PUPPY: PERTURBATION-AWARE TOPOLOGICAL PURIFICATION WITH PROXY RECTIFICATION FOR TEST-TIME GRAPH ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph anomaly detection (GAD) aims to identify anomalous patterns in graph-structured data, with conventional methods typically relying on the independent and identically distributed (i.i.d.) assumption. In real-world scenarios, previously unseen yet normal patterns may emerge during deployment, introducing normality shifts that degrade detection performance. However, existing test-time adaptation methods remain unsatisfactory due to heterogeneous semantic pattern conflicts and biased predictions under normality shifts. Towards this end, in this paper, we propose a novel approach named Perturbation-aware Topological Purification with Proxy Rectification (PUPPY) for test-time GAD. The core of our proposed PUPPY is to jointly suppress structural contamination and correct prediction bias under normality shifts. Given the pretrained GAD detector, on the one hand, we estimate the sensitivity of each node under graph perturbations, which further guides selective edge pruning for test graph structure transformation. On the other hand, we introduce a self-distillation framework that leverages a proxy model trained on high-confidence target nodes as a reliable teacher. In this way, we model the adapting detector's prediction drift relative to the pretrained source model and debias the student prediction with mutual adaptation distillation. Extensive experiments across diverse test-time graph anomaly detection benchmarks demonstrate the effectiveness of our proposed PUPPY against the baseline method. Our code is available at https://anonymous.4open.science/r/PUPPY-code-94F2/.

1 INTRODUCTION

Graph anomaly detection (GAD) aims to identify rare or malicious entities that deviate from dominant patterns in graph-structured data, with broad applications in financial fraud detection (Tang et al., 2022), malicious-user discovery (Rayana & Akoglu, 2015), and industrial network security (Wu et al., 2021). Unlike anomaly detection in Euclidean space, GAD must jointly reason over node attributes and relational structure, making anomalous patterns highly context-dependent (Akoglu et al., 2015; Ma et al., 2021). Recent advances have substantially improved graph anomaly modeling through more expressive architectures and learning objectives (Tang et al., 2023; Gao et al., 2023; Chen et al., 2024). In deployment, however, graph attributes and topology continuously evolve, creating target distributions that may differ markedly from those observed during training and consequently weakening source-learned detection patterns.

Test-time adaptation (TTA) offers a natural way to address this mismatch by adapting pretrained models directly on unlabeled target data. Early efforts mainly rely on self-supervised target optimization (Sun et al., 2020; Wang et al., 2020), while subsequent studies improve adaptation through selective or continual updating (Wang et al., 2022; Niu et al., 2022; 2023) and preservation of source knowledge (Niu et al., 2022). Graph TTA further extends these ideas to graph data through target graph transformation (Jin et al., 2022), structural augmentation (Ju et al., 2023), and adaptation guided by graph structure (Mao et al., 2024; Zhang et al., 2024). While these methods adapt target representations or structures to reduce distribution mismatch, GAD introduces an additional difficulty because anomaly decisions are defined relative to the normal patterns learned from source data. When previously unseen benign patterns emerge, they not only change the relational context used for message passing, but also challenge the prediction patterns inherited from the source detector.

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

Consequently, both the structural evidence used to form target representations and the prediction signals used to guide adaptation may become inconsistent with the evolved target normality.

This gives rise to two intertwined challenges in test-time GAD, as illustrated in Figure 1.

❶ ***Structural contamination under normality shift.*** Graph message passing couples each node with its local context. When evolved normal patterns no longer conform to source-learned semantics, their relational interactions can introduce mismatched information into neighborhood aggregation and distort target representations. The challenge is therefore not merely to transform the target graph, but to determine where structural denoising is needed and which interactions should be suppressed. ❷ ***Prediction bias under evolving target distributions.*** Normality shift also changes how source-learned predictions should be interpreted on the target graph. Benign but unfamiliar nodes may remain suspicious under the source detector, and reusing such predictions as adaptation supervision can preserve this source-induced bias. Target supervision must therefore evolve beyond its source-initialized state as adaptation progresses.

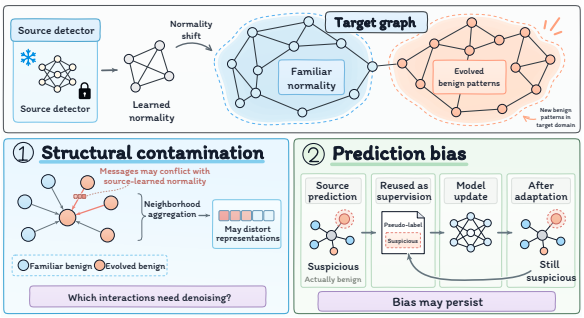


Figure 1: Motivation under test-time normality shift. Evolved benign patterns can contaminate neighborhood aggregation, while source-biased predictions reused for adaptation may preserve prediction bias.

Toward this end, we propose a novel approach named **P**ert**u**rbation-aware Topological **P**urification with **P**rox**y** (PUPPY) for test-time GAD. PUPPY addresses the two effects of normality shift through complementary structure and supervision adaptation. For structure denoising, we probe the target graph with joint feature and structural perturbations and measure the induced node-wise representation changes. The resulting perturbation sensitivity, together with local connectivity, guides selective edge pruning toward structurally vulnerable regions, producing denoised graph views for adaptation. For proxy refinement, we first initialize a target teacher from high-confidence target anchors. As the student adapts, its prediction drift relative to the frozen source detector is used to refine the target proxy, allowing supervision to evolve with the target model rather than remain fixed to its initial state. Through teacher–student adaptation, the refined proxy supervises the student on denoised graph views, while the updated teacher further refreshes structural sensitivity throughout adaptation. Extensive experiments across diverse GAD benchmarks and detector backbones demonstrate consistent gains, with additional results supporting the generalization of PUPPY and the complementary roles of structure purification and proxy refinement.

Our main contributions are summarized as follows: ❶ ***New Perspective.*** We recast test-time GAD under normality shift as a joint problem of repairing target structural context and evolving target supervision, moving beyond adaptation that treats the shifted graph merely as an unlabeled optimization target. ❷ ***Novel Methodology.*** We propose PUPPY, which couples perturbation-aware structure denoising with adaptive proxy refinement, perturbation-induced representation sensitivity localizes structural denoising, while source-referenced prediction drift progressively refines target supervision through teacher–student adaptation. ❸ ***State-of-the-art Performance.*** PUPPY achieves state-of-the-art performance across diverse GAD benchmarks and detector backbones, including both controlled normality shifts and natural temporal evolution.

2 Related Work

Graph Anomaly Detection. Graph anomaly detection has progressively moved from identifying deviations within a fixed graph toward learning anomaly semantics that remain useful beyond the observed distribution. Early deep methods jointly model node attributes and graph structure through reconstruction (Ding et al., 2019; Fan et al., 2020), while contrastive learning exposes local contextual inconsistency (Liu et al., 2021b). Subsequent studies address class imbalance with PC-GNN (Liu et al., 2021a), characterize anomalies from spectral perspectives (Tang et al., 2022; Gao et al., 2023), and mitigate misleading structural connections through SparseGAD (Gong et al., 2023). GADBench (Tang et al., 2023) further reveals substantial variation in detector behavior across graph

domains and architectures, motivating stronger generalization beyond individual benchmarks. Recent work follows this direction more explicitly: ARC (Liu et al., 2024) explores generalist detection through in-context learning, NSReg (Wang et al., 2025) targets open-set anomalies through normal-structure regularization, UNPrompt (Niu et al., 2024) transfers anomaly knowledge across graphs, and DEMO (Yu et al., 2026) strengthens open-set detection through adaptive anomaly synthesis. These advances improve robustness to previously unseen patterns, yet robustness acquired before deployment cannot directly account for later changes in what constitutes normal behavior. Once benign semantics evolve on the target graph, the problem shifts from *generalizing with a fixed detector* to *adapting the detector to evolving normality*.

Test-Time Adaptation for Graph Learning. Test-time adaptation addresses this setting by exploiting target observations after deployment. Early studies establish self-supervised test-time training (Sun et al., 2020) and entropy-based adaptation (Wang et al., 2020), while T3A (Iwasawa & Matsuo, 2021) performs lightweight classifier adjustment and TTT++ (Liu et al., 2021c) introduces feature-level regularization. MEMO (Zhang et al., 2022) further exploits augmentation consistency, whereas AdaContrast (Chen et al., 2022) combines contrastive learning with online pseudo-label refinement. For evolving test streams, NOTE (Gong et al., 2022), RMT (Döbler et al., 2023), TeSLA (Tomar et al., 2023), and RoTTA (Yuan et al., 2023) address temporal correlation, accumulated errors, and unstable updates from different perspectives. Source-free adaptation likewise studies how pretrained knowledge can be preserved without revisiting source data (Liang et al., 2020b; Yang et al., 2021; Tang et al., 2025). For graphs, distribution shifts further affect both node semantics and relational structure. GTrans (Jin et al., 2022) adapts target graph data directly, GraphPatcher (Ju et al., 2023) modifies local neighborhoods at test time, and Matcha (Bao et al., 2025) addresses structural shifts through adaptive graph aggregation. TUNE (Pan et al., 2026) further extends test-time adaptation to GAD by explicitly considering normality shift. Despite these advances, evolving normality can still make both relational context and prediction-derived supervision unreliable. `PUPPY` addresses these two aspects jointly through perturbation-aware structure denoising and adaptive proxy refinement.

3 METHODOLOGY

3.1 PRELIMINARIES AND OVERVIEW

Problem Setup. Let $\mathcal{G}_s$ and $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ denote the source and unlabeled target graphs drawn from $\mathbb{P}_s$ and $\mathbb{P}_t$. Under test-time normality shift, $\mathbb{P}_s \neq \mathbb{P}_t$ because the target graph may contain unseen benign patterns alongside seen normal patterns and anomalies. Given a GAD detector $f_s(\cdot)$ pretrained on $\mathcal{G}_s$, our goal is to adapt it to $\mathcal{G}$ without target labels so that seen and unseen normal nodes can be distinguished from anomalies. We keep f_s frozen and update residual feature adapters $g_\theta(\boldsymbol{x}_i) = \boldsymbol{x}_i + \boldsymbol{W}_\theta \boldsymbol{x}_i + \boldsymbol{b}_\theta$. For node v_i, let $m_i = z_{i,1} - z_{i,0}$ denote the anomaly margin and $p_i = \sigma(m_i)$ the anomaly probability. Evaluating the source detector on the target graph gives references m_i^s and p_i^s. We maintain teacher and student adapters θ_T and θ_S, with $m_{i,\text{orig}}^{T/S}$ and $m_{i,\text{view}}^{T/S}$ denoting their margins on the original graph and adaptation view. We use $\ell(m, p)$ for binary cross-entropy with logit m and soft target p.

Framework Overview. As illustrated in Figure 2, `PUPPY` adapts the shifted target graph via perturbation-aware structure denoising and adaptive proxy refinement within a teacher–student framework. Given a target graph and a frozen source detector, `PUPPY` first initializes the teacher from high-confidence target anchors, establishing the initial target proxy. It then probes the target graph with controlled feature and structural perturbations and measures the induced representation sensitivity. This sensitivity is propagated to pairwise interactions to guide edge pruning and construct denoised graph views. Meanwhile, the student's prediction drift relative to the frozen source detector is used to progressively refine the target proxy as adaptation proceeds. The refined proxy supervises the student on the denoised views, while the teacher is periodically updated via exponential moving average. As the teacher evolves, representation sensitivity is recomputed accordingly, allowing structure denoising and proxy refinement to evolve jointly throughout adaptation.

3.2 PERTURBATION-AWARE STRUCTURE DENOISING

As benign target behavior evolves, the relational context learned by the source detector can become unevenly informative across the graph. Rather than modifying structure uniformly, `PUPPY`

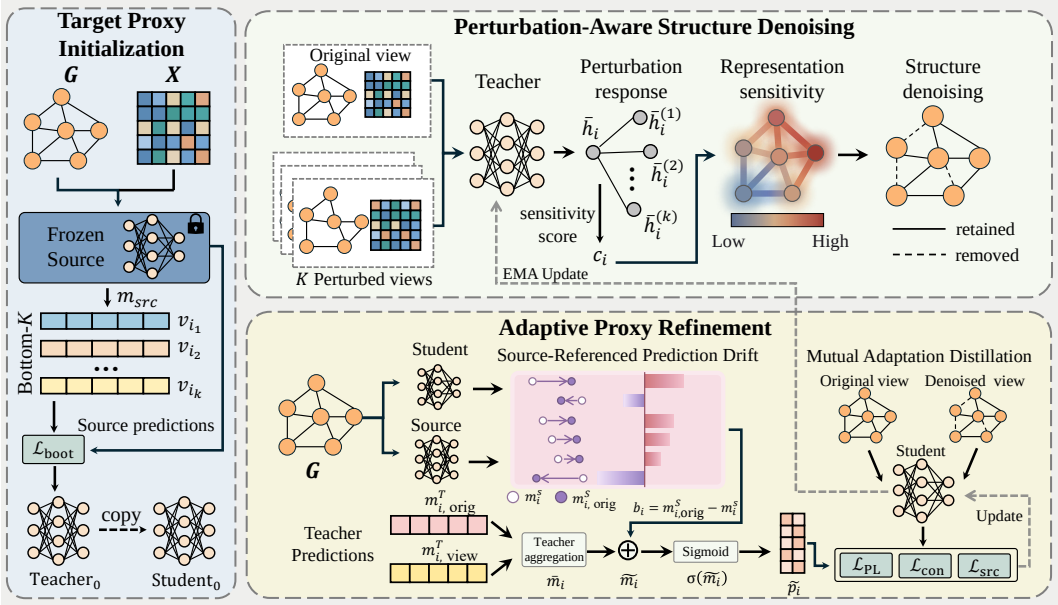


Figure 2: Overview of PUPPY. PUPPY performs test-time adaptation through perturbation-aware structure denoising and adaptive proxy refinement, where representation sensitivity guides selective edge pruning and source-referenced prediction drift updates the target proxy during adaptation.

uses perturbation-induced representation changes to locate regions that are more sensitive to local variation, and combines this signal with graph connectivity to guide selective structure denoising.

Perturbation-Guided Sensitivity Estimation. To expose local sensitivity, we construct K stochastic probe views by perturbing node features and graph structure. Feature perturbation applies random masking and Gaussian noise, while structural perturbation randomly removes edges subject to a minimum-degree constraint. Let $\boldsymbol{h}_i$ and $\boldsymbol{h}_i^{(k)}$ denote the proxy representations of node v_i on the original graph and the k-th probe view. The representation sensitivity of v_i is defined as:

$$c_i = \frac{1}{K}\sum_{k=1}^{K}\left\|\bar{\boldsymbol{h}}_i^{(k)} - \bar{\boldsymbol{h}}_i\right\|_2, \tag{1}$$

where $\bar{\boldsymbol{h}}$ denotes the ℓ_2-normalized representation. We further map c_i to its percentile rank $q_i \in [0,1]$ within the target graph, yielding a graph-relative sensitivity score that is comparable across nodes.

Sensitivity-Guided Structural Denoising. The node-level score indicates where structural denoising should concentrate, but the actual pruning strength also depends on the local connectivity pattern. Let d_i denote the non-self-loop degree of node v_i and $d_{\min}$ the minimum retained degree. We assign each node a pruning budget, which can be defined as:

$$u_i = q_i^{\gamma}\frac{\max(d_i - d_{\min}, 0)}{\max(d_i, 1)}, \tag{2}$$

where γ controls the emphasis on highly sensitive nodes. The degree factor progressively reduces pruning as a node approaches the minimum retained degree. To translate node sensitivity into edge-level pruning, we define sensitivity for an edge (v_i, v_j) as:

$$r_{ij} = \max\left(\frac{c_i}{d_j+1}, \frac{c_j}{d_i+1}\right), \tag{3}$$

which combines endpoint sensitivity with local connectivity. Each node allocates its pruning budget over incident edges according to these pairwise scores. For edge (v_i, v_j), the proposal from v_i is:

$$\pi_{ij}^{(i)} = u_i\frac{\exp(r_{ij}/\tau_e)}{\sum_{v_k\in\mathcal{N}(i)}\exp(r_{ik}/\tau_e)}, \qquad P_{ij}^{\text{drop}} = 1 - \left(1-\pi_{ij}^{(i)}\right)\left(1-\pi_{ij}^{(j)}\right), \tag{4}$$

where τ_e controls the concentration of the pairwise allocation process. Edges are then sampled according to P_{ij}^{drop}, with self-loops retained and edges restored when necessary to satisfy the

minimum-degree constraint. Together with feature perturbation, the pruned structure forms the denoised view used for subsequent adaptation.

3.3 Adaptive Proxy Refinement for Target Supervision

Structure denoising reshapes the target context used for adaptation, but the supervision driving the model must evolve as well. Source predictions provide a useful prior for target adaptation (Liang et al., 2020a; Iwasawa & Matsuo, 2021), while fixed supervision may become increasingly misaligned as adaptation proceeds under distribution shift (Wang et al., 2022; Chen et al., 2022). `PUPPY` therefore maintains a target proxy that is initialized from source-informed target anchors and progressively refined according to the student's departure from the source prediction state.

Reliable Teacher Initialization as Target Proxy. We identify normal anchors using the frozen source detector and use them to establish the initial target proxy. Given an anchor ratio ρ_a, we select the $k_a = \max(1, \lfloor \rho_a N \rfloor)$ nodes with the smallest source anomaly margins, defined as:

$$\mathcal{A} = \mathrm{BottomK}_{k_a}\{m_i^s : i \in \mathcal{V}\}, \tag{5}$$

and assign them the normal target. The teacher adapter is initialized on stochastic jointly perturbed views by optimizing:

$$\mathcal{L}_{\mathrm{boot}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \ell(m_{i,\mathrm{view}}^T, 0) + \lambda_s \frac{1}{N}\sum_{i\in\mathcal{V}} \ell(m_{i,\mathrm{view}}^T, p_i^s). \tag{6}$$

The anchor term introduces target-side normality into the teacher, while the second term preserves the prediction structure inherited from the source detector. The initialized teacher serves as the target proxy, and its parameters are copied to the student.

Source-Referenced Drift for Proxy Refinement. Once adaptation begins, the student gradually departs from the source prediction state, while a fixed proxy cannot reflect this evolution. We use the frozen source detector as a stationary reference and incorporate the student's target-side drift into the proxy. At each adaptation step, the teacher first aggregates its predictions from the original graph and the current denoised view as:

$$\bar{m}_i^T = \frac{1}{2} m_{i,\mathrm{orig}}^T + \frac{1}{2} m_{i,\mathrm{view}}^T. \tag{7}$$

The student's prediction drift relative to the source detector is measured as:

$$b_i = \mathrm{clip}\left(m_{i,\mathrm{orig}}^S - m_i^s, -C, C\right), \tag{8}$$

where C bounds extreme deviations. We refine the proxy margin by:

$$\widetilde{m}_i = \bar{m}_i^T + \alpha b_i, \tag{9}$$

where α controls the refinement strength. This update shifts the teacher estimate based on the student's deviation from the source state, allowing the target proxy to track adaptation.

Mutual Adaptation Distillation. The refined proxy is further distilled into the student. We form the soft target $\widetilde{p}_i = \sigma(\widetilde{m}_i)$ while keeping the normal targets of anchor nodes fixed. Nodes with the largest absolute refined margins are selected for supervision,

$$\mathcal{P} = \mathrm{TopK}_{k_p}\{|\widetilde{m}_i| : i \in \mathcal{V}\} \cup \mathcal{A}, \qquad k_p = \max(1, \lfloor \rho_p N \rfloor). \tag{10}$$

Each selected node is assigned a confidence weight w_i, where $w_i = 1$ for anchors and $w_i = |2\widetilde{p}_i - 1|$ otherwise. Teacher predictions, source references, and the resulting soft targets are detached when constructing the supervision. The student learns from these refined targets, and the teacher is periodically updated from the student. The two models thus evolve together: student adaptation reshapes the target proxy, while the updated proxy provides supervision for subsequent adaptation.

3.4 Overall Optimization and Inference

We now integrate structure denoising and proxy refinement into the overall adaptation procedure. The student is optimized using refined proxy supervision on denoised graph views, while the teacher

is periodically updated to refresh representation sensitivity and target supervision. After adaptation, teacher and student predictions are combined for final inference. Specifically, the student first learns from the selected proxy targets as:

$$\mathcal{L}_{\mathrm{PL}} = \frac{\sum_{i\in\mathcal{P}} w_i\, \ell(m^S_{i,\mathrm{view}}, \widetilde{p}_i)}{\max\left(\sum_{i\in\mathcal{P}} w_i, 1\right)}. \tag{11}$$

To align predictions across the original graph and its denoised view, we further impose the loss,

$$\mathcal{L}_{\mathrm{con}} = \frac{1}{N}\sum_{i\in\mathcal{V}} \left[\sigma(m^S_{i,\mathrm{view}}) - \sigma(m^S_{i,\mathrm{orig}})\right]^2. \tag{12}$$

Meanwhile, the source prediction structure on the original graph is retained through the loss:

$$\mathcal{L}_{\mathrm{src}} = \frac{1}{N}\sum_{i\in\mathcal{V}} \ell(m^S_{i,\mathrm{orig}}, p^s_i). \tag{13}$$

The overall adaptation objective is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{PL}} + \mathcal{L}_{\mathrm{con}} + \mathcal{L}_{\mathrm{src}}. \tag{14}$$

Only the student adapter is optimized, while the pretrained detector remains fixed. The teacher is periodically updated from the student through exponential moving average (EMA),

$$\theta_T \leftarrow \mu\theta_T + (1-\mu)\theta_S, \tag{15}$$

where μ denotes the update momentum. Each teacher update also refreshes the representation sensitivity used for structure denoising. Between updates, the sensitivity estimates are reused, while denoised views and refined proxy targets are regenerated at each adaptation step. As adaptation proceeds, the updated teacher reshapes the structural denoising process, while the evolving student provides new prediction drift for proxy refinement, allowing the two components to evolve jointly with the target model. After adaptation, teacher and student predictions on the original target graph are averaged to obtain the anomaly score,

$$s_i = \frac{1}{2}\sigma(m^T_{i,\mathrm{orig}}) + \frac{1}{2}\sigma(m^S_{i,\mathrm{orig}}). \tag{16}$$

The resulting s_i is used as the final anomaly score for node v_i.

3.5 Theoretical Analysis

We analyze the stability of perturbation-aware structure denoising under finite stochastic probing. Specifically, we bound the deviation between the empirical pruning policy induced by K probe views and its population counterpart.

Finite-Probe Stability of Structure Denoising. Fix the teacher and an undirected target graph $G = (V, E, \boldsymbol{X})$ with $N = |V| \geq 2$. For node i, define

$$Z_{ik} = \|\bar{\boldsymbol{h}}^{(k)}_i - \bar{\boldsymbol{h}}_i\|_2, \qquad c_i = \frac{1}{K}\sum_{k=1}^{K} Z_{ik}, \qquad c^\star_i = \mathbb{E}[Z_{ik}].$$

Since the representations are normalized, $0 \leq Z_{ik} \leq 2$. All expectations and probabilities below are conditioned on the fixed graph and teacher. Let

$$q_i = \frac{\mathrm{rank}(c_i) - 1}{N-1}$$

denote the ascending normalized rank of c_i, with ties broken by a fixed index-based rule. Applying the same ranking and denoising rules to $\boldsymbol{c}^\star$ defines the population quantities $q^\star_i$, $u^\star_i$, $r^\star_{ij}$, $\pi^{(i),\star}_{ij}$, and $P^{\mathrm{drop},\star}_{ij}$. For $j \neq i$, define

$$\Delta_{ij} = |c^\star_i - c^\star_j|, \qquad \Psi_i(K) = \frac{1}{N-1}\sum_{j\neq i} \exp\left(-\frac{K\Delta^2_{ij}}{8}\right).$$

We further define

$$a_i = \frac{(d_i - d_{\min})_+}{\max\{d_i, 1\}}, \qquad \kappa_i = \max_{v_j \in \mathcal{N}(i)} \frac{1}{\min\{d_i, d_j\} + 1},$$

where $(x)_+ = \max\{x, 0\}$ and $\kappa_i = 0$ for isolated nodes, and

$$\eta_i(K) = \int_0^2 \min\left\{1,\, 2(d_i+1)\exp\left(-\frac{Kt^2}{2}\right)\right\} dt.$$

Assume $d_{\min} \geq 0$, $\gamma > 0$, and $\tau_e > 0$, and let

$$\beta = \min\{1, \gamma\}, \qquad L_\gamma = \max\{1, \gamma\}.$$

Endpoint softmax is defined only for non-isolated nodes; isolated nodes receive zero pruning budget.

Theorem 3.1. *The finite-probe pruning policy satisfies*

$$\mathbb{E}|q_i - q_i^\star| \leq \Psi_i(K), \tag{17}$$

$$\mathbb{E}|u_i - u_i^\star| \leq a_i L_\gamma [\Psi_i(K)]^\beta, \tag{18}$$

$$\mathbb{E} \sum_{\{i,j\} \in E} |P_{ij}^{\text{drop}} - P_{ij}^{\text{drop},\star}| \leq L_\gamma \sum_i a_i [\Psi_i(K)]^\beta + \frac{1}{\tau_e} \sum_i u_i^\star \kappa_i \eta_i(K). \tag{19}$$

The expectation is over the K stochastic probe views, and the deletion probabilities are evaluated before degree restoration.

The proof is provided in Appendix A. Theorem 3.1 characterizes how finite-probe uncertainty affects the denoising policy. Each pairwise rank error decays exponentially with its population sensitivity gap Δ_{ij}, which in turn controls the deviation of pruning budgets. The final bound additionally captures local sensitivity-estimation error in edge allocation through $\eta_i(K)$, with its effect modulated by the softmax temperature τ_e.

4 Experiment

4.1 Experimental Settings

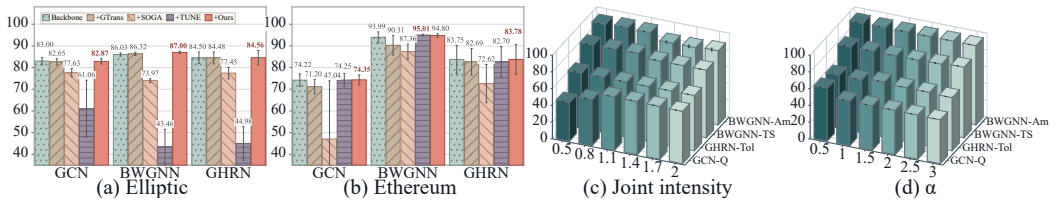
Datasets and Evaluation Metrics. We conduct experiments on eight public GAD benchmarks spanning diverse domains and scales, including Amazon (McAuley & Leskovec, 2013), YelpChi (Rayana & Akoglu, 2015), Reddit and Weibo (Kumar et al., 2019), Tolokers and Questions (Platonov et al., 2023), and T-Finance and T-Social (Tang et al., 2022). To construct controlled normality shifts, normal nodes are clustered by their features, with only the largest cluster retained for source training and validation and the remaining clusters reserved for testing (Pan et al., 2026). We evaluate anomaly detection using AUROC and AUPRC. Each experiment is repeated five times, and we report the mean and population standard deviation. Detailed dataset statistics are provided in the Appendix.

Baselines. We compare `PUPPY` with five representative supervised GAD methods, including CareGNN (Dou et al., 2020), H2-FDetector (Shi et al., 2022), GAGA (Wang et al., 2023), PMP (Zhuo et al., 2024), and ConsisGAD (Chen et al., 2024). We further include four test-time adaptation methods: GTrans (Jin et al., 2022), GraphPatcher (Ju et al., 2023), SOGA (Mao et al., 2024), and TUNE (Pan et al., 2026). Following prior GAD evaluations (Tang et al., 2023; Pan et al., 2026), all adaptation methods are evaluated with GCN (Kipf & Welling, 2016), BWGNN (Tang et al., 2022), and GHRN (Gao et al., 2023) as detector backbones.

Implementation Details. `PUPPY` is implemented in PyTorch and optimized with Adam. Each adaptation run starts from a pretrained source detector, which remains fixed while only the residual adapters are updated. For most tasks, we use $K = 4$ probe views for sensitivity estimation. The minimum retained degree is typically set to $d_{\min} \in \{1, 2\}$, and the teacher is updated every $I \in [1, 20]$ adaptation epochs according to the task configuration. Full configurations and implementation details are provided in our released code.

Table 1: Main AUROC results (%). Results are reported as mean±std over five runs. **Bold red** marks the best overall result for each dataset, while underlining marks the best TTA result within each backbone block. OOM and OOT denote out-of-memory and out-of-time, respectively.

Model	Amazon	YelpChi	Reddit	Weibo	Tolokers	Questions	T-Finance	T-Social
CareGNN	77.03±2.18	53.47±1.53	56.79±0.91	66.85±0.91	65.74±0.20	50.45±0.04	86.14±2.91	63.40±2.00
H2-FDetector	61.19±13.84	53.34±0.75	51.31±3.58	61.56±12.08	61.30±2.21	50.44±4.08	OOM	OOM
GAGA	87.44±1.26	52.03±0.64	52.25±4.54	94.43±1.76	60.53±2.40	51.80±1.34	93.64±0.89	OOT
PMP	80.78±2.17	55.62±2.42	56.31±0.50	80.25±1.59	66.87±1.03	58.14±4.93	80.67±6.69	OOM
ConsisGAD	90.32±0.73	50.97±0.42	62.93±1.21	93.43±0.52	59.55±1.28	48.35±0.83	85.57±0.79	70.36±4.40
GCN	21.21±1.02	47.70±0.16	62.88±0.44	86.30±14.06	63.20±0.26	51.58±0.44	84.08±1.32	69.79±8.03
+ GTrans	49.28±11.12	49.37±0.60	47.13±4.64	82.56±4.67	60.15±4.17	40.97±0.99	68.89±1.87	78.76±0.05
+ GraphPatcher	OOM	OOM	62.27±1.32	73.71±20.73	64.97±0.42	57.37±1.52	OOM	OOM
+ SOGA	22.81±2.90	49.30±0.32	61.08±0.65	79.17±20.35	64.67±2.01	50.91±0.45	77.18±2.07	76.56±0.02
+ TUNE	55.10±22.83	54.50±0.74	63.62±0.03	96.16±0.01	66.27±0.07	60.61±0.97	85.53±0.04	81.89±0.02
+ Ours	80.39±0.59	56.21±0.05	65.16±0.05	96.81±0.15	**67.04±0.00**	63.45±0.09	88.10±1.20	84.26±0.01
BWGNN	83.38±1.75	51.87±0.71	59.62±1.50	93.04±1.10	62.02±0.87	50.40±0.45	92.35±1.27	80.14±2.08
+ GTrans	76.85±3.50	51.41±0.55	52.92±0.88	88.43±3.04	60.06±2.51	49.05±1.27	84.07±1.71	82.51±0.02
+ GraphPatcher	OOM	OOM	60.52±1.78	84.83±12.48	58.72±1.51	58.67±0.50	OOM	OOM
+ SOGA	77.61±2.60	50.58±0.25	58.34±2.47	77.21±5.65	62.26±1.21	49.68±0.41	85.12±2.84	81.26±0.00
+ TUNE	92.19±0.67	60.58±0.87	61.34±0.20	95.26±0.49	65.51±0.04	63.21±0.37	94.28±0.53	82.72±0.01
+ Ours	94.43±0.20	**62.00±1.49**	64.45±0.19	**96.96±0.31**	66.77±0.23	**66.30±0.06**	**95.12±0.07**	84.77±0.05
GHRN	81.58±2.19	52.75±0.35	61.17±1.01	90.34±2.48	60.34±0.59	49.65±0.82	92.72±0.82	85.73±1.94
+ GTrans	79.12±4.03	52.05±0.40	51.05±2.86	88.14±5.02	63.44±1.74	48.75±0.78	78.52±7.40	88.13±0.01
+ GraphPatcher	OOM	OOM	59.57±4.49	95.38±0.49	62.67±2.76	58.68±0.74	OOM	OOM
+ SOGA	77.62±1.57	52.36±0.49	56.87±2.50	77.26±7.22	62.67±1.08	48.57±0.87	87.50±4.03	**88.74±0.00**
+ TUNE	82.11±6.71	58.36±0.66	63.46±0.10	92.31±0.01	66.17±0.42	58.88±0.39	93.28±0.04	88.42±0.01
+ Ours	**96.35±0.04**	60.60±0.07	**65.91±0.42**	94.57±0.16	66.25±0.01	64.90±1.97	93.92±0.02	88.61±0.01



(a) Elliptic (b) Ethereum (c) Joint intensity (d) α

Figure 3: Generalization and sensitivity analysis in AUROC (%). (a)–(b) Natural temporal shifts on Elliptic and Ethereum. (c)–(d) Sensitivity to joint perturbation intensity and refinement strength α.

4.2 Experimental Results

Performance under Normality Shifts. Table 1 summarizes the main AUROC results across eight GAD benchmarks and three detector backbones, from which several observations can be made. **Obs.❶** The relative performance of supervised GAD methods and frozen source detectors varies substantially across datasets, indicating that source-trained detection patterns do not transfer uniformly once target normality changes. **Obs.❷** Existing TTA methods improve performance in some settings but may also degrade the corresponding source detector, suggesting that generic graph adaptation does not consistently accommodate normality shifts. **Obs.❸** In contrast, `PUPPY` performs strongly across GCN, BWGNN, and GHRN, benefiting both weak and competitive source detectors. This behavior shows that its improvements stem from effective target adaptation rather than reliance on a particular backbone or source initialization.

Performance under Temporal Shifts. We further evaluate `PUPPY` on Elliptic (Weber et al., 2019) and Ethereum (Chen et al., 2020) under natural temporal shifts, where the source detector is trained on earlier transactions and adapted to later graph data, following prior temporal evaluation protocols for evolving graphs (Pareja et al., 2020). As shown in Figures 3(a)-(b), unlike the controlled shifts in the main benchmarks, the frozen source detectors remain competitive in most settings, leaving less room for adaptation and making negative transfer more likely. Several existing TTA methods degrade substantially in this regime (Niu et al., 2022), whereas `PUPPY` largely preserves the source performance while providing improvements when adaptation is beneficial. This sug-

Table 2: Component ablation on representative settings (%). Results are reported as mean±std over three seeds; the left and right columns correspond to AUROC and AUPRC, respectively.

Variant	GCN–Questions		BWGNN–T-Social		GHRN–Tolokers	
w/o Target Proxy	46.46±0.06	2.72±0.01	79.38±0.00	25.09±0.02	61.47±0.02	30.71±0.01
w/o Drift Refinement	48.19±0.02	5.14±0.01	81.90±0.02	32.99±0.04	63.31±0.01	31.68±0.00
w/o Proxy Supervision	51.45±0.00	4.87±0.00	81.49±0.01	28.72±0.17	63.31±0.01	31.67±0.00
w/o Structural Denoising	46.85±0.00	2.76±0.00	82.02±1.15	26.19±2.53	63.31±0.00	31.68±0.01
Full PUPPY	**63.48±0.10**	**6.08±0.01**	**84.79±0.04**	**35.54±0.07**	**66.25±0.01**	**34.09±0.02**

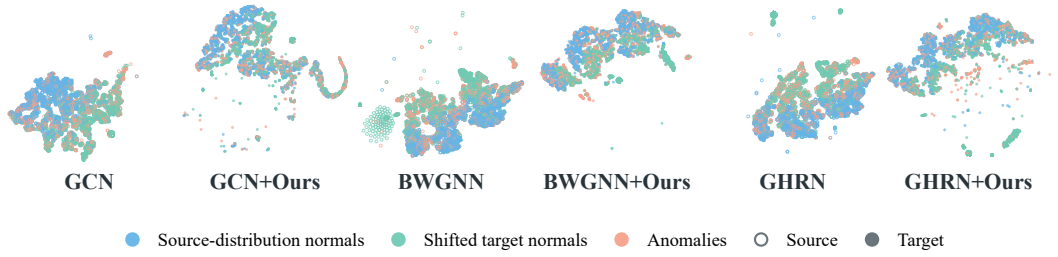


Figure 4: Representation visualization on Questions. t-SNE of node representations.

gests that perturbation-aware structure denoising and adaptive proxy refinement can accommodate evolving graph distributions without unnecessarily overriding source knowledge.

4.3 Further Analysis of PUPPY

Ablation Study. Table 2 compares the full PUPPY with four ablated variants that remove the target proxy, drift refinement, proxy supervision, or structure denoising. Several observations can be drawn: **Obs.❶** *Structure denoising is essential before adaptation.* Removing structure denoising consistently weakens performance, indicating that direct adaptation on the shifted graph can retain misleading relational context. Perturbation-aware denoising provides a cleaner structural basis for subsequent adaptation. **Obs.❷** *A useful proxy must both capture and transfer target normality.* Removing either the target proxy or proxy supervision leads to clear degradation, showing that target normality must not only be estimated but also effectively transferred to the adapting student. **Obs.❸** *Target supervision should evolve with adaptation.* Removing drift refinement further degrades performance, revealing the limitation of static proxy supervision. Source-referenced drift allows the proxy to evolve with the student, complementing structure denoising with adaptive supervision.

Parameter Analysis. Figures 3(c)–(d) reveal different sensitivities to the two adaptation factors. The refinement strength α is more task-dependent, indicating that the contribution of source-referenced drift should vary with the underlying shift rather than be increased uniformly. In contrast, joint perturbation intensity exhibits a clearer intermediate regime, balancing informative sensitivity exposure against excessive structural distortion.

Visualization. Figure 4 provides a representation-level view on Questions, where shifted target normals become better integrated with the learned normal structure while anomalies remain separated, especially for GCN and GHRN. This suggests that PUPPY better accommodates evolved target normality while preserving the discriminative structure needed for anomaly detection.

5 Conclusion

This paper presents PUPPY, a framework that combines perturbation-aware structure denoising with adaptive proxy refinement for test-time graph anomaly detection under normality shifts. PUPPY addresses evolving target normality by selectively suppressing unreliable structural interactions and progressively refining target supervision throughout adaptation. Extensive experiments across diverse graph settings demonstrate its effectiveness and generalizability. More broadly, our results highlight the importance of jointly adapting relational context and supervisory signals in test-time GAD, enabling pretrained detectors to accommodate evolving target normality rather than treating the shifted graph as a static adaptation target.

AI USE STATEMENT

Generative AI tools were used during this work to assist with research discussion and refinement, including feedback on the conceptual framing, experimental design, and interpretation of empirical results. They were also used to support literature search and organization, manuscript structuring, language editing, and the preparation and refinement of paper text. All AI-assisted suggestions were critically reviewed by the authors against the underlying methodology, experimental results, and relevant literature. The proposed method, experimental results, and final scientific claims were determined and validated by the authors. We take full responsibility for the final content of this work, including any text or research artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide the complete proofs of our theoretical results in Appendix A, the full adaptation procedure and implementation details in Appendix B, and detailed descriptions of the datasets and evaluation protocols in Appendix C. Complete experimental results, including AUPRC results, temporal-shift evaluation, and full component ablations, are reported in Appendix D, with computational complexity analyzed in Appendix E. The anonymous code repository linked in the abstract contains the implementation, task-specific configurations, and experimental scripts required to reproduce the reported results.

REFERENCES

Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688, 2015.

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*, volume 2025, pp. 56906–56944, 2025.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–305. IEEE, 2022.

Liang Chen, Jiaying Peng, Yang Liu, Jintang Li, Fenfang Xie, and Zibin Zheng. Phishing scams detection in ethereum transaction network. *ACM Transactions on Internet Technology (TOIT)*, 21 (1):1–16, 2020.

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*, volume 2024, pp. 30180–30207, 2024.

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

Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7704–7714. IEEE, 2023.

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 29th ACM international conference on information & knowledge management*, pp. 315–324, 2020.

Haoyi Fan, Fengbin Zhang, and Zuoyong Li. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5685–5689. IEEE, 2020.

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 2023*, pp. 1528–1538, 2023.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

Zheng Gong, Guifeng Wang, Ying Sun, Qi Liu, Yuting Ning, Hui Xiong, and Jingyu Peng. Beyond homophily: Robust graph anomaly detection via neural sparsification. In *Ijcai*, pp. 2104–2113, 2023.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. *arXiv preprint arXiv:2210.03561*, 2022.

Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. Graphpatcher: mitigating degree bias for graph neural networks via test-time augmentation. *Advances in neural information processing systems*, 36:55785–55801, 2023.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pp. 6028–6039. PMLR, 2020a.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pp. 6028–6039. PMLR, 2020b.

Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*, pp. 3168–3177, 2021a.

Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6):2378–2392, 2021b.

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

Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021c.

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, 2021.

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 17th ACM International Conference on Web Search and Data Mining*, pp. 520–528, 2024.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, 2013.

Chaoxi Niu, Hezhe Qiao, Changlu Chen, Ling Chen, and Guansong Pang. Zero-shot generalist graph anomaly detection with unified neighborhood prompts. *arXiv preprint arXiv:2410.14886*, 2024.

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*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

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*, volume 40, pp. 24691–24699, 2026.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5363–5370, 2020.

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? *arXiv preprint arXiv:2302.11640*, 2023.

Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pp. 985–994, 2015.

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*, pp. 1486–1494, 2022.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

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

Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in Neural Information Processing Systems*, 36:29628–29653, 2023.

Song Tang, Wenxin Su, Yan Gan, Mao Ye, Jianwei Zhang, and Xiatian Zhu. Proxy denoising for source-free domain adaptation. In *International Conference on Learning Representations*, volume 2025, pp. 82548–82569, 2025.

Devavrat Tomar, Guillaume Vray, Behzad Bozorgtabar, and Jean-Philippe Thiran. Tesla: Test-time self-learning with automatic adversarial augmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20341–20350. IEEE, 2023.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201. IEEE, 2022.

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

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 2023*, pp. 406–416, 2023.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.

Yulei Wu, Hong-Ning Dai, and Haina Tang. Graph neural networks for anomaly detection in industrial internet of things. *IEEE Internet of Things Journal*, 9(12):9214–9231, 2021.

Shiqi Yang, Joost Van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in neural information processing systems*, 34:29393–29405, 2021.

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

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15922–15932. IEEE, 2023.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

Zhen Zhang, Meihan Liu, Anhui Wang, Hongyang Chen, Zhao Li, Jiajun Bu, and Bingsheng He. Collaborate to adapt: Source-free graph domain adaptation via bi-directional adaptation. In *Proceedings of the ACM Web Conference 2024*, pp. 664–675, 2024.

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*, volume 2024, pp. 34329–34347, 2024.

A Proofs for the Theoretical Analysis

A.1 Proof of Theorem 3.1

Proof. All probabilities and expectations are conditioned on the fixed graph and teacher. Since the K probe views are sampled independently, $\{Z_{ik}\}_{k=1}^{K}$ are independent across k for each node i, with $0 \le Z_{ik} \le 2$.

We first bound the deviation of the sensitivity rank. For $j \neq i$, define the pairwise ordering indicator

$$I_{ji}(\boldsymbol{c}) = \mathbf{1}\{c_j < c_i\} + \mathbf{1}\{c_j = c_i\}\mathbf{1}\{j < i\}.$$

The fixed tie-breaking rule gives

$$q_i = \frac{1}{N-1}\sum_{j \neq i} I_{ji}(\boldsymbol{c}), \qquad q_i^\star = \frac{1}{N-1}\sum_{j \neq i} I_{ji}(\boldsymbol{c}^\star).$$

Consider a fixed pair (i, j) and let

$$Y_k = Z_{jk} - Z_{ik}, \qquad \mathbb{E}[Y_k] = c_j^\star - c_i^\star.$$

Although Z_{ik} and Z_{jk} may be dependent within the same probe view, the variables $\{Y_k\}_{k=1}^{K}$ are independent across views and satisfy $Y_k \in [-2, 2]$. If $c_i^\star \neq c_j^\star$, a change in their relative ordering requires the empirical mean of Y_k to cross its population ordering boundary. Hence, by the one-sided Hoeffding inequality (Hoeffding, 1963),

$$\Pr(I_{ji}(\boldsymbol{c}) \neq I_{ji}(\boldsymbol{c}^\star)) \le \exp\left(-\frac{K(c_i^\star - c_j^\star)^2}{8}\right).$$

When $c_i^\star = c_j^\star$, the same inequality holds trivially because its right-hand side equals one. Therefore,

$$\mathbb{E}|q_i - q_i^\star| \le \frac{1}{N-1}\sum_{j\neq i}\mathbb{E}\,|I_{ji}(\boldsymbol{c}) - I_{ji}(\boldsymbol{c}^\star)|$$
$$\le \frac{1}{N-1}\sum_{j\neq i}\exp\left(-\frac{K\Delta_{ij}^2}{8}\right) = \Psi_i(K),$$

which proves equation 17.

We next consider the pruning budget. For any $x, y \in [0,1]$,

$$|x^\gamma - y^\gamma| \le L_\gamma |x-y|^\beta, \qquad \beta = \min\{1,\gamma\}, \quad L_\gamma = \max\{1,\gamma\}.$$

Indeed, for $\gamma \ge 1$ this follows from the mean value theorem, whereas for $0 < \gamma < 1$ it follows from the γ-Hölder continuity of x^γ on $[0,1]$. Since

$$u_i = a_i q_i^\gamma, \qquad u_i^\star = a_i (q_i^\star)^\gamma,$$

we have

$$|u_i - u_i^\star| \le a_i L_\gamma |q_i - q_i^\star|^\beta.$$

Because $0 < \beta \le 1$, Jensen's inequality and equation 17 yield

$$\mathbb{E}|u_i - u_i^\star| \le a_i L_\gamma \mathbb{E}|q_i - q_i^\star|^\beta$$
$$\le a_i L_\gamma \left(\mathbb{E}|q_i - q_i^\star|\right)^\beta$$
$$\le a_i L_\gamma [\Psi_i(K)]^\beta,$$

proving equation 18.

It remains to control the edge-allocation probabilities. For a non-isolated node i, define the maximum sensitivity-estimation error over its closed neighborhood by

$$M_i = \max_{v_\ell \in \{v_i\}\cup\mathcal{N}(i)} |c_\ell - c_\ell^\star|.$$

For each node ℓ, Hoeffding's inequality gives

$$\Pr(|c_\ell - c_\ell^\star| > t) \le 2\exp\left(-\frac{Kt^2}{2}\right).$$

A union bound over the $d_i + 1$ nodes in the closed neighborhood therefore yields, for $0 \le t \le 2$,

$$\Pr(M_i > t) \le \min\left\{1,\, 2(d_i+1)\exp\left(-\frac{Kt^2}{2}\right)\right\}.$$

Since $M_i \le 2$, the tail-integral identity gives

$$\mathbb{E}[M_i] = \int_0^2 \Pr(M_i > t)\,dt \le \eta_i(K).$$

Recall that

$$r_{ij} = \max\left\{\frac{c_i}{d_j+1}, \frac{c_j}{d_i+1}\right\}.$$

Because the maximum is 1-Lipschitz with respect to the ℓ_∞ norm,

$$|r_{ij} - r_{ij}^\star| \le \frac{M_i}{\min\{d_i, d_j\}+1} \le \kappa_i M_i.$$

Let

$$\xi_{ij} = \frac{\exp(r_{ij}/\tau_e)}{\sum_{v_k\in\mathcal{N}(i)}\exp(r_{ik}/\tau_e)}, \qquad \boldsymbol{\xi}_i = (\xi_{ij})_{v_j\in\mathcal{N}(i)},$$

and define $\boldsymbol{\xi}_i^\star$ analogously.

For the temperature-scaled softmax map, its Jacobian is

$$\frac{1}{\tau_e}\left(\operatorname{diag}(p) - pp^\top\right).$$

For any vector v with $\|v\|_\infty \le 1$,

$$\left\|\left(\operatorname{diag}(p) - pp^\top\right)v\right\|_1 = \sum_j p_j|v_j - \bar{v}| \le \left(\sum_j p_j(v_j - \bar{v})^2\right)^{1/2} \le 1,$$

where $\bar{v} = \sum_j p_j v_j$. Integrating the Jacobian along the line segment between two inputs therefore gives

$$\|\operatorname{softmax}(x/\tau_e) - \operatorname{softmax}(y/\tau_e)\|_1 \le \frac{1}{\tau_e}\|x - y\|_\infty.$$

Consequently,

$$\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^\star\|_1 \le \frac{\kappa_i M_i}{\tau_e},$$

and hence

$$\mathbb{E}\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^\star\|_1 \le \frac{\kappa_i \eta_i(K)}{\tau_e}.$$

Writing

$$\boldsymbol{\pi}^{(i)} = u_i\boldsymbol{\xi}_i, \qquad \boldsymbol{\pi}^{(i),\star} = u_i^\star\boldsymbol{\xi}_i^\star,$$

and using $\|\boldsymbol{\xi}_i\|_1 = 1$, we obtain

$$\|\boldsymbol{\pi}^{(i)} - \boldsymbol{\pi}^{(i),\star}\|_1 \le |u_i - u_i^\star| + u_i^\star\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^\star\|_1.$$

Taking expectations and applying the preceding bounds gives

$$\mathbb{E}\|\boldsymbol{\pi}^{(i)} - \boldsymbol{\pi}^{(i),\star}\|_1 \le a_i L_\gamma[\Psi_i(K)]^\beta + \frac{u_i^\star\kappa_i\eta_i(K)}{\tau_e}.$$

Isolated nodes have zero pruning budget and contribute nothing.

Finally, let

$$F(a,b) = 1 - (1-a)(1-b).$$

For $a, b, a', b' \in [0,1]$,

$$|F(a,b) - F(a',b')| \le |a - a'| + |b - b'|.$$

Since

$$P_{ij}^{\text{drop}} = F(\pi_{ij}^{(i)}, \pi_{ij}^{(j)}),$$

summing over undirected edges gives

$$\sum_{\{i,j\}\in E} |P_{ij}^{\text{drop}} - P_{ij}^{\text{drop},\star}| \le \sum_i \|\boldsymbol{\pi}^{(i)} - \boldsymbol{\pi}^{(i),\star}\|_1.$$

Taking expectations and summing the node-wise bounds yields

$$\mathbb{E}\sum_{\{i,j\}\in E} |P_{ij}^{\text{drop}} - P_{ij}^{\text{drop},\star}| \le L_\gamma \sum_i a_i[\Psi_i(K)]^\beta + \frac{1}{\tau_e}\sum_i u_i^\star\kappa_i\eta_i(K),$$

which proves equation 19. □

B Algorithm and Implementation Details

Algorithm 1 summarizes the complete test-time adaptation procedure of PUPPY. The frozen source detector first provides the initial prediction reference and normal anchors, from which the target teacher is initialized. During adaptation, representation sensitivity is periodically recomputed to guide structure denoising, while source-referenced prediction drift is updated at each step to refine the target proxy. Only the student adapter is optimized by gradient descent, whereas the teacher is periodically updated from the student through exponential moving average.

All experiments are implemented in PyTorch. The pretrained detector remains frozen during adaptation, and only the lightweight residual adapters are optimized. Each denoised graph view is independently sampled from the original target graph, while the sensitivity statistics are refreshed according to the update schedule in Algorithm 1. Complete task-specific configurations and executable scripts are provided in the anonymous code repository linked in the abstract.

Algorithm 1 Test-Time Adaptation Procedure of PUPPY

Require: Target graph $\mathcal{G}$, frozen source detector f_s, anchor ratio ρ_a, probe number K, bootstrap epochs B, adaptation epochs T, update interval I
Ensure: Node anomaly scores $\{s_i\}_{i\in\mathcal{V}}$
Cache source predictions $\{m_i^s, p_i^s\}$ on $\mathcal{G}$
Construct normal anchor set $\mathcal{A}$ from the lowest source anomaly margins
Initialize teacher adapter θ_T
for $b = 1$ to B **do**
    Sample a perturbed target view and update θ_T with $\mathcal{L}_{\text{boot}}$
end for
Initialize student adapter $\theta_S \leftarrow \theta_T$
for $t = 0$ to $T - 1$ **do**
    if $t \bmod I = 0$ **then**
        Estimate representation sensitivity from K probe views
        Refresh perturbation-aware structure denoising
    end if
    Sample the current denoised target view
    Refine the target proxy using source-referenced student drift
    Update θ_S with the joint adaptation objective $\mathcal{L}$
    if $(t + 1) \bmod I = 0$ **then**
        $\theta_T \leftarrow \mu\theta_T + (1 - \mu)\theta_S$
    end if
end for
Compute final anomaly scores by averaging teacher and student predictions on $\mathcal{G}$
return $\{s_i\}_{i\in\mathcal{V}}$

Table 3: **Statistics of benchmark datasets.**

Dataset	Domain	Nodes	Edges	Features	Anomalies (%)
Amazon	Online product review	11,944	8,847,096	25	821 (6.87%)
YelpChi	Reviewer interaction	45,954	7,739,912	32	6,677 (14.53%)
Reddit	Social media interaction	10,984	168,016	64	366 (3.33%)
Weibo	Social media interaction	8,405	416,368	400	868 (10.33%)
Tolokers	Crowdsourcing collaboration	11,758	530,758	10	2,566 (21.82%)
Questions	Community question answering	48,921	202,461	301	1,460 (2.98%)
T-Finance	Financial transaction network	39,357	42,484,443	10	1,803 (4.58%)
T-Social	Social friendship network	5,781,065	151,992,081	10	174,280 (3.01%)

C Dataset and Evaluation Details

Benchmark Statistics. We evaluate PUPPY on eight public graph anomaly detection benchmarks spanning online reviews, social networks, crowdsourcing platforms, and financial networks. These datasets vary substantially in graph scale, feature dimension, and anomaly ratio. Detailed statistics, including the numbers of nodes, edges, features, and anomalies, are summarized in Table 3.

Controlled Normality Shift Construction. To evaluate adaptation under normality shifts, we construct target distributions in which previously unseen normal patterns emerge at test time. Normal nodes are clustered according to their feature representations, with the largest cluster used as the source-normal distribution for training and validation and the remaining clusters held out as shifted target normals. The resulting source and shifted normal statistics are reported in Table 4.

Temporal Shift Evaluation. Beyond controlled normality shifts, we further evaluate PUPPY under natural temporal evolution on Elliptic and Ethereum. Following chronological ordering, earlier transaction periods are used as source data and later periods as target data, requiring adaptation across naturally evolving graph distributions. The corresponding graph statistics are provided in Table 5.

Table 4: **Statistics of controlled normality shifts.**

Dataset	Source Normals	Shifted Normals	Anomalies	Shifted Ratio
Amazon	8,225	2,898	821	24.26%
YelpChi	10,804	28,473	6,677	61.96%
Reddit	5,082	5,536	366	50.40%
Weibo	7,009	528	868	6.28%
Tolokers	3,557	5,635	2,566	47.92%
Questions	21,064	26,397	1,460	53.96%
T-Finance	34,258	3,296	1,803	8.37%
T-Social	5,583,770	23,015	174,280	0.40%

Table 5: **Statistics of temporal-shift evaluation datasets.**

Dataset	Nodes		Edges		Features
	Source	Target	Source	Target	
Elliptic	136,264	67,504	156,841	77,512	165
Ethereum	2,067,709	905,780	9,554,447	1,162,869	4

Table 6: **Complete AUPRC results on the eight benchmark datasets (%).** Mean±population standard deviation over five runs.

Model	Amazon	YelpChi	Reddit	Weibo	Tolokers	Questions	T-Finance	T-Social
CareGNN	13.19±1.35	18.78±1.00	3.23±0.11	60.14±1.23	33.59±0.34	4.45±0.01	47.75±7.24	5.30±0.66
H2-FDetector	10.83±4.96	18.15±0.37	2.86±0.29	43.57±9.44	29.67±1.31	4.38±0.91	OOM	OOM
GAGA	55.77±25.96	16.96±1.09	2.88±0.21	82.27±2.60	28.36±2.20	4.26±0.63	77.41±1.96	OOT
PMP	19.40±3.46	16.42±1.15	**14.02±2.02**	28.70±2.33	31.39±1.11	7.47±1.89	36.84±10.08	OOM
ConsisGAD	71.85±3.06	16.85±0.15	4.11±0.09	80.35±1.78	28.80±1.16	5.97±0.45	74.26±1.04	51.16±1.14
GCN	3.97±0.06	15.72±0.06	4.54±0.16	77.13±11.69	34.79±0.34	4.81±0.08	37.69±4.30	7.53±2.56
+ GTrans	9.04±6.07	15.82±0.24	2.59±0.35	58.68±5.91	29.17±2.16	2.62±0.09	31.60±28.11	18.99±0.08
+ GraphPatcher	OOM	OOM	4.52±0.07	64.70±18.33	**37.29±0.89**	6.09±0.40	OOM	OOM
+ SOGA	4.04±0.14	16.57±0.36	3.78±0.11	69.96±15.36	35.47±0.99	4.64±0.08	11.31±2.26	11.09±0.01
+ TUNE	10.88±5.65	18.79±0.53	4.74±0.03	84.77±0.02	36.97±0.03	5.10±0.86	36.58±0.15	11.44±0.01
+ Ours	26.97±0.79	19.37±0.02	5.02±0.07	**86.78±0.31**	37.19±0.00	6.08±0.05	49.07±10.46	17.69±0.02
BWGNN	20.03±2.55	17.07±1.03	4.72±0.15	74.78±6.89	30.60±2.66	3.59±0.44	74.62±6.43	25.28±1.95
+ GTrans	13.13±2.14	16.49±0.19	2.89±0.10	74.01±3.13	28.83±2.22	3.01±0.10	34.85±7.39	27.51±0.02
+ GraphPatcher	OOM	OOM	4.46±0.36	51.72±1.46	32.46±0.70	5.19±0.13	OOM	OOM
+ SOGA	13.62±1.53	16.07±0.06	4.57±0.39	44.89±6.85	30.37±1.16	3.16±0.12	20.87±6.51	18.02±0.01
+ TUNE	78.39±4.05	22.81±0.38	4.76±0.18	78.55±0.91	33.65±0.04	8.17±0.12	**82.99±1.02**	28.91±0.03
+ Ours	78.76±1.98	**25.39±1.65**	5.84±0.02	84.57±2.56	33.85±0.19	**9.40±0.05**	76.96±2.21	35.50±0.09
GHRN	17.10±3.99	17.87±0.50	4.00±0.53	76.48±4.21	31.64±0.01	4.22±1.05	76.06±3.90	49.08±5.53
+ GTrans	17.32±8.27	16.93±1.15	2.66±0.33	74.00±6.52	30.55±0.35	3.03±0.09	24.80±15.78	51.67±0.03
+ GraphPatcher	OOM	OOM	3.83±0.75	83.28±1.08	34.09±1.32	5.62±0.22	OOM	OOM
+ SOGA	13.85±1.27	17.68±0.53	3.61±0.29	60.64±7.64	30.73±0.87	3.56±0.58	28.56±9.77	51.84±0.02
+ TUNE	25.79±22.05	21.63±0.28	4.31±0.13	75.48±0.02	34.22±0.21	5.00±0.05	79.18±0.28	54.38±0.02
+ Ours	**83.18±0.13**	21.05±0.02	4.58±0.10	84.56±0.38	34.09±0.02	6.59±0.81	81.16±0.02	**55.60±0.03**

D Complete Experimental Results

Complete AUPRC Results. Table 6 reports the complete AUPRC results on the eight benchmark datasets. The overall pattern is consistent with the AUROC results in the main text: `PUPPY` remains competitive across all three backbones and achieves strong TTA performance across diverse graph domains, showing that its advantage extends beyond a single evaluation metric.

Temporal-Shift Results. Table 7 complements the AUROC results in the main text with AUPRC under natural temporal shifts. `PUPPY` maintains strong performance across both temporal datasets and the three detector backbones, further showing that adaptation can accommodate temporal evolution without systematically sacrificing effective source knowledge.

Complete Ablation Results. Tables 8 and 9 extend the component analysis in the main text to all 24 backbone–dataset settings. The full model achieves strong overall performance across the complete evaluation, while removing individual components degrades performance in a substantial portion of the settings. The results further support the complementary contributions of perturbation-aware structure denoising and adaptive proxy refinement, with target proxy construction, proxy supervision, and drift refinement jointly shaping the latter.

Table 7: **AUPRC results under natural temporal shifts (%).** Mean±population standard deviation over five runs.

Backbone	Dataset	Base	+ GTrans	+ SOGA	+ TUNE	+ Ours
GCN	Elliptic	**28.36±3.22**	27.43±2.69	15.76±0.71	9.80±4.33	27.75±2.15
	Ethereum	1.79±1.08	1.67±1.09	0.17±0.06	1.72±1.05	**1.96±1.27**
BWGNN	Elliptic	28.18±3.30	29.78±3.76	13.13±0.55	5.32±0.66	**31.86±2.91**
	Ethereum	19.25±2.24	13.67±3.28	0.48±0.16	**19.43±2.12**	19.26±2.24
GHRN	Elliptic	28.16±8.13	28.05±7.54	16.03±2.07	5.50±0.89	**29.15±9.46**
	Ethereum	4.98±1.82	5.13±1.74	0.21±0.05	4.24±1.89	**5.24±1.89**

Table 8: **Complete component ablation results in AUROC (%).** Mean±population standard deviation over three seeds.

Variant	Amazon	YelpChi	Reddit	Weibo	Tolokers	Questions	T-Finance	T-Social
GCN								
w/o Target Proxy	78.57±0.30	47.08±0.10	47.68±0.48	95.54±0.04	65.70±0.04	46.46±0.06	75.72±5.64	78.36±0.02
w/o Drift Refinement	80.13±0.92	**56.24±0.03**	65.15±0.02	96.76±0.04	66.39±0.02	48.19±0.02	85.99±1.55	82.04±0.02
w/o Proxy Supervision	25.23±0.33	54.70±0.00	55.87±0.01	96.27±0.02	64.03±0.01	51.45±0.00	85.93±0.89	81.72±0.02
w/o Structure Denoising	79.66±0.00	53.59±0.00	55.88±0.00	86.77±0.00	**67.04±0.00**	46.85±0.00	**87.51±1.27**	75.10±0.00
Full PUPPY	**80.52±0.59**	56.20±0.03	**65.20±0.03**	**96.81±0.03**	**67.04±0.00**	**63.48±0.10**	87.11±0.36	**84.27±0.01**
BWGNN								
w/o Target Proxy	87.46±0.18	54.08±1.35	58.17±0.06	82.36±0.14	61.51±0.31	59.85±0.05	52.41±2.15	79.38±0.00
w/o Drift Refinement	94.38±0.16	62.22±1.83	64.12±0.24	96.95±0.35	65.30±0.06	51.76±0.03	92.91±0.12	81.90±0.02
w/o Proxy Supervision	94.00±0.13	**62.28±1.76**	58.52±0.01	70.06±2.04	65.31±0.07	52.03±0.00	91.07±0.25	81.49±0.01
w/o Structure Denoising	87.24±0.08	53.52±0.00	63.37±0.00	71.72±0.02	64.21±0.00	54.50±0.27	92.64±3.67	82.02±1.15
Full PUPPY	**94.44±0.20**	62.27±1.81	**64.19±0.35**	**97.06±0.14**	**66.89±0.15**	**66.21±0.19**	**95.12±0.06**	**84.79±0.04**
GHRN								
w/o Target Proxy	95.34±0.01	52.40±0.05	64.29±0.01	**95.04±0.24**	61.47±0.02	61.82±0.99	92.36±0.03	86.28±0.01
w/o Drift Refinement	83.01±0.09	**60.74±0.04**	65.67±0.17	94.88±0.20	63.31±0.01	50.15±0.02	93.55±0.00	88.58±0.00
w/o Proxy Supervision	81.87±0.01	49.89±0.05	65.22±0.08	93.55±0.06	63.31±0.01	49.73±0.04	93.56±0.00	88.54±0.00
w/o Structure Denoising	82.04±0.00	52.27±0.01	63.81±0.01	81.59±0.11	63.31±0.00	64.98±0.02	93.88±0.00	88.18±0.00
Full PUPPY	**96.35±0.05**	60.64±0.03	**65.81±0.50**	94.62±0.14	**66.25±0.01**	**65.90±0.09**	**93.92±0.02**	**88.61±0.00**

Table 9: **Complete component ablation results in AUPRC (%).** Mean±population standard deviation over three seeds.

Variant	Amazon	YelpChi	Reddit	Weibo	Tolokers	Questions	T-Finance	T-Social
GCN								
w/o Target Proxy	24.63±0.29	15.19±0.04	2.63±0.03	83.08±0.12	36.47±0.02	2.72±0.01	21.90±12.39	15.94±0.01
w/o Drift Refinement	26.43±1.00	**19.37±0.01**	5.00±0.09	86.71±0.09	36.96±0.01	5.14±0.01	37.26±5.87	11.77±0.03
w/o Proxy Supervision	4.14±0.02	18.81±0.00	4.09±0.00	84.88±0.14	35.80±0.00	4.87±0.00	34.27±5.07	12.59±0.03
w/o Structure Denoising	26.06±0.00	18.52±0.00	4.08±0.00	76.16±0.00	37.18±0.00	2.76±0.00	**43.20±13.18**	8.99±0.00
Full PUPPY	**27.11±0.78**	19.36±0.01	**5.07±0.01**	**86.84±0.11**	**37.19±0.00**	**6.08±0.01**	40.63±2.58	**17.70±0.02**
BWGNN								
w/o Target Proxy	33.97±0.75	17.48±0.60	5.33±0.07	25.54±0.21	31.46±0.18	**4.52±0.01**	4.90±0.42	25.09±0.02
w/o Drift Refinement	78.65±1.25	25.97±1.82	5.87±0.04	**85.21±2.96**	32.71±0.09	3.67±0.00	**80.87±0.70**	32.99±0.04
w/o Proxy Supervision	69.45±2.08	**26.03±1.75**	**5.93±0.00**	25.17±5.91	32.69±0.10	3.66±0.00	71.96±1.16	28.72±0.17
w/o Structure Denoising	25.07±0.22	16.73±0.00	5.04±0.00	52.83±0.02	31.88±0.00	4.09±0.02	74.23±7.99	26.19±2.53
Full PUPPY	**80.16±1.20**	25.99±1.77	5.88±0.01	84.82±1.27	**33.75±0.17**	9.23±0.01	77.72±2.38	**35.54±0.07**
GHRN								
w/o Target Proxy	82.37±0.03	17.38±0.04	4.05±0.00	**85.44±0.28**	30.71±0.01	5.08±0.16	72.62±0.65	51.36±0.02
w/o Drift Refinement	16.87±0.11	**21.15±0.02**	4.15±0.03	84.60±0.45	31.68±0.00	4.14±0.00	80.27±0.01	55.19±0.02
w/o Proxy Supervision	15.97±0.01	16.40±0.04	4.09±0.01	78.79±0.07	31.67±0.00	4.91±0.01	80.22±0.01	54.71±0.00
w/o Structure Denoising	16.23±0.00	17.21±0.01	3.90±0.00	64.79±0.04	31.68±0.01	**7.76±0.01**	80.17±0.01	52.43±0.00
Full PUPPY	**83.22±0.11**	21.06±0.01	**4.56±0.12**	84.53±0.47	**34.09±0.02**	6.96±0.03	**81.16±0.02**	**55.57±0.02**

E Complexity Analysis

Let N and M denote the numbers of nodes and edges in the target graph, respectively, and let $\mathcal{C}_f(N, M)$ denote the computational cost of one forward pass through the underlying detector. The additional computation introduced by PUPPY mainly arises from sensitivity estimation, structure denoising, and proxy refinement.

For sensitivity estimation, K perturbed graph views are evaluated whenever the sensitivity estimates are refreshed. Constructing these views requires node- and edge-wise operations, while

representation sensitivity is computed independently for each node. The resulting cost is $\mathcal{O}(K(\mathcal{C}_f(N, M) + NF + M))$ per refresh, where F is the input feature dimension. Since the sensitivity statistics are updated every I adaptation steps, this overhead is amortized by the refresh interval. Based on the resulting node sensitivities, computing pruning budgets, pairwise scores, and edge-drop probabilities only traverses nodes and observed edges, yielding $\mathcal{O}(N + M)$ complexity.

During each adaptation step, proxy refinement and pseudo-supervision construction operate independently over target nodes and therefore require $\mathcal{O}(N)$ additional computation. Teacher and student predictions require only a constant number of detector evaluations, preserving the asymptotic complexity of the underlying backbone.

Importantly, `PUPPY` does not construct dense node-pair relations or $N \times N$ auxiliary matrices. All structural interactions are restricted to observed edges, while the remaining adaptation statistics are maintained at the node level. Its method-specific storage therefore scales linearly with the target graph, $\mathcal{O}(NF + M + ND)$, where D denotes the representation dimension, in addition to the memory required by the underlying detector. With the small fixed number of probe views used in practice, `PUPPY` preserves the scalability characteristics of the backbone without introducing quadratic dependence on the number of nodes.