}$
end for
Freeze the trained encoder E and representation heads E_s, E_a
Mode organization and semantic-prior estimation
Establish mode assignments using the selected knowledge-based or data-driven organization
for each mode $k = 1, \ldots, K$ **do**
  Encode anomaly samples assigned to mode k and fit the diagonal Gaussian $p_k(z_a) = \mathcal{N}(\mu_k, \Sigma_k)$
  Build the empirical mask pool $\mathcal{M}_k$ from correspondences assigned to mode k
end for
Form $\Pi_A(z_a) = \sum_{k=1}^{K} \pi_k p_k(z_a)$ with $\pi_k = 1/K$
Residual-expert training
if Diagnosis is enabled **then**
  Train the lightweight router on paired representations $(z_{n,s}, z_a)$ using the established simulated mode assignments to produce normalized mode-association scores
end if
for each simulated training batch **do**
  Compute $z_{n,s} = E_s(E(x_n))$ and $z_a = E_a(E(x_a))$
  Form the observation-level effect $r = x_a - x_n$
  Hard-route each sample according to its established mode assignment k
  Update the selected expert using $\mathcal{L}_{\text{gen}}$ conditioned on $(z_{n,s}, z_a, m)$
end for
Mode selection for target generation
if Diagnosis is enabled and $\mathcal{D}_{\text{known}} \neq \emptyset$ **then**
  Evaluate the router on $(E_s(E(x)), E_a(E(x)))$ for each $x \in \mathcal{D}_{\text{known}}$ to obtain $q_k(x)$
  Compute $\bar{q}_k \leftarrow \frac{1}{|\mathcal{D}_{\text{known}}|} \sum_{x \in \mathcal{D}_{\text{known}}} q_k(x)$ for $k = 1, \ldots, K$
  Sort modes such that $\bar{q}_{k_{(1)}} \geq \cdots \geq \bar{q}_{k_{(K)}}$
  Set $J_\rho \leftarrow \min\left\{J : \sum_{j=1}^{J} \bar{q}_{k_{(j)}} \geq \rho\right\}$
  Set $\mathcal{K} \leftarrow \{k_{(1)}, \ldots, k_{(J_\rho)}\}$
else
  Set $\mathcal{K} \leftarrow \{1, \ldots, K\}$
end if
Target-domain supervision recovery
Initialize an empty pair list $\widehat{\mathcal{P}}_{\text{tar}}$
for each target reference $x_i^{\text{tar}} \in \mathcal{D}_{\text{tar}}^{\text{tr}}$ **do**
  Extract $z_{i,s}^{\text{tar}} = E_s(E(x_i^{\text{tar}}))$
  for each $k \in \mathcal{K}$ **do**
    Sample $\tilde{z}_a^{(k)} \sim p_k(z_a)$ and $m \sim \mathcal{M}_k$
    Generate $\hat{r}_i^{(k)}$ with U_k conditioned on $(z_{i,s}^{\text{tar}}, \tilde{z}_a^{(k)}, m)$
    Construct $\hat{x}_{i,a}^{(k)} = x_i^{\text{tar}} + \hat{r}_i^{(k)}$
    Append $\big((x_i^{\text{tar}}, \mathbf{0}), (\hat{x}_{i,a}^{(k)}, m)\big)$ to $\widehat{\mathcal{P}}_{\text{tar}}$
  end for
end for
Detector training
Train $D_\theta : \mathbb{R}^T \to [0,1]^T$ on $\widehat{\mathcal{P}}_{\text{tar}}$ using averaged point-wise BCE with equal weight on the reference and generated branches
Test-time scoring: Output $D_\theta(x)$ for each test sequence x.

B.2 Paired Representation Learning and Counterfactual Recomposition

Representation learning uses simulated correspondences (x_n, x_a, m, k), where x_n is the normal reference, x_a is its matched anomalous counterpart, m specifies the anomaly support, and k denotes a coarse simulated anomaly family. Each pair shares a temporal reference while exposing an anomaly-induced change. The family label k provides the default organization for subsequent prior estimation and expert training, while the representation-learning objective operates on the matched observations and their support.

As described in Sec. 4.1, the shared encoder and two representation heads produce $z_{n,s}$, $z_{a,s}$, $z_{n,a}$, and z_a. The reconstruction and background objectives constrain the representations to explain the observations and preserve their shared temporal background. The disentanglement objective $\mathcal{L}_{\text{dis}}$ encourages agreement between $z_{n,s}$ and $z_{a,s}$, while suppressing anomaly-semantic activation $z_{n,a}$ for the normal reference.

Within-pair counterfactual recomposition further constrains how the learned factors are used. The ordinary anomalous reconstruction uses the structure code $z_{a,s}$ extracted from x_a. For recomposition, we replace this code with $z_{n,s}$ from its matched normal reference, while retaining z_a and m:

$$\text{Dec}(z_{n,s}, z_a, m) = \text{base}(z_{n,s}) + \text{res}(z_{n,s}, z_a, m).$$

The reconstruction target remains the same matched anomalous observation x_a, giving

$$\mathcal{L}_{\text{cf}} = \text{MSE}\big(\text{Dec}(z_{n,s}, z_a, m), x_a\big).$$

Both representations originate from the same simulated pair. This constraint encourages anomaly information to remain usable with structure extracted from a normal observation. The residual decoder is conditioned on both factors, allowing the reconstructed effect to depend on the reference context.

The complete representation-learning objective is

$$\mathcal{L}_{\text{disen}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{cf}}.$$

All four terms are evaluated on matched-pair minibatches, and the encoder, representation heads, and auxiliary decoders are optimized jointly. After representation learning, the encoder and heads are fixed for semantic-prior estimation and residual-expert training.

B.3 Semantic Prior, Mode Organization, and Diagnosis

After paired representation learning, the encoder and representation heads are fixed, yielding a shared continuous anomaly-semantic space $\mathcal{Z}_A$. The mode index k provides a coarse organization of heterogeneous simulated anomaly semantics within this common space; it does not define separate latent spaces or an exhaustive taxonomy of target-domain anomalies. Continuous semantic variation is retained within each mode through z_a, while mode organization provides a coarse specialization structure for prior estimation and context-conditioned realization.

Mode organization. We consider two strategies for organizing the learned anomaly-semantic space. The default strategy is *knowledge-based*. Based on the temporal characteristics of the simulated anomaly operations, we organize them into three coarse modes,

$$K = 3: \qquad \text{trend}, \quad \text{periodic}, \quad \text{point}.$$

Here, $K = 3$ follows from this source-side organization rather than being selected through target-domain supervision or latent-space clustering. Each mode still contains continuous variation in its operation-specific configuration parameters.

As an alternative, we consider a fully *data-driven* organization that does not use the predefined trend–periodic–point assignments. After representation learning, the frozen anomaly-semantic codes z_a are organized directly in $\mathcal{Z}_A$. We evaluate

$$K \in \{1, 3, 5, 7\},$$

where $K = 1$ treats the semantic space as a single mode and $K > 1$ uses K-means clustering to obtain the corresponding mode assignments. These assignments replace the simulator-defined mode

labels throughout subsequent prior estimation and expert specialization. For the clustering-based variants, the same assignments also define the mode-specific mask pools and, when applicable, the supervision labels for the Diagnosis router.

The two strategies therefore examine different sources of semantic organization: the knowledge-based construction introduces a coarse inductive bias derived from temporal anomaly structure, whereas the data-driven construction derives the organization directly from the learned semantic representations. As evaluated in Appendix D.3, the knowledge-based three-mode organization achieves the strongest overall performance and is therefore adopted as the default setting.

Changing K in the data-driven variants changes not only the specialization granularity but, under our default generation protocol, also the total number of residual experts and the number of generated counterparts per target reference. Comparisons across different values of K should therefore be interpreted as end-to-end mode-organization configurations rather than as an isolated measure of specialization granularity. In contrast, the knowledge-based and data-driven configurations with $K = 3$ use the same number of modes and provide a more direct comparison of the two organization strategies.

Mode-wise semantic prior. For a given mode organization, let

$$\mathcal{Z}_k = \left\{ z_a^{(j)} : k_j = k \right\}$$

denote the anomaly-semantic codes assigned to mode k. We estimate a diagonal Gaussian

$$p_k(z_a) = \mathcal{N}(\mu_k, \Sigma_k),$$

where

$$\mu_k = \frac{1}{|\mathcal{Z}_k|} \sum_{z \in \mathcal{Z}_k} z,$$

and Σ_k is the diagonal empirical covariance of the corresponding semantic codes. The resulting simulation-induced semantic prior is

$$\Pi_A(z_a) = \sum_{k=1}^{K} \pi_k p_k(z_a), \qquad \pi_k = \frac{1}{K}.$$

The Gaussian components provide a coarse multimodal approximation within the same continuous semantic space rather than defining independent anomaly spaces. Under the standard generation procedure, each target reference receives one generated counterpart from each mode, consistent with the uniform mixture weights.

Expert specialization. During simulated-domain expert training, each sample is hard-routed according to its established mode assignment k. Under the knowledge-based organization, k is determined by the simulated anomaly family; under the data-driven organization, it is determined by the semantic-space clustering. The corresponding residual expert U_k is optimized only with samples assigned to that mode. Such coarse specialization is intended to reduce interference among heterogeneous semantic regions during context-conditioned realization.

The lightweight router used for Diagnosis neither determines the mode organization nor serves as the training gate for the residual experts. Mode assignments are established before expert training, whereas the router is used only to associate limited known target anomalies with the already constructed modes.

Diagnosis with limited known anomalies. The standard CAPS setting requires no target-domain anomaly labels and uses all available modes,

$$\mathcal{K} = \{1, \ldots, K\}.$$

When limited known target anomalies are available, CAPS(Diagnosis) uses a lightweight router to estimate their associations with the simulation-derived modes. During simulated training, the router takes $(z_{n,s}, z_a)$ from matched correspondences and is supervised by their established mode

assignments. For a known target anomaly x, its input is $(E_s(E(x)), E_a(E(x)))$, so no matched normal counterpart is required. It outputs normalized mode-association scores

$$q_k(x) \geq 0, \qquad \sum_{k=1}^{K} q_k(x) = 1.$$

Given a small known-anomaly set $\mathcal{D}_{\text{known}}$, we compute

$$\bar{q}_k = \frac{1}{|\mathcal{D}_{\text{known}}|} \sum_{x \in \mathcal{D}_{\text{known}}} q_k(x).$$

Let

$$\bar{q}_{k_{(1)}} \geq \bar{q}_{k_{(2)}} \geq \cdots \geq \bar{q}_{k_{(K)}}$$

denote the sorted associations. We use a cumulative-association threshold

$$\rho = 0.8$$

and define

$$J_\rho = \min \left\{ J : \sum_{j=1}^{J} \bar{q}_{k_{(j)}} \geq \rho \right\},$$

with the retained mode set

$$\mathcal{K}_\rho = \left\{ k_{(1)}, \ldots, k_{(J_\rho)} \right\}.$$

The selected modes are used equally for counterpart generation, corresponding to

$$\Pi_{A,\mathcal{K}_\rho}(z_a) = \frac{1}{|\mathcal{K}_\rho|} \sum_{k \in \mathcal{K}_\rho} p_k(z_a).$$

The known anomalies are used only for mode association and targeted counterpart generation rather than as detector-training labels. The router is unnecessary in the standard CAPS setting and is not used by the final detector at inference.

B.4 Architecture and Optimization Details

Table 3 summarizes the main architectural and optimization configurations used in CAPS. Unless otherwise specified, all reported results use an input length of 256 and the knowledge-based three-mode organization described in Appendix B.3. The representation and generation modules are trained in two stages. We first optimize the paired representation-learning objective and then freeze the encoder and representation heads. The fixed representations are subsequently used to estimate the semantic priors and train the hard-routed residual experts. The detector is trained independently on the recovered target-domain reference–counterpart pairs.

Two-stage representation and generation training. In the first stage, the shared encoder E, structure head E_s, anomaly-semantic head E_a, and auxiliary decoders are jointly optimized on minibatches of matched simulated normal–anomalous pairs using

$$\mathcal{L}_{\text{disen}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{cf}}.$$

We use Adam with a learning rate of 10^{-3}, a batch size of 32, and train the representation modules for 60 epochs. After this stage, E, E_s, and E_a are frozen, thereby fixing the learned structure and anomaly-semantic spaces.

In the second stage, the frozen representation modules are used to extract $z_{n,s}$ and z_a from the simulated correspondences. The mode-wise diagonal Gaussian priors $\{p_k\}_{k=1}^{K}$ are estimated from these fixed semantic codes, and the residual experts are optimized using $\mathcal{L}_{\text{gen}}$. Each simulated sample is hard-routed according to its established mode assignment. The residual-generation stage is trained for 100 epochs using Adam with the same learning rate and batch size. We use a diffusion horizon of 200 steps and run the full 200-step reverse process for counterpart generation. The lightweight router is trained only when CAPS(Diagnosis) is evaluated and does not participate in standard expert routing. The base and residual decoders in Table 3 are auxiliary reconstruction modules used during

representation learning. The diffusion experts operate directly in observation-level residual space; the clean-residual estimate defined in Sec. 4.2 is used to evaluate $\mathcal{L}_{\text{out}}$ during training, whereas counterpart generation uses the full reverse diffusion process.

Table 3: Main architectural and optimization configurations of CAPS.

Component	**Setting**
Input shape	$[B, 1, 256]$
Default mode organization	$K = 3$ (trend / periodic / point)
Latent dimensions	$\dim(z_s) = 128$, $\dim(z_a) = 48$
Encoder	hidden dims $[64, 128, 256]$, dropout 0.1
Multi-scale extractor	4 branches; kernels $(3, 5, 7, 9)$; dilations $(1, 2, 3, 4)$
Residual experts	K hard-routed experts ($K = 3$ by default)
Diagnosis router	hidden dim 128, $\rho = 0.8$ (optional)
Location embedding	64
Diffusion process	200 training timesteps; 200 reverse sampling steps
Diffusion U-Net	channels $[64, 128, 256, 256]$
Base decoder width	256
Residual decoder width	192
Detector	independent TCN with point-wise sigmoid output
Optimizer	Adam
Learning rate	1×10^{-3}
Batch size	32
Representation-learning epochs	60
Residual-generation epochs	100
Detector-training epochs	20

After residual-generator training, the representation and generation modules remain fixed while target-domain counterparts are constructed. The detector D_θ is then optimized independently for 20 epochs using Adam with the same learning rate and batch size. Training uses averaged point-wise binary cross-entropy on the recovered pair list, with equal weight assigned to the reference and generated branches. This balances the two branches of each recovered pair but does not imply balanced positive and negative labels across individual time points. No representation, generator, or semantic-prior parameters are updated during detector training.

Reproducibility. To facilitate reproducibility, we provide an anonymized repository containing the CAPS implementation, training and evaluation scripts, configuration files for the main experiments, and instructions for preparing the benchmark datasets: `https://github.com/gyf-0623/caps`. The configuration files specify the remaining implementation details, including the diffusion noise schedule. We additionally release the simulated normal–anomalous correspondences, including their mode assignments, anomaly supports, and reference identifiers, to support reproduction of paired representation learning and subsequent supervision construction.

C Experimental Setup and Detailed Results

C.1 Benchmark Datasets

We follow the TSB-AD benchmark [39] and conduct the main evaluation on nine public TSAD datasets: IOPS, MGAB, NAB, NEK, Power, SED, TODS, UCR, and YAHOO. We use the benchmark-defined training and test partitions throughout. For standard CAPS, the training portion serves as the unlabeled target-reference pool, with its anomaly labels unused, while the test portion is reserved for evaluation. These datasets span diverse application domains, including operations, sensors, web services, traffic, weather, and energy, and exhibit heterogeneous anomaly frequencies and temporal characteristics. Their basic statistics are summarized in Table 4, where AR denotes the anomaly ratio.

C.2 Baseline Methods

We compare CAPS with representative TSAD methods and controlled TCN-based training variants.

Table 4: Benchmark datasets used in the main experiments.

Name	Domain	#TS	Avg. Length	AR (%)
UCR	Misc.	228	67818.7	0.6
NAB	Web	28	5099.7	10.6
YAHOO	Web	259	1560.2	0.6
IOPS	Operations	17	72792.3	1.3
MGAB	Sensor	9	97777.8	0.2
SED	Energy	3	23332.3	4.1
TODS	Traffic	15	5000.0	6.3
NEK	Weather	9	1073.0	8.0
Power	Power Grid	1	35040.0	8.5

Conventional TSAD baselines. The external baselines include classical methods LOF [34] and IForest [35], together with deep TSAD models OmniAnomaly [5], USAD [3], TranAD [36], Anomaly Transformer [6], TimesNet [37], and FITS [38]. We follow the benchmark-recommended configurations whenever available. In particular, the input window is set to 10 for TranAD, 100 for USAD, and 100 for OmniAnomaly; LOF uses 50 neighbors, and IForest uses 200 estimators. The remaining benchmark baselines use their corresponding benchmark configurations.

Controlled TCN-based variants. To separate the effect of supervision construction from detector-backbone differences, we additionally construct four TCN-based variants using the same input length, preprocessing pipeline, and TCN detector backbone as CAPS. TCN-AE and TCN-Forecast use reconstruction and forecasting objectives, respectively. TCN-Injection employs perturbation-based synthetic anomaly training [28].

TCN-Supervised serves as a same-backbone label-supervised reference. It is trained on the labeled target-domain training split using point-wise binary cross-entropy, with the original temporal labels preserved; test labels are used only for evaluation. Unlike CAPS, which constructs matched reference–counterpart pairs and explicitly balances the reference and generated branches during detector training, TCN-Supervised follows the empirical anomaly frequency of the labeled training data and may therefore be affected by the severe class imbalance common in TSAD. We consequently treat it as a controlled supervised reference rather than as an optimized upper bound on supervised performance. Additional imbalance-aware supervised variants are examined separately in Appendix D.1.

These controlled variants allow us to distinguish the contribution of recovered supervision from gains attributable solely to the TCN architecture or exposure to synthetic anomalous samples. In particular, TCN-Injection examines whether the construction of synthetic supervision matters beyond synthetic anomaly exposure, while TCN-Supervised provides a reference for learning from factual target-domain labels under the same detector architecture.

We also report CAPS(Diagnosis). In this variant, 10% of anomalous windows from the target test partition are treated as limited known anomalies and are used only for mode association and targeted counterpart generation through the Diagnosis mechanism described in Appendix B.3. They are not used as detector-training labels. Evaluation is performed on the full test partition, including these known-anomaly windows; therefore, CAPS(Diagnosis) is treated as a target-guided diagnostic extension rather than part of the standard label-free setting.

C.3 Evaluation Metrics

We evaluate detection performance using Affiliation-F, $\mathrm{F1_T}$, Standard-F1, and VUS-PR. Following TSB-AD [39], we use Affiliation-F, Standard-F1, and VUS-PR, and follow [40] for $\mathrm{F1_T}$. Together, these metrics provide complementary point-wise, temporal, event-level, and range-aware perspectives on detection quality.

- **Standard-F1.** Standard-F1 is the conventional point-wise F1 score computed from precision and recall over individual timestamps.
- $\mathrm{F1_T}$**.** $\mathrm{F1_T}$ evaluates temporal anomaly detection while accounting for the correspondence between predicted and ground-truth anomaly regions, making it less dependent on exact point-wise alignment than Standard-F1.

- **Affiliation-F.** Affiliation-F evaluates predicted and ground-truth anomaly events through affiliation-based temporal matching [41], providing an event-oriented measure that is tolerant to moderate temporal misalignment.

- **VUS-PR.** VUS-PR is a threshold-independent and range-aware metric that integrates precision–recall performance over varying temporal tolerances [42], thereby accounting for uncertainty in anomaly boundaries and detection delay.

Threshold selection and VUS configuration. Following the TSB-AD evaluation protocol, Standard-F1, $\mathrm{F1_T}$, and Affiliation-F are evaluated using metric-specific oracle thresholds. For each test sequence and metric, we search over 1,500 equally spaced anomaly-score quantile levels and report the best value obtained using the test labels. VUS-PR is computed directly from continuous anomaly scores without selecting a single operating threshold. Its window length is automatically determined from the dominant period estimated using the autocorrelation function of the test sequence.

C.4 Implementation Settings

Unless otherwise specified, CAPS uses an input length of 256 and the knowledge-based three-mode organization described in Appendix B.3. The representation-learning, residual-generation, and detector-training stages are trained for 60, 100, and 20 epochs, respectively, using Adam with a learning rate of 10^{-3} and batch size 32. The diffusion model uses 200 training timesteps and the full 200-step reverse process for counterpart generation. CAPS-specific architectural, semantic-prior, mode-organization, and Diagnosis settings are provided in Appendix B.4.

For all controlled TCN variants, the target-domain input construction, preprocessing procedure, and detector backbone are kept consistent with CAPS wherever applicable. Their differences therefore arise primarily from the training objective and the form of supervision provided to the detector. External benchmark methods retain their benchmark-specific model configurations rather than being forced into the CAPS input protocol.

C.5 Detailed Per-Dataset Results

Table 5 reports the complete per-dataset results corresponding to the aggregate comparison in the main text. We report Affiliation-F, $\mathrm{F1_T}$, Standard-F1, and VUS-PR on all nine benchmark datasets. CAPS and CAPS(Diagnosis) results are averaged over five runs. Table 22 reports the mean and standard deviation over five independent runs for CAPS and the controlled TCN references. To keep the comparison with the standard label-free setting clear, CAPS(Diagnosis) is excluded when determining the best and second-best methods; we separately mark whether Diagnosis improves over standard CAPS for each dataset–metric pair.

The detailed results show that the aggregate advantage of CAPS is not driven by a single dataset or evaluation criterion. Standard CAPS remains particularly strong under both event-oriented and point-/range-aware metrics, while the controlled comparisons indicate that neither the shared TCN backbone nor synthetic anomaly exposure alone explains the observed gains. The results also illustrate substantial variation across datasets, reinforcing the value of reporting complementary metrics rather than relying on a single evaluation protocol.

CAPS(Diagnosis) improves over standard CAPS on many dataset–metric pairs, showing that limited known anomalies can provide useful information for targeted mode selection. The improvement is not universal, however. In particular, restricting generation to modes associated with a small known-anomaly subset can discard useful semantic coverage when that subset is not fully representative of the target anomaly distribution. We therefore treat Diagnosis as an optional target-guided extension rather than part of the standard label-free CAPS setting.

Table 5: Performance on nine datasets under four metrics (higher is better). CAPS variants are averaged over five runs. Best and second-best results excluding **CAPS(Diagnosis)** are shown in bold and underline, respectively; a star marks the better result between **CAPS** and **CAPS(Diagnosis)** for each dataset–metric pair.

Metric	Method	IOPS	MGAB	NAB	Power	SED	UCR	NEK	TODS	YAHOO
Affiliation-F	LOF	81.06	68.44	75.75	66.76	63.85	73.53	84.74	60.58	75.63
	IForest	52.81	68.82	39.84	0.00	70.09	50.56	71.15	44.17	33.30
	OmniAnomaly	80.32	67.35	92.35	78.16	61.26	73.53	86.30	50.73	71.31
	TranAD	83.19	67.28	90.28	71.56	61.03	73.31	85.02	52.76	76.08
	USAD	71.08	67.81	91.54	76.48	55.60	76.00	71.13	47.90	53.05
	AnomTrans.	70.79	67.65	79.03	71.57	68.21	80.03	75.32	44.57	64.75
	TimesNet	85.04	66.93	89.24	69.08	66.98	80.55	**89.79**	70.36	83.13
	FITS	86.36	67.77	89.03	67.96	66.99	74.36	87.76	71.77	81.19
	TCN-AE	83.91	67.61	85.74	83.34	67.75	74.54	85.74	72.61	82.47
	TCN-Injection	87.58	69.13	88.89	85.04	70.94	84.32	81.52	72.05	83.22
	TCN-Supervised	84.35	67.59	89.62	85.96	68.47	72.40	85.24	71.75	83.11
	TCN-Forecast	76.05	67.27	84.06	84.80	67.68	72.81	85.83	72.48	82.64
	CAPS	**89.24**	**69.22**	**93.25**	**85.98**	**74.09**	**88.41**	86.30	**86.85***	**94.63***
	CAPS(Diagnosis)	89.83*	69.61*	93.81*	86.35*	75.82*	89.72*	86.52*	81.32	94.57
$F1_T$	LOF	27.97	1.15	35.76	19.80	9.60	8.31	63.57	31.63	55.93
	IForest	7.64	0.84	21.44	0.00	9.54	6.36	65.56	11.06	4.90
	OmniAnomaly	51.17	1.61	40.09	23.48	9.68	8.47	**82.20**	14.33	24.16
	TranAD	22.63	1.65	37.28	22.36	9.57	7.75	69.97	13.51	8.41
	USAD	20.99	4.07	**61.46**	28.23	9.54	14.63	70.64	20.85	9.35
	AnomTrans.	2.56	0.56	8.13	7.81	15.07	4.09	22.62	6.18	1.39
	TimesNet	25.12	0.50	28.80	15.77	7.90	4.69	42.68	17.97	7.52
	FITS	15.38	0.54	31.98	15.77	7.95	4.95	42.90	16.39	4.86
	TCN-AE	39.38	3.25	36.63	19.92	10.23	13.87	68.58	24.85	58.94
	TCN-Injection	48.37	**23.89**	46.91	27.04	21.59	**44.79**	64.45	17.26	35.68
	TCN-Supervised	41.92	1.94	42.50	20.13	11.01	9.56	72.93	22.54	59.52
	TCN-Forecast	30.24	1.56	37.01	19.82	10.15	8.50	68.89	23.42	61.61
	CAPS	**60.62**	7.58	57.90	**29.20**	**24.27***	41.53	80.15	**46.17***	**71.35**
	CAPS(Diagnosis)	61.90*	7.99*	59.21*	30.36*	15.76	45.26*	83.33*	40.77	75.23*
Standard-F1	LOF	30.28	1.05	24.04	12.18	4.11	4.70	56.92	25.77	48.95
	IForest	8.37	0.73	29.41	19.77	3.81	4.09	58.10	13.35	3.20
	OmniAnomaly	47.05	1.44	28.81	23.50	0.43	5.11	74.03	12.65	21.40
	TranAD	34.85	1.46	27.33	22.36	2.63	4.40	60.36	11.94	5.70
	USAD	30.66	3.89	**56.15**	28.24	3.41	10.74	62.91	23.87	7.21
	AnomTrans.	3.74	0.58	19.35	15.77	16.87	3.44	30.23	13.58	2.31
	TimesNet	22.52	0.45	24.98	15.77	7.90	3.84	43.90	17.58	6.97
	FITS	15.81	0.49	29.00	15.77	7.95	3.86	44.27	18.08	4.58
	TCN-AE	27.15	2.26	28.53	20.08	10.22	9.06	58.61	20.72	49.27
	TCN-Injection	46.37	**14.61**	38.00	27.00	21.42	35.65	60.35	16.60	31.21
	TCN-Supervised	42.38	1.56	34.35	20.03	10.92	5.24	62.58	19.23	51.24
	TCN-Forecast	18.65	1.30	28.56	19.77	10.16	4.94	61.32	20.56	54.63
	CAPS	**51.63**	4.54	51.78	**29.37**	**24.11***	**37.70**	**76.23**	**41.24***	**69.58**
	CAPS(Diagnosis)	52.14*	6.84*	51.82*	30.30*	15.78	41.24*	76.78*	37.22	73.55*
VUS-PR	LOF	19.43	0.57	21.18	9.31	6.81	2.39	58.52	49.14	41.37
	IForest	8.59	0.62	23.57	11.56	7.71	2.88	56.50	46.62	10.47
	OmniAnomaly	25.35	0.64	27.17	14.32	6.20	2.40	74.51	45.55	29.26
	TranAD	21.61	0.64	24.82	13.04	5.75	2.25	61.63	47.33	25.78
	USAD	16.58	0.75	**55.03**	18.68	4.37	8.85	58.53	56.36	14.15
	AnomTrans.	4.60	0.43	13.11	8.84	8.29	2.12	22.67	44.74	8.00
	TimesNet	17.94	0.42	20.45	7.73	4.70	2.37	44.73	57.06	25.08
	FITS	13.94	0.40	25.43	7.46	5.08	2.20	38.77	56.00	18.81
	TCN-AE	17.27	0.87	21.56	14.95	8.99	3.63	61.04	46.30	41.03
	TCN-Injection	48.69	**3.27**	34.97	**21.87**	17.50	26.07	60.76	49.35	42.35
	TCN-Supervised	33.62	0.61	30.66	11.85	8.50	2.87	67.52	76.61	44.28
	TCN-Forecast	11.19	0.57	20.85	11.25	8.92	2.37	63.35	45.72	43.27
	CAPS	**49.00**	1.55	52.35	19.42	**20.09***	**28.47**	**80.52**	**86.26***	**83.70**
	CAPS(Diagnosis)	50.45*	1.68*	52.73*	21.00*	13.69	31.40*	81.91*	81.92	84.32*

D Extended Experiments and Analyses

D.1 Supervision Construction and Controlled Comparisons

This section provides detailed results and analyses for the supervision-construction comparisons summarized in Table 2. We examine three complementary aspects: matching generated effects to their target references, constructing counterparts through residual parameterization, and comparing with direct source-residual injection under the same simulated anomaly source. These comparisons assess how the construction of synthetic supervision contributes to detection performance. We additionally examine imbalance-aware label-supervised training and robustness to contaminated target references.

D.1.1 Effect of Context–Effect Matching

A central design of CAPS is that each anomaly effect is conditionally realized under the same target context on which its counterpart is constructed. To examine the contribution of this correspondence, we construct a shuffled variant that preserves the learned generator and the amount of generated supervision but mismatches each realized effect with its target reference.

Specifically, for a target reference x_i^{tar}, standard CAPS constructs $\hat{x}_{i,a}^{(k)} = x_i^{\mathrm{tar}} + \hat{r}_i^{(k)}$, where $\hat{r}_i^{(k)}$ is generated conditionally on the structure representation of x_i^{tar}. In the shuffled variant, we instead use an effect generated for another randomly selected reference within the same mode, yielding

$$\hat{x}_{i,a}^{(k),\mathrm{shuffle}} = x_i^{\mathrm{tar}} + \hat{r}_{\pi(i)}^{(k)}, \qquad \pi(i) \neq i.$$

The corresponding anomaly support is shuffled together with the residual, so each generated counterpart receives the mask associated with the applied effect. The number of generated counterparts and the underlying residual generator are otherwise unchanged. This is the *Shuffled residuals* variant in Table 2.

Table 6: Effect of disrupting context–effect matching in CAPS. Higher is better. The better result for each dataset–metric pair is bolded.

Metric	Method	IOPS	MGAB	NAB	Power	SED	UCR	NEK	TODS	YAHOO
Affiliation-F	CAPS	**89.24**	**69.22**	**93.25**	**85.98**	**74.09**	**88.41**	**86.30**	**86.85**	**94.63**
	Shuffled residuals	83.41	68.04	86.84	78.81	70.16	81.99	81.11	75.03	88.06
$\mathrm{F1_T}$	CAPS	**60.62**	**7.58**	**57.90**	**29.20**	**24.27**	**41.53**	**80.15**	**46.17**	**71.35**
	Shuffled residuals	53.32	1.76	52.75	20.38	22.88	37.18	74.98	38.15	69.50
Standard-F1	CAPS	**51.63**	**4.54**	**51.78**	**29.37**	**24.11**	**37.70**	**76.23**	**41.24**	**69.58**
	Shuffled residuals	46.85	1.62	46.31	20.38	22.81	32.29	68.40	34.12	67.26
VUS-PR	CAPS	**49.00**	**1.55**	**52.35**	**19.42**	20.09	**28.47**	**80.52**	**86.26**	**83.70**
	Shuffled residuals	44.78	0.61	45.33	17.22	**20.88**	23.17	72.68	72.60	74.46

Table 6 provides the per-dataset results underlying the aggregate comparison in the main text. Disrupting context–effect matching reduces the average Affiliation-F, $\mathrm{F1_T}$, Standard-F1, and VUS-PR from $85.33/46.53/42.91/46.82$ to $79.27/41.21/37.78/41.30$, respectively. The degradation occurs on nearly all dataset–metric pairs. The generator and number of generated counterparts are preserved, and moving each residual together with its mask maintains their support correspondence. These results support the importance of realizing an anomaly effect for the particular reference on which it is applied: effects generated under a different temporal context provide less effective supervision on average, even when the same learned generator is used.

D.1.2 Effect of Residual-Form Construction

We next examine the role of constructing counterparts through an observation-level residual effect. Standard CAPS generates $\hat{r}_i^{(k)}$ conditioned on the target reference representation and constructs the counterpart as $\hat{x}_{i,a}^{(k)} = x_i^{\mathrm{tar}} + \hat{r}_i^{(k)}$. As an ablation, we replace this residual formulation with direct anomalous-sequence generation while keeping the remaining framework as unchanged as

possible. Background-preservation supervision outside the intended anomaly support is retained in the direct-generation variant, so the comparison primarily changes how the anomalous counterpart is parameterized. This corresponds to *w/o residual parameterization* in Table 2.

Table 7: Effect of removing residual-form counterpart construction in CAPS. Higher is better. The better result for each dataset–metric pair is bolded.

Metric	Method	IOPS	MGAB	NAB	Power	SED	UCR	NEK	TODS	YAHOO
Affiliation-F	CAPS	**89.24**	**69.22**	**93.25**	**85.98**	74.09	**88.41**	**86.30**	**86.85**	**94.63**
	w/o residual parameterization	84.91	68.07	90.61	85.11	**77.54**	87.26	81.16	77.92	92.27
$F1_T$	CAPS	**60.62**	**7.58**	**57.90**	**29.20**	**24.27**	**41.53**	**80.15**	**46.17**	**71.35**
	w/o residual parameterization	44.06	1.26	54.96	24.03	19.96	39.59	76.53	39.91	68.37
Standard-F1	CAPS	**51.63**	**4.54**	**51.78**	**29.37**	**24.11**	**37.70**	**76.23**	**41.24**	**69.58**
	w/o residual parameterization	37.98	1.15	48.91	24.02	19.87	35.67	72.43	35.78	69.34
VUS-PR	CAPS	**49.00**	**1.55**	**52.35**	**19.42**	20.09	**28.47**	**80.52**	**86.26**	**83.70**
	w/o residual parameterization	37.39	0.57	46.35	19.31	**24.94**	26.80	72.90	74.85	77.32

Table 7 shows that direct anomalous-sequence generation reduces the four average metrics from $85.33/46.53/42.91/46.82$ to $82.76/40.96/38.35/42.27$. The degradation occurs on most dataset–metric pairs, although direct generation performs better on Affiliation-F and VUS-PR for SED. These results support residual-form construction as a useful reference-preserving parameterization. By retaining the observed reference explicitly, CAPS allows the generator to focus on the anomaly-induced observation-level change while preserving the surrounding temporal background. This comparison concerns the parameterization of generated counterparts; it does not assume that the underlying physical anomaly mechanism is additive.

D.1.3 Alternative Synthetic Supervision

We further examine whether the improvement of CAPS can be explained simply by access to a favorable simulated anomaly source or by exposure to synthetic anomalous samples. We compare alternative supervision construction strategies under the same TCN detector backbone, complementing the aggregate results in Tables 1 and 2.

Direct source-residual injection. We first construct a controlled baseline using the same simulated anomaly source as CAPS. Given a simulated normal–anomalous correspondence, its observation-level residual is directly applied to a target reference to construct a synthetic counterpart. This variant bypasses both anomaly-semantic learning and target-conditioned realization: it transfers an effect already realized under a source context, rather than generating an effect according to the target reference. We refer to this variant as *Direct source-residual injection*, consistent with Table 2.

Table 8: Comparison with synthetic supervision constructed from the same simulated anomaly source. Results are averaged over the nine benchmark datasets. Higher is better.

Method	Affiliation-F	$F1_T$	Standard-F1	VUS-PR
Direct source-residual injection	81.58	43.16	39.96	42.58
CAPS	**85.33**	**46.53**	**42.91**	**46.82**

As shown in Table 8, CAPS outperforms direct source-residual injection on all four aggregate metrics. Because the simulated anomaly source and detector backbone are shared, the improvement supports the value of learning transferable anomaly semantics and realizing them under target temporal contexts, beyond directly reusing source-context effects.

This comparison evaluates the combined contribution of semantic learning and target-conditioned realization, since both stages are bypassed together. The context–effect matching and residual-form comparisons above provide complementary evidence for individual construction choices. The same

Table 9: Comparison with label-supervised TCN training under different class-imbalance treatments. All methods use the same TCN detector backbone. Higher is better.

Dataset	Method	Affiliation-F	$F1_T$	Standard-F1	VUS-PR
YAHOO	TCN-Supervised (BCE)	83.11	59.52	51.24	44.28
	Weighted BCE	85.10	59.90	56.90	45.70
	Balanced Window Sampling	83.10	37.30	34.50	39.30
	Focal Loss	83.70	38.50	35.70	40.30
	CAPS	**94.63**	**71.35**	**69.58**	**83.70**
NEK	TCN-Supervised (BCE)	85.24	72.93	62.58	67.52
	Weighted BCE	77.30	47.10	36.10	30.70
	Balanced Window Sampling	76.50	43.50	31.80	26.80
	Focal Loss	76.40	44.20	31.70	26.50
	CAPS	**86.30**	**80.15**	**76.23**	**80.52**
TODS	TCN-Supervised (BCE)	71.75	22.54	19.23	76.61
	Weighted BCE	74.00	23.80	21.40	76.20
	Balanced Window Sampling	73.60	27.00	22.60	75.90
	Focal Loss	73.50	24.90	21.90	75.90
	CAPS	**86.85**	**46.17**	**41.24**	**86.26**

direct source-residual injection baseline is also reported in Sec. D.4 to contextualize the comparison of residual-generation mechanisms.

Perturbation-based injection. We additionally compare with TCN-Injection in Table 5, an independent perturbation-based synthetic-training strategy using the same TCN detector backbone. TCN-Injection achieves average Affiliation-F, $F1_T$, Standard-F1, and VUS-PR scores of $80.30/36.66/32.36/33.87$, compared with $85.33/46.53/42.91/46.82$ for CAPS. These results support the importance of how synthetic anomalous supervision is constructed.

Perturbation-based injection also constructs anomalous samples on target backgrounds, so this comparison does not isolate context–effect matching. Together with direct source-residual injection and the preceding controlled comparisons, the results support the central design of CAPS: learning reusable anomaly semantics and adapting their realization to each target reference provides useful supervision beyond the synthetic constructions evaluated here.

D.1.4 Recovered versus Label-Supervised Training

TCN-Supervised serves as a supervised reference with the same detector backbone, rather than an optimized upper bound on supervised TSAD. It is trained directly on observed target-domain labels using point-wise BCE and therefore inherits the naturally sparse anomaly frequency of the training data. CAPS instead constructs matched reference–counterpart pairs and assigns equal training weight to the reference and generated branches.

This construction provides controlled exposure to generated anomalous counterparts without requiring target anomaly labels. Equal branch weights, however, do not imply balanced point-wise classes: the anomaly support m generally occupies only part of each generated counterpart, and the remaining timestamps retain nominal labels. Any benefit from this training structure should therefore be distinguished from explicit balancing of anomalous and nominal timestamps.

To examine whether standard class-imbalance treatments account for the gap to the supervised reference, we evaluate additional supervised strategies using the same TCN backbone. This analysis covers YAHOO, NEK, and TODS, which together contain 283 time-series files and have training anomaly ratios of 0.299%, 3.17%, and 3.66%, respectively. Alongside standard BCE, we consider Weighted BCE, Balanced Window Sampling, and Focal Loss.

Table 9 shows that the effects of imbalance-aware training vary across datasets and metrics. Weighted BCE improves the YAHOO results over standard BCE, whereas the standard supervised configuration remains stronger than the tested alternatives on NEK. On TODS, all three treatments improve the F1-based metrics, while slightly reducing VUS-PR. Thus, these treatments do not uniformly improve supervised detection performance.

CAPS remains stronger across the reported metrics on all three datasets under the evaluated protocol. The tested imbalance-aware strategies therefore do not close the gap to CAPS, suggesting that the comparison with standard BCE reflects more than the absence of these particular treatments. This observation does not establish that recovered supervision is generally superior to ground-truth labels: a more extensively optimized supervised pipeline may benefit from different objectives, sampling policies, or dataset-specific tuning.

The comparison highlights the practical value of the supervision constructed by CAPS. Each generated counterpart is contrasted with its own nominal reference, concentrating the training distinction on an anomaly-induced change under a shared temporal background. Pair construction also controls the exposure to generated anomalous sequences through equal reference and generated branch weights. These properties support matched supervision as a useful approach in label-scarce settings, although the present comparison does not isolate their individual contributions.

D.1.5 Robustness to Reference Contamination

CAPS assumes that target-domain training data are normal-dominated rather than strictly anomaly-free. Reference contamination can therefore introduce supervision noise: an existing anomaly in a reference receives the nominal label **0**, and any such anomaly retained outside the generated support m is also unmarked in the counterpart. This reference-label noise is not explicitly modeled by the coverage and realization terms in Proposition 1, motivating an empirical robustness analysis.

Natural contamination in benchmark data. Several benchmarks contain anomalies in their target training segments. In YAHOO, 122 of the 259 time-series files contain training-segment anomalies. TODS presents a more demanding case, with an average contaminated training-window ratio of approximately 74%. This window-level ratio measures how frequently reference windows contain anomalies; it should be distinguished from the proportion of anomalous timestamps reported above. A high contaminated-window ratio can coexist with a relatively low point-wise anomaly ratio.

CAPS achieves strong results on both datasets in Table 5, indicating that contaminated references do not necessarily prevent effective detector learning. These benchmark observations nevertheless combine contamination with other dataset characteristics and do not isolate its effect.

Controlled contamination on IOPS. We conduct a controlled experiment on IOPS to examine reference contamination more directly. Its 17 time-series files provide approximately 21,184 clean training windows, with a natural training contamination ratio close to zero, while the test portions provide approximately 26,953 anomalous candidate windows.

For each file, we sample 100 target training windows of length 256 and replace 0%, 1%, 2%, 5%, 10%, 20%, or 30% of them with real anomalous windows drawn from the test portions of other IOPS files. Anomaly labels are used only to construct this controlled contamination experiment; the selected windows are treated as nominal references during CAPS training. This is a diagnostic stress test, separate from the standard training protocol without target anomaly labels. The total number of reference windows remains fixed at 100 per file across contamination levels, and all remaining representation, generation, detector-training, and evaluation settings are unchanged. Each level is repeated five times, with results averaged over the 17 files and five runs. Because this experiment uses a restricted reference set, its zero-contamination result differs from the main IOPS result.

Table 10: Robustness of CAPS to controlled target-reference contamination on IOPS using 100 reference windows per file. Results are averaged over 17 time-series files and five runs. Higher is better.

Contamination	Affiliation-F	$\mathrm{F1_T}$	Standard-F1	VUS-PR
0%	**85.74**	**34.17**	**37.54**	**34.05**
1%	84.29	33.08	34.98	31.19
2%	83.94	33.05	35.48	30.77
5%	84.48	31.06	32.69	29.44
10%	84.53	30.83	31.65	29.46
20%	84.20	30.61	29.22	26.52
30%	82.55	28.70	33.31	26.66

Table 10 shows an overall decline as contamination increases, although individual metrics are not strictly monotonic. Affiliation-F remains comparatively stable under mild contamination, while $\mathrm{F1_T}$, Standard-F1, and VUS-PR show larger reductions relative to the clean-reference setting. At 20%–30% contamination, CAPS continues to produce useful detection results, but the degradation is more apparent, particularly for VUS-PR.

These results suggest some tolerance to reference-label noise under the tested contamination protocol, while also demonstrating its cost. Clean windows remain the majority at every tested level, so the experiment does not establish robustness when contaminated references dominate. The findings support the use of normal-dominated target data and identify reference quality as a practical factor affecting recovered supervision.

D.2 Disentanglement and Semantic Recombination

We next examine how paired representation learning supports the separation of temporal structure and transferable anomaly semantics. We first evaluate the contribution of within-pair counterfactual recomposition, and then inspect the learned representations and their recombination across different reference trajectories. These analyses complement the aggregate ablation results in Table 2 and the qualitative examples in Fig. 2.

D.2.1 Effect of Counterfactual Recomposition

The representation-learning objective combines reconstruction, background preservation, paired disentanglement, and within-pair counterfactual recomposition. In particular, $\mathcal{L}_{\mathrm{cf}}$ reconstructs an anomalous sample using its anomaly-semantic code and the structure code of its matched normal reference:

$$\mathcal{L}_{\mathrm{cf}} = \mathrm{MSE}\big(\mathrm{Dec}(z_{n,s}, z_a, m), x_a\big).$$

Compared with reconstructing x_a from its own structure code $z_{a,s}$, this operation substitutes the matched normal structure $z_{n,s}$ while retaining the anomaly-semantic code z_a and support m. It encourages the two branches to jointly explain the observed anomalous change: the normal structure provides the reference context, while the anomaly-semantic code supplies the information needed to reconstruct its anomalous counterpart.

To assess this contribution, we remove $\mathcal{L}_{\mathrm{cf}}$ while retaining $\mathcal{L}_{\mathrm{rec}}$, $\mathcal{L}_{\mathrm{base}}$, and $\mathcal{L}_{\mathrm{dis}}$, with the remaining training procedure unchanged. This corresponds to the counterfactual-loss ablation summarized in Table 2.

Table 11: Effect of within-pair counterfactual recomposition. Results are averaged over the nine benchmark datasets. Higher is better.

Variant	Affiliation-F	$\mathrm{F1_T}$	Standard-F1	VUS-PR
w/o $\mathcal{L}_{\mathrm{cf}}$	83.65	42.76	39.11	43.11
CAPS	**85.33**	**46.53**	**42.91**	**46.82**

Table 11 shows that removing $\mathcal{L}_{\mathrm{cf}}$ reduces performance across all four aggregate metrics. The result supports the value of explicitly training the representations to reconstruct the anomalous outcome after substituting its matched normal structure. This requirement goes beyond reconstructing each observation from its own codes and encourages a representation organization suited to subsequent semantic transfer. The training constraint operates within matched simulated pairs; cross-context recombination is examined separately below.

D.2.2 Latent Representation Analysis

We inspect the learned representations using two-dimensional UMAP projections of the structure codes z_s and anomaly-semantic codes z_a. Figure 3 provides a complementary visualization to Fig. 2(a). The two visualizations use the same trained model but independently sampled simulated anomaly realizations, resulting in different projection inputs. Both exhibit similar qualitative organization of the learned representations.

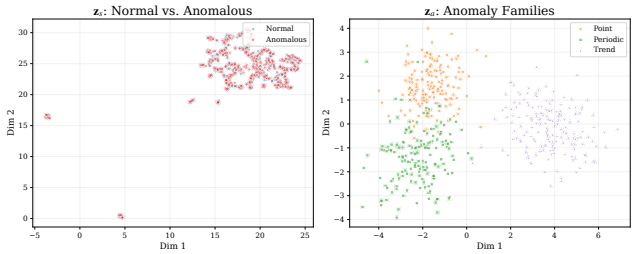


Figure 3: Learned representation spaces. *Left*: structure codes z_s colored by normal/anomalous status. *Right*: anomaly-semantic codes z_a colored by simulated anomaly family.

Normal and anomalous samples largely overlap in the projected structure space z_s, consistent with reduced sensitivity to anomaly presence in the structure branch. The anomaly-semantic space z_a, in contrast, exhibits clearer organization across the point, periodic, and trend families. This organization is consistent with the use of simulated families as coarse modes within a shared semantic space, supporting mode-wise prior estimation and expert specialization.

These projections provide qualitative evidence consistent with the intended division of representation roles. They do not by themselves establish statistical independence between z_s and z_a or semantic invariance across contexts. The recombination analysis below further examines how the learned semantic codes interact with different reference structures during counterpart construction.

D.2.3 Cross-Context Semantic Recombination

We further examine the use of learned anomaly semantics by recombining fixed semantic codes with different reference trajectories. Figure 4, corresponding to Fig. 2(b), presents a 3×3 set of generated counterparts. Each row uses a different reference trajectory, while each column reuses one fixed anomaly-semantic code z_a. The Point, Periodic, and Trend column labels identify the coarse simulated families associated with these codes. Colored diagonal cells indicate the original reference–semantic pairings, and off-diagonal cells show cross-context recombinations.

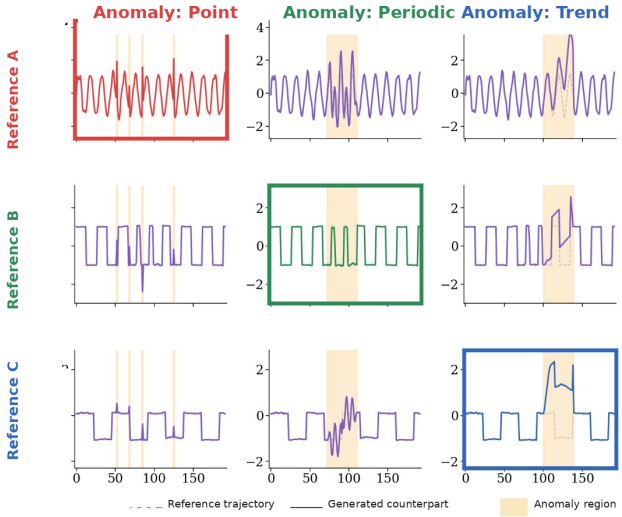


Figure 4: Cross-context semantic recombination. Rows use different references, and each column reuses a fixed anomaly-semantic code z_a. Colored diagonal cells mark the original pairings. Dashed curves denote references, solid curves denote generated counterparts, and shaded regions indicate anomaly supports.

Across the displayed references, each fixed semantic code produces a recognizable anomaly pattern, while the resulting counterparts exhibit different observation-level effects. These examples illustrate

the intended distinction between reusing anomaly semantics and reusing an already realized source residual. In CAPS, the semantic code supplies transferable anomaly information, while the generator also receives the reference structure and intended support to construct the counterpart.

The generated trajectories also largely preserve the reference background outside the intended anomaly support. Together with the counterfactual-loss ablation and latent visualization, these observations support the central role of paired representation learning in supervision recovery: learning anomaly information that can be recombined with reference structure and subsequently realized as context-anchored training counterparts. The cross-context examples are qualitative diagnostics of the learned model, rather than additional training constraints.

D.3 Semantic Coverage and Mode Organization

We next examine the coverage and organization of the simulation-induced semantic prior. CAPS uses coarse source-side modes to organize a shared continuous anomaly-semantic space and support expert specialization. These modes do not constitute an exhaustive taxonomy of target anomalies, which may exhibit mixed, context-dependent, or previously unseen patterns. The transferable representation is the continuous semantic code, while the mode index provides a coarse organization for prior estimation and generation. We study three aspects of this design: sensitivity to reduced coverage, alternative mode organizations, and targeted mode selection with limited target anomaly guidance.

D.3.1 Coverage Stress Test

The usefulness of simulation-derived supervision depends partly on whether the semantic prior captures anomaly variation relevant to the target domain. We examine this dependence by removing the source-side mode most strongly associated with each target dataset.

Given a small known-anomaly set used only for this diagnostic analysis, the router produces normalized mode-association scores $q_k(x)$. We compute

$$\bar{q}_k = \frac{1}{|\mathcal{D}_{\text{known}}|} \sum_{x \in \mathcal{D}_{\text{known}}} q_k(x), \qquad k^\star = \arg\max_k \bar{q}_k.$$

During target counterpart construction, we exclude the prior component and residual expert associated with $k^\star$. Additional counterparts are sampled from the remaining modes so that the total number of generated counterparts per reference is unchanged. The known anomalies identify the mode to exclude; they are not supplied as labeled examples for detector training.

Table 12: Semantic-coverage stress test. The source-side mode with the highest target association is excluded during counterpart generation while the total number of counterparts is preserved. Results are averaged over the nine benchmark datasets. Higher is better.

Method	Affiliation-F	$\mathrm{F1_T}$	Standard-F1	VUS-PR
CAPS w/o highest-associated mode	79.31	38.42	37.81	39.87
CAPS	**85.33**	**46.53**	**42.91**	**46.82**

Table 12 shows lower aggregate performance after removing the highest-associated mode. Because the number of generated counterparts is preserved, the degradation supports the importance of the anomaly variation available for supervision construction beyond the amount of synthetic training data. The remaining modes still provide useful supervision, although this does not establish coverage of arbitrary unseen anomaly patterns.

The excluded mode is defined by router association, rather than a ground-truth anomaly category for the target dataset. Real datasets may contain heterogeneous patterns that are only partially represented by any single source-side family. The stress test therefore examines the consequence of restricting an associated region of the source prior. Its results are consistent with the role of coverage discussed in Sec. 4.4, without treating router associations as a direct measurement of the theoretical coverage discrepancy.

D.3.2 Mode Construction and Granularity

We next examine the dependence of CAPS on its default knowledge-based mode organization. This organization uses $K = 3$ coarse modes corresponding to point, periodic, and trend variations in the simulated anomaly operations. All anomaly-semantic codes remain in the shared continuous space $\mathcal{Z}_A$. The modes organize prior estimation and expert specialization; they do not define separate latent spaces or assume an additive physical decomposition of target anomalies.

As a data-driven alternative, we freeze the learned representation modules and organize the simulated anomaly-semantic codes z_a directly in the latent space. We evaluate $K \in \{1, 3, 5, 7\}$, where $K = 1$ uses a single mode and $K > 1$ uses K-means clustering. The resulting assignments replace the predefined source-family labels in prior estimation, expert training, and mask-pool construction. When Diagnosis is used, they also provide the corresponding router supervision, as described in Appendix B.3.

Table 13: Comparison of semantic-mode organization and granularity. Results are averaged over the nine benchmark datasets. Higher is better.

Organization	K	Affiliation-F	$\mathrm{F1_T}$	Standard-F1	VUS-PR
Data-driven	1	82.34	39.92	37.23	41.36
Data-driven	3	84.98	43.75	40.12	45.76
Data-driven	5	83.10	44.04	40.43	44.76
Data-driven	7	81.61	42.38	38.78	43.38
Knowledge-based	3	**85.33**	**46.53**	**42.91**	**46.82**

Table 13 shows that data-driven organizations also yield effective detectors. In particular, clustering with $K = 3$ provides a competitive alternative, indicating that the learned semantic space supports mode organization without retaining the predefined family assignments in the subsequent generation stage.

At the same $K = 3$, the knowledge-based organization achieves higher aggregate performance across all four metrics. This comparison holds the number of experts and generated counterparts per reference constant, providing a more direct assessment of the organization strategy. The result supports the usefulness of coarse temporal characteristics as an inductive bias for specialization. One possible explanation is that these assignments group compatible simulated variations for residual generation, although the present comparison does not directly measure interference between experts.

Among the data-driven variants, increasing K does not produce uniform improvements. Relative to $K = 3$, $K = 5$ slightly improves $\mathrm{F1_T}$ and Standard-F1 but reduces Affiliation-F and VUS-PR, while $K = 7$ performs worse on all four. Changing K also changes the number of residual experts and, under the standard generation protocol, the number of generated counterparts per reference. These results therefore compare complete mode-organization configurations rather than isolating specialization granularity.

The findings support a coarse organization of continuous anomaly semantics without requiring target anomalies to be assigned to predefined categories. They do not imply that continuous latent variation alone guarantees coverage of anomaly patterns absent from simulation. The coverage stress test above highlights this remaining dependence on the source prior.

D.3.3 Diagnosis and Targeted Mode Selection

The mode organization also provides an optional interface for incorporating limited target anomaly guidance. Standard CAPS uses all source-side modes without target anomaly labels, preserving broad semantic coverage when the relevant target variation is unknown.

CAPS(Diagnosis) uses known target anomalies to select associated source-side modes for counterpart generation. In this diagnostic experiment, we randomly sample 10% of anomalous windows from the target test partition to form $\mathcal{D}_{\mathrm{known}}$. The router produces normalized association scores $q_k(x)$, which

are averaged as

$$\bar{q}_k = \frac{1}{|\mathcal{D}_{\text{known}}|} \sum_{x \in \mathcal{D}_{\text{known}}} q_k(x).$$

We sort the modes such that

$$\bar{q}_{k_{(1)}} \geq \cdots \geq \bar{q}_{k_{(K)}}$$

and retain the smallest subset whose cumulative association reaches ρ:

$$J_\rho = \min \left\{ J : \sum_{j=1}^{J} \bar{q}_{k_{(j)}} \geq \rho \right\}, \qquad \mathcal{K}_\rho = \{k_{(1)}, \ldots, k_{(J_\rho)}\}.$$

We use $\rho = 0.8$. Counterparts are generated uniformly from the retained modes; the association scores determine mode inclusion rather than generation mixture weights.

The known anomalies guide supervision construction through mode selection but are not directly included as labeled examples in detector training. Evaluation uses the full test partition, including the sampled known-anomaly windows. Accordingly, these results describe a diagnostic extension under partial test-set guidance, rather than the standard setting without target anomaly labels or an independent evaluation entirely on unseen test windows.

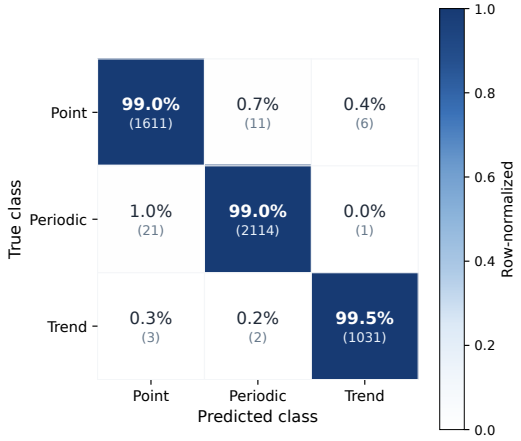


Figure 5: Router associations on simulated samples. Rows denote source-side modes and columns denote router-associated modes. Values are row-normalized percentages, with sample counts in parentheses.

Figure 5 shows that the reported router associations largely agree with the simulated mode assignments, with most entries concentrated on the diagonal. This supports consistency with the source-side organization on the displayed samples. It does not establish that real target anomalies belong to the same discrete categories. For target data, the scores serve as relative associations with the available source modes.

The detailed results in Table 5 show that CAPS(Diagnosis) improves over standard CAPS on many dataset–metric pairs, but not universally. Representative known anomalies may help focus generation on relevant semantic variation. Conversely, a small known subset may omit anomaly patterns present elsewhere in a heterogeneous dataset, causing mode selection to discard useful coverage. Under the one-counterpart-per-selected-mode protocol, selection also changes the amount of generated supervision, so the comparison evaluates the targeted-generation procedure as a whole.

Diagnosis therefore provides a way to adapt generation when limited target guidance is available, without guaranteeing a performance improvement. Standard CAPS retains all modes to preserve broader coverage when such guidance is unavailable.

D.4 Context-Conditioned Realization

Having examined semantic representation and coverage, we next study how anomaly semantics are realized on target references. In CAPS, z_a provides anomaly-semantic information, while the reference structure z_s and intended support m also condition the generated observation-level effect. This stage connects the simulation-induced semantic prior to context-anchored target supervision. We examine alternative residual-generation mechanisms and compare generated counterparts with real target anomalies.

D.4.1 Alternative Realization Mechanisms

We examine how the choice of realization mechanism affects the supervision constructed for detector learning. Standard CAPS uses conditional diffusion to generate residual effects from anomaly semantics, reference structure, and anomaly support. We compare this implementation with a VAE-style conditional residual generator. We additionally report *Direct source-residual injection*, the same baseline used in Table 8, which applies simulated source residuals directly to target references.

The VAE comparison replaces the conditional generation mechanism, whereas direct source-residual injection bypasses both semantic learning and target-conditioned realization. The latter therefore provides a reference for the complete transfer procedure rather than an isolated generator ablation. All variants use the same TCN detector backbone and evaluation protocol.

Table 14: Comparison of anomaly-effect construction strategies. Results are averaged over the nine benchmark datasets. Higher is better.

Strategy	Affiliation-F	$\mathrm{F1_T}$	Standard-F1	VUS-PR
Direct source-residual injection	81.58	43.16	39.96	42.58
VAE residual generator	83.28	42.32	38.82	42.63
CAPS (diffusion)	**85.33**	**46.53**	**42.91**	**46.82**

Table 14 shows that conditional diffusion achieves the highest aggregate performance across all four metrics. The two alternatives exhibit mixed relative performance: the VAE generator performs better on Affiliation-F and slightly better on VUS-PR, while direct source-residual injection performs better on $\mathrm{F1_T}$ and Standard-F1. Thus, introducing a learned conditional generator alone does not uniformly improve over direct injection; its implementation also matters for the resulting detector supervision.

The comparison with the VAE supports the diffusion implementation used in CAPS, while the comparison with direct source-residual injection supports the combined semantic-learning and target-conditioned realization procedure. These results do not establish diffusion as a necessary component of supervision recovery. Among the evaluated implementations, however, it provides the most effective residual generator for downstream detection. Its computational cost is examined in the efficiency analysis below; generation is used for offline supervision construction, while inference requires only the trained detector.

D.4.2 Generated-Counterpart Diagnostics against Real Anomalies

Downstream detection results do not directly characterize the relationship between generated counterparts and observed target anomalies. We therefore complement the detector comparisons with two diagnostics in detector-score and feature spaces.

We first train a detector using standard CAPS supervision. For held-out normal references not used in detector training, we construct equal numbers of counterparts using rule-based injection, direct source-residual injection, and CAPS. The three strategies share the same reference windows, anomaly supports, and sample counts. We then pass the generated counterparts and held-out real target anomalies through the same fixed detector.

Score measures the Wasserstein distance between the detector-score distributions of generated and real anomalies. *Feature* measures the average distance from each generated counterpart to its five nearest real-anomaly neighbors in the penultimate-layer feature space. Lower values indicate greater

proximity under the corresponding detector-derived representation. Real anomaly labels are used for this diagnostic evaluation, not for standard CAPS detector training.

Table 15: Detector-oriented diagnostics against real target anomalies, averaged over the nine benchmark datasets. Score measures Wasserstein distance between anomaly-score distributions; Feature measures average generated-to-real 5-NN distance in the penultimate-layer feature space. Lower is better.

Positive source	Score ↓	Feature ↓
Rule-based injection	0.4531	0.6218
Direct source-residual injection	0.4783	0.6712
CAPS	**0.3990**	**0.6123**

Table 15 shows that CAPS has the lowest average discrepancy under both diagnostics. Its generated counterparts are closer to real anomalies in detector-score distribution and in the selected feature-space measure. The feature-distance improvement over rule-based injection is modest, so these averages should not be interpreted as establishing a large or uniform advantage across datasets.

These observations complement the supervision-construction comparisons in Appendix D.1. Direct source-residual injection reuses effects already realized under simulated contexts, while rule-based injection constructs effects through predefined perturbations. CAPS instead learns anomaly-semantic representations and conditions their realization on the target reference. The smaller reported distances are consistent with this procedure producing supervision relevant to the real anomalies encountered by the detector.

The interpretation remains specific to the diagnostic setting. Because the detector is trained on CAPS-generated supervision, its scores and features are not independent of the method being evaluated and may favor aspects of CAPS counterparts. Moreover, generated-to-real nearest-neighbor distance measures local proximity rather than coverage of the full real-anomaly distribution. These diagnostics therefore provide complementary observations in a downstream representation space; they do not establish recovery of true counterfactual outcomes or complete agreement with the distribution of real target anomalies.

D.5 Robustness, Scalability, and Efficiency

We finally examine the sensitivity of CAPS to detector architecture and semantic-prior choices, its behavior under different amounts of simulated training data, and its computational cost. These analyses assess the flexibility of recovered supervision and distinguish the cost of offline representation learning and counterpart generation from that of detector inference.

D.5.1 Detector Architecture Variants

The main experiments use a lightweight TCN detector. To examine the applicability of recovered supervision to a different architecture, we replace the TCN with a Transformer while retaining the supervision-construction procedure and training protocol.

Table 16: Comparison of CAPS under different detector architectures. Results are averaged over the nine benchmark datasets. Higher is better.

Metric	CAPS (TCN)	CAPS (Transformer)
Affiliation-F	**85.33**	81.10
$F1_T$	**46.53**	42.12
Standard-F1	**42.91**	40.37
VUS-PR	**46.82**	43.28

Table 16 shows that the recovered pairs can also support training a Transformer detector, extending their use beyond the default TCN. The TCN achieves higher aggregate performance across all four metrics under the evaluated protocol and is retained as the default. The comparison supports flexibility

in detector choice, while also showing that the downstream architecture affects how effectively the recovered supervision is used.

D.5.2 Prior Sensitivity

CAPS uses a parametric prior to sample learned anomaly semantics during target-domain counterpart generation. The default implementation fits a diagonal Gaussian within each coarse mode. We compare this choice with an isotropic Gaussian, a full-covariance Gaussian, and a Student-*t* distribution, while keeping the mode organization and remaining settings unchanged.

Table 17: Sensitivity to the parametric form of the mode-wise anomaly-semantic prior. Results are averaged over the nine benchmark datasets. Higher is better.

Prior	Affiliation-F	$\mathrm{F1_T}$	Standard-F1	VUS-PR
Isotropic Gaussian	84.97	45.88	42.36	46.21
Diagonal Gaussian	85.33	46.53	**42.91**	46.82
Full-covariance Gaussian	**85.40**	**47.11**	42.74	46.70
Student-*t*	85.21	46.47	42.68	**46.95**

Table 17 shows similar aggregate performance across the tested prior families, with no single choice achieving the best result on every metric. The full-covariance Gaussian performs best on Affiliation-F and $\mathrm{F1_T}$, the diagonal Gaussian on Standard-F1, and the Student-*t* prior on VUS-PR. The diagonal Gaussian therefore offers a competitive default using only per-dimension means and variances.

These results suggest limited sensitivity to the tested within-mode parameterizations under the current semantic organization. They concern how the learned codes are approximated for sampling; they do not establish that changing the prior family can compensate for anomaly variation absent from the simulated source.

D.5.3 Different Amounts of Simulated Data

The representation and generation stages are trained using simulated normal–anomalous correspondences. To examine their dependence on the amount of source supervision, we compare the default 48,000 simulated pairs with smaller sets of 5,000, 10,000, and 20,000 pairs, keeping the remaining settings unchanged.

Table 18: Performance under different amounts of simulated training data, averaged over the nine benchmark datasets. Higher is better.

Metric	5,000	10,000	20,000	48,000
Affiliation-F	77.66	78.93	82.15	**85.33**
$\mathrm{F1_T}$	44.36	**47.24**	46.21	46.53
Standard-F1	41.05	**43.52**	42.78	42.91
VUS-PR	39.68	42.40	44.79	**46.82**

Table 18 shows different responses across metrics. Affiliation-F and VUS-PR improve consistently as the number of simulated pairs increases. In contrast, $\mathrm{F1_T}$ and Standard-F1 reach their highest values at 10,000 pairs and remain comparatively close at the larger scales. Thus, additional simulation benefits some aspects of detection more clearly than others.

Smaller simulated sets already support useful detector training. We retain 48,000 pairs as the default because this configuration achieves the strongest Affiliation-F and VUS-PR while maintaining comparable F1 performance. The results also identify smaller source sets as practical alternatives when offline training resources are limited.

Because the epoch counts and batch size remain fixed, larger source sets also entail more optimization updates. The comparison therefore measures the effect of increasing simulation scale under the adopted training schedule, rather than isolating data quantity at a fixed optimization budget. The associated training costs are reported below.

D.5.4 Computational Efficiency

The representation and generation modules are trained offline. Afterward, they are reused to construct counterparts for target references, and a separate detector is trained on the recovered supervision. At deployment, anomaly scoring requires only this detector; the representation encoder, semantic prior, and diffusion generator are not part of the inference path.

Table 19 reports simulated-domain training time at different source-data scales, together with per-pair generation and test-time detection costs. These measurements distinguish the cost of constructing supervision from the cost of scoring unseen sequences. Target-domain detector fitting is a separate stage and is not itemized in this table.

Table 19: Computational cost under different amounts of simulated training data. Pair generation is performed offline; test-time detection uses only the trained TCN.

Number of simulated pairs	Simulated-domain training	Pair generation	Test-time detection
5,000	23.6 min	50 ms / pair	2 ms / sequence
10,000	48.5 min	50 ms / pair	2 ms / sequence
20,000	95.1 min	50 ms / pair	2 ms / sequence
48,000	180.5 min	50 ms / pair	2 ms / sequence

Simulated-domain training time increases with the amount of source supervision. Under the fixed model and sampling configuration, the reported generation cost remains approximately 50 ms per pair, while the deployed detector takes approximately 2 ms per test sequence. The total cost of counterpart construction additionally depends on the number of target references and generated counterparts.

These measurements illustrate the computational separation between supervision recovery and deployment. CAPS incurs representation-learning and generation costs before detector deployment, allowing the conditional generator to support training without adding diffusion sampling to test-time scoring.

E Multivariate Extension

The main formulation of CAPS focuses on temporal anomaly effects without explicitly modeling dependencies among variables. We examine its extension to multivariate series through a variable-wise supervision-recovery procedure. This supplementary experiment evaluates the usefulness of the temporal supervision across variables, while leaving explicit dependency modeling for future work.

We evaluate on MSL, PSM, SMAP, and SMD. Table 20 summarizes the multivariate sequences included in these experiments.

Table 20: Multivariate datasets used in the supplementary experiments. #TS denotes the number of evaluated multivariate sequences, #Dim the number of input variables per sequence, and AR the anomaly ratio.

Name	Domain	#TS	#Dim	Avg. Length	AR (%)
MSL	Space	16	55	3119.4	5.1
PSM	Sensor	1	25	217624.0	11.2
SMAP	Space	27	25	7855.9	2.9
SMD	Server	22	38	25466.4	3.8

For CAPS, we apply the univariate pipeline independently to each variable. The pretrained representation and generation modules are shared across variables, while a separate TCN detector is trained for each variable without detector parameter sharing. The resulting variable-wise anomaly scores are aggregated at each timestamp to obtain a multivariate score.

We evaluate two aggregation strategies. **CAPS (Max)** takes the maximum variable-wise score, whereas **CAPS (Top5-mean)** averages the five largest scores. Neither strategy introduces a module

for learning cross-variable interactions. Because the extension uses one detector per variable, its parameter count and computational cost differ from those of the single-detector univariate setting.

We compare with the multivariate implementations of LOF, IForest, OmniAnomaly, TranAD, USAD, Anomaly Transformer, TimesNet, and FITS under the same data splits and evaluation protocol.

Table 21: Results on multivariate anomaly detection benchmarks. Higher is better. Best and second-best results within each dataset–metric pair are shown in **bold** and underline, respectively.

Metric	Model	MSL	PSM	SMAP	SMD
Affiliation-F	LOF	84.35	61.98	63.32	64.13
	IForest	63.36	63.78	59.96	69.71
	OmniAnomaly	83.15	58.17	91.38	85.82
	TranAD	79.91	73.83	87.39	**92.20**
	USAD	81.86	57.86	87.25	85.09
	AnomTrans.	74.38	66.04	74.82	73.44
	TimesNet	84.38	69.43	86.69	86.90
	FITS	82.32	75.55	81.41	89.53
	CAPS (Max)	86.64	**82.71**	**92.87**	89.37
	CAPS (Top5-mean)	**88.51**	79.68	91.78	88.62
$F1_T$	LOF	38.97	25.58	21.81	10.13
	IForest	20.73	25.39	14.32	16.20
	OmniAnomaly	49.36	30.42	46.63	51.84
	TranAD	39.42	25.49	29.12	37.98
	USAD	48.71	28.96	43.94	50.41
	AnomTrans.	5.91	19.55	7.18	4.82
	TimesNet	26.95	20.55	23.47	34.91
	FITS	23.86	20.15	21.47	29.92
	CAPS (Max)	**55.07**	**34.09**	**47.61**	54.23
	CAPS (Top5-mean)	54.27	32.22	44.37	**57.11**
Standard-F1	LOF	30.65	18.80	18.70	8.41
	IForest	14.68	24.15	13.61	16.89
	OmniAnomaly	39.10	30.43	40.50	57.06
	TranAD	29.60	25.63	25.11	43.99
	USAD	38.71	28.41	38.66	53.06
	AnomTrans.	10.65	23.37	8.02	9.86
	TimesNet	20.13	20.15	23.37	39.48
	FITS	16.96	20.15	19.99	34.18
	CAPS (Max)	**46.04**	**31.18**	**43.33**	52.39
	CAPS (Top5-mean)	40.04	25.40	40.84	**57.83**
VUS-PR	LOF	24.67	13.58	10.59	4.40
	IForest	11.29	15.85	7.55	8.88
	OmniAnomaly	31.57	18.58	28.07	37.44
	TranAD	14.78	16.49	13.37	28.34
	USAD	29.95	17.59	26.37	34.53
	AnomTrans.	8.15	16.90	4.39	5.68
	TimesNet	14.23	15.22	15.38	34.58
	FITS	13.98	14.02	14.85	30.32
	CAPS (Max)	**40.15**	**20.37**	**39.46**	50.06
	CAPS (Top5-mean)	31.93	19.54	35.04	**54.63**

Table 21 shows strong performance from the variable-wise extension despite its lack of explicit cross-variable modeling. CAPS (Max) achieves the best result on 11 of the 16 dataset–metric pairs, while CAPS (Top5-mean) achieves the best result on four. TranAD achieves the highest Affiliation-F on SMD.

Max aggregation performs better on most metrics for MSL, PSM, and SMAP, whereas Top5-mean is stronger on three of the four metrics for SMD. These differences illustrate the effect of score aggregation: Max emphasizes the strongest individual response, while Top5-mean combines evidence from several variables. Neither strategy is uniformly preferable across the evaluated datasets and metrics.

The results support the usefulness of recovered temporal supervision in a variable-wise multivariate pipeline. Each detector learns from reference-anchored pairs for its corresponding variable, and aggregation combines the resulting anomaly evidence at the decision level.

This construction cannot explicitly model anomalies expressed only through changing correlations or conditional relationships among otherwise plausible individual trajectories. The experiment therefore establishes a practical extension of the temporal supervision-recovery procedure, while leaving cross-variable anomaly structure unmodeled. Incorporating such dependencies into representation learning and counterpart construction remains a direction for future work.

Table 22: Performance variability of CAPS and controlled TCN references across nine datasets. Per-dataset results are reported as mean ± standard deviation over five independent runs. Avg. is the unweighted average of the nine dataset means. Higher is better. The best mean among the methods listed in each dataset–metric group is shown in bold.

Metric	Method	IOPS	MGAB	NAB	Power	SED	UCR	NEK	TODS	YAHOO	Avg.
Affiliation-F	TCN-AE	83.91 ± 0.80	67.61 ± 1.69	85.74 ± 1.66	83.34 ± 0.91	67.75 ± 1.08	74.54 ± 0.43	85.74 ± 1.43	72.61 ± 0.42	82.47 ± 0.43	78.19
	TCN-Injection	87.58 ± 0.72	69.13 ± 1.56	88.89 ± 1.37	85.04 ± 0.41	70.94 ± 0.46	84.32 ± 1.40	81.52 ± 0.84	72.05 ± 0.20	83.22 ± 1.66	80.30
	TCN-Supervised	84.35 ± 0.70	67.59 ± 0.98	89.62 ± 0.97	85.96 ± 1.36	68.47 ± 1.71	72.40 ± 1.12	85.24 ± 1.40	71.75 ± 0.67	83.11 ± 1.05	78.72
	TCN-Forecast	76.05 ± 1.47	67.27 ± 0.95	84.06 ± 1.09	84.80 ± 1.51	67.68 ± 0.55	72.81 ± 0.92	85.83 ± 0.81	72.48 ± 0.24	82.64 ± 1.18	77.07
	CAPS	$\mathbf{89.24 \pm 0.29}$	$\mathbf{69.22 \pm 0.70}$	$\mathbf{93.25 \pm 0.62}$	$\mathbf{85.98 \pm 0.97}$	$\mathbf{74.09 \pm 0.88}$	$\mathbf{88.41 \pm 0.66}$	$\mathbf{86.30 \pm 0.28}$	$\mathbf{86.85 \pm 0.97}$	$\mathbf{94.63 \pm 0.60}$	**85.33**
$F1_T$	TCN-AE	39.38 ± 1.02	3.25 ± 0.19	36.63 ± 1.23	19.92 ± 0.52	10.23 ± 0.28	13.87 ± 0.26	68.58 ± 1.43	24.85 ± 0.21	58.94 ± 0.97	30.63
	TCN-Injection	48.37 ± 0.92	$\mathbf{23.89 \pm 0.84}$	46.91 ± 0.43	27.04 ± 0.35	21.59 ± 0.76	$\mathbf{44.79 \pm 0.57}$	64.45 ± 1.00	17.26 ± 0.20	35.68 ± 0.50	36.66
	TCN-Supervised	41.92 ± 0.35	1.94 ± 0.17	42.50 ± 0.39	20.13 ± 0.35	11.01 ± 0.49	9.56 ± 0.55	72.93 ± 0.80	22.54 ± 0.38	59.52 ± 0.97	31.34
	TCN-Forecast	30.24 ± 0.44	1.56 ± 0.05	37.01 ± 0.68	19.82 ± 0.82	10.15 ± 0.21	8.50 ± 0.28	68.89 ± 0.75	23.42 ± 0.22	61.61 ± 1.29	29.02
	CAPS	$\mathbf{60.62 \pm 1.04}$	7.58 ± 0.45	$\mathbf{57.90 \pm 0.26}$	$\mathbf{29.20 \pm 1.03}$	$\mathbf{24.27 \pm 0.31}$	41.53 ± 0.22	$\mathbf{80.15 \pm 0.08}$	$\mathbf{46.17 \pm 0.78}$	$\mathbf{71.35 \pm 1.02}$	**46.53**
Standard-F1	TCN-AE	27.15 ± 0.60	2.26 ± 0.61	28.53 ± 0.72	20.08 ± 0.53	10.22 ± 0.39	9.06 ± 0.50	58.61 ± 0.50	20.72 ± 0.18	49.27 ± 1.09	25.10
	TCN-Injection	46.37 ± 0.68	$\mathbf{14.61 \pm 0.66}$	38.00 ± 0.89	27.00 ± 0.40	21.42 ± 0.74	35.65 ± 0.84	60.35 ± 1.36	16.60 ± 0.30	31.21 ± 0.50	32.36
	TCN-Supervised	42.38 ± 0.93	1.56 ± 0.05	34.35 ± 0.74	20.03 ± 0.51	10.92 ± 0.33	5.24 ± 0.29	62.58 ± 1.29	19.23 ± 0.43	51.24 ± 1.26	27.50
	TCN-Forecast	18.65 ± 0.53	1.30 ± 0.13	28.56 ± 0.53	19.77 ± 0.67	10.16 ± 0.36	4.94 ± 0.39	61.32 ± 1.02	20.56 ± 0.26	54.63 ± 1.31	24.43
	CAPS	$\mathbf{51.63 \pm 0.59}$	4.54 ± 0.45	$\mathbf{51.78 \pm 0.22}$	$\mathbf{29.37 \pm 1.71}$	$\mathbf{24.11 \pm 0.21}$	$\mathbf{37.70 \pm 0.18}$	$\mathbf{76.23 \pm 0.56}$	$\mathbf{41.24 \pm 0.45}$	$\mathbf{69.58 \pm 1.02}$	**42.91**
VUS-PR	TCN-AE	17.27 ± 0.59	0.87 ± 0.07	21.56 ± 0.55	14.95 ± 0.23	8.99 ± 0.53	3.63 ± 0.30	61.04 ± 1.04	46.30 ± 0.23	41.03 ± 0.60	23.96
	TCN-Injection	48.69 ± 1.16	$\mathbf{3.27 \pm 0.18}$	34.97 ± 1.02	$\mathbf{21.87 \pm 0.68}$	17.50 ± 0.30	26.07 ± 0.86	60.76 ± 0.77	49.35 ± 0.25	42.35 ± 0.82	33.87
	TCN-Supervised	33.62 ± 0.99	0.61 ± 0.06	30.66 ± 0.82	11.85 ± 0.28	8.50 ± 0.35	2.87 ± 0.16	67.52 ± 0.98	76.61 ± 0.23	44.28 ± 0.76	30.72
	TCN-Forecast	11.19 ± 0.35	0.57 ± 0.05	20.85 ± 0.48	11.25 ± 0.25	8.92 ± 0.34	2.37 ± 0.15	63.35 ± 0.62	45.72 ± 0.39	43.27 ± 0.73	23.05
	CAPS	$\mathbf{49.00 \pm 0.60}$	1.55 ± 0.23	$\mathbf{52.35 \pm 0.66}$	19.42 ± 1.45	$\mathbf{20.09 \pm 0.19}$	$\mathbf{28.47 \pm 0.14}$	$\mathbf{80.52 \pm 0.76}$	$\mathbf{86.26 \pm 1.17}$	$\mathbf{83.70 \pm 1.21}$	**46.82**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction describe CAPS as a supervision-recovery framework based on transferable anomaly semantics, target-conditioned realization, and context-anchored detector supervision. The problem formulation and experiments clarify the assumptions, scope, and distinction between standard CAPS and the optional Diagnosis setting.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The conclusion and supplementary analyses discuss dependence on simulated semantic coverage, sensitivity to contaminated target references, and the lack of explicit cross-variable dependency modeling. Additional experiments examine these limitations and the computational cost of supervision recovery.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 4.4 states Proposition 1 with its finite-risk, integrability, and uniform Lipschitz assumptions. Appendix A provides the proof, the derivation of the contextual realization gap, and additional discussion of the regularity conditions, learning-error term, and scope of the analysis.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The method sections and Algorithm 1 describe representation learning, semantic-prior estimation, residual-expert training, counterpart construction, detector training, and inference. Appendices B.4 and C provide implementation and evaluation details, complemented by the released code and experiment configurations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Reproducibility can be accomplished in various ways, including code release, detailed implementation instructions, model checkpoints, or other appropriate means.
   - If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
   - If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
   - If the contribution is a new model, there should either be a way to access this model for reproducing the results or a way to reproduce the model.

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

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

   Answer: [Yes]

   Justification: Appendix B.4 links the repository containing the CAPS implementation, training and evaluation scripts, experiment configurations, and data-preparation instructions. The released simulated normal–anomalous pairs include mode assignments, anomaly supports, and reference identifiers.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 and Appendices B.4 and C specify the data splits, preprocessing, metrics, threshold selection, architecture, and optimization settings. Mode organization and Diagnosis are described in Appendix B.3, with additional comparison protocols provided in the corresponding experimental subsections.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The appendix reports per-dataset means and standard deviations over five independent runs for CAPS and the controlled TCN references. These statistics quantify variability across repeated runs. The reported uncertainty is the standard deviation, and the comparisons are not presented as formal significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated.
   - The method for calculating the error bars should be explained.
   - The assumptions made should be given.
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Implementation settings and computational costs are provided in Appendices B.4 and D.5.4, covering simulated-domain training, counterpart generation, and detector inference.

   Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research uses public time-series anomaly detection benchmarks and simulated normal–anomalous pairs. It does not involve human-subject experiments or private personal-data collection, and is intended for methodological evaluation under the stated benchmark protocols.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity.

10. **Broader impacts**

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

Answer: [Yes]

Justification: The paper discusses potential benefits for monitoring applications and potential risks associated with false alarms and missed anomalies.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments.
- The authors should consider possible harms that could arise when the technology is used as intended and functioning correctly, when it gives incorrect results, and through misuse.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.

11. **Safeguards**

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

Answer: [N/A]

Justification: The released assets concern time-series anomaly detection code and simulated anomaly pairs. The work does not introduce high-risk general-purpose models or datasets containing scraped personal information.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model.
- Datasets that have been scraped from the Internet could pose safety risks.
- We recognize that providing effective safeguards is challenging, and many papers do not require this.

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The datasets and baseline implementations are credited to their original sources and used in accordance with their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license should be included for each asset.
- For scraped data from a particular source, the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: The repository linked in Appendix B.4 documents the CAPS implementation and released simulated pairs, including their mode assignments, anomaly supports, and reference identifiers. Training, evaluation, and data-preparation instructions accompany these assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets if applicable.

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

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

Answer: [N/A]

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

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so institutional human-subject review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval or equivalent may be required for human subjects research.
- We expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed method. Language-tool assistance is limited to writing, editing, or formatting and does not determine the method, experimental results, or theoretical analysis.

Guidelines:

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