Test-Time Graph Anomaly Detection via Shifted Augmentation with Dynamic Objective Scheduling

Chunjing Xiao[1,2], **Ruobing Fan**[1], **Tian Zhao**[1], **Fan Zhou**[3], **Ying Ma**[4], **Chong Tang**[5]*

[1]School of Computer and Information Engineering, Henan University
[2]State Key Laboratory of Spatial Datum, College of Remote Sensing and Geoinformatics Engineering, Faculty of Geographical Science and Engineering, Henan University
[3]School of Information and Software Engineering, University of Electronic Science and Technology of China
[4]Faculty of Computing, Harbin Institute of Technology
[5]School of Engineering Mathematics and Technology, University of Bristol

xiaocj@henu.edu.cn, ruobingfan@henu.edu.cn, zhaotian@henu.edu.cn, fan.zhou@uestc.edu.cn, y.ma@hit.edu.cn, chong.tang@bristol.ac.uk

Abstract

Graph anomaly detection (GAD) is critical to many real-world systems. However, supervised GAD models often degrade under distribution shifts due to the closed-world training assumption. Test-time adaptation (TTA) offers a promising solution by updating pre-trained models with unlabeled test data. Yet, applying TTA to GAD poses a unique dilemma. High-confidence samples provide reliable pseudo-labels, but they are scarce and biased toward low-shift regions. Low-confidence samples better reflect target shifts, but they are too noisy for direct cross-entropy optimization and often require label-free objectives that are misaligned with anomaly discrimination. To address this dilemma, we propose SADOS, a Shifted Augmentation-based Dynamic Objective Scheduling framework for test-time GAD. SADOS turns unreliable target samples into drift-carrying signals for reliable supervised adaptation. It distills prototypes from confidence-stratified unreliable samples and injects them into reliable samples to generate label-preserving yet shift-oriented augmentations. This expands coverage over shifted target regions. In parallel, SADOS dynamically schedules the adaptation objective. It uses contrastive learning to capture early-stage drift signals, and gradually anneals its weight so that classification-oriented anomaly detection dominates later adaptation. Extensive experiments demonstrate that SADOS robustly improves test-time GAD under distribution shifts.

1 Introduction

Graph anomaly detection (GAD) aims to identify anomalous entities in graph-structured data, enabling a wide range of real-world applications from fraud detection and abusive medical claim identification to social network analysis [32, 23]. Despite the appeal of learning without labels, unsupervised GAD methods can be easily confounded by noise due to the lack of prior knowledge about the anomalies of interest, leading to unstable detection performance [46, 42, 32]. Consequently, a growing body of recent work has moved toward supervised formulations that leverage available labels to learn more discriminative representations and decision boundaries [39, 36, 50, 45]. However, most existing GAD models are trained under a closed-world assumption, where training and test data are expected to follow the same distribution. When deployed in evolving environments with distribution shifts (e.g., emerging normal types or changing behavioral patterns), these models can suffer substantial performance degradation [28].

*Corresponding author.

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

To adapt models under distribution shifts, test-time adaptation (TTA) updates a pre-trained model using unlabeled target data at deployment time [21, 49]. Existing TTA methods typically rely on pseudo-label self-training, entropy-based prediction regularization, or self-supervised objectives [54, 51, 40, 1, 15]. While effective in general settings, applying them to graph anomaly detection (GAD) is non-trivial because target samples exhibit sharply different reliability: high-confidence samples provide useful supervision but cover only low-shift regions, whereas low-confidence samples better reflect target drift but are too noisy for direct classification training. This leads to two key challenges:

C1: *Coverage bias of reliable samples.* Reliable samples are inherently constrained by limited quantity and biased coverage, making it difficult to adapt to the most drifted regions of the target distribution. Reliable samples are routinely leveraged in self-training pipelines to refine pre-trained models for target-domain adaptation. However, their contribution is often limited in the presence of substantial distribution shift: they are typically sparse and biased toward low-shift regions of the target distribution, leaving the more drifted regions under-adapted. This issue arises because cross-entropy in self-training is highly sensitive to noisy pseudo-labels (even a small amount of noise can severely harm performance), thus existing methods usually adopt stringent confidence thresholds to obtain relatively uncontaminated labels [26, 22]. Such aggressive filtering inevitably reduces the number of selected samples (especially for rare anomalous cases) [54, 15], and may bias them toward target regions closer to the source distribution. Consequently, the more shifted portions of the target data remain insufficiently covered during adaptation.

C2: *Optimization-objective misalignment of unreliable samples.* Unreliable samples often require label-free objectives whose optimization may deviate from the GAD decision boundary. Unreliable samples tend to better cover the drifted target distribution and are therefore crucial for adapting to real-world shifts [54, 15]. However, their pseudo-labels are substantially noisier, making direct cross-entropy self-training unstable [10]. Consequently, many recent approaches resort to label-free objectives (e.g., contrastive/clustering losses or prototype-anchored regularization) to exploit unreliable samples during test-time fine-tuning [15, 12]. Nevertheless, these objectives primarily emphasize representation learning and distribution alignment (e.g., enforcing feature invariance/consistency), rather than directly optimizing the anomaly-discriminative decision boundary required by GAD, which may lead to objective misalignment. Moreover, generic label-free adaptation can introduce additional risks. For instance, entropy-minimization-style updates may bias target features toward dominant classes and thereby deteriorate performance [16, 15]. As a result, how to effectively leverage unreliable samples to capture target distribution drift while maintaining alignment with the original discriminative classification objective remains a key challenge.

To address these challenges, we propose SADOS, a Shifted Augmentation-based Dynamic Objective Scheduling framework for test-time GAD. SADOS jointly exploits reliable and unreliable target samples rather than treating them separately. It first partitions test nodes by node-adaptive confidence and further stratifies unreliable nodes into confidence groups. To overcome the limited coverage of reliable samples, SADOS distills uncertainty-aware prototypes from these unreliable groups and injects them into reliable samples through confidence-aware shifted augmentation. The resulting samples preserve reliable pseudo-labels while inheriting drift patterns from the target distribution, enabling supervised fine-tuning beyond low-shift regions. Meanwhile, to avoid over-optimizing label-free objectives on noisy unreliable samples, SADOS adopts dynamic objective scheduling: contrastive learning is emphasized early to capture target drift, and its weight is gradually annealed so that classification-oriented fine-tuning dominates later adaptation.

Our main contributions are summarized as follows:

- We propose SADOS, a test-time adaptation framework for GAD that jointly uses reliable samples for supervision and unreliable samples as carriers of target-shift information.
- We introduce confidence-aware shifted augmentation, which distills prototypes from confidence-stratified unreliable samples and injects them into reliable samples to generate label-preserving, drift-oriented augmentations.
- We design a dynamic objective scheduling strategy that transitions from early contrastive drift exploration to later supervised anomaly classification, and demonstrate consistent gains over strong GAD and graph TTA baselines on eight benchmark datasets under distribution shifts.

Extensive experiments demonstrate the effectiveness of SADOS compared with strong baselines. Source code is provided at `https://github.com/ChunjingXiao/SADOS`.

2 Related Work

Graph Anomaly Detection. Graph anomaly detection (GAD) aims to identify anomalous entities in graph-structured data. Depending on the availability of anomaly labels, existing methods can be broadly categorized into unsupervised and supervised approaches. Unsupervised GAD methods typically model normal patterns without anomaly annotations, leveraging representation learning, reconstruction, or contrastive objectives [2, 6, 20, 44, 53]. Recent advances further enhance unsupervised detection by modeling feature–structure inconsistencies or exploiting graph foundation models to improve generalization, such as CPF [18], AnomalyGFM [31], and HUGE [27]. In contrast, supervised GAD methods exploit labeled anomaly data to learn discriminative decision boundaries and have shown strong performance in applications such as fraud detection and cybersecurity [8, 20, 56, 4, 19]. Recent GAD studies have also explored attribute and structure-aware modeling to distinguish different anomaly patterns [47]. Despite their methodological differences, both paradigms are commonly developed under a closed-world assumption that training and test data follow similar distributions [28]. This assumption is frequently violated in real-world deployments, leading to degraded robustness under distribution shifts. To improve generalization, prior work has explored open-set GAD [32, 2] and cross-domain GAD [12], aiming to handle unseen domains [28] or anomaly types. Early cross-domain GAD studies, including COMMANDER [7] and ACT [41], established and developed the cross-domain anomaly detection setting. GTGD [12] represents a more recent development in this line of research. However, these approaches often require additional training objectives or specialized modules, which may limit their applicability and flexibility in continuously shifting deployment environments.

Graph Test-Time Adaptation. Test-time adaptation (TTA) has emerged as a promising paradigm for addressing distribution shifts by dynamically refining pre-trained models using unlabeled test streams. A prominent line of research leverages *cross-entropy self-training* to refine models using trustworthy pseudo-labels, such as FreeTTA [5], AdaPAC [15], TSA [11] and ASSESS [54]. Another line adopts *unsupervised entropy minimization* (or related prediction-regularization objectives) to reduce prediction uncertainty on the target stream, including GTRANS [13], COME [51], PGNDS [55], and Shift-ACT [16]. While effective, these paradigms are often subject to a coverage bias toward reliable samples. In parallel, *self-supervised* graph TTA methods exploit task-related constraints to adapt without explicit reliance on hard labels, such as Matcha [1], GraphPatcher [14], and GADT3 [29]. Nevertheless, these approaches face a fundamental trade-off when leveraging unreliable samples. Although such samples are crucial for capturing drifted target regions, they are typically optimized with generic label-free objectives geared toward representation alignment rather than anomaly discrimination.

3 Method

We now present SADOS and describe the design and functionality of its core components, as illustrated in Figure 1.

3.1 Confidence-Adaptive Node Partitioning

Given a test graph, we first use a pre-trained GAD detector to assign each node a pseudo-label and an associated confidence score. We then partition test nodes into a *reliable* set and an *unreliable* set based on confidence, and further stratify the unreliable set into multiple confidence groups. The reliable subset is treated as having relatively trustworthy pseudo-labels and is used for classification-oriented fine-tuning. In contrast, the unreliable groups exhibit higher uncertainty yet provide richer signals of distribution drift. This stratification facilitates subsequent prototype extraction and shifted augmentation, thereby expanding coverage toward drifted target regions.

Specifically, existing methods [28, 10, 48] typically adopt shared or class-wise thresholds to separate reliable and unreliable samples. However, such coarse thresholding ignores node-level heterogeneity and can lead to unstable selection under distribution shifts [54]. To address this issue, we propose a node-adaptive confidence calibration scheme and use the calibrated confidence to distinguish reliable nodes from unreliable ones and further stratify unreliable nodes into confidence tiers.

To calibrate the confidence, we introduce a node-adaptive correction term δ_v based on the mutual information (MI) between node v and its neighborhood $\mathcal{N}(v)$. We use MI as a structure-aware

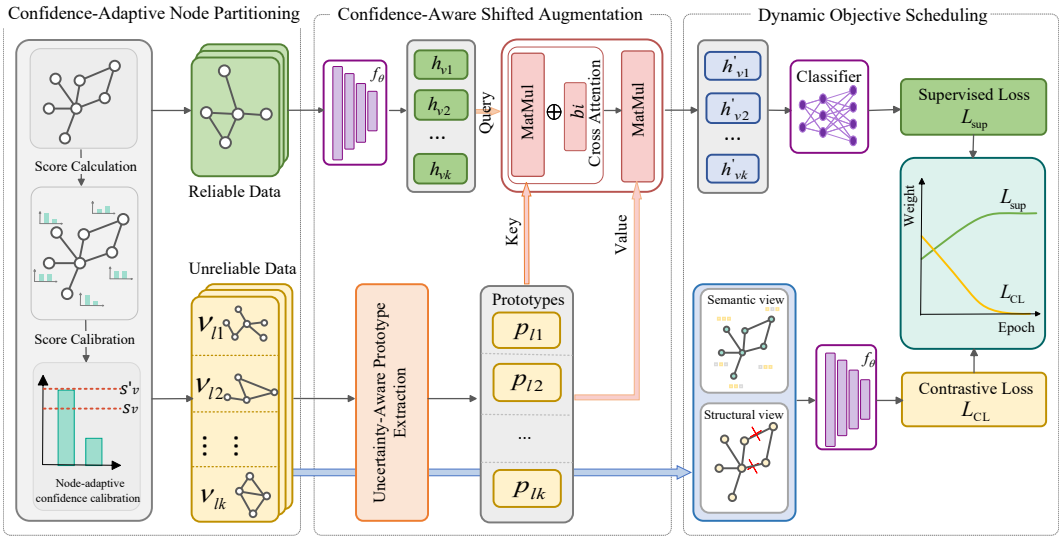


Figure 1: The Overview of the SADOS framework.

consistency signal between a node and its neighborhood, which correlates with the reliability of the model's prediction under structural heterogeneity [54]. We compute the correction term as:

$$\delta_v = \lambda \cdot \mathrm{norm}\big(I(v;\mathcal{N}(v))\big), \tag{1}$$

where $\mathrm{norm}(\cdot)$ is a normalization function and λ controls the correction strength. We estimate $I(v;\mathcal{N}(v))$ efficiently via an InfoNCE-style objective. Let $\mathbf{h}_v$ denote the representation of node v, and $\mathbf{h}_{\mathcal{N}(v)}$ be an aggregated neighborhood representation (e.g., mean pooling). Then the InfoNCE-style neighborhood agreement score is approximated by

$$I(v;\mathcal{N}(v)) \approx \log \frac{\exp\big(\mathrm{sim}(\mathbf{h}_v, \mathbf{h}_{\mathcal{N}(v)})/r\big)}{\sum_{u\in\mathcal{B}} \exp\big(\mathrm{sim}(\mathbf{h}_v, \mathbf{h}_{\mathcal{N}(u)})/r\big)}, \tag{2}$$

where $\mathrm{sim}(\cdot,\cdot)$ is a similarity function (e.g., cosine similarity), r is a temperature, and $\mathcal{B}$ is a mini-batch that provides negative samples through mismatched node–neighborhood pairs.

We treat the MI-derived correction δ_v as a node-wise calibration of the original confidence score s_v, which is the predicted probability produced by the pre-trained model. Instead of directly adding the correction in the probability space, we perform calibration in the logit space:

$$s'_v = \sigma\left(\mathrm{logit}(s_v) + \delta_v\right), \tag{3}$$

where $\mathrm{logit}(x) = \log \frac{x}{1-x}$, $\sigma(\cdot)$ denotes the sigmoid function, and s_v is clipped to $(\epsilon, 1-\epsilon)$ in implementation for numerical stability. Intuitively, a higher MI score increases the log-odds of the prediction confidence, making v more likely to be assigned to a higher-quality group while keeping $s'_v \in (0,1)$.

Calibrated confidence-based node partition. Given a base threshold τ_{L0}, we extract the reliable set $\mathcal{V}_h$ and the unreliable set $\mathcal{V}_l$ using the calibrated confidence:

$$\mathcal{V}_h = \{\, v \mid s'_v \geq \tau_{L0} \,\}, \quad \mathcal{V}_l = \{\, v \mid s'_v < \tau_{L0} \,\}. \tag{4}$$

We further stratify $\mathcal{V}_l$ into K confidence groups using a decreasing sequence of thresholds $\tau_{L1} > \tau_{L2} > \cdots > \tau_{LK}$:

$$\mathcal{V}_{li} = \{\, v \in \mathcal{V}_l \mid \tau_{L(i-1)} > s'_v \geq \tau_{Li} \,\}, \quad i = 1, \ldots, K. \tag{5}$$

As a result, we obtain a reliable node set $\mathcal{V}_h$ and K confidence-stratified unreliable node sets $\{\mathcal{V}_{l1}, \ldots, \mathcal{V}_{lK}\}$, where $\mathcal{V}_{l1}$ contains the highest-confidence unreliable nodes and $\mathcal{V}_{lK}$ contains the lowest-confidence ones.

3.2 Confidence-Aware Shifted Augmentation

Reliable samples selected from the test stream are typically regarded as having relatively trustworthy pseudo-labels and can thus be used to fine-tune a pre-trained detector for improved target adaptation [54, 22]. However, to suppress pseudo-label noise, selection strategies often adopt stringent criteria, which makes reliable samples scarce and biased toward low-shift regions close to the source distribution [54, 26]. As a result, relying solely on reliable samples provides limited coverage of the more drifted parts of the target distribution. To overcome this limitation, we transform reliable pseudo-normal samples into drift-oriented augmented samples and use them for supervised fine-tuning, thereby improving the model's ability to adapt to shifted target regions.

To this end, we propose a confidence-aware shifted augmentation strategy that generates augmentations satisfying two constraints: (i) preserving the pseudo-label of the original reliable sample, and (ii) shifting it toward drifted target regions. We achieve this by injecting uncertainty-aware prototypes distilled from confidence-stratified unreliable target groups into reliable samples. Specifically, for each unreliable group, we first extract an uncertainty-aware prototype that summarizes shared and drift-related patterns while suppressing sample-specific noise. We then adopt a confidence-aware injection scheme, where prototypes from higher-confidence groups contribute more, whereas those from lower-confidence groups contribute less. Since higher-confidence prototypes are more likely to be label-consistent with reliable samples, assigning them larger weights helps preserve the original pseudo-label. Meanwhile, even if some lower-confidence prototypes contain label-inconsistent patterns, their limited influence reduces the risk of label flipping. Importantly, because all prototypes are distilled from the target-domain unreliable set, they inherently encode the shifted target distribution. Injecting these target-derived prototypes into reliable samples therefore yields label-preserving yet drift-oriented augmentations, satisfying both constraints by design. Next, we first introduce uncertainty-aware prototype extraction, and then present the confidence-aware shifted augmentation procedure.

3.2.1 Uncertainty-Aware Prototype Extraction

Prototypes provide a compact and robust summary by aggregating shared patterns among unreliable nodes while suppressing instance-level noise induced by unreliable pseudo-labels [37, 33]. Under test-time distribution shifts, directly leveraging individual low-confidence nodes may amplify noise and destabilize self-training. Therefore, we extract uncertainty-aware prototypes from stratified unreliable groups as stable references to guide shifted augmentation.

Concretely, for each unreliable group in $\{\mathcal{V}_{\ell_1}, \ldots, \mathcal{V}_{\ell_K}\}$, we extract an uncertainty-aware prototype via confidence-guided selection and aggregation. To preserve normal-class semantics, we only use pseudo-normal nodes within each group, denoted by $\mathcal{V}_{\ell_i}^{\mathrm{nor}} = \{v \in \mathcal{V}_{\ell_i} \mid \hat{y}_v = y_{\mathrm{nor}}\}$. Throughout this section, $\mathbf{p}_{\ell_i}$ denotes the pseudo-normal prototype of group $\mathcal{V}_{\ell_i}$. We initialize a semantic prototype $\mathbf{p}_{\ell_i}^{(0)}$ for each group $\mathcal{V}_{\ell_i}$ ($i \in \{1, \ldots, K\}$). In other words, we compute the mean representation $\bar{\mathbf{h}}_{\ell_i}^{\mathrm{nor}}$ of all pseudo-normal nodes in group $\mathcal{V}_{\ell_i}$, and then blend it with the row weight vector $\mathbf{w}_{\mathrm{nor}}$ corresponding to the normal class in the pre-trained binary classifier. The resulting vector is normalized to obtain the initial prototype:

$$\mathbf{p}_{\ell_i}^{(0)} = \mathrm{norm}\left(\alpha \cdot \bar{\mathbf{h}}_{\ell_i}^{\mathrm{nor}} + (1-\alpha) \cdot \mathbf{w}_{\mathrm{nor}}\right), \quad i = 1, \ldots, K, \tag{6}$$

where $\alpha \in [0, 1]$ controls the mixing ratio and $\mathrm{norm}(\cdot)$ denotes a normalization function. This initialization anchors prototypes with normal-class semantics at the early adaptation stage, improving stability and preventing them from being dominated by noisy perturbations.

At the t-th iteration, for group $\mathcal{V}_{li}$, we select the top-M pseudo-normal nodes with the highest calibrated confidence s'_v and collect them into $\mathcal{T}_i^{(t)}$:

$$\mathcal{T}_i^{(t)} = \mathrm{Top}M\big(\mathcal{V}_{li}^{\mathrm{nor}};\, s'_v\big), \quad i \in \{1, \ldots, K\}, \quad |\mathcal{T}_i^{(t)}| = M. \tag{7}$$

We then aggregate their embeddings in a confidence-weighted manner to obtain the candidate prototype $\tilde{\mathbf{p}}_{li}^{(t)}$:

$$\tilde{\mathbf{p}}_{li}^{(t)} = \sum_{v \in \mathcal{T}_i^{(t)}} \alpha_v^{(t)} \mathbf{h}_v, \tag{8}$$

where the aggregation weight $\alpha_v^{(t)}$ is computed as:

$$\alpha_v^{(t)} = \frac{\exp(\gamma s'_v)}{\sum_{u \in \mathcal{T}_i^{(t)}} \exp(\gamma s'_u)}, \quad v \in \mathcal{T}_i^{(t)}, \tag{9}$$

where γ controls the sharpness of the weighting distribution.

To stabilize prototype refinement across iterations, we maintain a prototype memory for each confidence level and update it via exponential moving average:

$$\mathbf{p}_{li}^{(t)} = (1-\beta)\mathbf{p}_{li}^{(t-1)} + \beta\, \tilde{\mathbf{p}}_{li}^{(t)}, \tag{10}$$

where $\beta \in (0, 1)$ controls the update speed. This yields an uncertainty-aware prototype $\mathbf{p}_{li}^{(t)}$ for each unreliable group, serving as a stable and denoised reference for shifted augmentation.

3.2.2 Confidence-Aware Shifted Augmentation

For a given reliable pseudo-normal sample, we steer it toward the drifted target distribution by aggregating prototype information from the confidence-stratified unreliable groups (Eq. 10) to generate a shifted augmentation. To produce label-preserving yet drift-oriented augmentations, we adopt a confidence-aware aggregation scheme, where higher-confidence prototypes contribute more while lower-confidence prototypes contribute less. This design is motivated by the fact that higher-confidence prototypes are more likely to be label-consistent with reliable samples and thus preserve their pseudo-labels, whereas lower-confidence prototypes may contain label-inconsistent patterns but are heavily downweighted, preventing them from flipping the label. Meanwhile, since all prototypes are distilled from target-domain unreliable nodes, the aggregated prototype signal inherently encodes drifted target characteristics, yielding augmentations that both preserve labels and shift toward drifted regions.

In particular, for a reliable normal node $v \in \mathcal{V}_h^{\text{nor}}$, we use cross-attention to inject information from the pseudo-normal prototypes $\{\mathbf{p}_{\ell_i}\}_{i=1}^K$ into its representation $\mathbf{h}_v$ and obtain the shifted representation $\mathbf{h}'_v$. Specifically, we treat the node embedding $\mathbf{h}_v$ as the query and the prototypes as keys/values, and compute the content-based matching score $e_{v,i}^{\text{cnt}}$:

$$e_{v,i}^{\text{cnt}} = \frac{\mathbf{h}_v^\top \mathbf{p}_{li}}{\sqrt{d}}, \tag{11}$$

where d is the hidden dimension.

Since prototypes from different confidence groups have different reliability, we further introduce a confidence-wise reliability bias. We compute the average calibrated confidence within each group and normalize it to obtain the bias b_i:

$$b_i = \eta \cdot \text{norm}\left(\frac{1}{|\mathcal{V}_{li}|} \sum_{v \in \mathcal{V}_{li}} s'_v\right). \tag{12}$$

where η controls the strength of the reliability bias and $\text{norm}(\cdot)$ denotes a normalization function. We combine the content score and reliability bias to obtain the final attention logit:

$$e_{v,i} = e_{v,i}^{\text{cnt}} + b_i, \tag{13}$$

and compute the attention weight via softmax:

$$\alpha_{v,i} = \frac{\exp(e_{v,i})}{\sum_{j=1}^K \exp(e_{v,j})}. \tag{14}$$

Finally, we form the prototype-induced offset as the attention-weighted sum of prototypes and add a scaled offset to the original node representation to obtain the augmented sample $\mathbf{h}'_v$:

$$\mathbf{h}'_v = \mathbf{h}_v + \rho \sum_{i=1}^K \alpha_{v,i}\, \mathbf{p}_{li}, \tag{15}$$

where $\rho \in [0, 1]$ controls the strength of shifted augmentation.

3.3 Dynamic Objective Scheduling

Unreliable test samples, while often carrying richer variation patterns induced by distribution shifts, typically come with highly noisy pseudo-labels and are therefore unsuitable for direct supervised tuning. Following prior studies [15, 10, 3], we exploit these unreliable samples in a label-free manner via contrastive learning, which encourages the model to capture drift-related characteristics from the target stream without relying on noisy labels. However, contrastive learning mainly promotes representation invariance/consistency, which is not always well aligned with the classification objective of GAD, potentially leading to objective misalignment. To address this issue, we propose a dynamic objective scheduling strategy that smoothly transitions the optimization focus, i.e., we emphasize contrastive adaptation in the early stage to absorb shift-related information, and then progressively shift the emphasis to supervised learning on reliable samples to strengthen task alignment. This schedule enables the model to benefit from drift-informative unreliable data while ultimately optimizing toward robust anomaly-discriminative decision boundaries.

3.3.1 Contrastive Learning on Unreliable Data

To effectively leverage unreliable nodes during test-time adaptation, we perform contrastive learning over nodes $v \in \mathcal{V}_l$ by constructing two complementary views for each node, a semantic view and a structural view, thereby learning robust representations without relying on unreliable pseudo-labels.

Semantic view construction. Given the feature matrix $\mathbf{X}$, we generate an augmented feature matrix $\tilde{\mathbf{X}}$ by adding dimension-wise modulated noise. We then feed $\tilde{\mathbf{X}}$ together with the graph structure $\mathbf{E}$ into the graph encoder $f_\theta(\cdot)$ to obtain the semantic-view representation matrix, from which we extract the augmented representation of node v, denoted as $\tilde{\mathbf{h}}_v$:

$$\tilde{\mathbf{X}} = \mathbf{X} + \epsilon(\boldsymbol{\sigma} \odot \mathbf{S}_\xi), \qquad \tilde{\mathbf{h}}_v = f_\theta(\tilde{\mathbf{X}}, \mathbf{E})_v. \tag{16}$$

where $\mathbf{S}_\xi \in \mathbb{R}^{|V| \times d}$ is a noise matrix whose v-th row $\boldsymbol{\xi}_v$ is sampled from a standard multivariate Gaussian distribution, i.e., $\boldsymbol{\xi}_v \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The operator $\odot$ denotes the Hadamard product, ϵ controls the perturbation magnitude, and $\boldsymbol{\sigma} \in \mathbb{R}^d$ modulates the noise amplitude across feature dimensions.

Structural view construction. In parallel, we perturb the graph topology via random edge dropping and re-encode node representations on the perturbed graph:

$$E' = \{e \in E \mid \text{Bernoulli}(1-p) = 1\}, \qquad \hat{\mathbf{h}}_v = f_\theta(\mathbf{X}, \mathbf{E}')_v, \tag{17}$$

where p is the edge-drop probability, E is the original edge set, E' is the perturbed edge set, and $\hat{\mathbf{h}}$ denotes the structure-enhanced representation.

The two views jointly capture semantic and structural uncertainty. We treat $(\tilde{\mathbf{h}}_v, \hat{\mathbf{h}}_v)$ as a positive pair and $(\tilde{\mathbf{h}}_v, \hat{\mathbf{h}}_u)$ with $u \neq v$ as negative pairs, and optimize the InfoNCE loss:

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{V}_l|} \sum_{v \in \mathcal{V}_l} \log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_v, \hat{\mathbf{h}}_v)/r)}{\sum_{u \in \mathcal{V}_l} \exp(\text{sim}(\tilde{\mathbf{h}}_v, \hat{\mathbf{h}}_u)/r)}, \tag{18}$$

where $\text{sim}(\cdot, \cdot)$ is a similarity function and r is the temperature parameter.

3.3.2 Supervised Tuning on Reliable and Augmented Data

To adapt the detector to the test-time distribution, we perform supervised fine-tuning using the original representations of all reliable samples and the shifted augmentations of reliable pseudo-normal samples. Specifically, we collect the original representation $\mathbf{h}_v$ for every $v \in \mathcal{V}_h$ and the augmented representation $\mathbf{h}'_v$ only for $v \in \mathcal{V}_h^{\text{nor}}$ into a unified set:

$$\mathcal{H} = \{(v, h_v) \mid v \in V_h\} \cup \{(v, h'_v) \mid v \in V_h^{\text{nor}}\} \tag{19}$$

We then perform supervised fine-tuning on $\mathcal{H}$ with the corresponding pseudo-labels $\hat{y}_v$:

$$\mathcal{L}_{\text{sup}} = \frac{1}{|\mathcal{H}|} \sum_{(v, \tilde{\mathbf{h}}) \in \mathcal{H}} \ell(g_\theta(\tilde{\mathbf{h}}), \hat{y}_v), \tag{20}$$

where $g_\theta(\cdot)$ denotes the anomaly detector with parameters θ, and $\ell(\cdot)$ is a standard supervised loss (e.g., cross-entropy).

3.3.3 Dynamic Objective Optimization

We optimize the model with two complementary objectives: (i) the supervised loss on reliable samples and their shifted augmentations, $\mathcal{L}_{\text{sup}}$ (Eq. 20), and (ii) the contrastive loss on unreliable samples, $\mathcal{L}_{\text{con}}$ (Eq. 18). As these objectives may be imperfectly aligned, naively optimizing them with fixed weights can hinder learning a better anomaly-discriminative decision boundary. Therefore, we present a *dynamic objective scheduling* strategy, i.e., we assign a larger weight to contrastive regularization in the early stage to capture drift characteristics from unreliable data, and then progressively increase the weight of supervised learning on reliable and augmented samples to strengthen task alignment.

Specifically, we use a linear ramp-up schedule for the supervised and contrastive weights:

$$s(t) = \min\left(1, \frac{t}{T} + c_0\right), \tag{21}$$

where $c_0 \in [0, 1]$ controls the initial weight for the supervised objective, t denotes the current epoch, and T refers to the total adaptation epochs. After $t \geq T \times (1 - c_0)$, the weight saturates at 1 and the optimization relies solely on $\mathcal{L}_{\text{sup}}$, effectively down-weighting the contrastive loss.

The overall objective at epoch t is then

$$\mathcal{L}(t) = s(t)\,\mathcal{L}_{\text{sup}} \;+\; \big(1 - s(t)\big)\,\mathcal{L}_{\text{con}}. \tag{22}$$

As $s(t)$ increases over epochs, the supervised component gradually dominates, while the contrastive component is annealed, enabling stable adaptation that benefits from unreliable data early and remains aligned with anomaly classification at convergence.

4 Experiment Evaluation

4.1 Experimental Setup

Datasets. We evaluate on eight real-world benchmarks for GAD: Amazon[25], YelpChi[34], Reddit[17], Weibo[52], Tolokers[30], Questions[30], T-Finance[38], and T-Social[38]. These datasets cover diverse application domains, including e-commerce, social networks, crowdsourcing platforms, and financial transaction systems. All benchmarks are node-attributed graphs with binary anomaly labels and exhibit severe class imbalance, which is common in typical GAD settings.

Baselines. We compare SADOS with two groups of baselines. (1) *Supervised GAD models*: CareGNN [8], H2-FDetector [35], GAGA [43], PMP [56], and ConsisGAD [4]. (2) *Graph TTA methods*: GTrans [13], SOGA [24], GADT3 [29], and TUNE [28]. Note that GTrans was originally developed for graph classification, whereas TUNE is designed for graph anomaly detection. For a fair comparison, we use GHRN [9] as the common backbone for all graph TTA methods.

Experimental Settings. For SADOS, we adopt a source-only model-selection protocol. All experiments are run on an i9-12900K CPU, an RTX 3090 GPU and 64GB RAM. The hyperparameters are selected by grid search on a small labeled validation set drawn exclusively from the source domain, without using any target-domain data for tuning. Once selected, the hyperparameters are fixed across all test-time adaptation experiments.

4.2 Performance Comparison

In this section, we compare the performance of our method, SADOS, with supervised GAD baselines and graph TTA methods under various test-time distribution shift scenarios. Table 1 reports the AUROC, where the best performance is highlighted in bold. All results are reported as mean ± standard deviation over five independent runs with different random seeds. We have the following observations.

First, SADOS achieves the best overall performance, outperforming competing baselines on nearly all datasets and shift settings, with substantial AUROC improvements. This validates the effectiveness of our confidence-aware test-time optimization, i.e., reliable nodes provide relatively trustworthy

Table 1: Performance (AUROC in %) of our SADOS and baselines.

	Dataset	GAD					TTA				
		CareGNN	H2-FDetector	GAGA	PMP	ConsisGAD	GTrans	SOGA	GADT3	TUNE	SADOS
Cluster -based split	Amazon	72.12 $_{\pm 2.13}$	61.25 ±8.90	78.47 ±1.20	80.79 ±2.15	82.21 ±0.75	84.31 ±3.10	85.92 ±2.85	88.14 ±1.45	89.23 ±1.92	**91.45** ±0.42
	YelpChi	51.50 ±1.50	50.33 ±0.72	52.12 ±0.66	53.68 ±2.40	51.01 ±0.43	53.41 ±0.61	54.34 ±0.30	55.45 ±0.88	56.44 ±0.72	**58.82** ±0.35
	Reddit	54.90 ±0.92	51.34 ±3.55	52.20 ±4.50	56.25 ±0.52	58.89 ±1.20	60.20 ±2.60	61.12 ±0.63	62.28 ±1.30	63.60 ±0.06	**65.68** ±0.58
	Weibo	66.93 ±0.90	61.61 ±7.10	84.40 ±1.74	80.22 ±1.55	83.44 ±0.51	88.58 ±2.70	89.20 ±5.30	86.65 ±4.80	96.18 ±0.03	**97.12** ±0.15
	Tolokers	55.80 ±0.18	51.31 ±2.20	60.45 ±2.38	56.83 ±1.02	59.50 ±1.25	60.12 ±3.15	64.70 ±1.99	65.01 ±0.43	66.23 ±0.05	**68.94** ±0.21
	Questions	45.47 ±0.03	48.47 ±4.05	51.83 ±1.35	48.13 ±4.91	48.33 ±0.80	52.92 ±0.97	55.92 ±0.46	57.39 ±1.55	60.60 ±0.98	**62.27** ±0.85
	T-Finance	76.16 ±2.89	70.52 ±4.12	75.64 ±0.91	80.69 ±6.70	81.60 ±0.78	82.92 ±1.85	83.20 ±2.10	85.54 ±1.20	85.55 ±0.05	**86.38** ±0.32
	T-Social	68.34 ±3.15	62.11 ±5.45	72.48 ±1.85	70.21 ±4.52	70.34 ±4.45	75.12 ±3.20	78.45 ±2.12	80.82 ±2.34	81.88 ±0.08	**84.56** ±0.50
Density -based split	Amazon	68.15 ±3.50	58.42 ±8.12	75.34 ±2.10	76.54 ±3.20	77.67 ±1.10	79.23 ±4.45	81.12 ±4.10	82.45 ±2.12	84.62 ±0.50	**86.24** ±0.45
	YelpChi	49.24 ±1.80	50.15 ±1.15	50.32 ±1.20	51.42 ±2.80	49.56 ±0.85	51.12 ±1.20	52.56 ±0.90	51.22 ±1.05	53.80 ±0.85	**55.45** ±0.41
	Reddit	52.32 ±1.20	50.12 ±4.20	51.10 ±3.80	53.12 ±0.85	55.45 ±1.65	58.80 ±3.12	59.23 ±0.80	61.15 ±1.45	63.15 ±0.05	**65.82** ±0.62
	Weibo	60.21 ±1.15	60.23 ±9.50	82.15 ±2.10	78.45 ±2.10	81.23 ±0.95	84.12 ±3.20	85.45 ±4.50	82.15 ±6.20	91.45 ±0.05	**92.68** ±0.18
	Tolokers	58.12 ±0.25	55.15 ±3.10	58.95 ±2.95	59.23 ±1.45	57.34 ±1.65	61.45 ±2.80	62.34 ±2.15	63.54 ±0.65	65.85 ±0.10	**67.92** ±0.28
	Questions	44.12 ±0.10	45.56 ±4.85	48.23 ±1.85	50.45 ±5.20	47.12 ±1.10	54.45 ±1.20	55.15 ±0.75	55.80 ±1.85	59.42 ±1.15	**61.45** ±0.92
	T-Finance	72.45 ±3.15	65.28 ±5.22	78.15 ±1.25	78.42 ±7.20	80.15 ±1.10	81.45 ±2.15	82.12 ±2.50	84.42 ±1.85	84.12 ±0.12	**88.15** ±0.45
	T-Social	62.22 ±4.50	59.85 ±6.32	65.12 ±2.15	66.45 ±5.85	67.45 ±5.12	70.34 ±5.10	74.12 ±3.85	76.25 ±3.10	80.45 ±0.10	**82.12** ±0.62

pseudo-labels, while unreliable nodes supply informative cues about distribution drift. Second, compared with conventional supervised GAD models, TTA-based approaches typically yield more pronounced improvements, underscoring the benefit of adapting the detector at deployment time to mitigate distribution shifts and enhance generalization. Third, SADOS maintains stable and strong performance under both splitting protocols, indicating robustness to semantic shifts and structural shifts. We attribute this to our joint utilization of nodes with different confidence levels. Reliable nodes offer discriminative pseudo-supervision, whereas unreliable nodes are not discarded as noise but are leveraged to capture uncertainty and shift-related signals in the target distribution, thereby improving generalization across environments.

4.3 Ablation Study

To quantify the contribution of each core component in SADOS for anomaly detection, we conduct ablation experiments and consider four simplified variants that correspond to the key design choices: (1) *w/o WDI*: replace the proposed confidence-aware injection with uniform weights across the confidence-stratified unreliable groups; (2) *w/o UAP*: remove uncertainty-aware prototype distillation and instead inject features from randomly sampled unreliable nodes; (3) *w/o SA*: replace the proposed shifted augmentation with standard perturbations (e.g., feature dropout or edge dropout) applied to reliable samples; (4) *w/o DOS*: replace dynamic objective scheduling with constant loss weights, i.e., fixing both the supervised and contrastive loss weights to 0.5 throughout adaptation.

Figure 2 shows that the full SADOS consistently outperforms all variants, confirming that each module contributes to robust test-time adaptation. In particular, *w/o SA* incurs the largest performance drop, indicating that generic perturbations are insufficient for effective boundary exploration under distribution shifts, whereas drift-oriented shifted augmentations are crucial for capturing subtle anomalies. The degradations of *w/o WDI* and *w/o UAP* further highlight the importance of uncertainty-aware synthesis, i.e., without stratified weighting or prototype-guided injection, the generated augmentations fail to reliably encode drifted target characteristics from unreliable nodes. Finally, *w/o DOS* also yields noticeable degradation, suggesting that a fixed optimization objective is suboptimal for test-time adaptation. In contrast, dynamic scheduling better balances early-stage drift exploration from unreliable data with later-stage classification-oriented refinement.

4.4 Effect of the Initial Weight for Dynamic Scheduling

SADOS employs a dynamic objective scheduling strategy to balance the supervised loss and the contrastive loss during test-time optimization. In this experiment, we examine how the initial weight of the supervised objective affects adaptation performance and stability. Specifically, we vary the initial supervised weight c_0 in Equation 21 while keeping all other hyperparameters fixed, and evaluate the resulting detection performance under the same protocol as the main experiments.

Figure 3 reports the adaptation curves of SADOS under different initializations. We observe that c_0=0.5 and c_0=0.7 lead to smooth and stable convergence of AUROC, indicating reliable adaptation when the supervised objective is sufficiently emphasized. Compared with c_0=0.7, c_0=0.5

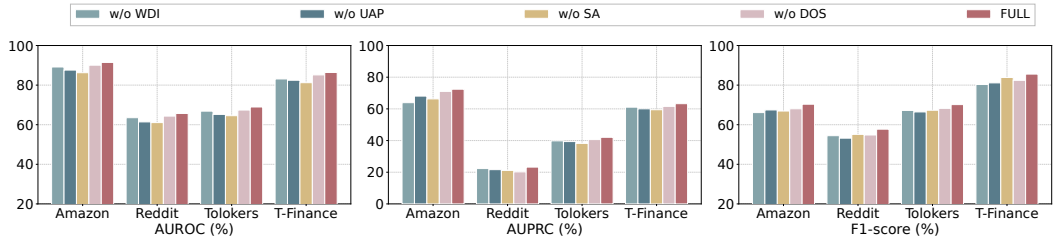


Figure 2: Ablation study of key components.

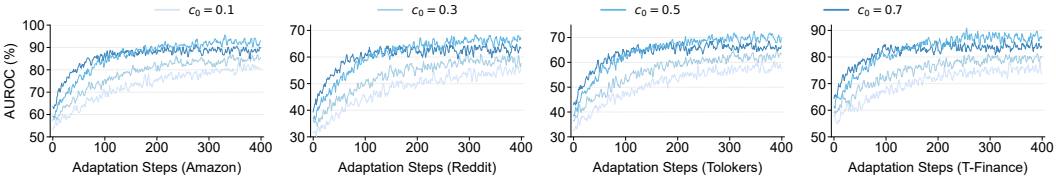


Figure 3: Performance with different initial weights.

converges faster and achieves better early-stage adaptation, suggesting that an overly large supervised initialization may suppress the model's ability to capture distribution drift from the target stream at the beginning of adaptation. In contrast, c_0=0.1 yields rapid initial improvement but introduces noticeable oscillations, reflecting unstable optimization when the supervised signal is too weak. The setting c_0=0.3 is relatively stable, yet adapts more conservatively and is less effective at capturing early shifts. Overall, these results reveal a clear trade-off between fast adaptation and optimization stability, and highlight the importance of a balanced initialization for robust test-time adaptation.

5 Conclusion

In this paper, we presented SADOS, a Shifted Augmentation-based Dynamic Objective Scheduling framework for test-time graph anomaly detection. SADOS jointly leverages reliable and unreliable test samples to adapt a pre-trained detector under distribution shifts. To expand the effective coverage of scarce reliable samples, SADOS introduces a confidence-aware shifted augmentation module that generates label-preserving yet drift-oriented augmentations for supervised fine-tuning. To better exploit unreliable samples without being hindered by objective mismatch, SADOS employs a dynamic objective scheduling strategy that gradually anneals the contrastive objective and ultimately prioritizes classification-oriented optimization. Experimental results show that SADOS consistently improves robustness under distribution shifts and achieves superior performance compared to strong test-time adaptation baselines.

SADOS has several limitations. First, it assumes transductive access to the complete target graph, limiting its applicability to fully online adaptation. Second, it relies on the source-trained detector retaining a meaningful confidence ranking on target nodes. If the detector produces nearly random predictions or is systematically overconfident on incorrect predictions, adaptation may degrade. Third, highly sparse or noisy graphs may undermine neighborhood information, while systematic pseudo-label errors among unreliable pseudo-normal nodes may bias prototype construction. Future work will explore gradually evolving distribution shifts and online graph adaptation, as well as extensions to multi-class and edge- or graph-level anomaly detection.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62677022 and the State Key Laboratory of Spatial Datum under Grant No. SKLSD2025-HNZZ-21.

References

[1] Wenxuan Bao, Zhichen Zeng, Zhining Liu, Hanghang Tong, and Jingrui He. Matcha: Mitigating graph structure shifts with test-time adaptation. In *International Conference on Learning Representations*, 2025.

[2] Jinyu Cai, Yunhe Zhang, Fusheng Liu, and See-Kiong Ng. Leveraging diffusion model as pseudo-anomalous graph generator for graph-level anomaly detection. In *International Conference on Machine Learning*, 2025.

[3] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022.

[4] Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *International Conference on Learning Representations*, 2024.

[5] Qiyuan Dai and Sibei Yang. Free on the fly: Enhancing flexibility in test-time adaptation with online em. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9538–9548, 2025.

[6] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *Proceedings of the SIAM international conference on data mining*, pages 594–602, 2019.

[7] Kaize Ding, Kai Shu, Xuan Shan, Jundong Li, and Huan Liu. Cross-domain graph anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2406–2415, 2022.

[8] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the ACM international conference on information & knowledge management*, pages 315–324, 2020.

[9] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference*, pages 1528–1538, 2023.

[10] Peiliang Gong, Mohamed Ragab, Min Wu, Zhenghua Chen, Yongyi Su, Xiaoli Li, and Daoqiang Zhang. Augmented contrastive clustering with uncertainty-aware prototyping for time series test time adaptation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 390–401, 2025.

[11] Hans Hao-Hsun Hsu, Shikun Liu, Han Zhao, and Pan Li. Structural alignment improves graph test-time adaptation. In *Proceedings of The International Conference on Artificial Intelligence and Statistics*, pages 829–837, 2026.

[12] Changqin Huang, Xinxing Shi, Chengling Gao, Qintai Hu, Xiaodi Huang, Qionghao Huang, and Ali Anaissi. Cross-domain graph anomaly detection via graph transfer and graph decouple. *CAAI Transactions on Intelligence Technology*, 10(4):1089–1103, 2025.

[13] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. In *International Conference on Learning Representations*, 2023.

[14] Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. Graphpatcher: Mitigating degree bias for graph neural networks via test-time augmentation. In *Neural Information Processing Systems*, 2023.

[15] Saeed Karimi and Hamdi Dibeklioglu. Adapac: Prototypical anchored contrastive test time adaptation for domain generalization. *Neurocomputing*, 650:130839, 2025.

[16] Saeed Karimi and Hamdi Dibeklioğlu. Shift-act: Test-time adaptation via shift-aware class-specific dynamic thresholding for domain generalization. *Neurocomputing*, 663:132013, 2026.

[17] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1269–1278, 2019.

[18] Guoming Li, Jian Yang, and Yifan Chen. Partition-wise graph filtering: A unified perspective through the lens of graph coarsening. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1353–1364, 2025.

[19] Xuan Li, Chunjing Xiao, Ziliang Feng, Shikang Pang, Wenxin Tai, and Fan Zhou. Controlled graph neural networks with denoising diffusion for anomaly detection. *Expert Systems with Applications*, 237:121533, 2024.

[20] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. ARC: A generalist graph anomaly detector with in-context learning. In *Neural Information Processing Systems*, 2024.

[21] Yuejiang Liu, Parth Kothari, Bastien Germain van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: When does self-supervised test-time training fail or thrive? In *Neural Information Processing Systems*, 2021.

[22] Jing Ma. Improved self-training for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23701–23710, 2024.

[23] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE transactions on knowledge and data engineering*, 35(12):12012–12038, 2023.

[24] Haitao Mao, Lun Du, Yujia Zheng, Qiang Fu, Zelin Li, Xu Chen, Shi Han, and Dongmei Zhang. Source free graph unsupervised domain adaptation. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 520–528, 2024.

[25] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the international conference on World Wide Web*, pages 897–908, 2013.

[26] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, pages 16888–16905, 2022.

[27] Junjun Pan, Yixin Liu, Xin Zheng, Yizhen Zheng, Alan Wee-Chung Liew, Fuyi Li, and Shirui Pan. A label-free heterophily-guided approach for unsupervised graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12443–12451, 2025.

[28] Junjun Pan, Yixin Liu, Chuan Zhou, Fei Xiong, Alan Wee-Chung Liew, and Shirui Pan. Correcting false alarms from unseen: Adapting graph anomaly detectors at test time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 24691–24699, 2026.

[29] Delaram Pirhayatifard and Arlei Silva. Cross-domain graph anomaly detection via test-time training with homophily-guided self-supervision. *Transactions on Machine Learning Research*, 2025.

[30] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023.

[31] Hezhe Qiao, Chaoxi Niu, Ling Chen, and Guansong Pang. Anomalygfm: Graph foundation model for zero/few-shot anomaly detection. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2326–2337, 2025.

[32] Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. Deep graph anomaly detection: A survey and new perspectives. *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5106–5126, 2025.

[33] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. General class-balanced multicentric dynamic prototype pseudo-labeling for source-free domain adaptation. *International Journal of Computer Vision*, page 3327–3348, 2025.

[34] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the Acm SIGKDD international conference on knowledge discovery and data mining*, pages 985–994, 2015.

[35] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM web conference 2022*, pages 1486–1494, 2022.

[36] Yudan Song, Yuecen Wei, Yuhang Lu, Qingyun Sun, Minglai Shao, Li-e Wang, Chunming Hu, Xianxian Li, and Xingcheng Fu. Mitigating message imbalance in fraud detection with dual-view graph representation learning. In *International Joint Conference on Artificial Intelligence*, pages 3281–3289, 2025.

[37] Haopeng Sun, Lumin Xu, Sheng Jin, Ping Luo, Chen Qian, and Wentao Liu. PROGRAM: PROtotype GRAph model based pseudo-label learning for test-time adaptation. In *International Conference on Learning Representations*, 2024.

[38] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, pages 21076–21089, 2022.

[39] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. GADBench: Revisiting and benchmarking supervised graph anomaly detection. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[40] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

[41] Qizhou Wang, Guansong Pang, Mahsa Salehi, Wray Buntine, and Christopher Leckie. Cross-domain graph anomaly detection via anomaly-aware contrastive alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4676–4684, 2023.

[42] Yili Wang, Yixin Liu, Xu Shen, Chenyu Li, Rui Miao, Kaize Ding, Ying Wang, Shirui Pan, and Xin Wang. Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. In *International Conference on Learning Representations*, 2025.

[43] Yuchen Wang, Jinghui Zhang, Zhengjie Huang, Weibin Li, Shikun Feng, Ziheng Ma, Yu Sun, Dianhai Yu, Fang Dong, Jiahui Jin, et al. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference*, pages 406–416, 2023.

[44] Zhihao Wu, Jinyu Cai, Yunhe Zhang, Jielong Lu, Zhaoliang Chen, Shuman Zhuang, and Haishuai Wang. Where graph meets heterogeneity: Multi-view collaborative graph experts. In *Neural Information Processing Systems*, 2025.

[45] Chunjing Xiao, Xovee Xu, Yue Lei, Kunpeng Zhang, Siyuan Liu, and Fan Zhou. Counterfactual graph learning for anomaly detection on attributed networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10540–10553, 2023.

[46] Chunjing Xiao, Shikang Pang, Xovee Xu, Xuan Li, Goce Trajcevski, and Fan Zhou. Counterfactual data augmentation with denoising diffusion for graph anomaly detection. *IEEE Transactions on Computational Social Systems*, 11(6):7555–7567, 2024.

[47] Chunjing Xiao, Jiahui Lu, Xovee Xu, Fan Zhou, Tianshu Xie, Wei Lu, and Lifeng Xu. Reconciling attribute and structural anomalies for improved graph anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 36(9):16661–16674, 2025.

[48] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Neural information processing systems*, 34:18408–18419, 2021.

[49] Jiaxin Zhang, Yiqi Wang, Xihong Yang, Siwei Wang, Yu Feng, Yu Shi, Ruichao Ren, En Zhu, and Xinwang Liu. Test-time training on graphs with large language models (llms). In *Proceedings of the ACM International Conference on Multimedia*, pages 2089–2098, 2024.

[50] Jinghui Zhang, Zhengjia Xu, Dingyang Lv, Zhan Shi, Dian Shen, Jiahui Jin, and Fang Dong. Dig-in-gnn: Discriminative feature guided gnn-based fraud detector against inconsistencies in multi-relation fraud graph. In *Proceedings of the AAAI conference on artificial intelligence*, pages 9323–9331, 2024.

[51] Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. COME: Test-time adaption by conservatively minimizing entropy. In *International Conference on Learning Representations*, 2025.

[52] Tong Zhao, Chuchen Deng, Kaifeng Yu, Tianwen Jiang, Daheng Wang, and Meng Jiang. Error-bounded graph anomaly loss for gnns. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 1873–1882, 2020.

[53] Yunfeng Zhao, Yixin Liu, Shiyuan Li, Qingfeng Chen, Yu Zheng, and Shirui Pan. Freegad: A training-free yet effective approach for graph anomaly detection. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 4379–4389, 2025.

[54] Yusheng Zhao, Qixin Zhang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. Test-time adaptation on graphs via adaptive subgraph-based selection and regularized prototypes. In *International Conference on Machine Learning*, 2025.

[55] Xin Zheng, Wei Huang, Chuan Zhou, Ming Li, and Shirui Pan. Test-time graph neural dataset search with generative projection. In *International Conference on Machine Learning*, 2025.

[56] Wei Zhuo, Zemin Liu, Bryan Hooi, Bingsheng He, Guang Tan, Rizal Fathony, and Jia Chen. Partitioning message passing for graph fraud detection. In *International Conference on Learning Representations*, 2024.

Appendix

A Empirical Properties of Shifted Augmentations

This section provides offline diagnostics to support the claim that our shifted augmentation is (i) label-preserving for reliable nodes and (ii) drift-oriented in the sense that the augmented representations move closer to the drifted (low-confidence) regions in the target distribution. Importantly, ground-truth (GT) labels are used only for post-hoc evaluation in this appendix and are never used during test-time adaptation.

For each target node, we obtain a pseudo-label and its confidence from the detector output. The confidence is defined as the predicted probability of the chosen pseudo-label (i.e., how certain the model is about its own prediction). We report the calibrated confidence used in our method.

We quantify distribution shift in the representation space. Specifically, we extract node embeddings using the (pre-trained) graph encoder and define the *drifted region* as the set of *low-confidence* target nodes selected by the same confidence-based partitioning used in adaptation. We then measure how close the augmented embeddings of reliable nodes are to the embedding of this unreliable set.

To assess whether augmentation preserves labels, we focus on nodes that are selected as reliable and have GT label normal. We apply shifted augmentation once to obtain an augmented embedding for each such node, and compute the fraction of nodes whose predicted label changes from normal to anomaly after augmentation. A lower label flip rate indicates better label preservation.

To assess whether augmentation is drift-oriented, we examine how the representations of reliable normal nodes move toward the drifted target region before and after augmentation. Specifically, we define shift proximity as the mean cosine similarity between reliable-normal embeddings and the centroid of the unreliable set. The label flip rates and the original and augmented shift scores are reported in Table 2. A larger shift proximity indicates that reliable representations are closer to the drifted target region.

Table 2: Offline diagnostics for shifted augmentations.

Dataset	Label Flip Rate (%)	Original Shift Score	Augmented Shift Score
Amazon	0.6	0.82	0.86
Reddit	1.1	0.67	0.72
Tolokers	1.2	0.69	0.74
T-Finance	0.8	0.81	0.86

B Experimental Details

B.1 Datasets

We evaluate anomaly detection performance on eight GAD benchmarks: Amazon[25], YelpChi[34], Reddit[17], Weibo[52], Tolokers[30], Questions[30], T-Finance[38], and T-Social[38]. These datasets are widely used in prior work as benchmarks for graph anomaly detection, and each contains labeled real-world anomalies spanning diverse domains.

- Amazon[25] dataset focuses on identifying fraudulent reviews in the Musical Instrument category on Amazon. The graph includes three types of relationships: user-product-user (U-P-U), user-star-user (U-S-U), and user-verified text similarity (U-V-U). These relationships help in detecting fake reviews by analyzing user behaviors and text patterns across the platform.
- YelpChi[34] consists of reviews from Yelp's restaurant and hotel categories, classified as either spam or legitimate. This dataset includes three key relationships: review-user-review (R-U-R), review-star-review (R-S-R), and review-time-review (R-T-R). The graph captures the social dynamics of review behaviors, providing a rich structure for anomaly detection.
- Reddit[17] dataset captures user interactions and discussions across various subreddits. The graph is designed to detect anomalous posts based on user behaviors and text patterns. Relationships

Table 3: Source and target anomaly ratios (%).

Dataset	Cluster Source	Cluster Target	Density Source	Density Target
Amazon	10.42	8.75	8.87	10.74
YelpChi	15.35	13.22	15.02	13.88
Reddit	3.68	2.96	2.94	3.87
Weibo	10.76	10.43	10.28	8.95
Tolokers	22.84	20.97	20.96	24.03
Questions	3.05	2.42	3.58	2.81
T-Finance	4.96	4.21	4.02	5.28
T-Social	3.29	2.61	3.92	3.01

within the graph focus on comment interactions, user reputation, and temporal patterns, which aid in identifying outliers or abnormal posts.

- Weibo[52] dataset represents social media interactions on Weibo, focusing on identifying anomalous behaviors within posts and comments. It captures diverse relationships such as user-comment-user and post-user interactions, which are key in detecting fraudulent or irregular activities like spam or propaganda.
- Tolokers[30] is a crowdsourcing dataset designed for detecting fraudulent tasks in platforms like Amazon Mechanical Turk. It includes interactions between workers and tasks, with a focus on detecting low-quality or unreliable workers. This dataset provides valuable insights into the dynamics of online crowdsourcing and the identification of anomalies in task completion patterns.
- Questions[30] dataset consists of a set of question-answer pairs from various domains, with a particular focus on detecting anomalies in user behavior or content quality. The relationships capture patterns of user submissions and answer correctness, making it useful for anomaly detection in educational or QA systems.
- T-Finance[38] dataset involves financial transactions, where the objective is to detect fraudulent behavior such as money laundering or suspicious financial activities. The graph is structured with various financial interactions, providing a strong foundation for detecting irregular patterns in financial data.
- T-Social[38] is focused on detecting social media-based anomalies, where relationships capture interactions between users and posts. It is particularly useful for detecting malicious behaviors such as coordinated misinformation campaigns or bot-driven activities on social media platforms.

In our experiments, we employed two distinct strategies to split the datasets, simulating test-time distribution shifts and enabling a comprehensive evaluation of model performance. Table 3 reports the source and target anomaly ratios.

Cluster-based split: We utilized the k-means clustering algorithm [28] to partition the normal nodes based on their feature similarities. The number of clusters was set to 3, and the smallest cluster is excluded from both the training and validation sets. This approach simulates a normality shift, where the test data contains unseen normal subtypes that do not appear during training.

Density-based split: We partitioned the graphs based on the average node degree, which serves as a measure of the graph's density. We employed the METIS algorithm [54] to partition the nodes into clusters with varying density levels. One cluster is used for training, while the other is used as the test set. This partitioning strategy simulates structural shifts, where variations in node connectivity and graph structure are present between the training and test distributions.

B.2 Hyper-Parameter Setting

We tune the main hyperparameters of SADOS, including the number of epochs, learning rate and weight decay, via grid search on a small labeled validation set drawn exclusively from the source domain. Once selected on the source-domain validation set, all hyperparameters are fixed for all test-time adaptation experiments. Concretely, we partition the unreliable set into K=4 confidence groups. For confidence partitioning, we consider two predefined threshold schedules, $(\tau_{L_0}, \tau_{L_1}, \tau_{L_2}, \tau_{L_3}, \tau_{L_4}) = (0.90, 0.80, 0.70, 0.60, 0.50)$ and $(0.95, 0.85, 0.75, 0.60, 0.50)$. Based on source-domain validation performance, the former is used for Amazon, Weibo, T-Finance, and

Table 4: Performance (AUPRC in %) of our SADOS and baselines.

	Dataset	GAD					TTA				
		CareGNN	H2-FDetector	GAGA	PMP	ConsisGAD	GTrans	SOGA	GADT3	TUNE	SADOS
Cluster -based split	Amazon	37.84 ±0.61	20.91 ±1.20	48.63 ±0.72	52.11 ±0.85	55.46 ±1.05	59.32 ±1.18	63.14 ±1.30	66.02 ±1.42	69.08 ±0.55	**72.41** ±1.68
	YelpChi	36.02 ±0.60	35.34 ±0.62	35.88 ±0.58	37.41 ±1.52	40.71 ±0.85	46.92 ±0.98	45.44 ±0.92	45.86 ±0.96	47.07 ±1.02	**48.86** ±1.10
	Reddit	8.22 ±0.28	7.56 ±0.32	8.82 ±0.36	9.98 ±0.42	13.47 ±0.52	14.49 ±0.58	15.51 ±0.62	17.24 ±0.70	20.24 ±0.78	**23.23** ±0.86
	Weibo	29.88 ±0.66	21.77 ±0.75	58.96 ±1.02	52.64 ±0.95	57.23 ±1.05	67.44 ±1.28	69.12 ±1.35	64.35 ±1.20	82.76 ±1.58	**84.31** ±1.65
	Tolokers	27.64 ±0.60	23.15 ±0.52	30.22 ±0.66	27.46 ±0.60	30.98 ±0.72	36.24 ±0.85	36.88 ±0.88	38.91 ±0.95	39.17 ±0.98	**42.05** ±1.05
	Questions	3.34 ±0.22	3.71 ±0.25	4.62 ±0.30	3.08 ±0.20	4.86 ±0.32	7.42 ±0.45	9.21 ±0.55	10.74 ±0.62	10.41 ±0.60	**12.92** ±0.72
	T-Finance	41.73 ±0.82	31.56 ±0.66	40.98 ±0.80	49.87 ±0.95	52.64 ±1.02	57.36 ±1.15	58.02 ±1.18	59.11 ±1.22	60.38 ±1.28	**63.27** ±1.35
	T-Social	29.91 ±0.64	27.22 ±0.58	33.85 ±0.72	33.67 ±0.70	35.44 ±0.78	42.52 ±0.92	45.11 ±0.98	49.26 ±1.08	52.07 ±1.15	**55.83** ±1.25
Density -based split	Amazon	41.45 ±0.82	38.12 ±0.75	43.27 ±0.88	45.62 ±0.92	47.11 ±0.98	50.83 ±1.05	54.26 ±1.12	55.94 ±1.18	59.47 ±1.25	**62.18** ±1.32
	YelpChi	34.11 ±0.66	35.56 ±0.70	34.72 ±0.68	35.39 ±0.72	41.86 ±0.85	43.17 ±0.90	46.42 ±0.96	46.68 ±0.98	48.06 ±1.02	**49.58** ±1.08
	Reddit	7.50 ±0.28	5.82 ±0.22	6.36 ±0.25	7.95 ±0.30	10.16 ±0.40	12.93 ±0.48	12.84 ±0.47	16.00 ±0.58	19.03 ±0.68	**22.92** ±0.78
	Weibo	41.16 ±0.80	39.88 ±0.78	54.72 ±1.00	47.96 ±0.92	53.48 ±1.00	59.12 ±1.12	61.03 ±1.18	55.68 ±1.05	72.86 ±1.40	**75.14** ±1.48
	Tolokers	28.41 ±0.60	25.09 ±0.52	29.12 ±0.62	29.76 ±0.64	27.58 ±0.58	31.66 ±0.70	32.41 ±0.72	34.88 ±0.98	37.64 ±0.85	**40.21** ±0.92
	Questions	6.95 ±0.26	6.72 ±0.25	7.94 ±0.30	6.36 ±0.24	8.89 ±0.34	8.91 ±0.35	10.74 ±0.62	12.05 ±0.48	12.66 ±0.30	**14.88** ±1.60
	T-Finance	40.91 ±0.78	33.72 ±0.66	44.88 ±1.26	46.37 ±0.90	49.11 ±0.95	51.46 ±1.10	53.27 ±1.05	57.52 ±1.02	56.89 ±1.10	**59.72** ±1.18
	T-Social	30.42 ±0.64	38.31 ±0.80	32.56 ±0.68	33.68 ±0.70	34.91 ±0.94	39.84 ±0.85	46.72 ±0.98	40.65 ±0.88	48.77 ±0.52	**52.18** ±1.10

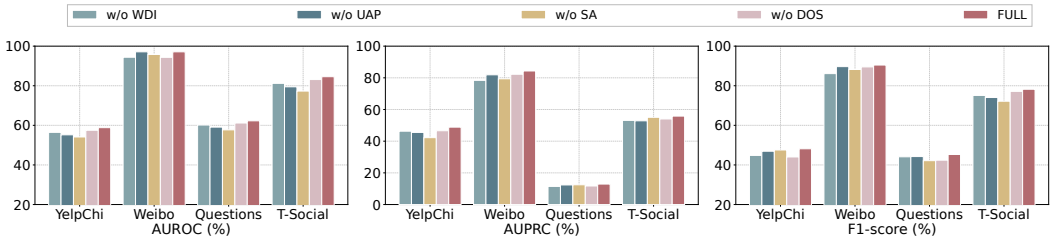


Figure 4: Ablation study on four datasets.

T-Social, and the latter for the remaining datasets. The thresholds remain fixed during adaptation, while node assignments are updated at each epoch. Backbone hyperparameters are selected using a labeled source-domain validation set. We search learning rates in $\{0.01, 0.005, 0.001\}$, weight decay in $\{0, 5 \times 10^{-5}, 10^{-4}\}$, dropout in $\{0, 0.2, 0.4, 0.6, 0.8\}$. Batch sizes are selected from $\{64, 128, 256, 1024\}$ for Amazon, YelpChi, Reddit, and Weibo, and from $\{512, 1024, 2048, 4096\}$ for the remaining datasets. For test-time adaptation, we use 100 epochs, a prototype EMA coefficient of 0.2, an augmentation strength of $\rho = 0.3$, a feature-noise magnitude of 0.1, an edge-drop probability of 0.2, and a contrastive temperature of 0.2 across datasets.

C Additional Experimental Results

Supplementary AUPRC Results. The results in Table 4 show that SADOS consistently outperforms supervised GAD and graph TTA baselines in terms of AUPRC, demonstrating its effectiveness under class imbalance and distribution shifts.

Supplementary Ablation Study Results. The ablation study results in Figure 4 (which complement the experiments in the main paper on YelpChi, Weibo, Questions and T-Social datasets) show that the full SADOS model consistently outperforms all variants across the four additional datasets.

Sensitivity to the Number of Unreliable Groups. We investigate how the number of confidence-stratified unreliable groups (i.e., K in Equation 5) affects SADOS. Specifically, we vary K from 1 to 5 to examine how the granularity of unreliable-node partitioning influences adaptation performance under distribution shifts.

Figure 5 shows that increasing K from 1 to 3 yields consistent improvements in AUROC, AUPRC, and F1-score. This indicates that a finer stratification enables more accurate characterization of distribution shifts within unreliable nodes, which in turn improves prototype extraction and the effectiveness of prototype-based shifted augmentation. When K reaches 4 or 5, performance across all datasets, especially T-Finance and Amazon, tends to stabilize with minimal fluctuation, suggesting that K=4 already provides sufficient granularity to capture the dominant drift patterns. Further

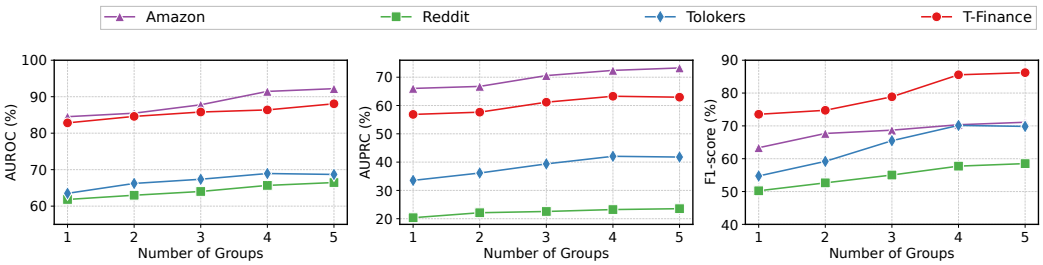


Figure 5: Sensitivity of SADOS to the number of unreliable groups (K).

Algorithm 1 Confidence-Aware Shifted Augmentation

Require: Node representations $\{\mathbf{h}_v\}_{v\in\mathcal{V}}$; reliable set $\mathcal{V}_h$; unreliable groups $\{\mathcal{V}_{l_i}\}_{i=1}^{K}$; calibrated confidences $\{s_v'\}$; pseudo-labels $\{\hat{y}_v\}$; prototype memories $\{\mathbf{p}_{l_i}^{(t-1)}\}_{i=1}^{K}$; top-$M$ size M.

Ensure: Updated prototypes $\{\mathbf{p}_{l_i}^{(t)}\}_{i=1}^{K}$; shifted augmented set $\mathcal{H}_{\text{aug}}$.

Initialize $\mathcal{H}_{\text{aug}} \leftarrow \varnothing$.
for $i = 1, 2, \ldots, K$ **do**
  if $\mathcal{V}_{l_i}^{\text{nor}}$ is empty **then**
    Keep the previous prototype and continue.
  end if
  Select up to M pseudo-normal nodes with the highest calibrated confidence scores from $\mathcal{V}_{\ell_i}^{\text{nor}}$.
  Aggregate selected nodes to obtain a candidate prototype.
  Update the prototype memory by exponential moving average.
  Estimate the group-level reliability bias.
end for
Select reliable normal nodes $\mathcal{V}_h^{\text{nor}} = \{v \in \mathcal{V}_h \mid \hat{y}_v = y_{\text{nor}}\}$.
for all $v \in \mathcal{V}_h^{\text{nor}}$ **do**
  Compute prototype attention with node-prototype similarity and reliability bias.
  Generate the shifted representation $\mathbf{h}_v'$ by injecting the prototype-induced offset into $\mathbf{h}_v$.
  Add $(v, \mathbf{h}_v')$ into $\mathcal{H}_{\text{aug}}$.
end for
return $\{\mathbf{p}_{l_i}^{(t)}\}_{i=1}^{K}$ and $\mathcal{H}_{\text{aug}}$.

increasing K offers diminishing returns, while maintaining robust performance, demonstrating that SADOS is not overly sensitive to the choice of grouping granularity.

D Algorithms and Complexity Analysis

Algorithms. We here provide the pseudo-code of the two main optimization modules in SADOS. Algorithm 1 summarizes the confidence-aware shifted augmentation module, which distills shift information from confidence-stratified unreliable nodes and injects it into reliable normal nodes. Algorithm 2 describes the overall test-time adaptation process with dynamic objective scheduling.

Algorithm 1 first updates one prototype for each confidence-stratified unreliable group. These prototypes summarize the drift patterns in unreliable target nodes while reducing the effect of noisy individual samples. Then, for each reliable normal node, SADOS computes its attention over the prototypes and generates a shifted representation by adding a prototype-induced offset. The resulting augmented samples are used together with reliable original samples for supervised test-time tuning.

Algorithm 2 presents the overall adaptation process. At each adaptation epoch, SADOS first refreshes node representations and partitions test nodes according to calibrated confidence. It then uses Algorithm 1 to generate shifted augmentations for reliable normal nodes. Meanwhile, SADOS constructs two perturbed graph views and applies contrastive learning on unreliable nodes to capture target-domain drift. The optimization objective is dynamically scheduled so that adaptation gradually shifts from early drift exploration to later anomaly-discriminative fine-tuning.

Algorithm 2 Dynamic Objective Scheduling for TTA

Require: Test graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}, \mathbf{X})$; pre-trained detector $g_\theta = c_\theta \circ f_\theta$; adaptation epochs T_{ada}; initial supervised weight c_0; learning rate ξ.
Ensure: Adapted detector g_θ and anomaly scores $\{a_v\}$.

for $t = 0, 1, \ldots, T_{\text{ada}} - 1$ **do**
  Encode the test graph:

$$\mathbf{H} = f_\theta(\mathbf{X}, \mathcal{E}).$$

  Partition nodes into reliable set $\mathcal{V}_h$ and unreliable groups $\{\mathcal{V}_{l_i}\}_{i=1}^{K}$ following Section 3.1.
  if $t = 0$ **then**
    Initialize prototype memories for $i = 1, \ldots, K$:
    $p_{\ell i}^{(0)} \leftarrow \text{norm}(\alpha \bar{h}_{\ell i}^{\text{nor}} + (1 - \alpha) w_{\text{nor}})$.
  end if
  Run Algorithm 1 to obtain shifted augmentations $\mathcal{H}_{\text{aug}}$.
  Construct semantic and structural graph views for unreliable nodes, and compute the contrastive loss $\mathcal{L}_{\text{con}}$.
  Construct the supervised tuning set:

$$\mathcal{H}_{\text{sup}} = \{(v, \mathbf{h}_v) \mid v \in \mathcal{V}_h\} \cup \mathcal{H}_{\text{aug}}.$$

  Compute the supervised loss $\mathcal{L}_{\text{sup}}$ on $\mathcal{H}_{\text{sup}}$ with pseudo-labels.
  Schedule the optimization objective:

$$\mathcal{L}^{(t)} = s(t)\mathcal{L}_{\text{sup}} + (1 - s(t))\mathcal{L}_{\text{con}}.$$

  Update model parameters θ.
end for
Compute anomaly scores a_v for all $v \in \mathcal{V}$ using the adapted detector g_θ.
return Adapted detector g_θ and anomaly scores $\{a_v\}$.

Complexity Analysis. We analyze the computational cost of the two main optimization modules in Algorithms 1 and 2. Let $N = |\mathcal{V}|$ be the number of test nodes, d be the hidden dimension, K be the number of unreliable confidence groups, M be the number of selected nodes per group for prototype extraction, B be the mini-batch size for contrastive learning, $N_l = |\mathcal{V}_l|$, and $N_h^+ = |\mathcal{V}_h^{\text{nor}}|$. We denote the cost of one graph encoder forward pass by C_{enc}, and the number of adaptation epochs by T_{ada}.

Confidence-aware shifted augmentation. Algorithm 1 updates the prototypes for the K unreliable groups and then injects them into reliable normal nodes. Selecting top-M nodes from unreliable groups costs $\mathcal{O}(N_l \log M)$. Aggregating selected nodes and updating prototype memories cost $\mathcal{O}(KMd)$. Computing prototype attention for reliable normal nodes costs $\mathcal{O}(N_h^+ Kd)$. Therefore, the augmentation module costs $\mathcal{O}(N_l \log M + KMd + N_h^+ Kd)$ per adaptation epoch. Since K and M are small in practice, this module is lightweight compared with graph encoding.

Dynamic objective scheduling. Algorithm 2 is dominated by graph encoder forward passes and contrastive similarity computation. At each epoch, SADOS encodes the original test graph and two augmented graph views, which costs $\mathcal{O}(3C_{\text{enc}})$. The contrastive loss over unreliable nodes costs $\mathcal{O}(N_l Bd)$ with mini-batch negatives. Other operations, such as view construction, supervised classification, and weight scheduling, are relatively lightweight. Therefore, the per-epoch cost of dynamic objective scheduling can be simplified as $\mathcal{O}(3C_{\text{enc}} + N_l Bd)$. When full-batch contrastive learning is used, the term $\mathcal{O}(N_l Bd)$ becomes $\mathcal{O}(N_l^2 d)$.

Combining the two modules, the total test-time adaptation cost over T_{ada} epochs is dominated by $\mathcal{O}\left(T_{\text{ada}}(3C_{\text{enc}} + N_l Bd)\right)$. The prototype-based shifted augmentation only adds a small lower-order overhead, because it operates on K group prototypes rather than all node pairs.

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 explicitly state the primary contributions, including the SADOS framework, the confidence-aware shifted augmentation, and the dynamic objective scheduling strategy. The scope is clearly defined as test-time adaptation for graph anomaly detection under distribution shifts.

   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 paper discusses limitations and trade-offs in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present formal theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All the implementation details, including dataset statistics, hyperparameter settings, and model architectures, can be found in Section 4.1 and Appendix B. We also provide detailed pseudocode for our core algorithms in Appendix D to ensure reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [Yes]

Justification: The paper provides open-source code and data in supplemental material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We list experimental setting in Section 4.1 and Appendix B.

   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: Tables 1 and 4 report mean ± standard deviation over five independent runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Section 4.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We promise that this paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [No]

    Justification: The paper does not explicitly discuss potential positive and negative societal impacts of the work. The research focuses on improving test-time graph anomaly detection under distribution shifts, which has general benefits for fraud detection and cybersecurity but could benefit from a discussion of broader implications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper properly cites and credits the original sources of the datasets, baseline methods, and backbone models used in the experiments, including Amazon, YelpChi, Reddit, Weibo, T-Finance, GHRN, and so on.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper provides the implementation code for SADOS and documents the proposed methodology to facilitate reproducibility.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or 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, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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