KU-GAD: RETHINKING SEMI-SUPERVISED GRAPH ANOMALY DETECTION WITH KNOWN–UNLABELED DISCRIMINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semi-supervised graph anomaly detection (GAD) typically has access to only a small set of verified normal nodes, while anomaly labels remain unavailable. Existing methods infer anomaly information indirectly through normality modeling or surrogate signals, leaving unclear whether the observed unlabeled population can provide direct supervision for anomaly ranking. In this work, we revisit this setting from a population perspective and show that, under a fixed representation and representative normal sampling, the optimal known-unlabeled discrimination score preserves the anomaly-to-normal likelihood-ratio ordering. Based on this insight, we propose KU-GAD, a theory-guided framework for estimating anomaly ranking scores from multi-scale graph representations. Since anomalies may arise from different structural contexts, KU-GAD constructs normal-calibrated multi-hop graph responses to capture anomaly evidence across propagation scales. A hop-sequence Transformer is then employed to adaptively aggregate these responses and produce anomaly scores optimized by the known-unlabeled objective. Extensive experiments on seven graph anomaly detection benchmarks demonstrate the effectiveness of KU-GAD, with further analyses validating the importance of graph context, calibration, and normal-reference quality.

1 INTRODUCTION

Graph anomaly detection (GAD) aims to identify nodes whose attributes, structures, or interactions deviate from normal patterns. In realistic scenarios, however, anomaly labels are often unavailable, while only a limited set of verified normal nodes can be obtained. Because node-level GAD typically assigns continuous anomaly scores to prioritize suspicious nodes, the ordering induced by these scores is central even when no fixed operating threshold is specified. We study this limited-normal semi-supervised setting, where anomaly scores must be learned without access to anomaly labels during training.

Existing semi-supervised GAD methods obtain anomaly-side information through different indirect routes. Normality-based methods such as RHO learn patterns from verified normals and score deviations from them (Ai et al., 2025); generation-based methods such as GGAD and TAQ-GAD construct pseudo anomalies for discriminative training (Qiao et al., 2024; Guo et al., 2026); and GraphNC extracts signals from unlabeled nodes through teacher-mediated normality calibration (Zeng et al., 2026). These approaches infer, synthesize, or mediate the anomaly-side signal. This leaves a more basic question: *can the observed unlabeled population itself provide a direct training signal for anomaly ranking?*

We revisit this setting from a population perspective. For a fixed graph-induced representation $\mathbf{z}$, let $p_N(\mathbf{z})$ and $p_A(\mathbf{z})$ denote the normal and anomalous densities, and let $0 < \pi_A < 1$ denote the anomaly prevalence. Although the identity of each anomalous node in the unlabeled set U is unknown, the observed target population satisfies

$$p_U(\mathbf{z}) = (1 - \pi_A)p_N(\mathbf{z}) + \pi_A p_A(\mathbf{z}),$$

and therefore

$$\frac{p_U(\mathbf{z})}{p_N(\mathbf{z})} = (1 - \pi_A) + \pi_A \frac{p_A(\mathbf{z})}{p_N(\mathbf{z})}.$$

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

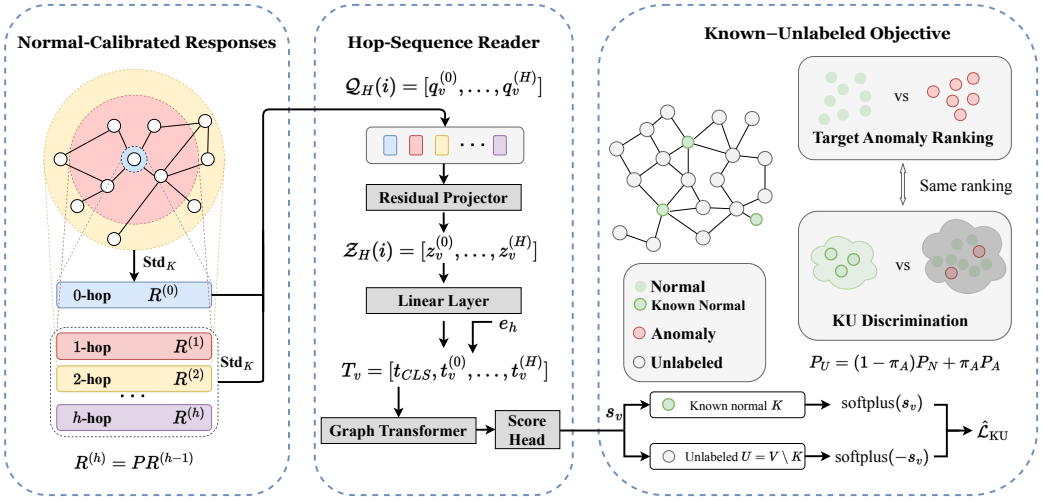

Figure 1: Overview of KU-GAD. The attributed graph is partitioned into known normal nodes K and an unlabeled population U. Although individual labels in U are unknown, the population is an observed mixture of normal and anomalous nodes. KU-GAD constructs normal-calibrated multi-hop graph responses, adaptively aggregates information across propagation scales, and produces anomaly score s_v through direct known–unlabeled discrimination. Under representative known-normal sampling, the corresponding population-optimal score preserves the target anomaly ordering.

Thus, individual anomaly identities may be unavailable while the unlabeled population still carries anomaly-side information through its distributional contrast with normality.

Viewing population discrimination through its density-ratio interpretation (Blanchard et al., 2010; Hido et al., 2008; 2011; Menon & Ong, 2016), we characterize the ranking induced by the known–unlabeled contrast in the limited-normal GAD setting. When the known normals are representative of the target normal population, the population-optimal balanced discriminator satisfies

$$s^*(\mathbf{z}) = \log \frac{p_U(\mathbf{z})}{p_N(\mathbf{z})} = \log \left[(1-\pi_A) + \pi_A \frac{p_A(\mathbf{z})}{p_N(\mathbf{z})} \right].$$

For a fixed representation, this score is strictly monotonic in the anomaly-to-normal likelihood ratio and therefore preserves its ordering, without requiring the identities of anomalous nodes in U or an estimate of π_A. This gives direct known–unlabeled discrimination a population-level anomaly-ranking interpretation.

This population result identifies a supervision signal, but it does not by itself provide a graph representation on which that signal can be effectively learned. In attributed graphs, node attributes alone may be insufficient when anomalous behavior arises from neighborhood interactions or appears at different structural scales. Realizing the population principle in such cases therefore requires representations that expose both node-level and graph-contextual deviations.

To this end, we propose *Known–Unlabeled Graph Anomaly Detection* (KU-GAD), which realizes direct known–unlabeled discrimination over normal-relative multi-scale graph representations. Figure 1 summarizes the resulting framework. Specifically, KU-GAD first constructs normal-calibrated multi-hop graph responses where node attributes are first expressed relative to the verified normal reference, and graph propagation then exposes neighborhood information over multiple depths. Then, KU-GAD employs a hop-sequence Transformer to aggregate information across propagation scales and produce a unified node representation. The Transformer serves as a cross-scale evidence fusion mechanism; the anomaly supervision itself comes from the observed known–unlabeled population contrast. Finally, KU-GAD learns the resulting anomaly score by contrasting the verified normal set K with the complete unlabeled population U.

We evaluate KU-GAD on seven graph anomaly detection benchmarks. Under established semi-supervised protocols, KU-GAD achieves strong overall performance against existing methods. Controlled ablations examine the roles of calibration, graph propagation, multi-scale retention, and

adaptive aggregation, while sensitivity analyses characterize the effects of normal-reference size and representativeness.

Our contributions are summarized as follows:

1. **A population view of anomaly-side information in semi-supervised GAD.** We revisit limited-normal GAD through the observed known–unlabeled population contrast and characterize its ranking implication: under representative normal sampling, the population-optimal discrimination score preserves the anomaly-to-normal likelihood-ratio ordering.

2. **Known–unlabeled graph anomaly detection.** We propose KU-GAD, which realizes this supervision principle using normal-calibrated multi-hop graph responses and adaptive cross-scale aggregation, learning anomaly scores directly from known-normal and unlabeled population membership.

3. **Extensive empirical evaluation.** Experiments on seven graph anomaly detection benchmarks demonstrate strong performance against representative methods. Ablation and sensitivity analyses further examine the proposed graph realization and characterize the effects of graph context and normal-reference quality.

2 Related Work

2.1 Graph Anomaly Detection with Limited Normal Supervision

Graph anomaly detection has been studied through reconstruction-based, one-class, self-supervised, generation-based, and discriminative paradigms. Representative methods include reconstruction-based DOMINANT (Ding et al., 2019) and AnomalyDAE (Fan et al., 2020), one-class methods such as OCGNN (Wang et al., 2021) and TAM (Qiao & Pang, 2023), and contrastive self-supervised methods such as CoLA (Liu et al., 2022) and ANEMONE (Jin et al., 2021). These approaches highlight the roles of representation learning and normality modeling in GAD.

Recent work more directly considers semi-supervised GAD with limited normal supervision. RHO learns homophily-aware filters from available normal labels (Ai et al., 2025); GGAD generates structure-aware pseudo anomalies (Qiao et al., 2024); TAQ-GAD constructs pseudo-anomaly supervision from topology-derived anomaly quantification (Guo et al., 2026); and GraphNC calibrates a pre-trained detector through teacher-guided score alignment and normal-node representation regularization (Zeng et al., 2026). KU-GAD uses a different training signal by directly contrasting membership in the observed known-normal and unlabeled populations.

2.2 Learning from Normal and Unlabeled Populations

Semi-supervised novelty detection studies a nominal sample together with a possibly contaminated unlabeled population and reduces novelty detection to discrimination between the two observed populations (Blanchard et al., 2010). Related inlier-based outlier detection methods directly estimate density ratios between inlier and contaminated test distributions (Hido et al., 2008; 2011), while probabilistic classification provides a general connection between binary discrimination and density-ratio estimation (Menon & Ong, 2016). Deep SVDD (Ruff et al., 2018) and Deep SAD (Ruff et al., 2020) provide related one-class and semi-supervised anomaly objectives, while DDAD exploits known-normal and unlabeled populations through distributional discrepancy modeling (Cai et al., 2022).

Positive–unlabeled (PU) learning instead seeks to recover a latent positive–negative classifier from labeled positive and unlabeled data (Elkan & Noto, 2008; Kiryo et al., 2017). Graph PU methods such as GPL (Wu et al., 2024) and BLD (Liang et al., 2025) extend this setting to graph data under class-prior, heterophily, and label-disambiguation challenges. KU-GAD differs in that it uses the two observed populations themselves as the training contrast and targets the anomaly ordering induced by their density ratio.

3 PROBLEM FORMULATION

We consider an attributed graph

$$G = (V, E, \mathbf{X}), \qquad \mathbf{A} \in \mathbb{R}^{N \times N}, \qquad \mathbf{X} \in \mathbb{R}^{N \times F},$$

where V is the node set, E is the edge set, $\mathbf{A}$ is the adjacency matrix, $\mathbf{X}$ contains node attributes, $N = |V|$, and F is the input feature dimension. Let $K \subset V$ denote the set of known normal nodes and let

$$U = V \setminus K$$

denote the unlabeled target population. The identities of anomalous nodes in U are unavailable during training.

The model may use graph structure, node attributes, and K/U membership, but it does not access outcome labels in U during representation construction, optimization, configuration selection, or score generation. Ground-truth anomaly labels for nodes in U are used only after model scores are frozen for evaluation.

Our goal is to learn a scoring function

$$s : V \to \mathbb{R},$$

where larger values indicate greater anomalousness. Evaluation concerns the ranking induced by these scores over nodes in U. A downstream application may threshold the scores according to its inspection budget or error costs; no fixed operating threshold is assumed in the benchmark setting considered here. We train a single model on the fixed target graph and score nodes in U, yielding a transductive node-ranking setting.

4 KU-GAD

4.1 OVERVIEW

KU-GAD consists of three stages: constructing normal-relative multi-hop representations, aggregating multi-scale anomaly-relevant evidence, and optimizing a known–unlabeled discrimination objective whose population optimum preserves the target anomaly ranking under representative normal sampling. Specifically, the model calibrates node features using statistics from the verified normal set, propagates the calibrated responses across multiple graph hops, adaptively aggregates the resulting hop sequence, and produces a scalar score trained by direct discrimination between K and U.

Figure 1 summarizes the pipeline. Section 4.2 constructs the graph representations, Section 4.3 describes the multi-hop evidence encoder, Section 4.4 defines the training objective, and Section 4.5 formalizes its population ranking interpretation.

4.2 NORMAL-CALIBRATED MULTI-HOP ANOMALY REPRESENTATIONS

To facilitate anomaly ranking under the KU objective, we first express node features in a coordinate system defined by the verified normal set. For any response matrix $\mathbf{M}$, we define

$$\mathrm{Std}_K(\mathbf{M}) = \frac{\mathbf{M} - \boldsymbol{\mu}_K(\mathbf{M})}{\max(\boldsymbol{\sigma}_K(\mathbf{M}), \varepsilon)}, \tag{1}$$

where $\boldsymbol{\mu}_K(\mathbf{M})$ and $\boldsymbol{\sigma}_K(\mathbf{M})$ are the feature-wise mean and standard deviation over rows indexed by K, with the maximum applied element-wise. The input response is

$$\mathbf{R}^{(0)} = \mathrm{Std}_K(\mathbf{X}).$$

Multi-hop graph propagation. Since anomalies may arise from neighborhood interactions, we propagate the calibrated features across multiple graph hops to capture information at different structural scales. With self-loops, let $\mathbf{P} = \widetilde{\mathbf{D}}^{-1/2}(\mathbf{A} + \mathbf{I})\widetilde{\mathbf{D}}^{-1/2}$ denote the symmetrically normalized propagation matrix. The undirected benchmarks use this form, while DGraph follows its directed normalization described in Appendix B.3. Multi-hop responses are constructed as

$$\mathbf{R}^{(h)} = \mathbf{P}\mathbf{R}^{(h-1)}, \qquad h = 1, \ldots, H. \tag{2}$$

Hop-wise recalibration. Graph propagation generally changes the location and scale of the known-normal responses, so each propagated representation is recalibrated against the same normal reference:

$$\mathbf{q}_v^{(0)} = \mathbf{R}_v^{(0)}, \qquad \mathbf{q}_v^{(h)} = \mathrm{Std}_K(\mathbf{R}^{(h)})_v, \quad h = 1, \ldots, H. \tag{3}$$

Input calibration defines normal-relative coordinates, while hop-wise recalibration realigns these coordinates after graph propagation.

The resulting response sequence is

$$\mathcal{Q}_H(v) = [\mathbf{q}_v^{(0)}, \mathbf{q}_v^{(1)}, \ldots, \mathbf{q}_v^{(H)}].$$

Because the anomaly-relevant structural scale is generally unknown, we retain responses from all propagation depths rather than collapsing them into a single terminal representation.

4.3 Multi-Hop Anomaly Evidence Encoder

The retained multi-hop responses can be viewed as a sequence of anomaly-relevant evidence extracted from different graph contexts, requiring an adaptive mechanism to determine their relative importance. We instantiate this cross-scale aggregation with a tokenized graph-Transformer design, following the general sequence pattern used by models such as NAGphormer (Chen et al., 2023).

Each hop response is independently projected before aggregation, allowing scale-specific information to be retained in a shared token space. We use a residual projector

$$\mathbf{z}_v^{(h)} = \mathbf{W}_h^{\mathrm{skip}} \mathbf{q}_v^{(h)} + \mathbf{W}_{h,2}\, \mathrm{GELU}\Big(\mathbf{W}_{h,1}\mathbf{q}_v^{(h)} + \mathbf{b}_{h,1}\Big) + \mathbf{b}_{h,2}.$$

The projected responses are then mapped to the Transformer token space and augmented with learnable hop embeddings:

$$\mathbf{t}_v^{(h)} = \mathbf{A}_d \mathbf{z}_v^{(h)} + \mathbf{e}_h.$$

For node v, the encoder receives

$$\mathbf{T}_v = [\mathbf{t}_{\mathrm{CLS}}, \mathbf{t}_v^{(0)}, \ldots, \mathbf{t}_v^{(H)}],$$

where $\mathbf{t}_{\mathrm{CLS}}$ is a learnable classification (CLS) token. We adopt a pre-norm Transformer encoder to model dependencies among different propagation scales and generate a unified node representation from multi-hop evidence. Its CLS representation is mapped to a scalar score:

$$s_v = \mathbf{w}^\top \mathrm{LN}(\mathrm{GT}(\mathbf{T}_v)_{\mathrm{CLS}}) + b, \tag{4}$$

where LN denotes layer normalization. Higher values of s_v indicate greater anomalousness. The Transformer therefore serves as an adaptive cross-scale evidence fusion mechanism; the anomaly supervision itself is provided by the known–unlabeled objective below.

4.4 Known–Unlabeled Discrimination Objective

Given the multi-scale graph representation, we optimize the model using known–unlabeled discrimination, where the learned score is encouraged to distinguish the verified normal population from the unlabeled target population. Specifically, we minimize

$$\widehat{\mathcal{L}}_{\mathrm{KU}} = \frac{1}{2|K|} \sum_{v \in K} \mathrm{softplus}(s_v) + \frac{1}{2|U|} \sum_{v \in U} \mathrm{softplus}(-s_v), \tag{5}$$

where $\mathrm{softplus}(x) = \log(1 + e^x)$. This is the balanced binary logistic loss for discriminating known normal nodes from the unlabeled target population. The two populations receive equal total weight despite the larger size of U, and membership in U denotes target-population membership rather than a node-level anomaly label.

According to the population analysis in Section 4.5, for a fixed representation and representative normal sampling, the population-optimal solution of this objective preserves the ordering induced by the anomaly-to-normal density ratio. This provides a theoretical ranking interpretation for the learned known–unlabeled discrimination score.

4.5 Population Ranking Analysis

We now formalize the ranking interpretation of the known–unlabeled objective introduced above. Let $\mathbf{Z} = \Phi(G, \mathbf{X}, v) \in \mathcal{Z}$ denote a fixed induced representation. Let P_N, P_A, and P_U denote the normal, anomalous, and unlabeled populations in this representation space, with densities satisfying

$$p_U(\mathbf{z}) = (1 - \pi_A)p_N(\mathbf{z}) + \pi_A p_A(\mathbf{z}), \qquad 0 < \pi_A < 1. \tag{6}$$

When the known normal nodes are representative of the target normal population P_N, the corresponding balanced population objective is

$$\mathcal{L}_{\mathrm{KU}}(s) = \frac{1}{2}\mathbb{E}_{\mathbf{Z}\sim P_N}[\mathrm{softplus}(s(\mathbf{Z}))] + \frac{1}{2}\mathbb{E}_{\mathbf{Z}\sim P_U}[\mathrm{softplus}(-s(\mathbf{Z}))]\,. \tag{7}$$

For an unrestricted score function on the common support, pointwise minimization of Equation 7 gives

$$s^*(\mathbf{z}) = \log \frac{p_U(\mathbf{z})}{p_N(\mathbf{z})}. \tag{8}$$

Balanced weighting makes this form exact; any fixed positive reweighting of the two populations changes only the additive intercept and therefore leaves the induced ranking unchanged.

Proposition 1 (Population Ranking Equivalence)**.** *For a fixed induced representation, suppose the known normal nodes are representative of the target normal population P_N. Then the population minimizer satisfies*

$$s^*(\mathbf{z}) = \log\left[(1 - \pi_A) + \pi_A \frac{p_A(\mathbf{z})}{p_N(\mathbf{z})}\right]$$

almost everywhere on the common support. Consequently, for any two points $\mathbf{z}$ and $\mathbf{z}'$ in that support,

$$s^*(\mathbf{z}) > s^*(\mathbf{z}') \quad \Longleftrightarrow \quad \frac{p_A(\mathbf{z})}{p_N(\mathbf{z})} > \frac{p_A(\mathbf{z}')}{p_N(\mathbf{z}')}.$$

Proposition 1 shows that the observed K/U population contrast is sufficient to identify the target anomaly ordering at the population level under representative normal sampling. It does not require identifying individual anomalies in U or estimating π_A. This density-ratio relation is supported by classical analyses of semi-supervised novelty detection and inlier-based outlier detection (Blanchard et al., 2010; Hido et al., 2008; 2011).

What the observed supervision identifies. The result also provides a common view of alternative supervision sources. A score determined only by the normal distribution cannot, in general, determine the anomaly likelihood-ratio ordering for every possible P_A: the same P_N can be paired with different anomaly distributions that induce conflicting rankings. A synthesized anomaly distribution Q_A supplies additional anomaly-side information, but its induced ordering depends on how q_A/p_N relates to the target ratio p_A/p_N. By contrast, the observed target population satisfies the mixture relation above, which directly aligns the P_U/P_N ordering with the target anomaly ordering when the normal reference is representative. Formal constructions for the normal-only and proxy cases are provided in Appendix A.3.

This is a population statement for a fixed representation. Finite normal data, reference bias, graph-dependent representation learning, and optimization can affect how closely the implemented model follows this ordering; the experiments and Appendix A examine these finite-model effects.

5 Experiments

5.1 Experimental Setup

Datasets and baselines. We evaluate KU-GAD on seven node-level graph anomaly detection benchmarks: Amazon (Dou et al., 2020), Photo (McAuley et al., 2015), Elliptic (Weber et al., 2019), T-Finance (Tang et al., 2022), Tolokers and Questions (Platonov et al., 2023), and DGraph (Huang et al., 2022). We compare against representative GAD baselines reported by RHO (Ai et al., 2025) and include GraphNC (Zeng et al., 2026) on overlapping datasets. Dataset statistics, preprocessing, and protocol boundaries are given in Appendix B.

AUROC

Method	Amazon	Photo	Elliptic	T-Finance	Tolokers	Questions	DGraph
DOMINANT	0.8867	0.5314	0.3256	0.6167	0.5121	0.5454	0.5851
AnomalyDAE	0.9171	0.5272	0.5409	0.6027	0.6074	0.5347	0.5866
TAM	0.8405	0.6013	0.4150	0.5923	0.4847	0.5222	–
AEGIS	0.7593	0.5936	0.5132	0.6728	0.4451	0.5344	0.4450
GAAN	0.6531	0.4355	0.2724	0.3636	0.3578	0.4840	–
GGAD	0.9443	0.6476	0.7290	0.8228	0.5340	0.5122	0.5943
OCGNN	0.8810	0.6461	0.2881	0.5742	0.4803	0.5578	–
GCN	0.7345	0.5727	0.6463	0.7262	0.4954	0.4556	0.5112
BWGNN	0.8312	0.6861	0.7241	0.7683	0.5821	0.5740	0.4958
PMP	0.8329	0.5844	0.5617	0.8321	0.5815	0.5790	0.5376
CONSISGAD	0.8715	0.5859	0.7354	0.8277	0.5974	0.5737	0.5735
RHO	0.9302	0.7129	0.8509	0.8623	0.6255	0.5833	0.6033
GraphNC	**0.9613**	0.7693	–	0.8340	0.6505	–	–
KU-GAD	0.9046	**0.9492**	**0.9068**	**0.8901**	**0.7345**	**0.5843**	**0.7477**

AUPRC

Method	Amazon	Photo	Elliptic	T-Finance	Tolokers	Questions	DGraph
DOMINANT	0.7289	0.1238	0.0652	0.0542	0.2217	0.0314	0.0076
AnomalyDAE	0.7748	0.1177	0.0949	0.0538	0.2697	0.0317	0.0071
TAM	0.5183	0.1087	0.0552	0.0551	0.2178	0.0391	–
AEGIS	0.2616	0.1110	0.0912	0.0685	0.1943	0.0313	0.0058
GAAN	0.0856	0.0768	0.0611	0.0324	0.1693	0.0359	–
GGAD	0.7922	0.1442	0.2425	0.1825	0.2502	0.0349	0.0082
OCGNN	0.7538	0.1501	0.0640	0.0492	0.2138	0.0354	–
GCN	0.1617	0.1257	0.1176	0.1387	0.2351	0.0405	0.0040
BWGNN	0.3797	0.2202	0.1869	0.1436	0.2687	0.0410	0.0041
PMP	0.3582	0.1254	0.1006	0.4084	0.2939	0.0383	0.0046
CONSISGAD	0.6523	0.1319	0.1765	0.4152	0.3059	0.0424	0.0050
RHO	0.7879	0.2337	0.5095	0.4893	0.3256	0.0430	0.0059
GraphNC	**0.8403**	0.3561	–	0.3667	0.3082	–	–
KU-GAD	0.7393	**0.8059**	**0.7691**	**0.7100**	**0.4354**	**0.0545**	**0.0341**

Table 1: AUROC and AUPRC on seven graph anomaly detection benchmarks. Baseline results are taken from RHO (Ai et al., 2025) under its reported protocols, with GraphNC results taken from its original paper (Zeng et al., 2026). Best results are boldfaced and second-best results are underlined; "–" denotes an unavailable result. Metric implementation details are provided in Appendix B.2.

Evaluation protocol. We report AUROC and AUPRC under established semi-supervised GAD protocols; metric details are in Appendix B.2. Following GGAD (Qiao et al., 2024) and RHO (Ai et al., 2025), we construct K using the 15% candidate-budget protocol. For Amazon, Photo, Elliptic, T-Finance, Tolokers, and Questions, the complete $U = V \setminus K$ is used for training and scoring; DGraph follows Appendix B.3. Ground-truth anomaly labels for nodes in U are used only after model scores are frozen.

Configuration and implementation. Training and configuration selection do not access anomaly outcomes. Propagation depth and optimizer checkpoint step are selected using held-out K/U membership; details are in Appendix D. Unless otherwise specified, KU-GAD uses $d = 32$, AdamW with weight decay 10^{-4}, learning rate 3×10^{-4}, batch size 4096, and $\varepsilon = 10^{-6}$. Further implementation and reference-budget details are provided in the appendix.

5.2 Overall Performance

Table 1 compares KU-GAD with existing graph anomaly detection methods on the adopted benchmark suite. KU-GAD outperforms the compared methods on five of the six primary datasets in both

	Amazon		Photo		T-Finance		Tolokers	
Variant	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**
Full KU-GAD	<u>0.905</u>	<u>0.739</u>	**0.949**	**0.806**	**0.890**	**0.710**	**0.735**	**0.435**
No calibration	0.848	0.651	0.722	0.232	0.772	0.188	<u>0.723</u>	0.419
No propagation ($H = 0$)	**0.933**	**0.835**	0.672	0.189	0.524	0.051	0.702	0.381
Permuted graph	0.855	0.515	0.590	0.139	0.521	0.056	0.638	0.327
Terminal hop only	0.757	0.227	0.899	0.666	0.877	<u>0.675</u>	0.683	0.392
Simple aggregation	0.855	0.430	<u>0.923</u>	<u>0.705</u>	<u>0.881</u>	0.515	0.720	<u>0.422</u>

Table 2: Controlled ablation of KU-GAD on four shared-protocol datasets. Values are five-seed means; best results are boldfaced and second-best results are underlined. Complete results are provided in Appendix C.

AUROC and AUPRC. The improvements are particularly pronounced on Photo, Elliptic, T-Finance, and Tolokers, especially in AUPRC, whereas Questions shows a more modest gain. Amazon remains the exception, where GraphNC achieves the strongest reported result. KU-GAD also achieves strong performance on DGraph. Because DGraph follows a distinct published evaluation protocol, we report its protocol details separately in Appendix B.3.

5.3 Ablation Study

We next evaluate the major design choices in KU-GAD. We ablate normal calibration, graph propagation, retention of multiple propagation scales, and adaptive aggregation, and include a permuted-graph control to distinguish useful graph propagation from arbitrary propagation. For each dataset, all variants use the frozen KU-GAD split, optimizer profile, and checkpoint step; the selected propagation depth is kept fixed except for the explicit $H = 0$ intervention. No anomaly labels are used to re-select configurations for individual variants.

Variants. *No calibration* removes both input calibration and hop-wise recalibration. *No propagation* sets $H = 0$ and uses only the input response. *Permuted graph* uses $\mathbf{A}_{\mathrm{perm}} = \mathbf{\Pi}\mathbf{A}\mathbf{\Pi}^{\top}$, where $\mathbf{\Pi}$ is a permutation matrix, while leaving node attributes, K/U membership, and evaluation identities unchanged; the graph remains isomorphic to the original while nodes are assigned to different graph positions. *Terminal hop only* retains only the response at the selected maximum propagation depth. *Simple aggregation* replaces adaptive aggregation over hop responses with their mean. Complete implementation details are provided in Appendix C.

Table 2 validates the effectiveness of the overall KU-GAD design. The full model achieves the best performance in six of the eight comparisons and ranks second in the remaining two. Removing calibration causes substantial degradation on Amazon, Photo, and T-Finance, while retaining only the terminal hop or replacing adaptive aggregation with a simple mean generally reduces performance. These results support the complementary roles of normal calibration, multi-hop responses, and adaptive aggregation in constructing effective representations for the known–unlabeled objective.

Graph propagation provides particularly large gains on Photo and T-Finance and also improves Tolokers. Amazon is the notable exception, where the H=0 variant performs best, consistent with a feature-dominant regime in which neighborhood propagation provides limited marginal benefit.

The permuted-graph control further verifies that the gains from propagation are tied to the observed graph rather than to propagation itself. On Photo, T-Finance, and Tolokers, replacing the original graph assignment with an isomorphic permutation substantially reduces performance, with the permuted variant performing at or below $H = 0$. Overall, these results validate the proposed representation design and show that, when propagation is beneficial, its gains are not reproduced by arbitrary graph propagation.

5.4 Sensitivity to Reference Bias

The population analysis assumes that the known normal set is representative of target normality. We test the practical importance of this condition with a controlled selection-bias intervention. For each dataset, we construct a normal-only score r_v from the first principal component (PC1) of the eligible

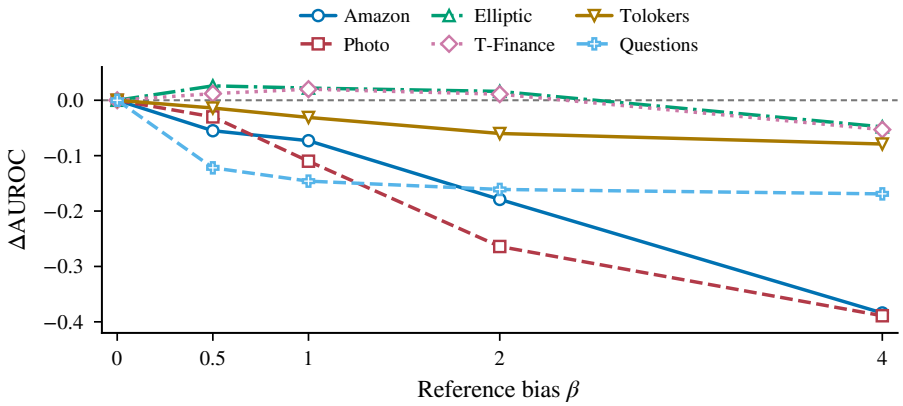

Figure 2: Sensitivity to reference-selection bias. Curves show the change in five-seed mean AUROC relative to the unbiased $\beta = 0$ setting. Larger β increasingly concentrates the known-normal reference.

normal candidates and sample the known normal set with weights

$$w_v(\beta) \propto \exp(\beta r_v), \qquad \beta \in \{0, 0.5, 1, 2, 4\}.$$

Here $\beta = 0$ is the unbiased baseline, while larger β increasingly concentrates the known-normal sample. All bias levels share the same frozen configuration and evaluation protocol. Labels are used only to construct the controlled intervention and fixed evaluation population; the model receives only the selected set K.

Figure 2 shows that severe reference bias materially degrades anomaly ranking. Amazon and Photo are particularly sensitive, while Elliptic and T-Finance tolerate mild bias before degrading under stronger selection shifts. Complete AUROC/AUPRC statistics and matched-seed analysis are reported in Appendix C.3.

These results empirically support representative normal sampling as an important condition for the finite KU-GAD model, consistent with the reference-shift analysis in Appendix A.

6 Conclusion

We studied semi-supervised GAD from the observation that the absence of anomaly labels does not imply the absence of anomaly-side information. Although individual nodes in U are unlabeled, the observed unlabeled population is anomaly-bearing. Under representative normal sampling, direct known–unlabeled discrimination therefore induces the target anomaly ordering at the population level.

Based on this result, KU-GAD combines normal-relative calibration, multi-hop graph responses, and a hop-sequence reader with the known–unlabeled objective. Experiments on seven graph anomaly detection benchmarks demonstrate strong performance under established semi-supervised protocols, while ablation and sensitivity studies validate the proposed design and clarify the role of the normal reference. The current study focuses on transductive node anomaly detection; extending KU-GAD toward inductive settings and stronger finite-sample guarantees remains an important direction for future work.

AI use statement

Generative AI tools were used to assist literature retrieval, research brainstorming and experimental design, refinement and presentation of mathematical arguments, implementation and research-workflow tasks, and drafting and polishing manuscript text. All AI-assisted mathematical claims, experimental designs, implementations, citations, and reported results were reviewed and verified by the authors, who take responsibility for the final content of the submission.

Ethics statement

This study uses established public graph benchmarks and does not collect new personal data, contact individuals, or make decisions about real people. Under the canonical benchmark protocol, benchmark normal labels are used to construct the verified-normal set K; training and per-run configuration then use K/U membership without anomaly outcomes in U, which are introduced only after score freezing for evaluation. Controlled and oracle diagnostics explicitly use labels to construct interventions or fixed evaluation populations and are not part of the canonical training procedure. The results should therefore be interpreted as methodological benchmark evidence rather than deployment guidance for high-stakes decisions.

Reproducibility statement

The appendix records the dataset profiles, split construction, metric implementation, DGraph protocol boundary, label-free configuration procedure, and sensitivity analyses used in this study. Public datasets are identified by their original citations, and all reported KU-GAD results use the score direction and evaluation populations described in the manuscript. Code and experiment artifacts will accompany the submission where permitted by the review process.

References

Guoguo Ai, Hezhe Qiao, Hui Yan, and Guansong Pang. Semi-supervised graph anomaly detection via robust homophily learning. In *Advances in Neural Information Processing Systems*, 2025.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(99):2973–3009, 2010.

Yu Cai, Hao Chen, Xin Yang, Yu Zhou, and Kwang-Ting Cheng. Dual-distribution discrepancy for anomaly detection in chest x-rays. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, 2022.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8KYeilT3Ow.

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, 2019. doi: 10.1137/1.9781611975673.67.

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 and Knowledge Management*, pp. 315–324, 2020.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–220, 2008. doi: 10.1145/1401890.1401920.

Haoyi Fan, Fengbin Zhang, and Zuoyong Li. AnomalyDAE: Dual autoencoder for anomaly detection on attributed networks. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5685–5689, 2020. doi: 10.1109/ICASSP40776.2020.9053387.

Ting Guo, Yangrui Fan, Caixia Cui, Jiye Liang, Jiao Zhao, and Da Wang. Topological anomaly quantification for semi-supervised graph anomaly detection. In *International Conference on Learning Representations*, 2026.

Shohei Hido, Yuta Tsuboi, Hisashi Kashima, Masashi Sugiyama, and Takafumi Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 223–232. IEEE, 2008. doi: 10.1109/ICDM.2008.49.

Shohei Hido, Yuta Tsuboi, Hisashi Kashima, Masashi Sugiyama, and Takafumi Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26(2):309–336, 2011. doi: 10.1007/s10115-010-0283-2.

Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. DGraph: A large-scale financial dataset for graph anomaly detection. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22765–22777, 2022.

Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. ANEMONE: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3122–3126, 2021. doi: 10.1145/3459637.3482057.

Ryuichi Kiryo, Gang Niu, Marthinus C. du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1675–1685, 2017.

Chunquan Liang, Luyue Wang, Xinyuan Feng, Yuying Cheng, Mei Li, Shirui Pan, and Hongming Zhang. Graph positive-unlabeled learning via bootstrapping label disambiguation. *Neural Networks*, 190:107630, 2025. doi: 10.1016/j.neunet.2025.107630.

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, 2022. doi: 10.1109/TNNLS.2021.3068344.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, 2015.

Aditya Krishna Menon and Cheng Soon Ong. Linking losses for density ratio and class-probability estimation. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 304–313. PMLR, 2016. URL https://proceedings.mlr.press/v48/menon16.html.

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.

Hezhe Qiao and Guansong Pang. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hezhe Qiao, Qingsong Wen, Xiaoli Li, Ee-Peng Lim, and Guansong Pang. Generative semi-supervised graph anomaly detection. In *Advances in Neural Information Processing Systems*, 2024.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 2018. URL https://proceedings.mlr.press/v80/ruff18a.html.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkgH0TEYwH.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21076–21089, 2022.

Xuhong Wang, Baihong Jin, Ying Du, Ping Cui, Yingshui Tan, and Yupu Yang. One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications*, 33(18):12073–12085, 2021. doi: 10.1007/s00521-021-05924-9.

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. In *KDD '19 Workshop on Anomaly Detection in Finance*, 2019.

Yuhao Wu, Jiangchao Yao, Bo Han, Lina Yao, and Tongliang Liu. Unraveling the impact of heterophilic structures on graph positive-unlabeled learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53928–53943. PMLR, 2024.

Guolei Zeng, Hezhe Qiao, Guoguo Ai, Jinsong Guo, and Guansong Pang. Normality calibration in semi-supervised graph anomaly detection. In *International Conference on Machine Learning*, 2026.

A Population Analysis of Known–Unlabeled Discrimination

A.1 Scope and Distributional Assumptions

We condition on a fixed attributed graph and view each population distribution as induced by drawing a node index from the corresponding node population and then applying a fixed representation $\mathbf{Z} = \Phi(G, \mathbf{X}, v)$. This construction does not require graph nodes to be independent. Let P_N, P_A, and P_U be probability measures on $(\mathcal{Z}, \mathcal{F})$ dominated by a common measure μ, with densities p_N, p_A, and p_U. We analyze the common support

$$\Omega = \{\mathbf{z} : p_N(\mathbf{z}) > 0,\ p_U(\mathbf{z}) > 0\}.$$

The target mixture is

$$P_U = (1 - \pi_A)P_N + \pi_A P_A, \qquad 0 < \pi_A < 1.$$

The population result assumes that the finite known normal set K is representative of P_N. This population assumption is distinct from the finite sampling process used to construct K in the experiments.

A.2 Proof of Population Ranking Equivalence

For a fixed $\mathbf{z} \in \Omega$, the integrand of Equation 7, up to the common measure μ, is

$$\ell_{\mathbf{z}}(t) = \frac{p_N(\mathbf{z})}{2} \log(1 + e^t) + \frac{p_U(\mathbf{z})}{2} \log(1 + e^{-t}).$$

Its first and second derivatives are

$$\ell'_{\mathbf{z}}(t) = \frac{p_N(\mathbf{z})e^t - p_U(\mathbf{z})}{2(1 + e^t)}, \qquad \ell''_{\mathbf{z}}(t) = \frac{[p_N(\mathbf{z}) + p_U(\mathbf{z})]e^t}{2(1 + e^t)^2} > 0.$$

The integrand is therefore strictly convex, and its unique minimizer satisfies $e^t = p_U(\mathbf{z})/p_N(\mathbf{z})$. Pointwise minimization yields

$$s^*(\mathbf{z}) = \log \frac{p_U(\mathbf{z})}{p_N(\mathbf{z})}.$$

For class weights $w_N, w_U > 0$, the same calculation adds the intercept $\log(w_U/w_N)$ and leaves pairwise orderings unchanged.

Substituting the target mixture gives

$$s^*(\mathbf{z}) = g_{\pi_A}\left(\frac{p_A(\mathbf{z})}{p_N(\mathbf{z})}\right), \qquad g_{\pi_A}(t) = \log[(1 - \pi_A) + \pi_A t].$$

Because

$$g'_{\pi_A}(t) = \frac{\pi_A}{(1 - \pi_A) + \pi_A t} > 0,$$

for any $\mathbf{z}, \mathbf{z}' \in \Omega$,

$$s^*(\mathbf{z}) > s^*(\mathbf{z}') \quad \Longleftrightarrow \quad \frac{p_A(\mathbf{z})}{p_N(\mathbf{z})} > \frac{p_A(\mathbf{z}')}{p_N(\mathbf{z}')}.$$

Thus the two scores have the same strict ordering and the same ties. The ordering does not require estimating π_A, although small anomaly prevalence can compress finite score differences. This is a population statement in the fixed representation space, not a calibrated estimate of the anomaly posterior. □

A.3 Identifiability under Alternative Supervision

Normality underidentification. Consider a representation space with two points $\mathbf{z}_1, \mathbf{z}_2$ and a fixed normal distribution with masses $n_1, n_2 > 0$. Any scoring rule determined solely by P_N returns two fixed values t_1, t_2. If $t_1 \geq t_2$, choose a constant $0 < c < 1$ and define an anomaly distribution through

$$\frac{p_A(\mathbf{z}_1)}{p_N(\mathbf{z}_1)} = c, \qquad \frac{p_A(\mathbf{z}_2)}{p_N(\mathbf{z}_2)} = \frac{1 - n_1 c}{n_2}.$$

The second ratio is greater than one and therefore greater than c, while $n_1 c + n_2(1 - n_1 c)/n_2 = 1$ ensures that P_A is a valid distribution. The desired anomaly ordering ranks $\mathbf{z}_2$ above $\mathbf{z}_1$, contradicting the normal-only order. If $t_1 < t_2$, the symmetric construction reverses the contradiction. Hence no score determined by P_N alone recovers the anomaly likelihood-ratio ordering for every possible P_A. Extra assumptions or side information can make a particular normality objective effective, but that information is not identified by P_N itself. □

Proxy ranking condition. Let Q_A be a synthesized anomaly distribution with density q_A. Balanced discrimination between P_N and Q_A yields

$$s_Q^*(\mathbf{z}) = \log \frac{q_A(\mathbf{z})}{p_N(\mathbf{z})}.$$

This objective preserves the strict target anomaly ordering exactly when, for all evaluated $\mathbf{z}, \mathbf{z}'$, it satisfies

$$\frac{p_A(\mathbf{z})}{p_N(\mathbf{z})} > \frac{p_A(\mathbf{z}')}{p_N(\mathbf{z}')} \quad \Longleftrightarrow \quad \frac{q_A(\mathbf{z})}{p_N(\mathbf{z})} > \frac{q_A(\mathbf{z}')}{p_N(\mathbf{z}')}.$$

Equivalently, on the realized likelihood-ratio range, q_A/p_N must be an order-preserving transform of p_A/p_N. A pseudo-anomaly mechanism may encode useful structural priors or improve finite-sample optimization, but proxy discrimination alone does not establish this ordering condition.

A.4 Reference Shift and the Finite Implementation

When the known normal set is selected from a biased population, let P_K denote the population represented by that sampling mechanism, with density p_K. The population discriminator then targets

$$\log \frac{p_U(\mathbf{z})}{p_K(\mathbf{z})} = \log \frac{p_U(\mathbf{z})}{p_N(\mathbf{z})} + \log \frac{p_N(\mathbf{z})}{p_K(\mathbf{z})}.$$

The first term is a strictly increasing transform of p_A/p_N, while the second can alter pairwise orderings whenever the known normal sampling mechanism is not representative of target normality. This decomposition identifies reference shift as a practical failure mode for the finite model.

The implemented model introduces additional finite effects. It uses empirical calibration statistics estimated from the same finite K, learns graph-dependent representations, and minimizes an empirical objective within a restricted neural function class using nonconvex optimization. Consequently, the population derivation does not by itself prove convergence of the end-to-end score under finite K, graph dependence, representation error, regularization, or optimization error. The sensitivity analyses below characterize parts of this empirical gap without turning the population result into a consistency theorem.

A.5 Effect of Graph Propagation on Known-Normal Calibration

The first calibration stage places the input response in a coordinate system whose feature-wise empirical mean over K is zero and whose nondegenerate feature-wise empirical variance is one. This property is not generally invariant to graph propagation. Define

$$m_K(\mathbf{M}) = \frac{1}{|K|} \mathbf{1}_K^\top \mathbf{M}.$$

Input calibration gives $m_K(\mathbf{R}^{(0)}) = \mathbf{0}$, whereas after h propagation steps,

$$m_K(\mathbf{R}^{(h)}) = \frac{1}{|K|} \mathbf{1}_K^\top \mathbf{P}^h \mathbf{R}^{(0)}.$$

There is no general reason for this quantity to remain zero because $\mathbf{P}^h$ mixes responses according to node-dependent neighborhoods; equivalently, the left action of $\mathbf{1}_K^\top \mathbf{P}^h$ need not preserve the subspace centered over K. The feature-wise covariance is likewise altered by this mixing, so unit scale on K is not generally preserved either.

This observation motivates aligning propagated responses before comparing or combining different hops.

Dataset	$\lvert V\rvert$	$\mathrm{nnz}(\mathbf{A})$	F	Role	Labels
Amazon	11,944	8,796,784	25	fraud	$0/1$
Photo	7,535	238,163	745	recommendation	$0/1$
Elliptic	46,564	73,248	93	AML	$0/1$
T-Finance	39,357	42,445,086	10	finance	$0/1$
Tolokers	11,758	530,758	8	heterophily	$0/1$
Questions	48,921	202,461	8	heterophily	$0/1$
DGraph	3,700,550	4,300,999	17	finance	$0/1$ target; $2/3$ background

Table 3: Bounded profiles of the exact dataset artifacts used in the main-table protocol. F is the input feature dimension after loader conversion and before K-fitted response standardization. The roles refer to the benchmark domains in the cited sources; AML denotes anti-money laundering, and the DGraph background labels are not outcome classes in the main-table metric population.

B Dataset Profiles and Protocol Boundaries

The seven benchmarks cover recommendation, fraud, anti-money-laundering, financial, and heterophily settings with substantially different scales and attribute spaces. Table 3 records a bounded profile of the exact loader artifacts used by the adopted main-table sweep. The edge column reports the number of nonzero entries in the loader adjacency; it is therefore not necessarily the number of undirected edges in a source release.

Amazon is the camouflaged-fraud graph (Dou et al., 2020), Photo is an image-based recommendation graph (McAuley et al., 2015), Elliptic is a Bitcoin anti-money-laundering transaction graph (Weber et al., 2019), and T-Finance is a financial transaction benchmark (Tang et al., 2022). Tolokers and Questions are heterophily benchmarks (Platonov et al., 2023). The loader applies project row-normalization to Amazon, Elliptic, and T-Finance, while Photo, Tolokers, Questions, and DGraph retain their raw attributes. The evaluated Elliptic label array contains only $0/1$; no separate unknown label is consumed by this loader.

B.1 Shared Semi-supervised Construction

For each dataset and split seed, the loader first forms a deterministic random ordering of the eligible split pool $\mathcal{P}_s$ and takes the first $\lfloor 0.15|\mathcal{P}_s| \rfloor$ candidates. It then retains only candidates with benchmark label 0 as the known-normal reference set K; the remaining nodes form $U = V \setminus K$. Thus, 15% is a candidate-prefix budget, not a claim that exactly 15% of all normal nodes are sampled. The size of K can vary across seeds. The model uses the graph, attributes, and K/U membership during training; outcome labels for nodes in U are joined only after scores are frozen. A higher score always denotes a more anomalous node. The main-table runs use seeds $0, \ldots, 4$ and the metric implementation described in Appendix B.2.

B.2 Evaluation Metrics

All anomaly metrics are computed only after the model scores are frozen, with larger scores denoting greater anomalousness. AUROC is the area under the receiver-operating-characteristic curve over the stated evaluation population. For AUPRC, we construct the empirical precision–recall curve by sweeping score thresholds and compute its area by trapezoidal integration. This matches the AUPRC convention used by RHO in the reported benchmark comparison. Trapezoidal AUPRC is distinct from step-wise average precision (AP); whenever AP appears in an appendix diagnostic, it is labeled explicitly and is not substituted for the main AUPRC metric.

B.3 DGraph-specific Protocol and Comparison Boundary

DGraph is the protocol exception in the main table. The loader uses the DGraph target-node pool ($y \in \{0, 1\}$) as the eligible split pool for constructing the 15% candidate prefix, retains only $y = 0$ candidates in K, and leaves the $y = 2/3$ background nodes in the full graph and in the U-side training risk. The main-table score population is then filtered after score freezing to $U \cap \{y \in \{0, 1\}\}$, with $y = 0$ treated as normal and $y = 1$ as anomalous. The official DGraph train/validation/test

Variant	Amazon	Photo	Elliptic	T-Finance	Tolokers	Questions
Full KU-GAD	0.905/0.739	0.949/0.806	0.907/0.769	0.890/0.710	0.735/0.435	0.584/0.055
No calibration	0.848/0.651	0.722/0.232	0.907/0.805	0.772/0.188	0.723/0.419	0.618/0.062
No propagation ($H = 0$)	0.933/0.835	0.672/0.189	0.907/0.769	0.524/0.051	0.702/0.381	0.584/0.072
Permuted graph	0.855/0.515	0.590/0.139	–	0.521/0.056	0.638/0.327	0.545/0.046
Terminal hop only	0.757/0.227	0.899/0.666	0.907/0.769	0.877/0.675	0.683/0.392	0.545/0.046
Simple aggregation	0.855/0.430	0.923/0.705	0.598/0.132	0.881/0.515	0.720/0.422	0.585/0.053

Table 4: Complete ablation results on the six shared-protocol datasets. Each entry reports five-seed mean AUROC/AUPRC. For *Permuted graph*, five deterministic permutations are averaged within each seed; Elliptic is omitted because its selected model uses $H = 0$.

Setting	Photo	T-Finance	Tolokers
Full	0.949/0.806	0.890/0.710	0.735/0.435
Input only	0.941/0.759	0.703/0.094	0.735/0.428
Hop only	0.904/0.681	0.896/0.724	0.720/0.424
None	0.722/0.232	0.772/0.188	0.723/0.419

Table 5: Five-seed mean AUROC/AUPRC for the two calibration stages. "Input" denotes standardization before propagation and "Hop" denotes recalibration of nonzero-hop responses.

masks are recorded in the source artifact but are not used as the main-table split masks. The main-table graph mode is directed; it is not silently symmetrized.

Accordingly, KU-GAD is evaluated on DGraph with a directed, background-retaining training graph and a target-only score population. The RHO entry in Table 1 is taken from its published benchmark results, so the DGraph column is interpreted as a reported-protocol comparison rather than a matched reimplementation.

C Additional Ablations and Sensitivity Analyses

For completeness, Table 4 reports the main ablation across all six shared-protocol datasets. Additional analyses below further examine calibration, the sequence reader, and normal-reference sensitivity.

C.1 Detailed Calibration Ablation

The main ablation removes both calibration stages together. To separate their roles, we additionally evaluate the 2×2 combination of input calibration before propagation and hop-wise recalibration after propagation on three representative multi-hop benchmarks.

Table 5 separates the contributions of the two calibration stages. Photo benefits from retaining both calibration stages, while T-Finance benefits especially from hop-wise recalibration. Tolokers changes only modestly across the calibrated variants. Overall, the two stages provide complementary ways of maintaining a useful normal reference across propagation.

C.2 Reader Implementation Analysis

We next test whether KU-GAD depends sensitively on two implementation choices in the sequence reader. The *No CLS* variant removes the CLS token and mean pools the Transformer outputs over hop tokens before applying the score head. The *No hop embedding* variant retains the CLS reader but removes the learnable hop-identity embeddings. All other dataset-specific configurations remain fixed. The experiment covers six datasets, three reader arms, and five seeds, for 90 trials in total.

Table 6 shows that both reader variants remain close to the full model. Hop embeddings have negligible effect, while mean pooling yields small mixed changes, indicating that KU-GAD is robust to these reader choices.

AUROC

Variant	Amazon	Photo	Elliptic	T-Finance	Tolokers	Questions
Full	0.9046 ± 0.0163	0.9492 ± 0.0117	0.9068 ± 0.0254	0.8901 ± 0.0115	0.7345 ± 0.0179	0.5843 ± 0.0580
No CLS	0.9030 ± 0.0129	0.9506 ± 0.0079	0.9089 ± 0.0224	0.8878 ± 0.0166	0.7349 ± 0.0194	0.5837 ± 0.0582
No hop embedding	0.9047 ± 0.0163	0.9491 ± 0.0117	0.9095 ± 0.0194	0.8898 ± 0.0112	0.7346 ± 0.0181	0.5843 ± 0.0578

AUPRC (trapezoidal)

Variant	Amazon	Photo	Elliptic	T-Finance	Tolokers	Questions
Full	0.7393 ± 0.0258	0.8059 ± 0.0289	0.7691 ± 0.0967	0.7100 ± 0.0131	0.4354 ± 0.0212	0.0545 ± 0.0080
No CLS	0.7221 ± 0.0070	0.8349 ± 0.0225	0.7812 ± 0.0832	0.7075 ± 0.0207	0.4452 ± 0.0219	0.0553 ± 0.0097
No hop embedding	0.7405 ± 0.0261	0.8061 ± 0.0289	0.7628 ± 0.0854	0.7098 ± 0.0126	0.4355 ± 0.0213	0.0545 ± 0.0080

Table 6: Reader implementation analysis. Entries are five-seed means $\pm$ sample standard deviations.

AUROC

Dataset	$\beta = 0$	0.5	1	2	4
Amazon	0.917 ± 0.021	0.862 ± 0.025	0.844 ± 0.044	0.738 ± 0.151	0.533 ± 0.152
Photo	0.933 ± 0.014	0.903 ± 0.023	0.823 ± 0.013	0.669 ± 0.059	0.544 ± 0.026
Elliptic	0.897 ± 0.032	0.923 ± 0.006	0.919 ± 0.013	0.913 ± 0.007	0.849 ± 0.060
T-Finance	0.870 ± 0.032	0.882 ± 0.009	0.890 ± 0.006	0.881 ± 0.012	0.817 ± 0.056
Tolokers	0.729 ± 0.016	0.715 ± 0.012	0.698 ± 0.011	0.669 ± 0.016	0.650 ± 0.012
Questions	0.597 ± 0.034	0.475 ± 0.018	0.451 ± 0.015	0.436 ± 0.006	0.428 ± 0.006

AUPRC (trapezoidal)

Dataset	$\beta = 0$	0.5	1	2	4
Amazon	0.874 ± 0.026	0.810 ± 0.079	0.766 ± 0.108	0.558 ± 0.192	0.337 ± 0.199
Photo	0.910 ± 0.019	0.878 ± 0.024	0.772 ± 0.030	0.492 ± 0.115	0.337 ± 0.015
Elliptic	0.886 ± 0.033	0.904 ± 0.005	0.899 ± 0.017	0.884 ± 0.010	0.742 ± 0.155
T-Finance	0.802 ± 0.037	0.820 ± 0.011	0.822 ± 0.006	0.778 ± 0.045	0.507 ± 0.181
Tolokers	0.755 ± 0.019	0.752 ± 0.015	0.729 ± 0.016	0.691 ± 0.022	0.668 ± 0.017
Questions	0.200 ± 0.045	0.117 ± 0.005	0.111 ± 0.003	0.108 ± 0.001	0.107 ± 0.002

Table 7: Controlled reference-bias results. Larger β increasingly concentrates the known-normal reference. Severe bias ($\beta = 4$) reduces both AUROC and AUPRC relative to unbiased sampling on all six datasets, while intermediate responses vary across datasets.

C.3 Reference-Bias Analysis

Table 7 reports the complete five-seed results for the intervention in Section 5.4. Bias levels share the same split seeds, enabling paired comparison; all metrics are computed after the model and configuration are frozen.

Because the bias levels use matched split seeds, we also examine paired changes relative to $\beta = 0$. At $\beta = 4$, the paired mean AUROC change is negative on all six datasets, with the largest declines on Amazon and Photo. These results are consistent with the sensitivity trends shown in Figure 2.

C.4 Reference-Budget Sensitivity

To examine sensitivity to the amount of verified normal supervision, we vary the nominal candidate-prefix budget over $\{1\%, 5\%, 10\%, 15\%\}$ while keeping the shared fixed optimization profile and the label-free-selected propagation-depth/checkpoint profiles fixed. The four reference sets are nested within each split, and all budget levels are evaluated on the same $U_{\text{eval}} = V \setminus K_{15\%}$ population after score freezing. The matrix covers six datasets, four budget levels, and five seeds, for a total of 120 formal trials. DGraph is omitted because its target-node/background protocol is not population-equivalent to the all-node candidate-prefix construction used for the other six datasets.

Figure 3 summarizes the reference-budget sensitivity. Overall, increasing the reference budget substantially improves performance: the 15% arm improves both metrics over the 1% arm in every paired seed comparison across all six datasets. Intermediate gains vary across datasets, with 15% giving the highest mean AUPRC on five of the six datasets.

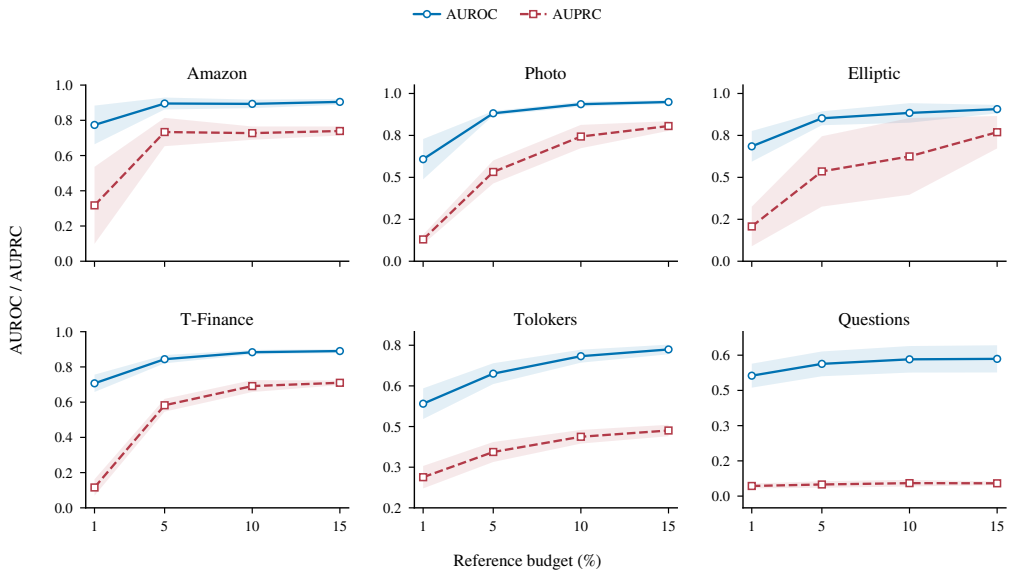

Figure 3: Reference-budget sensitivity across six shared-protocol datasets. Curves show five-seed mean AUROC and AUPRC as the verified-normal candidate-prefix budget increases from 1% to 15%; shaded regions indicate sample standard deviation. Y-axis ranges are dataset-specific for readability.

Because each candidate prefix is filtered to retain only benchmark-label-0 nodes, the actual reference fraction is slightly below the nominal budget. At the 15% setting, the mean $|K|/N$ ranges from 11.7% to 14.6% across the six datasets.

C.5 Oracle Mixture-Strength Analysis

The Population Ranking Equivalence proposition shows that the anomaly mixture preserves the ideal likelihood-ratio ordering under its population assumptions, whereas a small anomaly contribution compresses the score differences available to a finite learner. We examine this finite-model effect with an oracle intervention over $\widetilde{\pi} \in \{0.15, 0.25, 0.40, 0.60, 0.80, 1.00\}$ and compare it with the natural-mixture anchor π_{train}. The intervention keeps U membership, K, initialization, optimization profile, and training budget fixed; ground-truth anomaly identities are used to construct the oracle weights and for post-hoc analysis only.

Table 8 reports the oracle mixture-strength results. The fixed-reader and end-to-end trajectories increase throughout the oracle sweep, and every Full-U dose improves over the natural-mixture anchor in all four datasets. Full-U shows two small local AP reversals at intermediate doses, but the overall trajectory is strongly positive. The diagnostic therefore supports the finite-model mixture-dilution interpretation.

$\widetilde{\pi}$	Amazon	Photo	Elliptic	T-Finance
Fixed reader on strict U_{cross}				
π_{train}	0.9313/0.8610	0.8439/0.6017	0.5407/0.1905	0.8862/0.6922
0.15	0.9419/0.8697	0.8639/0.6460	0.5581/0.1967	0.8918/0.7148
0.25	0.9508/0.8778	0.8920/0.7124	0.5988/0.2130	0.8948/0.7215
0.40	0.9580/0.8858	0.9167/0.7736	0.6463/0.2361	0.8978/0.7283
0.60	0.9642/0.8945	0.9396/0.8328	0.6818/0.2555	0.9007/0.7349
0.80	0.9690/0.9029	0.9614/0.8876	0.7025/0.2689	0.9032/0.7407
1.00	0.9736/0.9152	0.9894/0.9537	0.7150/0.2786	0.9058/0.7442
End-to-end on strict U_{cross}				
π_{train}	0.9276/0.8662	0.9049/0.7628	0.8983/0.8216	0.8516/0.6486
0.15	0.9483/0.8847	0.9363/0.8255	0.9155/0.8621	0.9037/0.7896
0.25	0.9590/0.8986	0.9600/0.8881	0.9375/0.8967	0.9181/0.8144
0.40	0.9673/0.9104	0.9704/0.9136	0.9533/0.9158	0.9273/0.8232
0.60	0.9720/0.9172	0.9765/0.9241	0.9640/0.9263	0.9314/0.8293
0.80	0.9748/0.9189	0.9803/0.9304	0.9681/0.9302	0.9351/0.8324
1.00	0.9765/0.9253	0.9820/0.9321	0.9717/0.9346	0.9361/0.8328
Full-U				
π_{train}	0.9478/0.8452	0.9032/0.6733	0.9060/0.8011	0.8499/0.5817
0.15	0.9605/0.8603	0.9362/0.7525	0.9195/0.8278	0.9100/0.7541
0.25	0.9685/0.8752	0.9621/0.8362	0.9416/0.8610	0.9250/0.7799
0.40	0.9730/0.8867	0.9725/0.8720	0.9564/0.8896	0.9298/0.7917
0.60	0.9755/0.8929	0.9804/0.8923	0.9681/0.9073	0.9362/0.7908
0.80	0.9781/0.8947	0.9854/0.9001	0.9725/0.8967	0.9417/0.8022
1.00	0.9797/0.9029	0.9877/0.9038	0.9753/0.9097	0.9453/0.8043

Table 8: Label-controlled oracle target-mixture-strength diagnostic. Entries are five-seed mean AUROC/step-wise AP. Fixed-reader and end-to-end results use strict U_{cross}; Full-U uses its complete scored U. All aligned oracle doses use ground-truth anomaly identities and are excluded from the label-free detection protocol.

D Label-Free Configuration Protocol

The label-free configuration procedure is applied in two stages. It first chooses H by maximizing mean held-out K/U membership AUC over $H \in \{0, \ldots, 20\}$, then fixes that depth and selects the optimizer checkpoint by the same criterion. Neither stage uses anomaly outcomes. The selected configurations and proxy AUCs are summarized below.

	Amazon	Photo	Elliptic	T-Finance	Tolokers	Questions	DGraph
H	9	7	0	15	11	20	5
Step	32	32	1024	256	64	64	4096
$\mathrm{AUC}_H/\mathrm{AUC}_{\mathrm{step}}$	.532/.532	.550/.554	.541/.543	.521/.524	.557/.557	.507/.505	.694/.707

Values aggregate five development seeds and five deterministic membership folds; checkpoint margins range from 0.00025 to 0.00815. Questions is the weakest proxy case and selects the search-boundary value $H = 20$. On Amazon, a finer post-freeze diagnostic still selects step 32 by membership AUC, while a later checkpoint yields better anomaly metrics despite lower membership AUC, indicating proxy–outcome misalignment.