

# Correcting False Alarms from Unseen: Adapting Graph Anomaly Detectors at Test Time

Junjun Pan<sup>1</sup>, Yixin Liu<sup>1</sup>, Chuan Zhou<sup>2</sup>, Fei Xiong<sup>3</sup>, Alan Wee-Chung Liew<sup>1</sup>, Shirui Pan<sup>1\*</sup>

<sup>1</sup>School of Information and Communication Technology, Griffith University, Queensland, Australia

<sup>2</sup>Academy of Mathematics and Systems Science,

<sup>3</sup>School of Electronic and Information Engineeri  
{junjun.pan, yixin.liu, a.liew, s.pan}@griffith.edu.a

## Abstract

Graph anomaly detection (GAD), which aims to detect outliers in graph-structured data, has received increasing research attention recently. However, existing GAD methods assume identical training and testing distributions, which is rarely valid in practice. In real-world scenarios, unseen but normal samples may emerge during deployment, leading to a **normality shift** that degrades the performance of GAD models trained on the original data. Through empirical analysis, we reveal that the degradation arises from (1) semantic confusion, where unseen normal samples are misinterpreted as anomalies due to their novel patterns, and (2) aggregation contamination, where the representations of seen normal nodes are distorted by unseen normals through message aggregation. While retraining or fine-tuning GAD models could be a potential solution to the above challenges, the high cost of model retraining and the difficulty of obtaining labeled data often render this approach impractical in real-world applications. To bridge the gap, we proposed a lightweight and plug-and-play **Test-time adaptation** framework for correcting **Unseen Normal patterns (TUNE)** in GAD. To address semantic confusion, a graph aligner is employed to align the shifted data to the original one at the graph attribute level. Moreover, we utilize the minimization of representation-level shift as a supervision signal to train the aligner, which leverages the estimated aggregation contamination as a key indicator of normality shift. Extensive experiments on 10 real-world datasets demonstrate that TUNE significantly enhances the generalizability of pre-trained GAD models to both synthetic and real unseen normal patterns.

**Code** — <https://github.com/CampanulaBells/TUNE>

## Introduction

Graph anomaly detection (GAD) aims to identify abnormal entities that deviate from the dominant patterns in a graph (Ma et al. 2021). It has gained significant attention due to its critical role in safeguarding a wide range of graph-based applications, including social networks (Bian et al. 2020), online services (Wang and Zhu 2022), and financial transaction networks (Li et al. 2022). In recent years, graph neural networks (GNNs) (Liu et al. 2025; Li et al. 2026,

\*Corresponding Author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

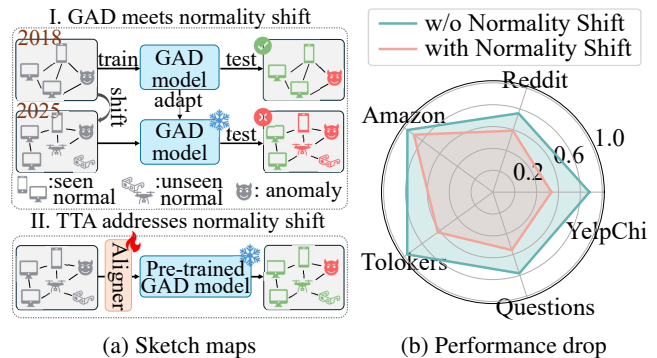


Figure 1: (a) Sketch maps of conventional GAD methods under normality shift and our solution. (b) Performance drop of BWGNN under data with normality shift.

2024; Shen et al. 2025) have gained popularity in GAD due to their strong capabilities in capturing complex graph anomaly patterns. While several unsupervised GNN-based GAD methods can bypass the costly process of obtaining labeled anomalies (Liu et al. 2021; Ding et al. 2019; Pan et al. 2023), mainstream approaches still predominantly focus on supervised paradigms, which often yield higher detection accuracy in real-world GAD scenarios, such as fraud detection, intrusion detection, and misinformation spotting (Tang et al. 2022; Feng et al. 2025; Chen et al. 2024).

Despite the promising performance of existing GAD methods, their reliance on the assumption of identical training and testing data distributions may limit their performance in real-world scenarios, where new categories of normal entities can emerge and change the data distribution (Baitieva et al. 2025; Tan et al. 2025; Chen et al. 2025; Li et al. 2025). Taking Figure 1a-I as an example, in a product recommendation graph for electronics, the training nodes may previously consist of traditional devices such as laptops and smartphones, while at test time, new product categories like drones or smartglasses may emerge and become prevalent. We refer to this phenomenon as **normality shift** (Han et al. 2023), where the definition of “normal” evolves due to the emergence of previously unseen but benign categories. Given its ubiquity in real-world applications, this shift raises a fundamental question: *Can GAD*

*models pre-trained on the original normality distribution effectively adapt to new and evolving definitions of normality?*

To answer this question, we first conduct a preliminary experiment to examine how GAD models pre-trained on original data perform when directly adapted to data with unseen normal samples. As shown in Figure 1b, the normality shift leads to a substantial performance drop on all datasets, highlighting the limitation of the generalizability of GAD models. To better understand the causes behind such performance degradation, we conduct a series of empirical analyses to investigate how unseen normal samples affect pre-trained GAD models. Our studies reveal a two-fold cause: *on the one hand*, due to the unfamiliar patterns of the novel normal categories, the model suffers from **semantic confusion** between unseen normal samples and anomalies, leading to incorrect predictions on unseen normals; *on the other hand*, the message propagation mechanism introduces **aggregation contamination**, where representations of seen normals are distorted by neighboring unseen normals, further degrading the model to distinguish known normal patterns. To sum up, the misclassification of both seen and unseen normal samples degrades the overall performance of GAD models under normality shift.

To address the above issues, a straightforward approach is to perform supervised fine-tuning or retrain the GAD models on the shifted data. Unfortunately, in real-world applications where data continuously evolves, frequently retraining the model from scratch is often impractical (Ding et al. 2021). Furthermore, identifying and labeling unseen normal categories is both costly and time-consuming, preventing us from fine-tuning GAD models on shifted data. To overcome these barriers, a more feasible solution is to leverage unsupervised test-time adaptation (TTA) techniques to adjust the shifted data for better alignment with the pre-trained GAD models through unsupervised objectives. Nevertheless, existing graph TTA methods (Ju et al. 2023; Jin et al. 2023; Mao et al. 2024) are typically designed for node classification tasks and rely on several assumptions, such as graph homophily or balanced label distributions, to guide the adaptation. These assumptions do not hold in GAD settings, making it suboptimal to directly apply such methods. Moreover, these methods overlook the semantic confusion and aggregation contamination issues inherent in GAD with normality shift, thereby limiting their adaptation effectiveness.

To fill the gap, we introduce a **TTA** framework for correcting **Unseen Normal patterns** (TUNE) in GAD, a plug-and-play lightweight method that can be integrated with diverse GAD model architectures without requiring any model retraining or fine-tuning. To deal with semantic confusion, we employ a graph aligner to correct the normality shift at the graph attribute level, which corrects the semantic deviation while preserving integration flexibility. For the aggregation contamination problem, rather than directly addressing it, we in turn leverage it to construct the supervision signals for unsupervised TTA. Specifically, TUNE employs an aggregation-free dual branch to simulate the undistorted node representations, and utilizes the divergence between the aggregation-free and aggregation-based branches to indicate the degree of aggregation contamination. Through min-

imizing this divergence, the framework optimizes the graph aligner, thereby reducing the impact of normality shift and hence enabling unsupervised TTA for GAD models. In summary, this paper makes the following contributions:

**Problem.** We identify the challenge of normality shift in real-world GAD scenarios and conduct comprehensive analyses to uncover the underlying difficulties, namely semantic confusion and aggregation contamination.

**Methodology.** We introduce TUNE, a novel unsupervised TTA approach that can be integrated with any pre-trained GAD model in a plug-and-play manner to mitigate the impact of normality shift in GAD.

**Experiments.** We perform extensive experiments on 10 benchmark datasets, and the results demonstrate the effectiveness and generalizability of TUNE.

## Related Work

In this section, we briefly review two related research directions, i.e., graph anomaly detection (GAD) and test-time adaptation (TTA). A more comprehensive literature review is available in Appendix A.

**Graph Anomaly Detection (GAD)** aims to identify outliers in graph-structured data. Unsupervised GAD methods typically identify anomalies via unsupervised learning techniques (Ding et al. 2019; Liu et al. 2021, 2023; Cai et al. 2025; Wu et al. 2025a; Zhao et al. 2025), while supervised GAD methods utilize annotations to build a more reliable anomaly classifier-based anomaly detector (Dou et al. 2020; Liu et al. 2024). Existing methods either explicitly suppress it with carefully designed model architectures (Gao et al. 2023; Shi et al. 2022; Zhuo et al. 2024; Yu et al. 2025; Wu et al. 2025b), or adopt spectral GNNs that are less sensitive to the local heterophilic connections (Tang et al. 2022; Gao et al. 2023; Xu et al. 2024; Pan et al. 2025a). Despite their performance, these methods assume a shared distribution between training and testing, limiting performance under normality shift (Kim, Park, and Choo 2024).

While recent works on open-set GAD (Wang et al. 2023; Cai et al. 2025) or cross-domain GAD (Ding et al. 2021) focus on enhancing generalization to unseen domains (Pan et al. 2025b), they require additional modules and training losses during the pre-training phase. This limits their effectiveness in adapting GAD models to normality shift.

**Graph Test Time Adaptation (TTA)** enables a pre-trained node classifier to adapt to distribution shift in unseen testing graphs, without fine-tuning the model itself. GTrans (Jin et al. 2023) pioneers the use of a data-centric graph TTA that can be seamlessly integrated with any model architecture, which is trained with a contrastive learning paradigm (Deng et al. 2025; Fu et al. 2025). Similarly, GraphPatcher (Ju et al. 2023) mitigates degree bias by iteratively generating virtual nodes to repair corrupted low-degree graphs. With the assumptions on graph homophily and label balance, SOGA (Mao et al. 2024) directly fine-tunes the pre-trained model by enforcing structure consistency and maximizing mutual information between the input and output. Despite their promising results in classification tasks, they are not directly applicable to GAD due to challenges such as strong label imbalance and the camouflage effect (Dou et al. 2020).

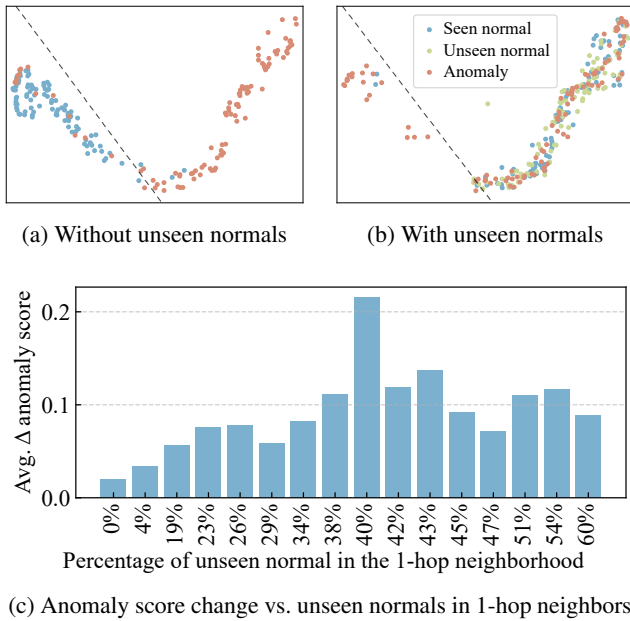


Figure 2: Motivate experiments on BWGNN.

### Problem Definition

**Notations.** Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$  denote an attributed graph with  $n$  nodes and  $m$  edges, where  $\mathcal{V} = \{v_1, \dots, v_n\}$  is the node set,  $\mathcal{E}$  is the edge set, and  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  is the node feature matrix, with  $\mathbf{x}_i \in \mathbb{R}^d$  representing the feature vector of node  $v_i$ . The adjacency matrix is denoted by  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , where  $\mathbf{A}_{ij} = 1$  if  $(v_i, v_j) \in \mathcal{E}$  and 0 otherwise.

**Conventional GAD Problem.** GAD aims to classify anomalous nodes  $\mathcal{V}_a$  from normal nodes  $\mathcal{V}_n = \mathcal{V} \setminus \mathcal{V}_a$ . This is achieved by learning an anomaly scoring function  $f: \mathcal{V} \rightarrow \mathbb{R}$  that predicts anomaly score  $s_i$  for each node  $v_i$ , where a higher score indicates a higher likelihood of being anomalous. In the conventional setting, it is assumed that the training graph  $\mathcal{G}_{\text{train}}$  and the test graph  $\mathcal{G}_{\text{test}}$  are drawn from the same underlying distribution, i.e.,  $\mathcal{G}_{\text{train}} \sim \mathbb{P}$  and  $\mathcal{G}_{\text{test}} \sim \mathbb{P}$ . This assumption enables the scoring function  $f$  learned on  $\mathcal{G}_{\text{train}}$  to generalize effectively to  $\mathcal{G}_{\text{test}}$ .

**GAD with Test-Time Normality Shift.** Formally, let the training and testing graphs be denoted as  $\mathcal{G}_{\text{train}}$  and  $\mathcal{G}_{\text{test}}$  respectively. The normality shift problem in GAD can be defined as the scenario where the underlying distribution  $\mathbb{P}$  is non-stationary between training and testing (Kim, Park, and Choo 2024), i.e.,  $\mathbb{P}_{\text{train}} \neq \mathbb{P}_{\text{test}}$ , due to the emergence of previously unseen normal nodes  $\mathcal{V}_{\text{un}} \subset \mathcal{V}_{\text{test}}$ . These unseen normal nodes can introduce novel and semantically valid patterns into the test graph that were absent during training, resulting in a distribution shift that leads existing GAD models to mistakenly assign them a high anomaly score, as illustrated in Figure 1b.

### Motivation and Analysis

Normality shift frequently occurs in real-world applications such as e-commerce, driven by the continuous emergence of

new products. In order to understand how normality shift affects GAD models, in this section, we expose the impact of normality shifts on the effectiveness of GAD models pre-trained on original data. Specifically, we first examine the generalization capability of pre-trained GAD models on unseen normal samples and find that **semantic confusion** between unseen normals and anomalies is a key factor that leads to generalization failure. Furthermore, we observe that the presence of unseen normal instances also affects the identification of previously seen normal data, caused by the **aggregation contamination** phenomenon. We provide empirical analysis and discussion to explain these issues.

### Semantic Confusion of Unseen Normals

In existing GAD methods, a common assumption is that the training and testing sets are drawn from the same distribution. Yet in real-world scenarios, new but benign node categories often emerge and follow valid behavioral patterns. Although these instances may appear novel to the model, they should not be treated as anomalies. This raises a key question that we seek to explore: *Can GAD methods effectively generalize to unseen normal categories?* To answer this question, we evaluate the generalizability of BWGNN (Tang et al. 2022) on five real-world GAD datasets to assess how the pre-trained model is affected by unseen normal categories introduced in the test split. As shown in Figure 1b, we observe a consistent and significant performance drop on datasets from different domains, showing that the presence of unseen normal classes severely reduces performance.

**Analysis:** From the perspective of unseen normal samples, we attribute the performance degradation to the **semantic confusion** between new normals and anomalies. Specifically, GAD models are typically trained to assign low anomaly scores to seen normal instances and high scores to anomalies, based on the observed training distribution. However, due to the inherent sparsity of anomalies and the limited diversity of normal categories, the learned pattern of normality tends to be overly specific. Consequently, when novel but benign categories emerge during testing, their unfamiliar patterns often lead the model to assign inflated anomaly scores. This indicates that pre-trained GAD models tend to overfit to seen normal patterns and struggle to generalize to unseen ones.

**Empirical Discussion:** To investigate how unseen normal samples are misclassified by the model, we visualize the learned representations and the corresponding decision boundary on Tolokers dataset (Platonov et al. 2023), comparing the model behavior without (Figure 2a) and with (Figure 2b) unseen normal samples. In Figure 2a, we observe that the seen normal samples (in blue) and anomalies (in red) are well-separated, indicating the reliable performance of the GAD model on data without normality shift. Nevertheless, in Figure 2b, the representations of unseen normal nodes (in green) are found to overlap significantly with those of anomalies, revealing that the model struggles to differentiate novel normal patterns from truly abnormal ones. The findings highlight that the semantic confusion between unseen normals and anomalies severely impairs the discrimination ability of GAD models under normality shift.

## Aggregation Contamination on Seen Normals

According to Figure 2b, we not only observe semantic confusion between the unseen normal samples and anomalies, but also an unexpected effect: the representations of seen normal nodes (in blue), which were previously well separated from the anomalies in Figure 2a, also undergo a noticeable shift. Such a shift, in turn, degrades the model performance in distinguishing anomalies from previously learned normal patterns. This effect brings forth a puzzling question: *Since the attributes of seen normal samples usually remain unchanged, why do their representations shift in the presence of unseen normal samples?*

**Analysis:** We suggest that this effect stems from **aggregation contamination**, where the representations of seen normal nodes are distorted due to the presence of unseen normals. This effect arises from the message aggregation mechanism in graph neural networks, where a node updates its representation by integrating information from its neighbors (Wu et al. 2020). Consequently, unseen normals can influence nearby seen normals through aggregation, causing their representations to deviate from expected normal patterns. Despite several GAD methods employing techniques like spectral GNN (Tang et al. 2022) or weighted aggregation (Shi et al. 2022) to address similar issues caused by anomalies during message aggregation, these approaches fall short when dealing with unseen normals. A key distinction is that unseen normals emerge only at test time, leaving the model unable to learn how to suppress their influence. Moreover, unlike anomalies that often exhibit distinct or extreme behaviors, unseen normals tend to follow partially familiar patterns, making their interference more subtle and thus harder to identify and mitigate.

**Empirical Discussion:** To assess the aggregation contamination effect, we examine how the presence of unseen normals alters the anomaly scores of seen normal nodes. Specifically, we group nodes based on the proportion of unseen normals in their 1-hop neighborhoods and compute the average change in anomaly score before and after introducing these unseen nodes. As shown in Figure 2c, the presence of unseen normals consistently leads to increased anomaly scores for seen normal nodes. Moreover, the magnitude of this increase grows with a higher proportion of novel normal neighbors, supporting our hypothesis that message aggregation amplifies the effect. Notably, even nodes with no unseen normals in their 1-hop neighborhoods experience elevated anomaly scores, suggesting that the influence can propagate through multi-hop interactions. This confirms that aggregation from unseen normals can distort the representation of seen normals, leading to incorrect abnormality prediction.

## Methodology

Our empirical study demonstrates that normality shift hinders GAD performance in real-world scenarios by causing both *semantic confusion of unseen normals* and *aggregation contamination on seen normals*. To address these challenges while minimizing deployment costs, we propose TUNE, an unsupervised test-time adaptation framework that can be seamlessly integrated with various pre-trained GAD models

to enable generalization to unseen normals, without requiring supervised fine-tuning. As illustrated in Figure 3, TUNE employs a graph aligner to transform node features, alleviating the semantic confusion phenomenon by reducing the distribution shift of unseen normals. To train the aligner without labels, we design an alignment loss that leverages aggregation contamination as an indicator of normality shift, and then minimize the discrepancy between aggregated and uncontaminated node representations.

## Feature Transformation-based Graph Aligner

To overcome the semantic confusion caused by normality shift in real-world applications, test-time adaptation offers a promising solution by aligning the feature distributions of unseen normal samples with those observed during training, while preserving the GAD models pre-trained on original data unchanged. Nevertheless, the diverse architectures of pre-trained GAD models make it challenging to design adaptation strategies that generalize across different GAD architecture designs. To this end, we introduce a simple yet effective graph aligner that enables data-centric test-time adaptation based on feature transformation. This design allows TUNE to be compatible with diverse GAD models without requiring supervised fine-tuning or architectural modifications.

In essence, normality shift arises because unseen normal nodes exhibit attribute patterns that deviate from those encountered during training. To address this, we project the features of unseen normals toward the distribution of seen normals, which realizes a model-agnostic adaptation. Specifically, we model this deviation as a learnable residual and train a multilayer perceptron (MLP) to estimate it based on the node feature matrix  $\mathbf{X}$ . The estimated shift  $\Delta\mathbf{X}$  is then applied to produce the aligned features  $\mathbf{X}'$ :

$$\mathbf{X}' = f_{\text{align}}(\mathbf{X}; \theta) = \mathbf{X} + \Delta\mathbf{X}, \quad \Delta\mathbf{X} = \text{MLP}_{\theta}(\mathbf{X}). \quad (1)$$

The aligned features are subsequently passed into the pre-trained GAD model to produce anomaly scores  $\mathbf{s}_{\text{gnn}}$ :

$$\mathbf{H} = f_{\text{enc}}(\mathbf{A}, \mathbf{X}'), \quad \mathbf{s}_{\text{test}} = f_{\text{det}}(\mathbf{H}), \quad (2)$$

where  $f_{\text{enc}}(\cdot)$  and  $f_{\text{det}}(\cdot)$  denote the encoder and detector components of the pre-trained GAD model, respectively.

To optimize the parameters  $\theta$  of the graph aligner, an ideal learning objective is to maximize the supervised detection accuracy on the aligned test graph:

$$\underset{\theta}{\text{argmin}} \mathcal{L}(\mathbf{s}_{\text{test}}, \mathbf{y}_{\text{test}}), \quad (3)$$

where  $\mathcal{L}(\cdot, \cdot)$  denotes the supervised anomaly detection loss. This formulation guides the aligner to reduce the impact of normality shift by minimizing its negative impacts on downstream GAD performance, which can handle the semantic confusion issue. However, since the test-time ground-truth labels  $\mathbf{y}_{\text{test}}$  are usually hard to acquire in the context of GAD, directly optimizing for detection performance is infeasible. Motivated by our empirical findings, we instead leverage aggregation contamination as an implicit indicator to estimate and mitigate the influence of normality shift in an unsupervised setting.

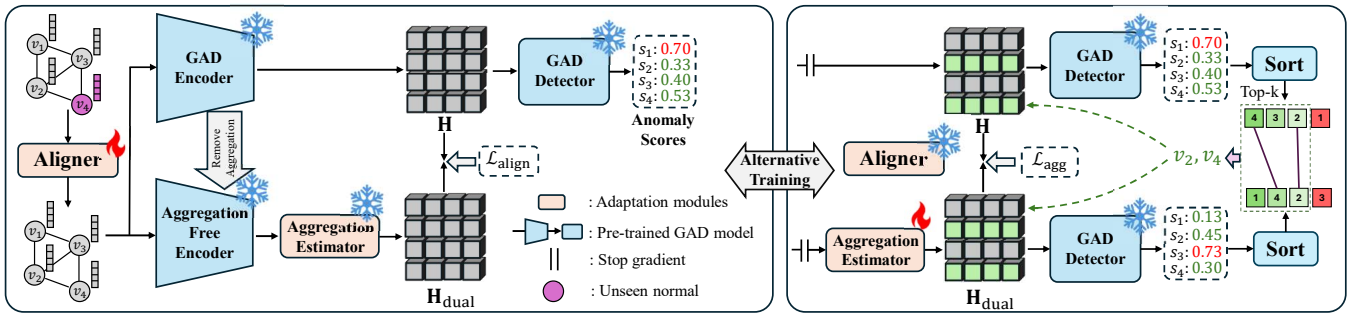


Figure 3: Overall framework of TUNE. TUNE addresses normality shift by leveraging a graph aligner and a dual-branch architecture. It captures aggregation contamination caused by unseen normals by measuring the discrepancy between representations from a main branch and an aggregation-free auxiliary branch. To ensure that the auxiliary branch provides contamination-free representations, an aggregation estimator is jointly trained with the aligner in an alternating manner, using high-confidence normal nodes.

### Aggregation Contamination-Guided Alignment

Our empirical analysis demonstrates that unseen normals can distort the embedding of seen normal nodes through message aggregation, leading to elevated anomaly scores. If we could obtain representations that are not affected by such contamination, the normality shift could be estimated by comparing the contaminated representations with their uncontaminated counterparts, thereby guiding the optimization of the graph aligner. In other words, we can optimize the aligner by minimizing the representation shift caused by contaminated aggregation, which uses the contamination itself as feedback to counteract its effect.

To obtain contamination-free representations without label supervision, we construct an aggregation-free auxiliary branch from the GNN backbone in GAD models, where message aggregation is explicitly removed. In this way, we thereby isolate node representations from the influence of unseen normals. Specifically, the auxiliary branch replicates the encoder of the pre-trained GAD model, but removes message passing and aggregation, yielding an encoder  $f_{\text{dual}}(\cdot)$  that generates node representations solely from ego features. This adjustment can be applied to a wide range of GNNs by replacing their aggregation operators with identity matrices. To further compensate for the absence of contextual signals, we incorporate an aggregation estimator  $g(\cdot)$  that reconstructs aggregated embeddings  $\mathbf{H}_{\text{dual}}$  using only ego representations, i.e.:

$$\mathbf{H}_{\text{dual}} = g(f_{\text{dual}}(\mathbf{X}')). \quad (4)$$

Assuming the estimator  $g(\cdot)$  can accurately approximate clean embeddings, the divergence between the outputs of the main branch and auxiliary branch reflects the impact of aggregation contamination. This signal is then used to optimize the graph aligner via an alignment loss:

$$\mathcal{L}_{\text{align}} = \text{KLD}(\mathbf{H}|\mathbf{H}_{\text{dual}}), \quad (5)$$

where  $\mathbf{H}$  denotes the contaminated embedding from the main branch, and KLD is the Kullback–Leibler divergence. Minimizing this loss encourages the aligner to reduce the influence of normality shift in the input features, providing

an effective unsupervised adaptation strategy across GAD architectures to handle shifted normal patterns.

### Training Aggregation Estimator

To reduce the influence of aggregation contamination on seen normals, our auxiliary branch encodes each node using only its ego features. However, while this eliminates contamination from unseen normals, it also discards potentially informative signals from clean ones. To compensate for the missing information, we introduce an aggregation estimator  $g(\cdot)$  to reconstruct the aggregated representations from the aggregation-free ones.

To avoid reintroducing contamination during reconstruction, the estimator is trained using a carefully selected set of clean nodes with high-confidence normality. As illustrated in right of Figure 3, we obtain anomaly scores from two branches,  $\mathbf{s}$  and  $\mathbf{s}_{\text{dual}}$ , by feeding  $\mathbf{H}$  and  $\mathbf{H}_{\text{dual}}$  into the GAD detector. We then select nodes that rank in the top- $k$  percentile of normal confidence under both scores as training samples. A high confidence score under the auxiliary branch suggests the ego features of a node are normal, while a high score under the main branch implies its normality is not changed due to unseen normals, together indicating a clean supervision signal. The corresponding embeddings,  $\mathbf{H}^{\text{top-}k}$  and  $\mathbf{H}_{\text{dual}}^{\text{top-}k}$ , are used to optimize the aggregation estimator  $g(\cdot)$  via another KL-divergence loss:

$$\mathcal{L}_{\text{agg}} = \text{KLD}(\mathbf{H}^{\text{top-}k}|\mathbf{H}_{\text{dual}}^{\text{top-}k}), \quad (6)$$

where  $g(\cdot)$  is implemented as a linear transformation. Note that  $\mathcal{L}_{\text{agg}}$  is only used to update the aggregation estimator rather than the graph aligner. By supervising the estimator with high-confidence normal nodes, we ensure the reconstructed representations can be sufficient to approximate contamination-free aggregation, thereby improving the detection of normality shift.

### Overall Algorithm

There are two learnable modules in the TUNE pipeline, graph aligner  $f_{\text{align}}(\cdot)$  and aggregation estimator  $g(\cdot)$ . The optimization of these two modules is tightly coupled: while

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

Table 1: Performance (AUROC in %) with standard deviation of all models on GFD datasets. **Bold** indicates the best overall performance across all models, while underline highlights the best result among TTA variants for each base model.

$g(\cdot)$  plays a critical role in the optimization objective of  $f_{\text{align}}(\cdot)$  (i.e., Eq. (5)),  $f_{\text{align}}(\cdot)$  also serves as the feature transformation input to  $g(\cdot)$ . In this case, to ensure the stable optimization of TUNE, we adopt an **alternative training** strategy. Concretely, we first fix the parameters of the aggregation estimator  $g(\cdot)$  and optimize the graph aligner  $f_{\text{align}}(\cdot)$  using the alignment loss in Eq. (5) (left part of Figure 3). Then, with  $f_{\text{align}}(\cdot)$  frozen, we update  $g(\cdot)$  under the supervision of Eq. (6) (right part of Figure 3). These two steps are alternated iteratively until the training converges. The alternative training strategy results in an overall complexity of  $\mathcal{O}(n + m)$ . The algorithmic description and detailed complexity analysis are provided in Appendix B.

## Experiments

### Experimental Settings

**Datasets.** We conduct experiments on eight public GAD datasets from different domains, including Amazon (McAuley and Leskovec 2013), YelpChi (Rayana and Akoglu 2015), Reddit (Kumar, Zhang, and Leskovec 2019), Weibo (Kumar, Zhang, and Leskovec 2019), Tolokers (Platonov et al. 2023), Questions (Platonov et al. 2023), T-Finance (Tang et al. 2022), and T-Social (Tang et al. 2022). Since these datasets only contain binary labels and do not explicitly define subtypes within the normal class, we apply clustering on the features of normal nodes and select the smallest cluster as unseen normals. To simulate a normality shift, the unseen normals are excluded from the training and validation sets. To further evaluate generalizability against natural unseen normals, we include two imbalanced node

classification datasets, Photo and Computers (Shchur et al. 2018). We treat classes with fewer than 5% of the total nodes as anomalies, while the largest normal class is utilized as the unseen normal. More details are given in Appendix C.

**Baseline and Evaluation Metrics.** We compare our approach against five state-of-the-art supervised GAD methods: CareGNN (Dou et al. 2020), H2-FDetector (Shi et al. 2022), GAGA (Xu et al. 2024), PMP (Zhuo et al. 2024), and ConsisGAD (Yu et al. 2025). Moreover, we compare TUNE with three graph TTA baselines, including GTrans (Jin et al. 2023), GraphPatcher (Ju et al. 2023), and SOGA (Mao et al. 2024). Furthermore, we evaluate all TTA methods alongside TUNE with three popular GNN/GAD methods as backbones, including GCN (Kipf and Welling 2017), BWGNN (Tang et al. 2022), and GHRN (Gao et al. 2023). To simplify the implication, we adapt the implementation provided by GADbench (Tang et al. 2023). We utilize the suggested hyperparameters in their paper for all baselines. Similar to previous works (Tang et al. 2023), we employ AUROC and AUPRC (results see Appendix E) as evaluation metrics. We run each experiment 5 times and report the average and standard deviation. More experimental details are given in Appendix D.

### Experimental Results

**Performance Comparison.** The comparison results on eight real-world GFD datasets with constructed unseen normals are reported in Table 1. From the table, we can see that: ① TUNE consistently and significantly improves performance on the three GNN backbones across all datasets, and outperforms SOTA methods in most datasets except

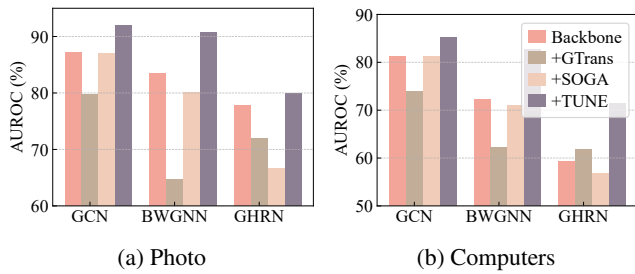


Figure 4: Performance on Photo and Computers datasets.

Model	Tolokers	Questions	Weibo
GCN	63.20±0.26	51.58±0.44	86.30±14.06
+ Ours	<b>66.27±0.07</b>	<b>60.61±0.97</b>	96.16±0.01
- Agg. Estimator	66.18±0.01	42.95±0.01	96.16±0.01
+ Affinity Loss	59.94±0.01	54.51±0.01	<b>96.17±0.01</b>
BWGNN	62.02±0.87	50.40±0.45	93.04±1.10
+ Ours	<b>65.51±0.04</b>	<b>63.21±0.37</b>	<b>95.26±0.49</b>
- Agg. Estimator	63.74±0.01	62.65±0.01	95.01±0.04
+ Affinity Loss	58.31±0.01	51.16±0.02	85.75±0.71
GHRN	60.34±0.59	49.65±0.82	90.34±2.48
+ Ours	<b>66.17±0.42</b>	<b>58.88±0.39</b>	92.31±0.01
- Agg. Estimator	64.54±0.01	58.05±0.01	92.28±0.01
+ Affinity Loss	56.71±0.02	55.24±0.06	<b>92.32±0.01</b>

Table 2: Performance (AUROC in %) with standard deviation of TUNE and its variants.

on Tolokers. These results demonstrate the effectiveness of TUNE in diverse real-world GFD datasets. While existing Graph TTA methods (GTrans, GraphPatcher, and SOGA) exhibit solid performance on classification tasks, they failed to achieve positive transfer in GAD with normality shift. This is mainly due to the unique challenges in GAD, such as severe label imbalance and heterophilic structures, which violate the core assumptions of these methods (Mao et al. 2024; Feng and Dai 2023). The scalability of TTA methods is hindered by out-of-memory (OOM) issues on large-scale real-world GFD datasets such as T-Finance and T-Social. In contrast, TUNE demonstrates better scalability, underscoring its effectiveness in handling large-scale GAD applications like e-commerce and social networks.

**Real-World Unseen Normals.** To evaluate the effectiveness of our method under normality shift introduced by real-world unseen normal categories, we test it on two imbalanced classification datasets: Photo and Computer. As shown in Figure 4, while other Graph TTA methods struggle to achieve positive transfer, our method consistently improves performance across both datasets and all three backbones. These promising results highlight not only the effectiveness of TUNE in real-world scenarios, but also underscore its versatility across various GAD architectures.

**Ablation Study.** To verify the contribution of key design in TUNE, we conduct ablation studies on the feature adapter and the alignment loss. To verify the effectiveness and robustness of our aggregation estimator module, we remove it in the variant (- Agg. Estimator). As shown in Table 2,

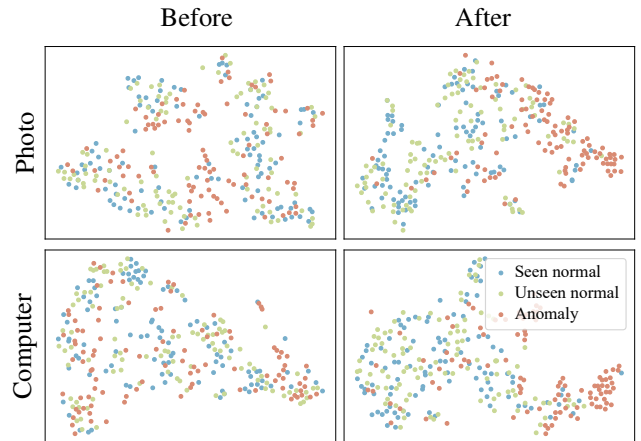


Figure 5: t-SNE visualizations of node representations by BWGNN (before and after applying TUNE) on two datasets.

while the variant still achieves positive transfer on most datasets, its performance consistently drops compared to TUNE, highlighting the effectiveness of both the adapter module and dual branch design. Moreover, we further investigate the alignment loss by replacing it with an unsupervised GAD loss (+ Affinity Loss) as proposed in (Qiao and Pang 2023). As Table 2 demonstrated, this variant leads to a performance reduction on all datasets except Weibo, where homophily dominates and GAD loss has limited additional benefit. In contrast, our dual-branch alignment loss does not rely on the homophily assumption of the graph, making it more suitable to mitigate the normality shift in GAD.

**Case Study.** To better understand why our proposed TUNE is more effective, we visualize the representations before and after adaptation. As shown in Figure 5, before adaptation, anomaly representations are indistinguishable from those of seen and unseen normals. This issue occurs because the pre-trained GAD model fails to generalize to unseen normals, and their presence also affects the representations of seen normals through aggregation. With the help of TUNE, representations become much more discriminative, with anomalies clearly separated from all types of normal nodes. This is because our aligner is trained to mitigate the normality shift from the data level, helping the model preserve embedding quality and separability in the presence of unseen normals.

## Conclusion

In this paper, we propose a novel TTA framework, TUNE, that can adapt to various GAD models to generalize to novel normal categories unseen during training, thereby guarding real-world graph applications under evolving normal definition. By utilizing the aggregation contamination as an indicator of normality shift, TUNE optimizes a graph aligner that operates at the graph level, making it compatible with a wide range of model architectures. Comprehensive empirical results show the effectiveness of TUNE in improving the generalizability of pre-trained GAD models compared to other graph TTA methods on 10 graph datasets against both artificial and real-world unseen normals.

## Acknowledgments

The work of S. Pan was partially supported by the Australian Research Council (ARC) under Grant Nos. DP240101547 and FT210100097. The work of C. Zhou was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62472416.

## References

- Baitieva, A.; Bouaouini, Y.; Briot, A.; Ameln, D.; Khalfouli, S.; and Akcay, S. 2025. Beyond Academic Benchmarks: Critical Analysis and Best Practices for Visual Industrial Anomaly Detection. *arXiv preprint arXiv:2503.23451*.
- Bian, T.; Xiao, X.; Xu, T.; Zhao, P.; Huang, W.; Rong, Y.; and Huang, J. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 549–556.
- Cai, J.; Zhang, Y.; Liu, F.; and Ng, S.-K. 2025. Leveraging Diffusion Model as Pseudo-Anomalous Graph Generator for Graph-Level Anomaly Detection. In *Forty-second International Conference on Machine Learning*.
- Chen, N.; Liu, Z.; Hooi, B.; He, B.; Fathony, R.; Hu, J.; and Chen, J. 2024. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *The Twelfth International Conference on Learning Representations*.
- Chen, Q.; Li, S.; Liu, Y.; Pan, S.; Webb, G. I.; and Zhang, S. 2025. Uncertainty-Aware Graph Neural Networks: A Multi-hop Evidence Fusion Approach. *IEEE Transactions on Neural Networks and Learning Systems*.
- Deng, B.; Wang, T.; Fu, L.; Huang, S.; Chen, C.; and Zhang, T. 2025. THESAURUS: Contrastive Graph Clustering by Swapping Fused Gromov-Wasserstein Couplings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 16199–16207.
- 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.
- Ding, K.; Shu, K.; Shan, X.; Li, J.; and Liu, H. 2021. Cross-domain graph anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6): 2406–2415.
- Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; and Yu, P. S. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 315–324.
- Feng, X.; and Dai, M. 2023. Leveraging Node Attributes for Link Prediction via Meta-path Based Proximity. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8.
- Feng, X.; Zhang, H.; Zhang, Y.; Zhang, L. Y.; and Pan, S. 2025. BiMark: Unbiased Multilayer Watermarking for Large Language Models. In *Forty-second International Conference on Machine Learning*.
- Fu, L.; Deng, B.; Huang, S.; Liao, T.; Pan, S.; and Chen, C. 2025. Less is More: Federated Graph Learning with Alleviating Topology Heterogeneity from A Causal Perspective. In *Forty-second International Conference on Machine Learning*.
- 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.
- Han, D.; Wang, Z.; Chen, W.; Wang, K.; Yu, R.; Wang, S.; Zhang, H.; Wang, Z.; Jin, M.; Yang, J.; et al. 2023. Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation. In *NDSS*.
- Jin, W.; Zhao, T.; Ding, J.; Liu, Y.; Tang, J.; and Shah, N. 2023. Empowering Graph Representation Learning with Test-Time Graph Transformation. In *The Eleventh International Conference on Learning Representations*.
- Ju, M.; Zhao, T.; Yu, W.; Shah, N.; and Ye, Y. 2023. Graph-Patcher: Mitigating Degree Bias for Graph Neural Networks via Test-time Augmentation. *Advances in neural information processing systems*.
- Kim, D.; Park, S.; and Choo, J. 2024. When model meets new normals: test-time adaptation for unsupervised time-series anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 13113–13121.
- Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.
- Kumar, S.; Zhang, X.; and Leskovec, J. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1269–1278.
- Li, J.; Li, Y.; Fu, Y.; Liu, J.; Liu, Y.; Yang, M.; and King, I. 2025. CLIP-Powered Domain Generalization and Domain Adaptation: A Comprehensive Survey. *arXiv preprint arXiv:2504.14280*.
- Li, R.; Liu, Z.; Ma, Y.; Yang, D.; and Sun, S. 2022. Internet financial fraud detection based on graph learning. *IEEE Transactions on Computational Social Systems*, 10(3): 1394–1401.
- Li, S.; Liu, Y.; Chen, Q.; Webb, G. I.; and Pan, S. 2024. Noise-resilient unsupervised graph representation learning via multi-hop feature quality estimation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 1255–1265.
- Li, S.; Liu, Y.; Wen, Q.; Zhang, C.; and Pan, S. 2026. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Liu, Y.; Ding, K.; Lu, Q.; Li, F.; Zhang, L. Y.; and Pan, S. 2023. Towards self-interpretable graph-level anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36, 8975–8987.

- Liu, Y.; Li, S.; Zheng, Y.; Chen, Q.; Zhang, C.; and Pan, S. 2024. Arc: A generalist graph anomaly detector with in-context learning. In *Advances in Neural Information Processing Systems*, volume 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, Y.; Zhang, G.; Wang, K.; Li, S.; and Pan, S. 2025. Graph-augmented large language model agents: Current progress and future prospects. *IEEE Intelligent Systems*.
- 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.
- Mao, H.; Du, L.; Zheng, Y.; Fu, Q.; Li, Z.; Chen, X.; Han, S.; and Zhang, D. 2024. Source free graph unsupervised domain adaptation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 520–528.
- McAuley, J. J.; and Leskovec, J. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, 897–908.
- Pan, J.; Liu, Y.; Zheng, X.; Zheng, Y.; Liew, A. W.-C.; Li, F.; and Pan, S. 2025a. A Label-Free Heterophily-Guided Approach for Unsupervised Graph Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12443–12451.
- Pan, J.; Liu, Y.; Zheng, Y.; and Pan, S. 2023. Prem: A simple yet effective approach for node-level graph anomaly detection. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1253–1258. IEEE.
- Pan, J.; Zheng, Y.; Tan, Y.; and Liu, Y. 2025b. A Survey of Generalization of Graph Anomaly Detection: From Transfer Learning to Foundation Models. *Proceedings-2025 IEEE International Conference on Knowledge Graph, ICKG 2025*.
- Platonov, O.; Kuznedelev, D.; Diskin, M.; Babenko, A.; and Prokhorenkova, L. 2023. A critical look at evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*.
- 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.
- Rayana, S.; and Akoglu, L. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, 985–994.
- Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.
- Shen, X.; Liu, Y.; Dai, Y.; Wang, Y.; Miao, R.; Tan, Y.; Pan, S.; and Wang, X. 2025. Understanding the Information Propagation Effects of Communication Topologies in LLM-based Multi-Agent Systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Shi, F.; Cao, Y.; Shang, Y.; Zhou, Y.; Zhou, C.; and Wu, J. 2022. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM web conference 2022*, 1486–1494.
- Tan, Y.; Hu, X.; Xue, H.; De Melo, C.; and Salim, F. D. 2025. Bisecl: Binding and Separation in Continual Learning for Video Language Understanding. In *Advances in Neural Information Processing Systems*.
- 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, C.; and Zhu, H. 2022. Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security. *IEEE Transactions on Information Forensics and Security*, 17: 2703–2718.
- Wang, Q.; Pang, G.; Salehi, M.; Xia, X.; and Leckie, C. 2023. Open-set graph anomaly detection via normal structure regularisation. *arXiv preprint arXiv:2311.06835*.
- Wu, Z.; Cai, J.; Zhang, Y.; Lu, J.; Chen, Z.; Zhuang, S.; and Wang, H. 2025a. Where Graph Meets Heterogeneity: Multi-View Collaborative Graph Experts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Wu, Z.; Cheng, Y.; Sun, T.; Ji, X.; and Xu, W. 2025b. MY-OPIA: Protecting Face Privacy from Malicious Personalized Text-to-Image Synthesis via Unlearnable Examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 905–913.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.
- Xu, F.; Wang, N.; Wu, H.; Wen, X.; Zhao, X.; and Wan, H. 2024. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 9214–9222.
- Yu, Z.; Liang, C.; Chang, X.; He, D.; Jin, D.; and Wei, J. 2025. Dynamic Neighborhood Modeling via Node-Subgraph Contrastive Learning for Graph-Based Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 13115–13123.
- Zhao, Y.; Liu, Y.; Li, S.; Chen, Q.; Zheng, Y.; and Pan, S. 2025. Freegad: A training-free yet effective approach for graph anomaly detection. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*.
- Zhuo, W.; Liu, Z.; Hooi, B.; He, B.; Tan, G.; Fathony, R.; and Chen, J. 2024. Partitioning Message Passing for Graph Fraud Detection. In *The Twelfth International Conference on Learning Representations*.