arXiv:2608.10699v1 [cs.LG] 11 Aug 2026

ProTAGAD: A Foundation Model for TAG Anomaly Detection with Decoupled Topological and Textual Prototypes

Ziyan Wang†, Liwen Wu†, Cheng Xie*, Song Gao, Zhenli He, Xin Jin

School of Software and AI, Yunnan University, Kunming, China
wangziyan_iaab@stu.ynu.edu.cn, lwwu@ynu.edu.cn, xiecheng@ynu.edu.cn,
gaos@ynu.edu.cn, hezl@ynu.edu.cn, xinjin@ynu.edu.cn

Abstract

Text-Attributed Graphs (TAGs), endowed with abundant textual content along with topological structures, have emerged as a versatile backbone for real-world anomaly detection spanning large language model security, social network moderation, and cyber threat identification. Unlike conventional Graph Anomaly Detection (GAD), which relies primarily on structural irregularities, TAG anomaly detection must jointly leverage both topological patterns and fine-grained textual semantics to capture nuanced anomalous behaviors. The current GNN-based anomaly detectors adopt holistic message-passing schemes that indiscriminately fuse structural proximity and textual semantics during propagation, leading to deep cross-modality coupling. This entanglement acts as a noise amplifier, obscuring subtle anomalous signals and directly giving rise to the ***Blurred-Anomaly-Boundary (BAB)*** issue by rendering normal-anomalous decision boundaries poorly separable. This challenge is further amplified for graph foundation models that require robust cross-domain generalization. To bridge this gap, we introduce a novel foundation model for TAG anomaly detection featuring decoupled topological and textual prototypes. Our framework constructs dual prototype banks to independently model structural normality and semantic consistency, effectively isolating anomaly cues that are otherwise diluted during coupled aggregation. Extensive experiments across 14 diverse benchmark datasets demonstrate that our method consistently achieves state-of-the-art performance in cross-domain settings. Notably, the ablation studies further corroborate the prevalence of the ***BAB*** issue in conventional coupled TAG anomaly detectors, and show that our decoupled prototype design effectively mitigates this challenge.

1 Introduction

Graph anomaly detection (GAD) aims to identify nodes that deviate from dominant graph patterns and has been widely applied to fraud detection, social network moderation, cybersecurity, and recommendation systems (Ding et al. 2019; Liu et al. 2021; Ma et al. 2021; Qiao et al. 2025b). In real-world graphs, nodes are often associated with rich textual content in addition to relational structures, such as paper abstracts, product descriptions, and user posts. Such data are commonly formulated as text-attributed graphs (TAGs) (Zhu et al. 2026; Bandyopadhyay, Lokesh, and Murty 2019), where anomalies may arise from irregular connections, subtle semantic inconsistencies, or both. Effective TAG anomaly detection therefore requires jointly exploiting textual semantics and graph topology (Yang et al. 2021; Zhao et al. 2022).

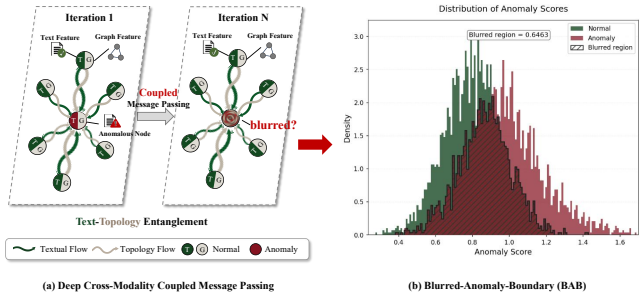


Figure 1: Motivation: deep cross-modality coupling obscures subtle anomaly cues, leading to the Blurred-Anomaly-Boundary (BAB) problem.

Existing GAD methods are primarily designed for numerical attributes and structural irregularities. Although recent TAGAD methods employ language models to encode raw text, most still rely on holistic GNN message passing to jointly propagate textual and topological information. As illustrated in Figure 1, this coupled propagation progressively smooths or distorts subtle anomaly cues, reducing the distinction between normal and anomalous nodes and thereby giving rise to the *Blurred-Anomaly-Boundary* (*BAB*) issue.

Recent graph pre-training and prompt-learning methods have explored transferable graph representations across tasks and domains (Hou et al. 2022; Sun et al. 2022; Liu et al. 2023, 2025; Xiong et al. 2025). However, the *BAB* issue becomes more severe in the generalist setting, where one model is trained on multiple source graphs and directly applied to unseen domains. Cross-domain variations in textual semantics and connectivity patterns make jointly propagated representations prone to retaining domain-specific neighborhood information, further weakening anomaly cues under distribution shifts and hindering zero-shot generalization.

To address this issue, we propose **ProTAGAD**, a prototype-based foundation model that decouples textual and topological representation learning. ProTAGAD separately learns transferable textual anomaly prototypes and topological normality prototypes without exchanging hidden representations, and combines their anomaly scores only at the

decision level. This design avoids cross-modal interference while preserving the complementary anomaly evidence of both modalities.

We evaluate ProTAGAD on 14 TAG datasets under a source-to-target zero-shot protocol. ProTAGAD achieves the best AUROC on seven of eight unseen target graphs, with an average rank of 1.12, demonstrating strong cross-domain generalization. Additional analyses further reveal the complementary roles of textual and topological prototypes and verify that decoupled prototype modeling effectively alleviates the *BAB* issue.

Our main contributions are summarized as follows:

- We formally identify the Blurred-Anomaly-Boundary (*BAB*) issue in generalist TAG anomaly detection, attributing its root cause to entangled cross-modal fusion via holistic GNN message passing.
- We propose ProTAGAD, a foundation model for TAG anomaly detection that leverages decoupled textual anomaly prototypes and topological normality prototypes to isolate modality-specific anomaly evidence without cross-modal interference.
- Extensive experiments on 14 diverse TAG benchmarks achieve state-of-the-art zero-shot cross-domain performance; ablations further verify dual prototype complementarity and empirically confirm both the prevalence of the BAB issue and the efficacy of our decoupled design.

2 Related Work

2.1 Graph Anomaly Detection

GAD aims to identify nodes that deviate from the dominant attribute or structural patterns (Wang et al. 2023). Existing methods mainly include reconstruction-, self-supervised-, spectral-, affinity-, and augmentation-based approaches. Reconstruction-based methods detect anomalies through the reconstruction of attributes or structures (Ding et al. 2019; Fan, Zhang, and Li 2020; Luo et al. 2022; Duan et al. 2026). Self-supervised methods learn anomaly-sensitive node-context or neighborhood patterns, represented by CoLA (Liu et al. 2021) and HCM-A (Huang et al. 2022). Spectral methods exploit high-frequency graph signals, including BWGNN (Tang et al. 2022) and GHRN (Gao et al. 2023), while affinity-based methods model one-class normality and suppress suspicious connections, such as TAM (Qiao and Pang 2023) and GCTAM (Zhang et al. 2025). Augmentation-based methods, including Semi-GGAD (Qiao et al. 2024) and CAGAD (Xiao et al. 2024a), synthesize anomalies to enhance model robustness. However, these methods generally follow a one-for-one paradigm with fixed numerical attributes, requiring retraining on new graphs and overlooking the fine-grained semantics of raw text.

2.2 Generalist Graph Anomaly Detection

Generalist graph anomaly detection (GGAD) extends conventional GAD from graph-specific learning toward cross-domain generalization, aiming to train a unified detector that can be transferred to unseen graphs (Pan et al. 2025; Wang et al. 2023). ARC (Liu et al. 2024) introduces in-context learning for transferable anomaly detection, while UNPrompt (Niu et al. 2024) and AnomalyGFM (Qiao et al. 2025a) explore zero/few-shot detection through unified prompts and graph foundation models. Recent studies further investigate domain shifts through invariant representation learning and prototype-based knowledge transfer, such as IA-GGAD (Zhang et al. 2026), DR-GGAD (Fu et al. 2026), OWLEYE (Zheng et al. 2026), and ProMoS (Xu et al. 2026).

However, existing GGAD methods generally focus on numerical node representations and structural distribution shifts. The semantic information contained in raw texts is not explicitly modeled, and the interaction between textual and topological anomaly evidence remains unexplored, limiting their applicability to text-attributed graphs.

2.3 Text-Attributed Graph Anomaly Detection

Text-attributed Graph Anomaly Detection (TAGAD) identifies anomalous nodes by jointly modeling textual attributes and graph topology. Conventional methods typically encode texts into fixed-dimensional representations, which may miss subtle anomaly cues. Recent approaches improve textual modeling through contrastive learning or large language model reasoning. CMUCL (Xu et al. 2025c) captures textual and structural inconsistencies through multi-scale contrastive learning, CoLL (Xu et al. 2025b) extracts textual anomaly evidence through collaborative LLM reasoning, and TAG-AD (Xu et al. 2025a) studies realistic anomaly construction and retrieval-augmented zero-shot detection.

However, existing methods often entangle textual and topological information, causing semantic anomalies to be smoothed by neighborhood aggregation, while LLM-based approaches rarely model transferable structural normality. Most are also tailored to individual graphs. Instead, ProTAGAD learns decoupled textual and topological prototype banks and combines their scores for cross-domain detection on unseen graphs.

3 Preliminaries

Notations. Let $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathcal{T}, \mathbf{X})$ denote a text-attributed graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set, $\mathbf{A} \in \{0,1\}^{N\times N}$ is the adjacency matrix, and $\mathcal{T} = \{t_i\}_{i=1}^{N}$ contains the raw textual attributes associated with the nodes. An entry $\mathbf{A}_{ij} = 1$ indicates an edge between v_i and v_j. The matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$ denotes the initial node feature matrix, whose i-th row $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional feature vector associated with node v_i.

For each node v_i, we denote its one-hop neighborhood by $\mathcal{N}_i$ and construct a local textual subgraph containing the text of the target node and its neighbors:

$$\mathcal{G}_i^t = \{t_i\} \cup \{t_j \mid j \in \mathcal{N}_i\}. \tag{1}$$

The local textual subgraph enables the agent to assess both the semantic content of the target node and its consistency with neighboring texts.

Generalist TAG Anomaly Detection. Given a collection of source-domain TAGs $\mathcal{T}_{\text{train}} = \{\mathcal{G}_{\text{train}}^{(1)}, \ldots, \mathcal{G}_{\text{train}}^{(n_s)}\}$ and a disjoint collection of unseen target-domain TAGs $\mathcal{T}_{\text{test}} = \{\mathcal{G}_{\text{test}}^{(1)}, \ldots, \mathcal{G}_{\text{test}}^{(n_t)}\}$, where n_s and n_t denote the numbers of

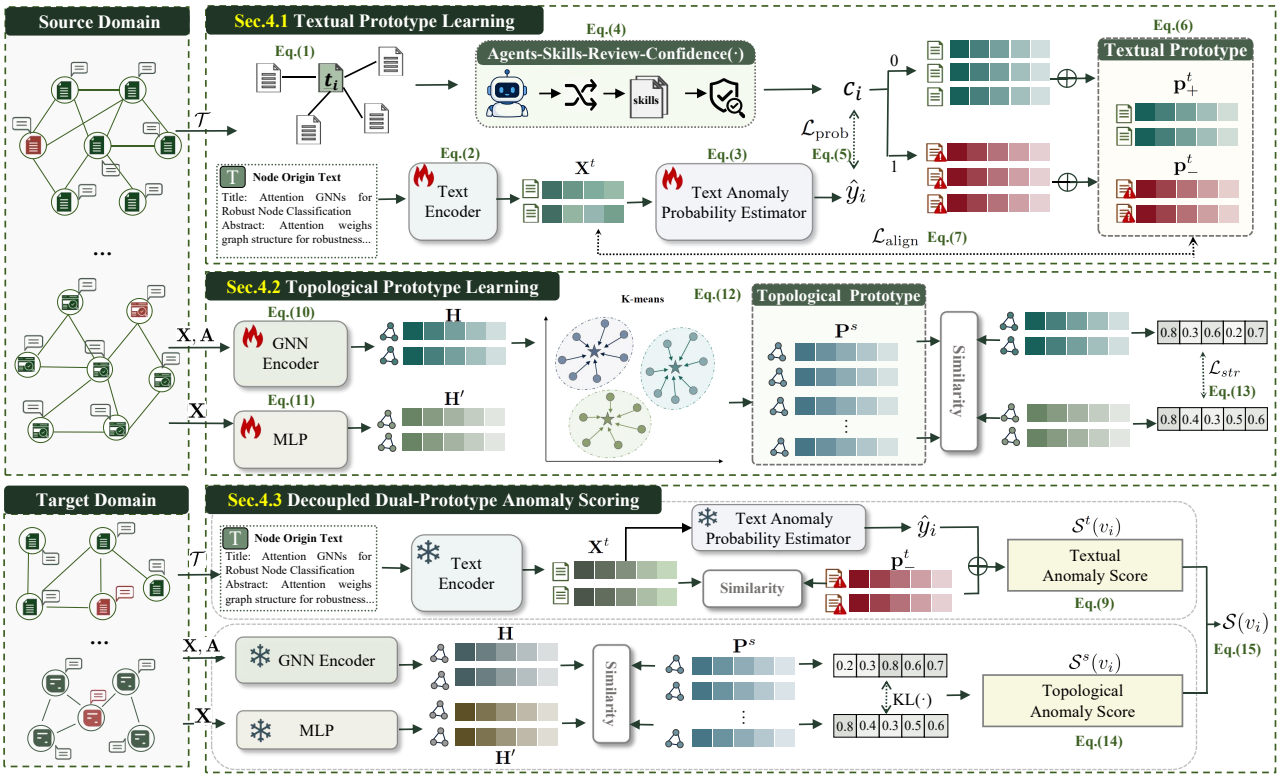


Figure 2: Overview of ProTAGAD.

source and target graphs, respectively, our goal is to learn a unified anomaly detector from $\mathcal{T}_{\text{train}}$ and directly generalize it to $\mathcal{T}_{\text{test}}$. During inference, the model parameters are frozen, and no target-domain labels or additional fine-tuning are available. For each node v_i in a target graph, the detector outputs an anomaly score $\mathcal{S}(v_i) \in \mathbb{R}$, where a larger value indicates a higher likelihood of being anomalous.

4 Methodology

To mitigate the *BAB* problem, we propose ProTAGAD, which decouples textual and topological representation learning by avoiding shared cross-modal message propagation. As illustrated in Figure 2, ProTAGAD separately learns textual and topological prototypes and fuses their anomaly scores for zero-shot inference on unseen graphs.

4.1 Textual Prototype Learning

The textual module aims to extract transferable semantic anomaly patterns from raw node texts. We first employ a text encoder f_{txt}^{Θ} to map the textual attributes into a continuous representation space:

$$\mathbf{X}^t = f_{txt}^{\Theta}(\mathcal{T}) = \left[\ \mathbf{x}_1^t, \ldots, \mathbf{x}_N^t\ \right]^\top \in \mathbb{R}^{N \times d_t}, \quad (2)$$

where $\mathbf{x}_i^t$ denotes the textual representation of node v_i.

Based on these textual representations, we employ a lightweight text anomaly probability estimator to estimate the textual anomaly probability of each node:

$$\hat{\mathbf{Y}} = f_{prob}^{\Theta}(\mathbf{X}^t) = [\hat{y}_1, \ldots, \hat{y}_N]^\top, \quad (3)$$

where $\hat{y}_i \in [0, 1]$ denotes the estimated textual anomaly probability of node v_i.

Existing text encoders often emphasize general semantics, making fine-grained anomaly cues difficult to capture. To address this issue, we introduce the `Agents-Skills-Review-Confidence` mechanism illustrated in Figure 3. Given a local textual subgraph $\mathcal{G}_i^t$, the anomaly agent first performs a coarse assessment. The Skill Router then identifies the graph domain, activates the corresponding domain-specific skill, and conducts fine-grained analysis of whether the target text is semantically consistent with its topic or content (Schick et al. 2023). The reviewer further verifies the decision consistency, supporting textual evidence, and confidence reliability (Madaan et al. 2023; Gou et al. 2023). After reviewing all nodes, we identify an elbow point in the anomaly confidence distribution as the threshold. Nodes with confidence scores above the threshold are assigned a pseudo-label of 1, while the remaining nodes are assigned 0. The resulting binary indicator for node v_i is defined as

$$c_i = \texttt{Agents-Skills-Review-Confidence}\left(\mathcal{G}_i^t\right), \quad (4)$$

where $\mathbf{C} = [c_1, \ldots, c_N]^\top$, with $c_i = 1$ indicating a high-confidence textual anomaly and $c_i = 0$ indicating low textual anomaly confidence. These agent-derived binary indicators are used only as pseudo-labels to supervise the text anomaly probability estimator through the following binary

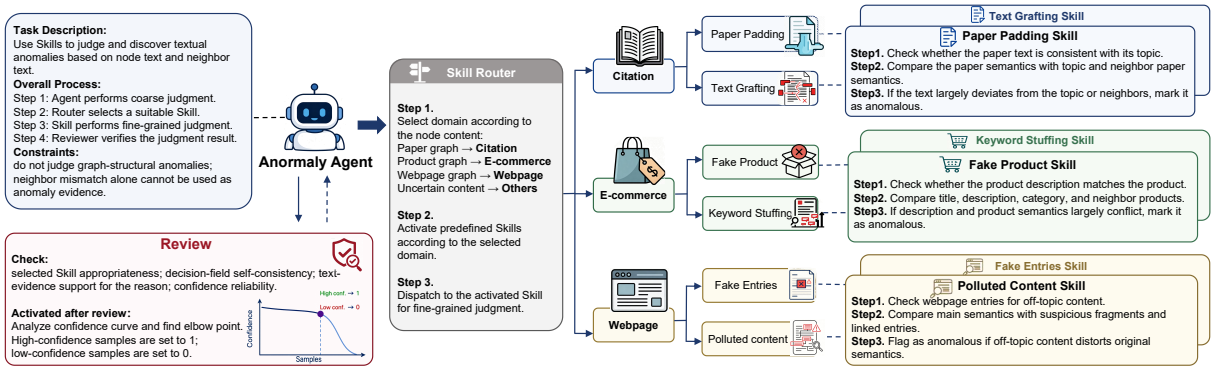


Figure 3: The overview of the function Agents-Skills-Review-Confidence(·). The input is textual subgraphs for each node. The output is the activated textual anomaly confidence c_i for each node.

cross-entropy loss:

$$\mathcal{L}_{\text{prob}} = -\frac{1}{N}\sum_{i=1}^{N}\left[c_i \log(\hat{y}_i) + (1-c_i)\log(1-\hat{y}_i)\right]. \quad (5)$$

We further use these pseudo-labels to partition the node representations into anomalous and normal groups, which are then aggregated to construct the corresponding textual prototypes:

$$\mathbf{p}_-^t = \frac{1}{\|C\|_1}\sum_{i=1}^{N} c_i \cdot \mathbf{x}_i^t,\ \mathbf{p}_+^t = \frac{1}{\|1-C\|_1}\sum_{i=1}^{N}(1-c_i)\cdot \mathbf{x}_i^t, \quad (6)$$

where $\mathbf{p}_-^t$ and $\mathbf{p}_+^t$ are obtained by averaging the textual representations of the inferred anomalous and normal nodes, respectively, thereby capturing the representative semantic patterns of the two groups.

However, constructing the prototypes alone does not explicitly constrain the positions of individual nodes in the representation space. To further enhance the separation between anomalous and normal textual patterns, we introduce a prototype alignment objective:

$$\mathcal{L}_{\text{align}} = \frac{1}{N}\Bigg[\sum_{i=1}^{N} c_i \log\left(1+\exp\left(\phi(\mathbf{x}_i^t,\mathbf{p}_+^t) - \phi(\mathbf{x}_i^t,\mathbf{p}_-^t)\right)\right) + (1-c_i)\log\left(1+\exp\left(\phi(\mathbf{x}_i^t,\mathbf{p}_-^t) - \phi(\mathbf{x}_i^t,\mathbf{p}_+^t)\right)\right)\Bigg], \quad (7)$$

where $\phi(\cdot,\cdot)$ denotes cosine similarity. For $c_i = 1$, the objective pulls $\mathbf{x}_i^t$ toward $\mathbf{p}_-^t$ and pushes it away from $\mathbf{p}_+^t$, while for $c_i = 0$, it pulls $\mathbf{x}_i^t$ toward $\mathbf{p}_+^t$ and pushes it away from $\mathbf{p}_-^t$. This objective yields a more discriminative textual representation space with a clearer normal-anomalous boundary.

The probability estimation and prototype alignment objectives optimize different components of the textual module:

$$\mathcal{L}_{prob} \xrightarrow{\texttt{update}} f_{prob}^{\Theta}(\cdot),\ \mathcal{L}_{align} \xrightarrow{\texttt{update}} f_{txt}^{\Theta}(\cdot)\,, \quad (8)$$

where $\mathcal{L}_{\text{prob}}$ optimizes the text anomaly probability estimator to estimate textual anomaly probabilities, while $\mathcal{L}_{\text{align}}$ optimizes the text encoder to learn discriminative semantic representations of anomalies.

After training, the textual anomaly score of node v_i is defined as

$$\mathcal{S}^t(v_i) = f_{prob}^{\Theta}(\mathbf{x}_i^t) + \phi\left(\mathbf{x}_i^t, \mathbf{p}_-^t\right) + \|\mathbf{x}^{\mathbf{t}}{}_i - \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\mathbf{x}_j^t\|_2^2, \quad (9)$$

where the first term is the textual anomaly probability estimated by f_{prob}^{Θ}, while the second measures the semantic affinity between the node representation and the textual anomaly prototype learned from the source graphs. The third term captures the deviation between a node's textual representation and its local textual neighborhood. Before summation, all score terms are standardized using Z-score normalization to eliminate their scale differences. By integrating these standardized signals, $\mathcal{S}^t(v_i)$ provides textual evidence for the final decoupled anomaly scoring.

4.2 Topological Prototype Learning

Following ProMoS (Xu et al. 2026), we learn two node representations to model transferable structural normality. A Graph Transformer is first trained with self-supervised contrastive learning to obtain topology-aware node representations:

$$\mathbf{H} = f_{\texttt{GNN}}^{\Theta}(\mathbf{X},\mathbf{A}) = \texttt{Graph Transformer}(\mathbf{X},\mathbf{A}), \quad (10)$$

where $\mathbf{H} = [\mathbf{h}_1,\ldots,\mathbf{h}_N]^\top$ encodes the node attributes together with their topological contexts.

To obtain a complementary representation, we further employ an MLP that takes only the node features as input and learns to approximate the topology-aware representations through knowledge distillation:

$$\mathbf{H}' = f_{\texttt{MLP}}^{\Theta}(\mathbf{X}) = \sigma((\sigma(\mathbf{X}\mathbf{W}+\mathbf{b}))\mathbf{W}'+\mathbf{b}') \quad (11)$$

Although the MLP does not directly use the adjacency matrix, the distillation process transfers topology-related knowledge from $\mathbf{H}$ to $\mathbf{H}'$. The two representations therefore provide complementary views for subsequent topological prototype construction.

To capture diverse structural normality patterns, we apply K-means clustering to the topology-aware representations

$\mathbf{H}$ and use the resulting cluster centers as topological prototypes:

$$\mathbf{P}^s = \texttt{K-means}(\mathbf{H}, K) = \{\mathbf{p}_1^s, \ldots, \mathbf{p}_K^s\}, \tag{12}$$

where each $\mathbf{p}_k^s$ represents a typical structural normality pattern in the topology-aware representation space. To ensure that the projected representation $\mathbf{h}_i'$ preserves the structural patterns encoded in $\mathbf{h}_i$, we align their similarity distributions over the shared topological prototypes. Specifically, the structural consistency objective is defined as

$$\mathcal{L}_{str} = \frac{1}{N}\sum_{i=1}^{N} \texttt{KL}\big(\mathrm{softmax}\big(\mathbf{h}_i \cdot (\mathbf{P}^s)^\top\big) \| \,\mathrm{softmax}\big(\mathbf{h}_i' \cdot (\mathbf{P}^s)^\top\big)\big) \tag{13}$$

where $\texttt{KL}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence. This objective encourages $\mathbf{h}_i'$ to retain the relative affinities of $\mathbf{h}_i$ to different topological prototypes, thereby preserving its topology-aware structural semantics. Based on these prototypes and the two complementary node representations, we define the topological anomaly score of node v_i as

$$\begin{aligned}\mathcal{S}^s(v_i) = &\texttt{KL}\left(\mathrm{softmax}\left(\mathbf{h}_i \cdot (\mathbf{P}^s)^\top\right) \parallel \mathrm{softmax}\left(\mathbf{h}_i' \cdot (\mathbf{P}^s)^\top\right)\right) \\ &+ \|\mathbf{h}_i - \mathbf{p}_m^s\|_2^2\,,\ m = \arg\min_k \|\mathbf{h}_i - \mathbf{p}_k^s\|_2^2\,.\end{aligned} \tag{14}$$

Specifically, the two terms measure the discrepancy between the similarity distributions of the two node representations over the topological prototypes and the deviation from the nearest topological prototype, respectively. Before summation, both score terms are standardized using Z-score normalization to eliminate scale differences.

4.3 Decoupled Dual-Prototype Anomaly Scoring

The textual and topological branches model semantic and structural anomalies in separate prototype spaces. The two branches produce the textual anomaly score and the topological anomaly score, respectively. ProTAGAD combines the two scores only at the final scoring stage, thereby avoiding the interference caused by deeply coupled representations. The overall anomaly score of node v_i is defined as

$$\mathcal{S}(v_i) = \mathcal{S}^t(v_i) + \mathcal{S}^s(v_i). \tag{15}$$

Before final aggregation, the textual and topological anomaly scores are independently standardized using Z-score normalization to ensure comparable scales. Consequently, a node is considered more anomalous when it exhibits a large standardized deviation in either textual semantics or topological structure, resulting in a higher final anomaly score $\mathcal{S}(v_i)$.

5 Experiments

5.1 Experimental Settings

Datasets. We adopt a cross-domain source/target split across 14 text-attributed graphs spanning the citation, e-commerce, web, and encyclopedia domains (Sen et al. 2008; Hu et al. 2020; Yan et al. 2023, 2024; Mernyei and Cangea 2020; Wang et al. 2025). We synthesize and inject realistic anomalies that mirror real-world scenarios, including off-topic papers, citation manipulation, misleading products, fraudulent co-purchases, fake encyclopedia entries, and promotional content. Building on CMUCL (Xu et al. 2025c), we further introduce contextual and structural anomalies to construct a challenging benchmark to evaluate zero-shot cross-domain generalization.

Dataset	Train	Test	Nodes	Edges	AvgLen	Anomaly
Citation network						
Cora	✓	–	2,791	10,990	135.45	164
Citeseer	–	✓	3,262	3,684	153.94	192
Pubmed	–	✓	21,248	92,154	256.08	1,184
Arxiv	✓	–	176,514	1,225,722	179.70	10,162
E-commerce network						
History	✓	–	43,503	373,468	228.36	2,494
Children	✓	–	85,137	1,591,763	209.12	4,612
Grocery	–	✓	17,908	147,332	67.36	1,024
Movies	–	✓	17,625	166,690	81.85	1,000
Toys	–	✓	21,283	117,424	74.50	1,240
Fitness	–	✓	187,918	3,113,588	21.87	10,384
Products	✓	–	46,617	117,064	110.65	2,752
Web network						
Cornell	–	✓	197	596	261.27	21
Texas	–	✓	192	608	192.47	21
WikiCS	✓	–	14,485	443,558	422.00	704

Table 1: Statistics of datasets.

Baselines. We compare ProTAGAD with 18 representative baselines: (1) *GAD methods*— DOMINANT (Ding et al. 2019), BGNN (Ivanov and Prokhorenkova 2021), BWGNN (Tang et al. 2022), GHRN (Gao et al. 2023), CoLA (Liu et al. 2021), HCM-A (Huang et al. 2022), TAM (Qiao and Pang 2023), Semi-GGAD (Qiao et al. 2024), CAGAD (Xiao et al. 2024a) and GCTAM (Zhang et al. 2025); (2) *GGAD methods*—ARC (Liu et al. 2024), IA-GGAD (Zhang et al. 2026), AnomalyGFM (Qiao et al. 2025a), UNPrompt (Niu et al. 2024), OWLEYE (Zheng et al. 2026) and ProMoS (Xu et al. 2026); (3) *TAGAD methods*—CMUCL (Xu et al. 2025c) and CoLL (Xu et al. 2025b).

Implementation. We report AUROC and AUPRC as mean $\pm$ standard deviation over five random seeds (Tang et al. 2023). Each method is trained once on $\mathcal{T}_{\text{train}}$ and directly evaluated on $\mathcal{T}_{\text{test}}$ under a pretrain-only protocol. We use BGE(Xiao et al. 2024b), GraphTransformer, and DeepSeek4-Flash as the text encoder, graph encoder, and LLM agent, respectively. The agent is used only for offline preprocessing, with its cached outputs shared across all seeds; no LLM calls are required during training or inference. This preprocessing takes approximately 395.8 seconds and costs USD 6.64. Additional implementation details are provided in Appendix B.

5.2 Main Results

Table 2 summarizes the AUROC results on eight unseen target graphs. ProTAGAD achieves the best performance on seven datasets and attains the lowest average rank of 1.12, demonstrating stable zero-shot cross-domain generalization. The most substantial improvements are observed on Citeseer (+8.65%) and Pubmed (+8.17%), indicating that the textual prototype bank effectively preserves fine-grained semantic anomalies and prevents them from being weakened during

Method	Citeseer	Pubmed	Texas	Grocery	Movies	Toys	Fitness	Cornell	Rank
GAD Methods									
DOMINANT (2019)	54.18±1.11	56.24±1.96	67.96±1.19	55.89±1.01	55.87±1.07	58.38±0.92	67.66±0.78	66.77±0.87	10.12
BGNN* (2021)	55.23±5.72	43.56±12.80	53.05±7.14	53.19±2.30	56.27±4.00	51.94±2.43	50.51±2.17	57.74±9.96	14.00
BWGNN* (2022)	61.26±1.82	54.16±4.11	49.60±7.91	59.20±1.12	55.35±0.97	57.82±0.77	49.76±1.53	50.28±3.46	12.50
GHRN* (2023)	59.34±1.56	50.96±4.61	50.71±6.07	58.65±2.44	52.97±2.31	56.98±2.45	49.69±2.49	56.28±2.60	13.00
CoLA (2021)	57.56±1.91	48.91±1.10	47.86±6.40	53.22±0.95	52.34±0.81	55.90±0.77	54.27±0.44	47.16±4.27	15.00
HCM-A (2022)	60.68±0.75	60.13±1.18	57.57±1.77	64.68±0.46	64.93±0.36	66.55±0.39	65.11±0.79	67.16±1.64	8.00
TAM (2023)	28.01±0.02	74.90±0.24	59.47±1.18	70.67±0.20	66.77±0.13	74.73±0.13	OOM	61.07±2.46	8.56
Semi-GGAD (2024)	46.51±14.23	35.87±6.46	42.62±6.75	42.81±7.73	42.93±6.33	44.74±6.77	43.08±7.35	36.75±5.28	18.62
CAGAD (2024)	59.56±4.00	42.11±6.44	59.69±6.49	55.67±1.61	56.32±3.27	56.18±2.60	48.14±6.08	54.41±2.69	12.88
GCTAM (2025)	49.60±1.11	70.76±0.84	67.30±2.50	65.50±0.56	64.34±0.26	68.64±0.54	OOM	63.93±2.29	9.56
GGAD Methods									
ARC* (2024)	73.23±0.01	75.22±1.03	75.66±1.07	70.95±0.36	68.16±0.74	74.22±1.01	77.93±0.61	**81.25±1.94**	2.50
IA-GGAD* (2025)	62.29±1.38	60.11±1.92	55.27±3.55	66.42±0.41	68.67±0.87	69.85±0.55	78.14±0.50	56.84±2.45	6.62
AnomalyGFM* (2025)	56.19±2.18	68.59±2.14	58.84±2.38	55.93±2.01	51.49±1.68	56.21±1.14	59.97±1.49	74.50±1.93	10.75
UNPrompt* (2025)	65.23±1.53	66.99±6.99	46.80±5.72	54.43±1.93	55.56±1.86	53.56±1.63	66.86±2.35	54.94±1.62	11.75
OWLEYE* (2026)	73.02±0.20	71.32±0.03	77.54±0.06	66.66±0.01	68.24±0.01	72.82±0.01	72.82±0.01	74.83±0.06	3.50
ProMoS (2026)	64.28±1.47	70.06±0.58	74.88±1.03	65.73±0.75	65.77±0.39	69.34±0.26	77.24±0.33	74.76±1.20	5.12
TAGAD Methods									
CMUCL (2025)	57.58±1.28	53.13±1.79	51.86±6.57	49.63±1.56	50.69±0.43	48.17±0.29	49.92±0.70	62.41±3.74	14.25
CoLL (2025)	55.79±1.91	56.26±2.33	45.55±2.49	57.32±2.09	59.80±2.46	58.94±0.86	60.02±1.65	51.20±2.83	12.12
ProTAGAD	**81.88±0.38**	**83.39±0.42**	**81.81±1.09**	**74.85±0.17**	**72.08±0.33**	**77.39±0.53**	**79.37±0.09**	80.31±0.84	**1.12**
Δ	↑8.65	↑8.17	↑4.27	↑3.90	↑3.41	↑2.66	↑1.23	↓0.94	–

Table 2: Performance comparison on different target domains (AUROC, %, mean ± std). "Rank" is the average rank over eight targets; Δ is the AUROC difference between ProTAGAD and the strongest baseline. Methods marked with * are supervised methods. The best and second-best results are shown in bold and underlined, respectively. OOM denotes out of memory.

topological aggregation. The consistent gains on Texas, Grocery, Movies, Toys, and Fitness further suggest that the decoupled topological prototype bank can capture transferable structural normality across different domains. The only exception is Cornell, where ProTAGAD achieves an AUROC of 80.31%, which is 0.94% lower than the best baseline. This small gap may result from the extremely limited graph size, which provides insufficiently diverse patterns for stable prototype estimation. Overall, the results show that independently modeling semantic consistency and topological normality enables ProTAGAD to preserve more discriminative anomaly cues under domain shifts. In addition, the standard deviations remain below 1.1% on all target graphs, confirming the model's stable performance across random seeds. Full AUPRC results appear in Appendix D.

5.3 Ablation Study

To disentangle the contribution of each prototype module, we evaluate four variants: (i) **Backbone**, which removes both prototype modules; (ii) **+ SP**, which incorporates only the structural prototype module; (iii) **+ TP**, which incorporates only the textual prototype module; and (iv) **Ours**, which combines both modules. As shown in Table 3, structural prototypes consistently improve the Backbone across all target graphs, with particularly large gains on Fitness (+33.82%), Cornell (+25.44%), and Texas (+24.02%), demonstrating their ability to capture transferable structural normality. Textual prototypes also yield notable improvements on Pubmed (+11.89%) and Fitness (+9.33%), indicating their effectiveness in capturing semantic anomaly cues. Combining both modules achieves the best performance on every target graph, confirming that textual and structural prototypes provide complementary evidence for cross-domain anomaly detection.

Datasets	Backbone	+ SP	+ TP	Ours
Citeseer	56.53±2.51	76.75±2.23	60.69±1.08	**81.88±0.38**
Pubmed	63.14±4.10	75.35±0.79	75.03±0.57	**83.39±0.42**
Grocery	49.63±1.67	71.19±0.39	56.85±1.08	**74.85±0.17**
Movies	52.96±2.03	68.93±0.18	54.11±1.64	**72.08±0.33**
Toys	55.68±1.67	74.57±0.36	58.80±1.40	**77.39±0.53**
Fitness	44.03±2.53	77.85±0.31	53.36±1.06	**79.37±0.09**
Cornell	54.50±5.31	79.94±0.74	52.91±3.42	**80.31±0.84**
Texas	54.69±2.13	78.71±0.64	52.88±2.72	**81.81±1.09**

Table 3: Ablation study on different target domains in terms of AUROC (%).

5.4 Effect of Decoupled Prototype Modeling

To evaluate decoupled prototype modeling, we compare ProTAGAD with a coupled variant under identical settings. In the coupled variant, the textual features and original node features are first projected to the same dimensionality, separately ℓ_2-normalized, and then averaged. The fused features replace the original node features as the input to the graph encoder.

Both textual and topological prototypes are constructed from the resulting fused representations, and anomaly scores are computed in this shared prototype space.

We further introduce Anomaly Boundary Separability ($\mathcal{ABS}$) to quantify the *BAB* issue. Let $\hat{f}^+(s)$ and $\hat{f}^-(s)$ denote the kernel density estimates of the anomaly scores for normal and anomalous nodes, respectively. Both densities are estimated using a Gaussian kernel with a shared bandwidth determined by applying Scott's rule to the pooled scores of the two groups. We define

$$\mathcal{ABS} = \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(\hat{f}^+(s) \,\|\, M\right) + \frac{1}{2} D_{\mathrm{KL}}\left(\hat{f}^-(s) \,\|\, M\right)} \tag{16}$$

where $M = \frac{1}{2}(\hat{f}^+ + \hat{f}^-)$ denotes the mixture distribution, and $D_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ represents the Kullback–Leibler divergence. A larger $\mathcal{ABS}$ indicates clearer separation between normal and anomalous nodes and thus a less severe *BAB* issue.

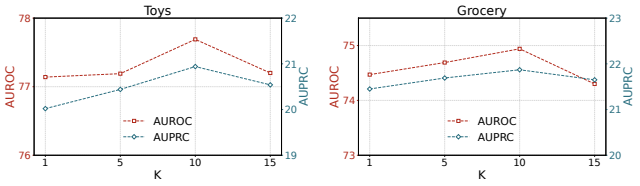
As shown in Table 4, ProTAGAD consistently outperforms the coupled variant on all eight target graphs, increasing the average AUROC from 61.36% to 78.89% and the average $\mathcal{ABS}$ from 0.2260 to 0.4553. The lower $\mathcal{ABS}$ of the coupled variant indicates that normal and anomalous nodes exhibit more similar anomaly-score distributions, leading to weaker boundary separability and a more severe *BAB* issue. In contrast, decoupled prototype modeling preserves the complementary anomaly information of the two modalities, resulting in more discriminative anomaly scores and clearer normal–anomalous boundaries.

	AUROC (%) ↑			$\mathcal{ABS}$ ↑		
Dataset	**Coupled**	**Decoupled**	Δ	**Coupled**	**Decoupled**	Δ
Citeseer	74.51	**81.88**	7.37	0.4529	**0.5416**	0.0887
Pubmed	62.40	**83.39**	20.99	0.1876	**0.5336**	0.3460
Grocery	58.12	**74.85**	16.73	0.1227	**0.3861**	0.2634
Movies	60.09	**72.08**	11.99	0.1729	**0.3536**	0.1807
Toys	59.73	**77.39**	17.66	0.1523	**0.4253**	0.2730
Fitness	62.28	**79.37**	17.09	0.1723	**0.4948**	0.3225
Cornell	59.65	**80.31**	20.66	0.3043	**0.4369**	0.1326
Texas	54.13	**81.81**	27.68	0.2428	**0.4709**	0.2281

Table 4: Comparison between coupled and decoupled variants in terms of AUROC (%) and anomaly boundary separability ($\mathcal{ABS}$). Higher values indicate better detection performance and clearer anomaly boundaries.

5.5 Parameter Sensitivity

We investigate the effect of the number of structural prototypes K in Eq. (12) on the Toys and Grocery datasets. As shown in Figure 4, both AUROC and AUPRC gradually improve as K increases from 1 to 10, and achieve their best performance at $K = 10$. A small K provides insufficient prototypes to characterize the diverse structural normality patterns across graphs. In contrast, an excessively large K may fragment the normal structural distribution and introduce domain-specific noise, leading to performance degradation when $K = 15$. Overall, ProTAGAD remains relatively stable under different values of K, while $K = 10$ provides the best balance between prototype diversity and cross-domain generalization.

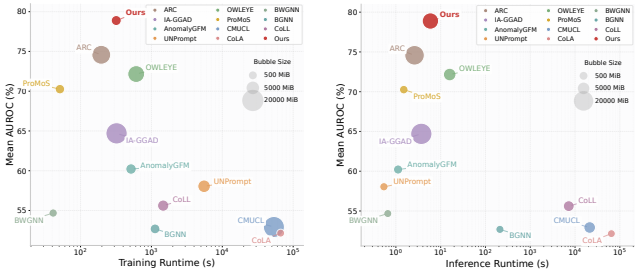


(a) Toys (b) Grocery

Figure 4: Impact of K for ProTAGAD on Toys and Grocery datasets.

5.6 Efficiency Analysis

Figure 5 compares the runtime, GPU memory usage, and mean AUROC of different methods during training and inference, where bubble size denotes memory consumption. ProTAGAD lies near the upper-left region in both panels, indicating that it achieves the highest detection performance with competitive computational efficiency. It also maintains relatively low GPU memory usage during training, while its inference memory consumption is comparatively high. Overall, ProTAGAD achieves a favorable trade-off between detection performance and computational efficiency.



(a) Training (b) Inference

Figure 5: Efficiency comparison of different methods in terms of runtime and GPU memory usage during training and inference.

6 Conclusion

We presented **ProTAGAD**, a zero-shot foundation model for generalist text-attributed graph anomaly detection on unseen graphs. By decoupling textual and topological representation learning and separately learning textual and structural prototypes, ProTAGAD alleviates the *Blurred-Anomaly-Boundary* (*BAB*) issue. Across 14 TAG datasets, ProTAGAD achieves the best AUROC on seven of eight target graphs with an average rank of 1.12. Ablation studies and coupled-versus-decoupled comparisons further validate the complementarity of the two prototype branches and the effectiveness of the proposed design. Future work will explore more memory-efficient prototype learning and extend ProTAGAD to large-scale graph domains.

References

Bandyopadhyay, S.; Lokesh, N.; and Murty, M. N. 2019. Outlier aware network embedding for attributed networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 12–19.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM international conference on data mining*, 594–602. SIAM.

Duan, H.; Yu, B.; Xie, C.; Li, L.; He, Z.; and Jin, X. 2026. Designated Masking Propagation Learning for Self-Supervised Heterogeneous Graph Representation. *ACM Transactions on Knowledge Discovery from Data*, 20(5): 1–23.

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

Fu, C.; He, Z.; Zhang, X.; Xie, C.; Jin, X.; and Yang, Y. 2026. DR-GGAD: Dual Residual Centering for Mitigating Anomaly Non-Discriminativity in Generalist Graph Anomaly Detection. In *The Fourteenth International Conference on Learning Representations*.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference 2023*, 1528–1538.

Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; and Chen, W. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 594–604.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.

Huang, T.; Pei, Y.; Menkovski, V.; and Pechenizkiy, M. 2022. Hop-count based self-supervised anomaly detection on attributed networks. In *Joint European conference on machine learning and knowledge discovery in databases*, 225–241. Springer.

Ivanov, S.; and Prokhorenkova, L. 2021. Boost then convolve: Gradient boosting meets graph neural networks. *arXiv preprint arXiv:2101.08543*.

Liu, J.; Yang, C.; Lu, Z.; Chen, J.; Li, Y.; Zhang, M.; Bai, T.; Fang, Y.; Sun, L.; Yu, P. S.; et al. 2025. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, Y.; Li, S.; Zheng, Y.; Chen, Q.; Zhang, C.; and Pan, S. 2024. Arc: A generalist graph anomaly detector with in-context learning. *Advances in Neural Information Processing Systems*, 37: 50772–50804.

Liu, Y.; Li, Z.; Pan, S.; Gong, C.; Zhou, C.; and Karypis, G. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6): 2378–2392.

Liu, Z.; Yu, X.; Fang, Y.; and Zhang, X. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, 417–428.

Luo, X.; Wu, J.; Beheshti, A.; Yang, J.; Zhang, X.; Wang, Y.; and Xue, S. 2022. Comga: Community-aware attributed graph anomaly detection. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 657–665.

Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q. Z.; Xiong, H.; and Akoglu, L. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE transactions on knowledge and data engineering*, 35(12): 12012–12038.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36: 46534–46594.

Mernyei, P.; and Cangea, C. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Niu, C.; Qiao, H.; Chen, C.; Chen, L.; and Pang, G. 2024. Zero-shot generalist graph anomaly detection with unified neighborhood prompts. *arXiv preprint arXiv:2410.14886*.

Pan, J.; Zheng, Y.; Tan, Y.; and Liu, Y. 2025. A survey of generalization of graph anomaly detection: From transfer learning to foundation models. *arXiv preprint arXiv:2509.06609*.

Qiao, H.; Niu, C.; Chen, L.; and Pang, G. 2025a. Anomalygfm: Graph foundation model for zero/few-shot anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2326–2337.

Qiao, H.; and Pang, G. 2023. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. *Advances in Neural Information Processing Systems*, 36: 49490–49512.

Qiao, H.; Tong, H.; An, B.; King, I.; Aggarwal, C.; and Pang, G. 2025b. Deep graph anomaly detection: A survey and new perspectives. *IEEE Transactions on Knowledge and Data Engineering*.

Qiao, H.; Wen, Q.; Li, X.; Lim, E.-P.; and Pang, G. 2024. Generative semi-supervised graph anomaly detection. *Advances in neural information processing systems*, 37: 4660–4688.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36: 68539–68551.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Sun, M.; Zhou, K.; He, X.; Wang, Y.; and Wang, X. 2022. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 1717–1727.

Tang, J.; Hua, F.; Gao, Z.; Zhao, P.; and Li, J. 2023. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in Neural Information Processing Systems*, 36: 29628–29653.

Tang, J.; Li, J.; Gao, Z.; and Li, J. 2022. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, 21076–21089. PMLR.

Wang, H. P.; Liu, S.; Wei, R.; and Li, P. 2025. Generalization Principles for Inference over Text-Attributed Graphs with Large Language Models. In *Forty-second International Conference on Machine Learning*.

Wang, Q.; Pang, G.; Salehi, M.; Buntine, W.; and Leckie, C. 2023. Cross-domain graph anomaly detection via anomaly-aware contrastive alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4676–4684.

Xiao, C.; Pang, S.; Xu, X.; Li, X.; Trajcevski, G.; and Zhou, F. 2024a. Counterfactual data augmentation with denoising diffusion for graph anomaly detection. *IEEE Transactions on Computational Social Systems*, 11(6): 7555–7567.

Xiao, S.; Liu, Z.; Zhang, P.; Muennighoff, N.; Lian, D.; and Nie, J.-Y. 2024b. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 641–649.

Xiong, Z.; Xie, C.; Duan, H.; and Yu, B. 2025. NoisehGNN: Synthesized similarity graph-based neural network for noised heterogeneous graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21725–21733.

Xu, H.; Qian, R.; Yao, Z.; Liu, Z.; Li, L.; Li, Y.; Li, Y.; Zheng, W.; Rosa, D.; Barcklow, D.; et al. 2025a. LLM-Powered Text-Attributed Graph Anomaly Detection via Retrieval-Augmented Reasoning. *arXiv preprint arXiv:2511.17584*.

Xu, Y.; Chen, J.; Peng, Z.; Chen, Z.; Lin, Q.; Ma, L.; Shi, B.; and Dong, B. 2025b. Court of LLMs: Evidence-Augmented Generation via Multi-LLM Collaboration for Text-Attributed Graph Anomaly Detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2437–2446.

Xu, Y.; Chen, Z.; Peng, Z.; Wang, S.; Shi, B.; Dong, B.; and Shen, C. 2026. Generalist Graph Anomaly Detection via Prototype-Based Distillation. *arXiv preprint arXiv:2605.26857*.

Xu, Y.; Hua, X.; Peng, Z.; Shi, B.; Chen, J.; Fu, X.; Wang, S.; and Dong, B. 2025c. Text-Attributed Graph Anomaly Detection via Multi-Scale Cross-and Uni-Modal Contrastive Learning. *arXiv preprint arXiv:2508.00513*.

Yan, H.; Li, C.; Long, R.; Yan, C.; Zhao, J.; Zhuang, W.; Yin, J.; Zhang, P.; Han, W.; Sun, H.; et al. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36: 17238–17264.

Yan, H.; Li, C.; Yin, J.; Yu, Z.; Han, W.; Li, M.; Zeng, Z.; Sun, H.; and Wang, S. 2024. When graph meets multimodal: Benchmarking and meditating on multimodal attributed graphs learning. *arXiv preprint arXiv:2410.09132*.

Yang, J.; Liu, Z.; Xiao, S.; Li, C.; Lian, D.; Agrawal, S.; Singh, A.; Sun, G.; and Xie, X. 2021. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34: 28798–28810.

Zhang, X.; He, Z.; Fu, C.; and Xie, C. 2026. Ia-ggad: Zero-shot generalist graph anomaly detection via invariant and affinity learning. *Advances in Neural Information Processing Systems*, 38: 83728–83760.

Zhang, X.; Peng, H.; He, Z.; Xie, C.; Jin, X.; and Jiang, H. 2025. GCTAM: Global and Contextual Truncated Affinity Combined Maximization Model For Unsupervised Graph Anomaly Detection. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 3642–3650. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhao, J.; Qu, M.; Li, C.; Yan, H.; Liu, Q.; Li, R.; Xie, X.; and Tang, J. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*.

Zheng, L.; Fu, D.; Li, Z.; and He, J. 2026. OWLEYE: Zero-Shot Learner for Cross-Domain Graph Data Anomaly Detection. *arXiv preprint arXiv:2601.19102*.

Zhu, Z.; Huang, T.; Wang, K.; Ye, J.; Chen, X.; and Luo, S. 2026. Graph-based approaches and functionalities in retrieval-augmented generation: A comprehensive survey. *ACM Computing Surveys*, 58(10): 1–38.

A Description of Datasets

We evaluate ProTAGAD on 14 text-attributed graph datasets spanning three domains: citation networks, e-commerce networks, and web networks. Following the cross-domain evaluation setting, Cora, Arxiv, Children, Products, History, and WikiCS are used as source graphs for training, while Citeseer, Pubmed, Grocery, Movies, Toys, Fitness, Cornell, and Texas serve as unseen target graphs for testing. These datasets exhibit substantial variations in graph scale, topology, textual content, and domain semantics, providing a diverse benchmark for evaluating zero-shot cross-domain generalization.

- **Cora** (Sen et al. 2008), **Citeseer**, and **Pubmed** are citation networks in which nodes represent academic papers and edges denote citation relationships. The textual attributes of each node are constructed from the title and abstract of the corresponding paper.
- **Arxiv** (Hu et al. 2020) is constructed from the ogbn-arxiv dataset, where nodes correspond to research papers and directed edges indicate citation links. We adopt its raw-text version released by the TAG benchmark (Yan et al. 2023), which augments the original citation topology with the title and abstract of each paper.
- **Products** (Hu et al. 2020) is a subset of the ogbn-products graph, where nodes represent commercial products, edges indicate co-purchase relationships, and node attributes contain product titles and descriptions.
- **Children**, **History** (Yan et al. 2023), **Grocery**, **Movies**, and **Toys** (Yan et al. 2024) are category-specific Amazon product graphs, where nodes represent products, edges capture co-purchase relationships, and product titles and descriptions serve as textual attributes.
- **Fitness** (Hu et al. 2020; Yan et al. 2023) is a large-scale Amazon product graph, where nodes represent fitness-related products, edges capture co-purchase relationships, and product descriptions serve as textual attributes.
- **WikiCS** (Mernyei and Cangea 2020) is a web network composed of Wikipedia pages related to computer science. Nodes represent Wikipedia entries, and edges correspond to hyperlinks between pages. The textual attributes are derived from entry names and page contents.
- **Cornell** (Wang et al. 2025) is a web network collected from the Cornell University website. Nodes represent webpages, and edges denote hyperlinks between them. The node attributes are constructed from the original textual contents of the webpages.
- **Texas** (Wang et al. 2025) is a web network collected from the computer science website of the University of Texas. Nodes correspond to webpages, and edges represent hyperlink relationships. The textual attributes consist of webpage titles and body texts.

Anomaly Injection. For the datasets requiring anomaly injection, we follow CMUCL (Xu et al. 2025c) and construct anomalous nodes through an inner-injection stage and an outer-injection stage. The inner-injection stage perturbs the original textual attributes and graph topology to generate two contextual and two structural anomaly types. For each contextual anomaly, we randomly select a normal target node and sample $K = 50$ candidate nodes from the same graph. The candidate whose text embedding has the lowest cosine similarity to that of the target node is selected as the anomaly source. Contextual insertion inserts textual content from the source node into an eligible field of the target text, whereas contextual replacement substitutes an existing sentence or text block with source content. Field boundaries, including titles, descriptions, and abstracts, are preserved during both operations. For structural perturbation, dense-clique anomalies are generated by randomly sampling groups of four nodes and fully connecting the nodes within each group. Random-edge anomalies are constructed by connecting a selected target node to randomly sampled nodes, where the number of newly added edges is drawn from the empirical degree distribution of the original graph.

The outer-injection stage further introduces domain-grounded anomalies using external anomaly-source nodes. Their textual content is used for contextual insertion or replacement, while structural anomalies are generated by appending external nodes to the graph and either forming dense four-node cliques with selected target nodes or connecting them to target nodes through randomly added edges. Existing anomaly labels are retained throughout the process, and the target nodes assigned to different anomaly types are kept disjoint. By default, the inner-injection stage introduces approximately 4% anomalous nodes in total, while the outer-injection stage further injects approximately 2% anomalous nodes. The labels produced by the two stages are combined to obtain the final anomaly annotations.

B Details of Implementation

Hyper-parameters. We use a shared hyper-parameter configuration for all datasets without target-specific tuning. The aligned feature dimension and hidden dimension are both set to 64, and the 768-dimensional BGE embeddings are projected through a two-layer text adapter. The Graph Transformer contains 2 layers and 4 attention heads with a dropout rate of 0.2. The hidden dimension and dropout rate of the multilayer perceptrons are set to 128 and 0.1, respectively. The number of topological prototypes is selected from $\{1, 5, 10, 15\}$ and set to 10. The warm-up and main training epochs are set to 5 and 3. The feature dropout rates are 0.1 and 0.2, the edge dropout rates are 0.2 and 0.4, and the contrastive temperature is 0.4. At most 4,096 nodes are used for each contrastive loss computation. We use Adam, with the learning rates for the model parameters and prototypes searched over $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$ and both set to 5×10^{-3}. The weight decay is set to 5×10^{-5}, and the maximum gradient norm is 5.0. For each source graph, we sample 64 subgraphs per epoch with at most 256 nodes and an expansion length of 3. We additionally use 64 subgraphs for prototype initialization and at most 64 skill-guided anchor subgraphs. All score components are normalized using z-score normalization.

Baseline Implementation. We use the official implementations and recommended hyper-parameter settings whenever they are publicly available. All baselines use the same source–target dataset split, input features, and five random

seeds as ProTAGAD. Each method is trained on $\mathcal{T}_{\text{train}}$ and directly evaluated on the unseen graphs in $\mathcal{T}_{\text{test}}$ without target-specific fine-tuning or hyper-parameter selection. We retain the original supervision setting and anomaly-scoring mechanism of each baseline to avoid method-specific modifications that could affect the comparison.

LLM Agent Implementation. The LLM agent uses DeepSeek4-Flash through the API provided by the DeepSeek Open Platform and is invoked only during offline preprocessing on the source graphs. For each source graph, approximately 6% of nodes are selected through stratified random sampling over source-graph node groups. For each selected node, the input contains its original textual attribute and the textual context of its one-hop neighborhood, with at most 10 neighboring nodes retained. The center-node text is truncated to at most 5,000 characters, and each neighboring node text is truncated to at most 320 characters. The maximum API concurrency is set to 1,000. All generated outputs are cached and shared across the five random seeds. The complete preprocessing involves 30,996 API requests and consumes 2.003×10^8 tokens. The entire process takes approximately 395.8 seconds and costs USD 6.64. No LLM calls are required during model training or inference.

Implementation Details. All experiments were conducted on a server equipped with an Intel Xeon Platinum 8458P CPU and four NVIDIA A40 GPUs, each with 48 GB of memory. The implementation was developed using Python 3.10.19, PyTorch 2.6.0 with CUDA 11.8, and PyTorch Geometric 2.7.0. We used `bge-base-en-v1.5` as the text encoder, a Graph Transformer as the graph encoder, and DeepSeek4-Flash as the LLM agent.

Metrics. We evaluate anomaly detection performance using AUROC and AUPRC. AUROC measures the ranking quality of anomaly scores, while AUPRC is more informative under class imbalance. Higher AUROC and AUPRC indicate better detection performance. All results are reported as mean$\pm$standard deviation over five independent runs.

C Description of Baselines

To ensure a fair and comprehensive evaluation of ProTAGAD, we compare it against 18 representative baselines spanning three categories: conventional graph anomaly detection (GAD), generalized graph anomaly detection (GGAD), and text-attributed graph anomaly detection (TAGAD). These categories represent common methodological frameworks in graph anomaly detection and include several state-of-the-art (SOTA) models, providing a broad basis for performance comparison under different assumptions and settings.

C.1 GAD Methods

- **BGNN** (Ivanov and Prokhorenkova 2021) integrates gradient-boosted decision trees with graph neural networks to jointly model node attributes and graph structure.
- **BWGNN** (Tang et al. 2022) employs localized Beta-wavelet filters to capture anomalous signals across different graph frequency bands.
- **GHRN** (Gao et al. 2023) suppresses potentially heterophilic edges using high-frequency graph signals to improve anomaly discrimination.
- **DOMINANT** (Ding et al. 2019) jointly reconstructs node attributes and graph structure, using their reconstruction errors to identify anomalous nodes.
- **CoLA** (Liu et al. 2021) performs anomaly detection through contrastive learning between target nodes and their sampled neighborhood subgraphs.
- **HCM-A** (Huang et al. 2022) adopts hop-count prediction as a self-supervised task and estimates anomalies based on prediction errors and uncertainty.
- **TAM** (Qiao and Pang 2023) learns anomaly-discriminative representations by maximizing local affinity while truncating suspicious graph interactions.
- **Semi-GGAD** (Qiao et al. 2024) generates pseudo-anomalous nodes from a small set of labeled normal nodes to train a one-class detector.
- **CAGAD** (Xiao et al. 2024a) uses counterfactual graph augmentation to construct anomaly-aware neighborhoods and improve anomaly separability.
- **GCTAM** (Zhang et al. 2025) combines contextual and global affinity with adaptive truncation to reduce incorrect anomaly judgments.

C.2 GGAD Methods

- **ARC** (Liu et al. 2024) performs cross-graph anomaly detection through feature alignment, ego-neighbor residual encoding, and in-context reconstruction.
- **AnomalyGFM** (Qiao et al. 2025a) aligns neighborhood residual representations with graph-independent prototypes for zero-shot and few-shot anomaly detection.
- **UNPrompt** (Niu et al. 2024) learns transferable neighborhood prompts and detects anomalies according to node-attribute prediction discrepancies.
- **IA-GGAD** (Zhang et al. 2026) learns anomaly-relevant invariant representations while reducing feature and structural shifts across graphs.
- **OWLEYE** (Zheng et al. 2026) captures transferable normal patterns through cross-domain feature alignment and multi-pattern dictionary learning.
- **ProMoS** (Xu et al. 2026) employs prototype-guided knowledge distillation and a mixture-of-students architecture for zero-shot anomaly detection.

C.3 TAGAD Methods

- **CMUCL** (Xu et al. 2025c) jointly models textual attributes and graph topology through multi-scale cross-modal and uni-modal contrastive learning.
- **CoLL** (Xu et al. 2025b) combines multi-LLM semantic reasoning with GNN-based topology modeling to detect contextual and structural anomalies.

Method	Citeseer	Pubmed	Texas	Grocery	Movies	Toys	Fitness	Cornell	Rank
GAD Methods									
DOMINANT (2019)	9.45±1.32	9.37±0.93	21.14±1.93	7.60±0.23	6.97±0.29	8.10±0.23	13.23±1.62	31.16±3.65	10.50
BGNN* (2021)	9.01±3.31	5.85±3.20	12.14±2.10	6.61±0.65	6.99±0.91	6.54±0.92	5.66±0.53	18.69±6.67	14.88
BWGNN* (2022)	20.81±3.69	13.63±5.85	12.93±2.68	9.36±0.82	8.17±1.27	8.28±0.42	8.59±1.65	15.66±1.35	11.12
GHRN* (2023)	18.52±1.34	15.18±10.03	13.45±2.50	9.27±1.26	8.11±1.34	8.18±0.92	9.04±1.88	16.24±1.59	10.62
CoLA (2021)	8.94±1.42	5.43±0.17	13.88±3.37	6.35±0.21	6.04±0.21	7.14±0.17	6.33±0.13	10.28±0.77	16.00
HCM-A (2022)	10.72±0.67	9.41±0.61	13.99±0.97	12.27±0.67	11.14±0.46	15.73±0.60	8.86±0.50	18.89±2.69	9.50
TAM (2023)	3.92±0.00	11.87±0.22	15.93±0.88	10.01±0.09	8.55±0.05	12.64±0.09	OOM	14.29±0.93	12.19
Semi-GGAD (2024)	5.78±1.96	4.20±0.53	14.49±7.91	4.84±0.82	4.78±0.70	5.16±0.81	4.71±0.81	11.68±4.61	17.62
CAGAD (2024)	17.03±3.90	5.59±1.58	21.59±7.46	7.97±1.28	8.52±1.38	7.37±0.54	7.97±3.75	15.73±2.27	11.62
GCTAM (2025)	7.69±0.23	26.77±1.78	24.51±4.05	13.34±0.72	10.14±0.39	12.35±0.69	OOM	21.83±3.98	8.81
GGAD Methods									
ARC* (2024)	51.36±0.14	**29.70±0.86**	**31.44±1.77**	19.79±0.38	17.31±0.81	19.86±1.18	32.29±0.87	40.11±3.90	2.00
IA-GGAD* (2025)	35.21±0.33	10.57±1.58	13.40±1.65	18.82±0.71	18.59±0.51	17.67±1.10	32.26±0.63	12.49±1.57	7.50
AnomalyGFM* (2025)	28.61±0.95	13.88±3.92	16.81±3.21	7.46±0.83	6.46±0.26	7.25±0.42	8.34±0.81	29.45±3.44	10.38
UNPrompt* (2025)	12.04±1.11	9.98±2.40	18.23±4.92	6.07±0.27	6.60±0.24	5.81±0.23	13.22±3.14	24.83±1.67	11.88
OWLEYE* (2026)	43.90±0.01	21.98±0.02	27.93±0.09	12.95±0.40	17.72±0.01	17.26±0.01	22.38±0.01	35.23±0.11	3.75
ProMoS (2026)	25.65±0.94	12.21±0.27	22.86±0.20	12.24±0.34	12.59±0.25	12.28±0.58	29.61±1.62	29.83±1.02	6.25
TAGAD Methods									
CMUCL (2025)	14.29±0.66	5.90±0.32	13.61±3.48	5.49±0.20	5.50±0.07	5.27±0.04	5.17±0.10	15.03±3.28	15.62
CoLL (2025)	7.93±0.72	19.99±1.70	15.01±1.84	11.17±0.74	13.18±1.47	11.12±1.38	16.98±0.95	22.85±1.97	8.50
ProTAGAD	**52.43±0.97**	27.20±0.69	29.65±0.53	**21.92±0.12**	**19.17±0.36**	**20.30±1.00**	**41.04±0.43**	**44.88±2.88**	**1.25**

Table 5: Performance comparison on different target domains (AUPRC, %, mean ± std). "Rank" is the average rank over eight targets; Methods marked with * are supervised methods. The best and second-best results are shown in bold and underlined, respectively. OOM denotes out of memory.

D Performance Comparison of AUPRC

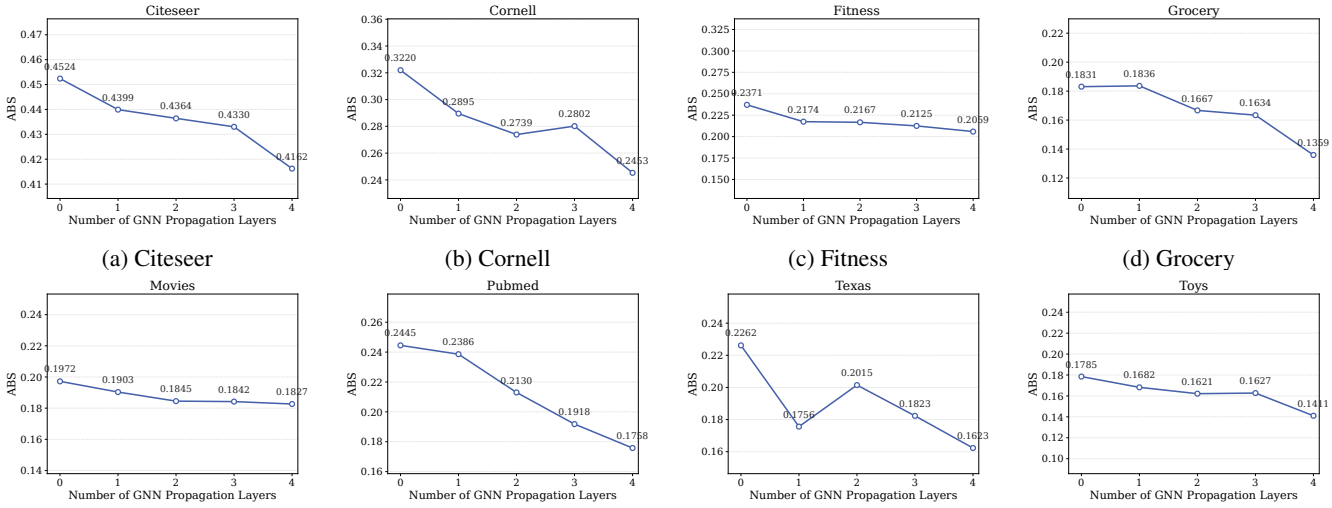
As shown in Table 5, ProTAGAD achieves the best overall AUPRC performance on the eight unseen target graphs, ranking first on six datasets and second on the remaining two, with the lowest average rank of 1.25. Compared with the strongest baseline ARC, ProTAGAD improves the average AUPRC from 30.23% to 32.07%. The largest gains are observed on Fitness (+8.75%) and Cornell (+4.77%), while consistent improvements on Citeseer, Grocery, Movies, and Toys demonstrate its effectiveness across citation, e-commerce, and web domains. Although ProTAGAD is slightly inferior to ARC on Pubmed and Texas, the gaps remain limited to 2.50% and 1.79%, respectively. Overall, these results indicate that separately preserving textual anomaly cues and topological normality patterns enables ProTAGAD to maintain robust anomaly discrimination under diverse graph structures and anomaly distributions, further validating the effectiveness of the decoupled prototype design in alleviating the *BAB* issue.

E Effect of GNN Propagation on $\mathcal{ABS}$

To directly examine whether graph propagation induces the blurred-anomaly-boundary issue, we evaluate the coupled variant with zero to four GNN propagation layers while keeping all other settings unchanged. Layer 0 measures $\mathcal{ABS}$ before graph propagation, whereas Layers 1–4 measure it after successive propagation steps. As shown in Figure 6, the four-layer $\mathcal{ABS}$ is lower than its pre-propagation value on all eight target graphs. On average, $\mathcal{ABS}$ decreases from 0.2551 at Layer 0 to 0.2082 at Layer 4, corresponding to a relative reduction of 18.41%. Particularly pronounced reductions are observed on Texas, Pubmed, and Grocery, where $\mathcal{ABS}$ decreases by 28.25%, 28.10%, and 25.78%, respectively. Although several datasets exhibit minor intermediate fluctuations, the overall tendency remains downward, and every dataset obtains its lowest or near-lowest separability after deeper propagation. Since a lower $\mathcal{ABS}$ indicates greater overlap between the normal and anomalous score distributions, these results provide direct evidence that repeated coupled propagation weakens anomaly boundary separability and aggravates the BAB issue. This observation further motivates the decoupled design of ProTAGAD, which avoids repeatedly propagating textual anomaly cues through graph neighborhoods and preserves their discriminability before decision-level fusion.

F Algorithmic description

The algorithmic description of the training and inference process of ProTAGAD is summarized in Algorithm 1, and Algorithm 2, respectively.



(a) Citeseer (b) Cornell (c) Fitness (d) Grocery

(e) Movies (f) Pubmed (g) Texas (h) Toys

Figure 6: Effect of coupled GNN propagation on Anomaly Boundary Separability ($\mathcal{ABS}$) across eight target graphs. Layer 0 denotes the anomaly scores before graph propagation, while Layers 1–4 denote the scores after successive propagation layers. A lower $\mathcal{ABS}$ indicates a more severe blurred-anomaly-boundary issue.

Algorithm 1: Training Algorithm of ProTAGAD

Input: Source graphs $\mathcal{T}_{\text{train}}$.
Parameters: Training epochs T; number of topological prototypes K.

Initialize all learnable parameters.
for each epoch $t = 1, \ldots, T$ **do**
  for each graph $\mathcal{G} \in \mathcal{T}_{\text{train}}$ **do**
    Derive textual representations $\mathbf{X}^t$ via Eq. (2).
    Estimate anomaly probabilities $\widehat{\mathbf{Y}}$ via Eq. (3).
    Obtain agent-derived pseudo-labels $\mathbf{C}$ via Eq. (4).
    Compute the probability estimation loss $\mathcal{L}_{\text{prob}}$ via Eq. (5).
    Construct textual prototypes $\mathbf{P}^t$ via Eq. (6).
    Compute the textual prototype alignment loss $\mathcal{L}_{\text{align}}$ via Eq. (7).
    Derive topology-aware representations $\mathbf{H}$ via Eq. (10).
    Derive projected representations $\mathbf{H}'$ via Eq. (11).
    Construct topological prototypes $\mathbf{P}^s$ from $\mathbf{H}$ via Eq. (12).
    Compute the structural consistency loss $\mathcal{L}_{\text{str}}$ via Eq. (13).
    Update model parameters using $\mathcal{L}_{\text{prob}}$, $\mathcal{L}_{\text{align}}$, and $\mathcal{L}_{\text{str}}$.
  end for
end for

Algorithm 2: Zero-Shot Inference Algorithm of ProTAGAD

Input: Target graphs $\mathcal{T}_{\text{test}}$; trained model; textual prototypes $\mathbf{P}^t$; topological prototypes $\mathbf{P}^s$.
Parameters: Well-trained ProTAGAD model weight parameters.

for each graph $\mathcal{G} \in \mathcal{T}_{\text{test}}$ **do**
  Derive textual representations $\mathbf{X}^t$ and anomaly probabilities $\widehat{\mathbf{Y}}$ via Eqs. (2) and (3).
  Compute the textual anomaly score $S^t(v_i)$ via Eq. (9).
  Derive node representations $\mathbf{H}$ and $\mathbf{H}'$ via Eqs. (10) and (11).
  Compute the topological anomaly score $S^s(v_i)$ via Eq. (14).
  Compute the final anomaly score $S(v_i)$ via Eq. (15).
end for
return $\{S(v_i)\}$.